

Figure 1. Framework based on RCCDA for face recognition.

Algorithm 1 RCCDA

Input:

Labeled data $\{X, y\}$ and dimension of random feature m .

Output:

- 1: Construct the matrix of classes Y for label y by one-of- k label encoding defined in Eq. (13).
- 2: Generate random features \hat{X} by the Nyström method as defined in Eq. (16).
- 3: Compute optimal discriminant vector matrix W_X by solving $[W_X, W_Y] = \text{CCA}(\hat{X}, Y)$.
- 4: Obtain random optimal discriminant features by $Z \leftarrow \hat{X}W_X$.
- 5: **return** Z ;

2.4 RCCDA for Face Recognition

Based on the theory of RCCDA proposed above, as shown in Fig. 1, we present a simple framework for face recognition.

Firstly, we preprocess all face images and split images into training set and test set, then represent images with extracted local features.

Secondly, Algorithm 1 has been used to obtain random optimal discriminant features for face images. We extract random features and calculate optimal discriminant vectors only by training set, then project all original face representations to optimal discriminant subspace, which can reduce the dimension of face features while preserving as much discriminatory information as possible.

Finally, we use random optimal discriminant features of training set to train a classifier to divided test set into different subjects.

3 Experiment Results

In this section, we evaluate the performance of the proposed RCCDA method on the Extended Yale B[19], AR[24], ORL[29] and FERET[27] face databases. These databases are widely used in evaluating the performances of various face recognition algorithms. As a comparison, we consider the basis methods Nearest Neighbor

(NN), and some state-of-the-art methods: spatially smooth LPP (S-LPP)[5], discriminative Graph regularized Extreme Learning Machine (GELM)[26], Relaxed Collaborative Representation (RCR)[42], Regularized Robust Sparse Coding (RRSC)[41], Kernel Sparse Representation (KSR)[15] and Kernel Collaborative Representation (KCR)[38]. The results of GELM and KCR are from their corresponding papers for no source codes found.

All face images are transformed into grayscale form and preprocessed by the code proposed in [36]. The local features are WTP descriptors[44] extracted from 3×3 image patches. For RCCDA, our shift-invariant kernel is Gaussian and dimension of random features is around $n/4$ (where n is number of training samples). We adopt the LIBSVM[6] classifier which is used in SLPP, KSR with the same parameter settings. NN used unsupervised nearest neighbor classifier. GELM trained an extreme learning machine as a classifier. Other methods used representation-based classifiers, which can be consider as least squares regressions with different constraints. We measure performance based on recognition accuracy rate, averaged over 10 random training/testing splits. All experiments are carried out on an Intel Pentium D processor with 3.4 GHZ frequency and 4 GB RAM.

3.1 Computational Complexity Analysis

RCCDA is extremely fast. Obviously, the computational complexity of KCCA and KFDA are both $O(n^3)$, where n is the size of training samples. CCA is $O((p^2 + q^2)n)$ and RCCA is $O((m_x^2 + m_y^2)n)$, where p, q are original dimensions of samples and m_x, m_y are dimensions of random features generated by Nyström method. CCA and RCCA are both linear in sample size n .

By connecting KCCA and KFDA with KCCDA, RCCA can be used to speed up the calculation of KFDA. That is to say, instead of calculating the KFDA directly whose computational complexity is $O(n^3)$, RCCDA can be calculated with only $O((m^2 + c^2)n)$ computational complexity (where c is number of classes).

We study the effect of random feature's dimensions m on Extended Yale B database, where number of classes $c = 38$. Randomly 10 images for each subject are selected for training, and the others are

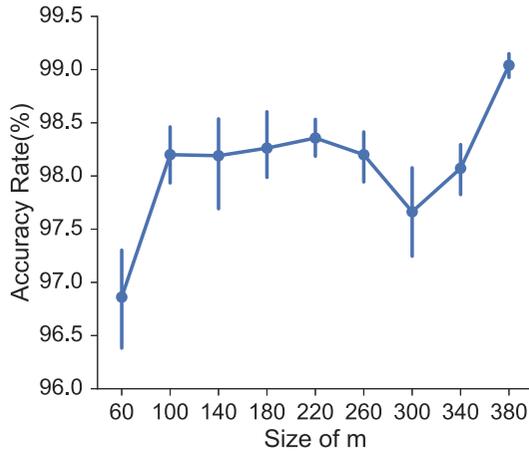


Figure 2. Recognition rates of RCCDA versus different size of m with 10 training samples on Extended Yale B database.

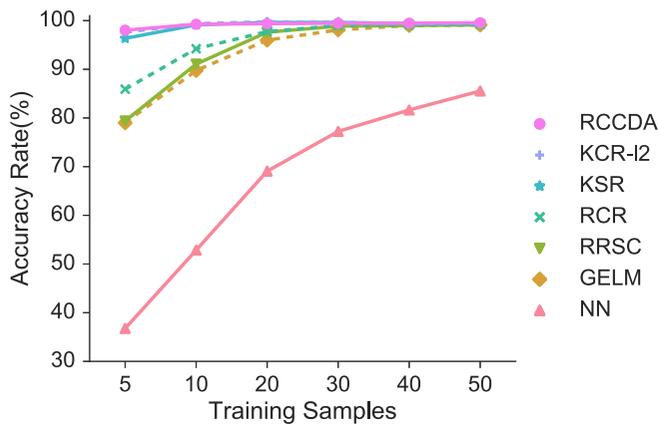


Figure 3. The face recognition results obtained on Extended Yale B database with different number of training samples.

used for test. As shown in Fig. 2, we can get good performance with a low dimension $m = 100$. Comparing to the recognition rate with $m = 380$, which is equal to KCCA or KFDDA, the proposed method can yield comparable generalization performance 12 times faster.

3.2 Experiments on Extended Yale B Database

The extended YaleB database contains 2414 frontal face images of 38 subjects taken under 64 illumination conditions, which has been widely used to evaluate the illumination robustness of face recognition methods. We collect the cropped and normalized face images of 192×168 pixels for our experiment (see Fig. 4 for some examples). A random subset with l images per individual is collected to form a training set ($l = 5, 10, 20, 30, 40$), and the rest is considered as the corresponding test set.

Our experimental results are shown in Fig. 3, in which we can observe that our method can achieve better performance than others especially with small number of training samples ($l = 5, 10$). Such results benefit from that our method retains discriminatory information of samples as much as possible.



Figure 4. Sample images in the extended Yale B database with variations of illumination.



Figure 5. Samples un-occluded (first row), with sunglasses (second row) and scarves (third row) in the AR database.

3.3 Experiments on AR Database

The AR face database contains over 4,000 color images corresponding to 126 people's faces (70 men and 56 women). All images are frontal view with different facial expressions, illumination conditions, and occlusions. In this experiment, we choose a subset that contains 50 men and 50 women. For each subject, 26 images are chosen (14 un-occluded, 6 occluded with sunglasses and 6 occluded with scarves). Each image is cropped to 165×120 pixels, as shown in Fig. 5.

In order to evaluate the robustness of our method under occlusion, we use 1400 un-occluded images as a training set and randomly splits l images ($l = 2, 4, 6, 8, 10$) per individual to training a classifier. We first consider the rest of un-occluded images as a test set. Then, 600 images of subjects wearing sunglasses and 600 images of subjects wearing scarves are used to form other two test sets. Fig. 6 shows the recognition rates of occlusion tests, we can learn from them that because of robustness for WTP on expression and occlusion our approach obtains higher accuracy rate than other methods when the number of training samples are 4, 6 and 8.

3.4 Experiments on ORL Database

ORL database consists of 400 different images of 40 individuals. For each subjects, there are 10 images were taken at different times, varying the lighting, facial expressions and facial details. All image are normalized to 112×92 in our experiment (see Fig. 7 for some examples).

We randomly select l images ($l = 2, 3, 4, 5$) of each individual images as training data and the rest as test data. The average recognition rates are reported in Tab. 1. With 5 training samples, proposed method gets close to 100% accuracy rate.

3.5 Experiments on FERET Database

The FERET database consists of 13539 face images of 1565 subjects who are diverse across age, gender, and ethnicity. We use a subset contains 1400 images from 200 subjects with 7 frontal images per

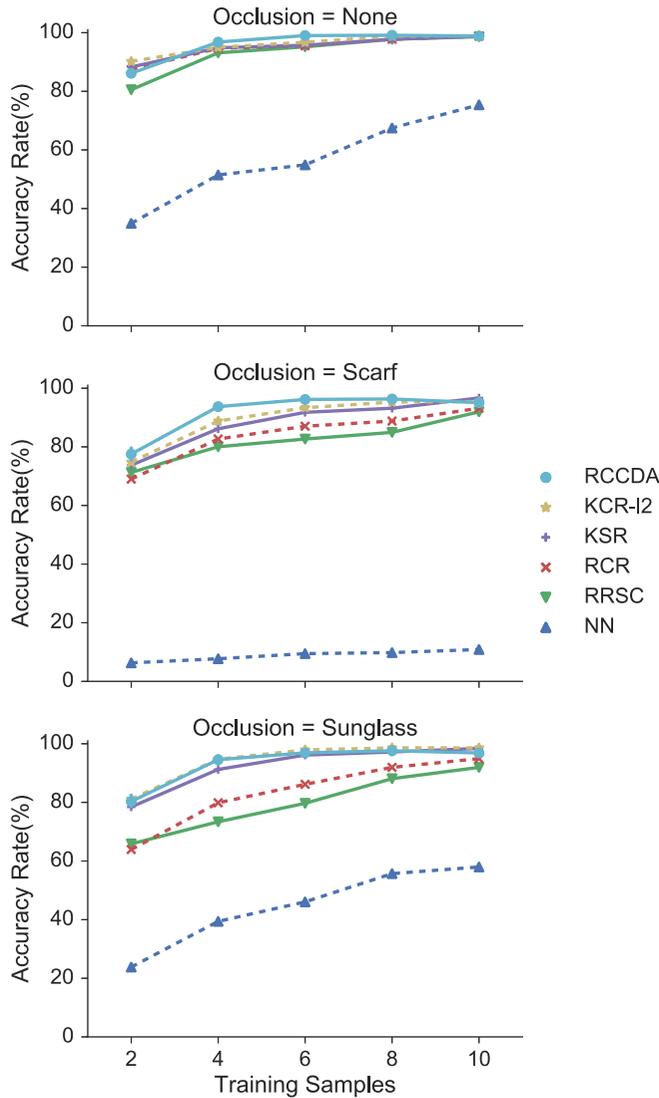


Figure 6. The face recognition results obtained on AR un-occluded, occluded with sunglasses and scarves subset with different number of training samples.



Figure 7. Sample images in ORL database with variations of expression and pose.

subject in this paper. Images are scaled and cropped to the size of 80×80 . Fig. 8 shows the normalized and cropped images.

In our experiment, 5 images per individual are randomly selected to form a training set, and the rest is used for tests. Tab. 2 demonstrates

Table 1. The face recognition results obtained on ORL database with different number of training samples (mean(std-dev))%.

Methods	2 Trains	3 Trains	4 Trains	5 Trains
NN	66.9	76.6	82.1	86.3
S-LPP	82.9	91.9	95.9	97.7
GELM	84.2	90.7	94.3	96.3
RCCDA	91.2(1.9)	95.6(1.6)	97.6(2.6)	99.3(0.7)

Table 2. The face recognition results obtained on FERET subset (mean(std-dev))%.

Methods	NN	RCR	RRSC	KSR	KCR-I2	RCCDA
Accuracy	69.5	82.5	81.3	83.0	88.3	91.5(0.9)



Figure 8. A subset of normalized images from FERET database.

that recognition rates of the proposed method is higher than other algorithms.

4 Conclusion and Further Work

In this paper, an efficient RCCDA method is presented for face recognition. Our method extracts local features firstly, then applies RCCDA to reduce the dimension and map the local feature into an optimal discriminant subspace. The main advantage of the proposed method is that RCCDA preserves as much discriminatory information as possible and greatly accelerates the computation through randomized method. The result of experiments on Extended Yale B, AR, ORL and FERET face databases demonstrates that our method achieve better performance than several state-of-the-art methods.

As an effective feature extraction method, the proposed RCCDA can also be used to other recognition tasks, such as visual tracking, image retrieval and image classification. For these tasks, the process of feature extraction can be replaced by our method. Necessary distinguishing information can be obtained efficiently with only small number of training samples. Our future work will focus on apply the proposed method to other recognition problems and optimize the combination of local features, kernel functions and other RCCDA encoding methods (such as [33]).

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (No. 61472036). We would like to thank both the editor and the reviewers for the invaluable comments. Many thanks to David Lopez-Paz for very helpful discussions.

REFERENCES

- [1] Yael Adini, Yael Moses, and Shimon Ullman, 'Face recognition: The problem of compensating for changes in illumination direction', *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **19**(7), 721–732, (1997).
- [2] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen, 'Face recognition with local binary patterns', in *Computer vision-ecv 2004*, 469–481, Springer, (2004).

- [3] Peter N Belhumeur, João P Hespanha, and David J Kriegman, 'Eigenfaces vs. fisherfaces: Recognition using class specific linear projection', *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **19**(7), 711–720, (1997).
- [4] Roberto Brunelli and Tomaso Poggio, 'Face recognition: Features versus templates', *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (10), 1042–1052, (1993).
- [5] Deng Cai, Xiaofei He, Yuxiao Hu, Jiawei Han, and Thomas Huang, 'Learning a spatially smooth subspace for face recognition', in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–7. IEEE, (2007).
- [6] Chih-Chung Chang and Chih-Jen Lin, 'LIBSVM: A library for support vector machines', *ACM Transactions on Intelligent Systems and Technology*, **2**, 27:1–27:27, (2011). Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [7] Tijl De Bie, Nello Cristianini, and Roman Rosipal, 'Eigenproblems in pattern recognition', in *Handbook of Geometric Computing*, 129–167, Springer, (2005).
- [8] Petros Drineas and Michael W Mahoney, 'On the nystrom method for approximating a gram matrix for improved kernel-based learning', *The Journal of Machine Learning Research*, **6**, 2153–2175, (2005).
- [9] Ronald A Fisher, 'The use of multiple measurements in taxonomic problems', *Annals of eugenics*, **7**(2), 179–188, (1936).
- [10] Yunhui He, Li Zhao, and Cairong Zou, 'Face recognition based on pca/kpca plus cca', in *Advances in Natural Computation*, 71–74, Springer, (2005).
- [11] Bernd Heisele, Purdy Ho, Jane Wu, and Tomaso Poggio, 'Face recognition: component-based versus global approaches', *Computer vision and image understanding*, **91**(1), 6–21, (2003).
- [12] Harold Hotelling, 'Relations between two sets of variates', *Biometrika*, **28**(3/4), 321–377, (1936).
- [13] Xudong Jiang, Bappaditya Mandal, and Alex Kot, 'Eigenfeature regularization and extraction in face recognition', *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **30**(3), 383–394, (2008).
- [14] Björn Johansson, 'On classification: simultaneously reducing dimensionality and finding automatic representation using canonical correlation', (2001).
- [15] Cuicui Kang, Shengcai Liao, Shiming Xiang, and Chunhong Pan, 'Kernel sparse representation with local patterns for face recognition', in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pp. 3009–3012. IEEE, (2011).
- [16] Kwang In Kim, Keechul Jung, and Hang Joon Kim, 'Face recognition using kernel principal component analysis', *Signal Processing Letters, IEEE*, **9**(2), 40–42, (2002).
- [17] G Kukharev and E Kamenskaya, 'Application of two-dimensional canonical correlation analysis for face image processing and recognition', *Pattern Recognition and Image Analysis*, **20**(2), 210–219, (2010).
- [18] Olcay Kursun, Ethem Alpaydin, and Oleg V Favorov, 'Canonical correlation analysis using within-class coupling', *Pattern Recognition Letters*, **32**(2), 134–144, (2011).
- [19] Kuang-Chih Lee, Jeffrey Ho, and David J Kriegman, 'Acquiring linear subspaces for face recognition under variable lighting', *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **27**(5), 684–698, (2005).
- [20] Chengjun Liu, 'Gabor-based kernel pca with fractional power polynomial models for face recognition', *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **26**(5), 572–581, (2004).
- [21] Chengjun Liu and Harry Wechsler, 'Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition', *Image processing, IEEE Transactions on*, **11**(4), 467–476, (2002).
- [22] David Lopez-Paz, Suvirit Sra, Alex Smola, Zoubin Ghahramani, and Bernhard Schölkopf, 'Randomized nonlinear component analysis', *arXiv preprint arXiv:1402.0119*, (2014).
- [23] Gui-Fu Lu, Zhong Lin, and Zhong Jin, 'Face recognition using discriminant locality preserving projections based on maximum margin criterion', *Pattern Recognition*, **43**(10), 3572–3579, (2010).
- [24] Aleix M Martinez, 'The ar face database', *CVC Technical Report*, **24**, (1998).
- [25] Brian McWilliams, David Balduzzi, and Joachim M Buhmann, 'Correlated random features for fast semi-supervised learning', in *Advances in Neural Information Processing Systems*, pp. 440–448, (2013).
- [26] Yong Peng, Suhang Wang, Xianzhong Long, and Bao-Liang Lu, 'Discriminative graph regularized extreme learning machine and its application to face recognition', *Neurocomputing*, **149**, 340–353, (2015).
- [27] P Jonathon Phillips, Hyeonjoon Moon, Syed A Rizvi, and Patrick J Rauss, 'The feret evaluation methodology for face-recognition algorithms', *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **22**(10), 1090–1104, (2000).
- [28] Lishan Qiao, Songcan Chen, and Xiaoyang Tan, 'Sparsity preserving projections with applications to face recognition', *Pattern Recognition*, **43**(1), 331–341, (2010).
- [29] Ferdinando S Samaria and Andy C Harter, 'Parameterisation of a stochastic model for human face identification', in *Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on*, pp. 138–142. IEEE, (1994).
- [30] Bernhard Scholkopf and Klaus-Robert Mullert, 'Fisher discriminant analysis with kernels', *Neural networks for signal processing IX*, **1**(1), 1, (1999).
- [31] Quan-Sen Sun, Pheng-Ann Heng, Zhong Jin, and De-Shen Xia, 'Face recognition based on generalized canonical correlation analysis', in *Advances in intelligent computing*, 958–967, Springer, (2005).
- [32] Quan-Sen Sun, Sheng-Gen Zeng, Yan Liu, Pheng-Ann Heng, and De-Shen Xia, 'A new method of feature fusion and its application in image recognition', *Pattern Recognition*, **38**(12), 2437–2448, (2005).
- [33] Tingkai Sun and Songcan Chen, 'Class label versus sample label-based cca', *Applied Mathematics and computation*, **185**(1), 272–283, (2007).
- [34] Tingkai Sun, Songcan Chen, Jingyu Yang, Xuelei Hu, and Pengfei Shi, 'Discriminative canonical correlation analysis with missing samples', in *Computer Science and Information Engineering, 2009 WRI World Congress on*, volume 6, pp. 95–99. IEEE, (2009).
- [35] Xiaoyang Tan and Bill Triggs, 'Fusing gabor and lbp feature sets for kernel-based face recognition', in *Analysis and Modeling of Faces and Gestures*, 235–249, Springer, (2007).
- [36] Xiaoyang Tan and Bill Triggs, 'Enhanced local texture feature sets for face recognition under difficult lighting conditions', *Image Processing, IEEE Transactions on*, **19**(6), 1635–1650, (2010).
- [37] Matthew Turk and Alex Pentland, 'Eigenfaces for recognition', *Journal of cognitive neuroscience*, **3**(1), 71–86, (1991).
- [38] Dong Wang, Huchuan Lu, and Ming-Hsuan Yang, 'Kernel collaborative face recognition', *Pattern Recognition*, **48**(10), 3025–3037, (2015).
- [39] Christopher Williams and Matthias Seeger, 'Using the nystrom method to speed up kernel machines', in *Proceedings of the 14th Annual Conference on Neural Information Processing Systems*, number EPFL-CONF-161322, pp. 682–688, (2001).
- [40] Jian Yang, David Zhang, Alejandro F Frangi, and Jing-yu Yang, 'Two-dimensional pca: a new approach to appearance-based face representation and recognition', *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **26**(1), 131–137, (2004).
- [41] Meng Yang, Lei Zhang, Jian Yang, and Dejing Zhang, 'Regularized robust coding for face recognition', *Image Processing, IEEE Transactions on*, **22**(5), 1753–1766, (2013).
- [42] Meng Yang, Lei Zhang, David Zhang, and Shenlong Wang, 'Relaxed collaborative representation for pattern classification', in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 2224–2231. IEEE, (2012).
- [43] Tianbao Yang, Yu-Feng Li, Mehrdad Mahdavi, Rong Jin, and Zhi-Hua Zhou, 'Nystrom method vs random fourier features: A theoretical and empirical comparison', in *Advances in neural information processing systems*, pp. 476–484, (2012).
- [44] Zuodong Yang, Yinyan Jiang, Yong Wu, Zongqing Lu, Weifeng Li, and Qingmin Liao, 'Weber binary pattern and weber ternary pattern for illumination-robust face recognition', in *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pp. 1050–1053. IEEE, (2015).
- [45] Wencho Zhang, Shiguang Shan, Wen Gao, Xilin Chen, and Hongming Zhang, 'Local gabor binary pattern histogram sequence (lgbphs): A novel non-statistical model for face representation and recognition', in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pp. 786–791. IEEE, (2005).
- [46] Wenyi Zhao, Rama Chellappa, P Jonathon Phillips, and Azriel Rosenfeld, 'Face recognition: A literature survey', *ACM computing surveys (CSUR)*, **35**(4), 399–458, (2003).

Reconsidering AGM-Style Belief Revision in the Context of Logic Programs

Zhiqiang Zhuang¹ and James Delgrande² and Abhaya Nayak³ and Abdul Sattar¹

Abstract.

Belief revision has been studied mainly with respect to background logics that are monotonic in character. In this paper we study belief revision when the underlying logic is non-monotonic instead—an inherently interesting problem that is under explored. In particular, we will focus on the revision of a body of beliefs that is represented as a logic program under the answer set semantics, while the new information is also similarly represented as a logic program. Our approach is driven by the observation that unlike in a monotonic setting where, when necessary, consistency in a revised body of beliefs is maintained by jettisoning some old beliefs, in a non-monotonic setting consistency can be restored by adding new beliefs as well. We will define two revision functions through syntactic and model-theoretic methods respectively and subsequently provide representation theorems for characterising them.

1 Introduction

The ability to change one's beliefs when presented with new information is crucial for any intelligent agent. In the area of *belief change*, substantial effort has been made towards the understanding and realisation of this process. Traditionally, it is assumed that the agent's reasoning is governed by a monotonic logic. For this reason, traditional belief change is inapplicable when the agent's reasoning is non-monotonic. Our goal in this research program is to extend the established (AGM) belief set [1] and belief base [13] approaches in belief revision to nonmonotonic setting. In this paper, we focus on *disjunctive logic programs*, as a well-studied and well-known approach to nonmonotonic reasoning that also has efficient implementations.

Much, if not most, of our day-to-day reasoning involves non-monotonic reasoning. To illustrate issues that may arise, consider the following example. In a university, professors generally teach, unless they have an administrative appointment. Assume we know that John is a professor. Since most faculty do not have an administrative appointment, and there is no evidence that John does, we conclude that he teaches. This reasoning is a classical form of non-monotonic reasoning, namely using the *closed world assumption*. It can be represented by the following logic program under the *answer set semantics*.

$$\text{Teach}(\text{John}) \leftarrow \text{Prof}(\text{John}), \text{not Admin}(\text{John}). \quad (1)$$

$$\text{Prof}(\text{John}) \leftarrow . \quad (2)$$

The *answer set* $\{\text{Prof}(\text{John}), \text{Teach}(\text{John})\}$ for this logic program corresponds exactly to the facts we can conclude.

Suppose we receive information that John does not teach, which we can represent by the rule

$$\leftarrow \text{Teach}(\text{John}). \quad (3)$$

Now our beliefs about John are contradictory; and it is not surprising that the logic program consisting of rules (1) – (3) has no answer set. For us or any intelligent agent in this situation to function properly, we need a mechanism to resolve this inconsistency. This is a typical belief revision problem; however, the classical (AGM) approach can not be applied, as we are reasoning non-monotonically.

It is not hard to suggest possible causes of the inconsistency and to resolve it. It could be that some of our beliefs are wrong; perhaps professors with administrative duties may still need to do teaching or perhaps John is not a professor. Thus we can restore consistency by removing rule (1) or (2). Alternatively and perhaps more interestingly, it could be that assuming that John is not an administrative staff via the absence of evidence is too adventurous; that is he may indeed be an administrative staff member but we don't know it. Thus we can also restore consistency by adding the missing evidence of John being an administrative staff member by

$$\text{Admin}(\text{John}) \leftarrow . \quad (4)$$

The second alternative highlights the distinction for belief revision in monotonic and non-monotonic settings. In the monotonic setting, an inconsistent body of knowledge will remain inconsistent no matter how much extra information is supplied. On the other hand, in the non-monotonic setting, inconsistency can be resolved by either removing old information, or adding new information, or both. Therefore, belief revision functions in a non-monotonic setting should allow a mixture of removal and addition of information for inconsistency-resolution. In this paper, we will define two such revision functions for disjunctive logic programs under the answer set semantics.

Our first revision function called *slp-revision*⁴ is like belief base revision which takes syntactic information into account. In revising P by Q , a slp-revision function first obtains a logic program R that is consistent with Q and differs minimally from P , then combines R with Q . For example, if $P = \{(1), (2)\}$ and $Q = \{(3)\}$, then R could be $\{(1)\}$ (i.e., resolving inconsistency by removing (2)); $\{(2)\}$ (i.e., resolving inconsistency by removing (1)); or $\{(1), (2), (4)\}$ (i.e., resolving inconsistency by adding (4)). Our second revision function called *llp-revision function*⁵ is like AGM belief set revision which ignores syntactic difference and focuses on the logical content of a knowledge base. So in revising P by Q , a llp-revision function

¹ IIS, Griffith University, Australia

² School of Computing Science, Simon Fraser University, Canada

³ Department of Computing, Macquarie University, Australia

⁴ “s” stands for syntactic and “lp” for logic program.

⁵ “l” stands for logical content and “lp” for logic program.

will instead obtain a logic program R whose logical content differ the least from that of P where the logical content is characterised by *strong equivalent (SE) models* [27].

The next section gives logical preliminaries. The following two sections develop our approach to slp-revision and llp-revision, in each case providing postulates, a semantic construction, and a representation result. This is followed by a comparison to other work, and a brief conclusion.

2 Preliminary Considerations

In this paper, we consider only fully grounded disjunctive logic programs. Thus a logic program (or program for short) here is a finite set of rules of the form:

$$a_1; \dots; a_m \leftarrow b_1, \dots, b_n, \text{not } c_1, \dots, \text{not } c_o$$

where $m, n, o \geq 0$, $m + n + o > 0$, and $a_i, b_j, c_k \in \mathcal{A}$ for \mathcal{A} a finite set of propositional atoms. We denote the set of all logic programs by \mathcal{P} . For each rule r , let $H(r) = \{a_1, \dots, a_n\}$, $B^+(r) = \{b_1, \dots, b_m\}$, and $B^-(r) = \{c_1, \dots, c_o\}$. The letters P, Q and R are used to denote a logic program throughout the paper.

An interpretation is represented by the subset of atoms in \mathcal{A} that are true in the interpretation. A *classical model* of a program P is an interpretation in which all rules of P are true according to the standard definition of truth in propositional logic, and where default negation is treated as classical negation. The set of classical models of P is denoted as $Mod(P)$. Given an interpretation Y , we write $Y \models P$ to mean Y is a classical model of P . The *reduct* of a program P with respect to an interpretation Y , denoted P^Y , is the set of rules:

$$\{H(r) \leftarrow B^+(r) \mid r \in P, B^-(r) \cap Y = \emptyset\}.$$

An *answer set* Y of P is a subset-minimal classical model of P^Y . The set of all answer sets of P is denoted as $AS(P)$.

An *SE interpretation* [27] is a pair (X, Y) of interpretations such that $X \subseteq Y \subseteq \mathcal{A}$. The set of all SE interpretations (over \mathcal{A}) is denoted \mathcal{SE} . The letters M and N are used to denote a set of SE interpretations throughout the paper. An SE interpretation is an *SE model* of a program P if $Y \models P$ and $X \models P^Y$. The set of all SE models of P is denoted as $SE(P)$. SE models are first proposed to capture *strong equivalence* [18] between programs that is $SE(P) = SE(Q)$ iff P and Q are strongly equivalent, thus they contain more informations than answer sets. For this reason, SE models have been used by many to characterise the logical content of a program [5, 7]. The following definitions and results are given in [10, 8]. A set M of SE interpretations is *well-defined* if $(X, Y) \in M$ implies $(Y, Y) \in M$. M is *complete* if it is well-defined and if $(X, Y) \in M$, $(Z, Z) \in M$ and $Y \subseteq Z$, then $(X, Z) \in M$. For each disjunctive logic program P , $SE(P)$ is complete and for each complete set M , there is a unique (up to strong equivalence) disjunctive logic program P such that $SE(P) = M$.

In this paper, we work with two notions of closure. The *closure* of a set M under completeness is denoted as $Cl(M)$. Formally, $Cl(M)$ is the minimal superset of M such that

1. if $(X, Y) \in M$, then $(Y, Y) \in Cl(M)$ and
2. if $(X, Y), (Z, Z) \in M$ and $Y \subseteq Z$, then $(X, Z) \in Cl(M)$.

The *closure* of a program P which intends to capture all logical consequence of P is denoted as $cl(P)$. Formally

$$cl(P) = \{r \mid SE(P) \subseteq SE(\{r\}) \text{ and } r \text{ is a program rule}\}.$$

We say P is *closed* if $P = cl(P)$.

The following two properties of SE models [27] are crucial to this paper:

1. $Y \in AS(P)$ iff $(Y, Y) \in SE(P)$ and there is no $(X, Y) \in SE(P)$ such that $X \subset Y$.
2. $(Y, Y) \in SE(P)$ iff $Y \in Mod(P)$.

So $SE(P) \neq \emptyset$ iff $Mod(P) \neq \emptyset$ but $SE(P) \neq \emptyset$ does not imply $AS(P) \neq \emptyset$. This gives rise to two notions of consistency.

Definition 1. P is consistent iff $AS(P) \neq \emptyset$ and P is m-consistent⁶ iff $SE(P) \neq \emptyset$.

It follows from the SE model properties that consistency implies m-consistency and m-inconsistency implies inconsistency. In other words, a consistent program is m-consistent but not vice versa. For convenience, we say a set M of SE interpretations is consistent iff the program P such that $SE(P) = Cl(M)$ is consistent; we say (Y, Y) is an “answer set” in M iff $(Y, Y) \in M$ and there is no $(X, Y) \in M$ such that $X \subset Y$. Clearly, M is consistent iff $Cl(M)$ is consistent; the consistency of M indicates the consistency of its corresponding logic program; and (Y, Y) being an answer set in M indicates Y is an answer set of the corresponding program.

In subsequent sections, we will need to describe the difference between two logic programs and between their sets of SE models. For this purpose, we use the *symmetric difference* operator \ominus which is defined as

$$X \ominus Y = (X \setminus Y) \cup (Y \setminus X)$$

for any sets X and Y .

3 SLP-Revision Functions

In this section, we give a syntax-based revision function $*$: $\mathcal{P} \times \mathcal{P} \mapsto \mathcal{P}$ for revising one logic program by another. The function maps a logic program P called the *original logic program* and a logic program Q called the *revising logic program*, to another logic program $P * Q$ called the *revised logic program*. Following AGM belief revision, we want to have Q contained in $P * Q$ (i.e., $Q \subseteq P * Q$) and $P * Q$ is consistent whenever possible.

A main task in defining $*$ is to deal with the possible inconsistency between Q and P . As illustrated in the teaching example, one means of ensuring that $P * Q$ is consistent is to remove a minimal set of beliefs from P so that adding Q to the result is consistent. Of course there may be more than one way to remove beliefs from P . Following this intuition, we obtain all maximal subsets of P that are consistent with Q , which we call the *s-removal compatible programs* of P with respect to Q .

Definition 2. The set of s-removal compatible programs of P with respect to Q , denoted $P \downarrow Q$, is such that $R \in P \downarrow Q$ iff

1. $R \subseteq P$,
2. $R \cup Q$ is consistent, and
3. if $R \subset R' \subseteq P$, then $R' \cup Q$ is inconsistent.

The notion of s-removal compatible programs is not new, classical revision functions [1, 11] are based on more or less the same notion. The difference is that this notion alone is sufficient to capture the inconsistency-resolution strategy of classical belief revision, but there is more that one can do in non-monotonic belief revision.

In our non-monotonic setting, we are able to express assumptions (i.e., *negation as failure*) and to reason with them. Earlier, we assumed John is not an administrator, in the absence of evidence to the

⁶ “m” stands for “monotonic” which indicates that the notion of m-consistency is based on a monotonic characterisation (i.e., SE models) for logic programs.

contrary. With this, we came to the conclusion that he has to teach. Consequently, if we learn that John does not teach, as in our example, one way of resolving this inconsistency is by adding a minimal set of information so that our assumption does not hold. Following this intuition, we obtain all the minimal supersets of P that are consistent with Q , which we call the *s-expansion compatible programs* of P with respect to Q .

Definition 3. The set of s-expansion compatible programs of P with respect to Q , denoted $P \uparrow Q$, is such that $R \in P \uparrow Q$ iff

1. $P \subseteq R$,
2. $R \cup Q$ is consistent, and
3. if $P \subseteq R' \subset R$, then $R' \cup Q$ is inconsistent.

Since the s-expansion and s-removal compatible programs are consistent with Q and are obtained by removing or adding minimal sets of rules from or to P , the union of Q with any of these sets is consistent and comprises a least change made to P in order to achieve consistency. These programs clearly should be candidates for forming the revised logic program $P * Q$; however, they do not form the set of all candidates. In particular, we can obtain a program that differs the least from P and is consistent with Q by removing some beliefs of P and at the same time adding some new beliefs to P . Thus we consider all those logic programs that differ the least from P and are consistent with Q ; these are called the *s-compatible programs* of P with respect to Q .

Definition 4. The set of s-compatible programs of P with respect to Q , denoted $P \downarrow Q$, is such that $R \in P \downarrow Q$ iff

1. $R \cup Q$ is consistent and
2. if $P \ominus R' \subset P \ominus R$, then $R' \cup Q$ is inconsistent.

For example, let $P = \{a \leftarrow b, \text{not } c., b., e \leftarrow f, \text{not } g., f.\}$ and $Q = \{\leftarrow a., \leftarrow e.\}$. Then $P \cup Q$ is inconsistent since a and e can be concluded from P but they contradict the rules of Q . To resolve the inconsistency via making the least change to P , we could remove $b \leftarrow$ from P (which eliminates the contradiction about a) and add $g \leftarrow$ to P (which eliminates the contradiction about e). The program thus obtained (i.e., $(P \setminus \{b.\}) \cup \{g.\}$) is a s-compatible program in $P \downarrow Q$.

It is obvious, but worth noting that the notion of s-compatible program subsumes those of s-removal and s-expansion compatible programs. In the above example, $P \downarrow Q$ also contains $P \setminus \{b., f.\}$ and $P \cup \{c., g.\}$, which are respectively an s-removal and an s-expansion compatible program of P with respect to Q .

Proposition 1. $(P \uparrow Q) \cup (P \downarrow Q) \subseteq P \downarrow Q$.

There are cases in which we cannot resolve inconsistency by only adding new beliefs which means the set of s-expansion compatible programs is empty. For example, if $P = \{a.\}$ and $Q = \{\leftarrow a.\}$, then $P \cup Q$ is inconsistent and we cannot restore consistency without removing $a \leftarrow$ from P . In these cases, the inconsistency is due to contradictory facts that can be concluded without using any reasoning power beyond that of classical logic. Clearly, the inconsistency is of a monotonic nature, that is, in our terminology, m-inconsistency.

Proposition 2. If $P \cup Q$ is m-inconsistent, then $P \uparrow Q = \emptyset$.

So far, we have identified the candidates for forming $P * Q$. It remains to pick the ‘‘best’’ one. Such extralogical information is typically modelled by a *selection function*, which we do next.

Definition 5. A function γ is a selection function for P iff for any program Q , $\gamma(P \downarrow Q)$ returns a single element of $P \downarrow Q$ whenever $P \downarrow Q$ is non-empty; otherwise it returns P .

The revised logic program $P * Q$ is then formed by combining Q with the s-compatible program picked by the selection function for P . We call the function $*$ defined in this way a *slp-revision function* for P .

Definition 6. A function $*$: $\mathcal{P} \times \mathcal{P} \mapsto \mathcal{P}$ is a slp-revision function iff

$$P * Q = \gamma(P \downarrow Q) \cup Q$$

where γ is a selection function for P .

In classical belief revision, multiple candidates maybe chosen by a selection function, and their intersection is combined with the new belief to form the revision result. There, a selection function that picks out a single element is called a *maxichoice* function [1]. In classical logic, maxichoice selection functions lead to undesirable properties for belief set revision but not for belief base revision. In our non-monotonic setting, picking multiple candidates does not make sense, as intersection of s-compatible programs may not be consistent with the revising program. For example, let $P = \{a \leftarrow \text{not } b, \text{not } c.\}$ and $Q = \{\leftarrow a.\}$. We can restore consistency of P with Q by, for instance, adding the rule $b \leftarrow$ to P which corresponds to the s-compatible program $P \cup \{b.\}$ or by adding the rule $c \leftarrow$ which corresponds to the s-compatible program $P \cup \{c.\}$. However, the intersection of the two s-compatible programs is inconsistent with Q .

We turn next to properties of slp-revision functions. Consider the following set of postulates for a function $*$: $\mathcal{P} \times \mathcal{P} \mapsto \mathcal{P}$.

- (s*s) $Q \subseteq P * Q$.
- (s*c) If Q is m-consistent, then $P * Q$ is consistent.
- (s*f) If Q is m-inconsistent, then $P * Q = P \cup Q$.
- (s*rr) If $R \neq \emptyset$ and $R \subseteq P \setminus (P * Q)$, then $(P * Q) \cup R$ is inconsistent.
- (s*er) If $E \neq \emptyset$ and $E \subseteq (P * Q) \setminus (P \cup Q)$, then $(P * Q) \setminus E$ is inconsistent.
- (s*mr) If $R \neq \emptyset$, $R \subseteq P \setminus (P * Q)$, $E \neq \emptyset$, and $E \subseteq (P * Q) \setminus (P \cup Q)$, then $((P * Q) \cup R) \setminus E$ is inconsistent.
- (s*u) If $P \downarrow Q = P \downarrow R$, then $P \setminus (P * Q) = P \setminus (P * R)$ and $(P * Q) \setminus (P \cup Q) = (P * R) \setminus (P \cup R)$.

These postulates are obtained by adapting the classical belief base revision postulates [13] to the present non-monotonic setting. (s*s) (*Success*) states that a revision is always successful in incorporating the new beliefs. (s*c) (*Consistency*) states that a revision ensures consistency of the revised logic program whenever possible. In the monotonic setting, a revision results in inconsistency only when the new beliefs are themselves inconsistent. This is not the case in the non-monotonic setting. For example, consider the revision of $P = \{a.\}$ by $Q = \{b \leftarrow \text{not } b.\}$. Although Q is inconsistent, we have $P \cup \{b.\}$ as a s-compatible program of P with respect to Q . Thus we can have $P \cup \{b.\} \cup Q$ as the revised logic program, which contains Q and is consistent. Here, a revision results in inconsistency only when the revising logic program is m-inconsistent. In such a case, (s*f) (*Failure*) states that the revision corresponds to the union of the original and revising logic program.

(s*rr) (*Removal Relevance*) states that if some rules are removed from the original logic program for the revision, then adding them to the revised logic program results in inconsistency. It captures the

intuition that nothing is removed unless its removal contributes to making the revised logic program consistent. (*s*er*) (*Expansion Relevance*) states that if some new rules other than those in the revising logic program are added to the original logic program for the revision, then removing them from the revised logic program results in inconsistency. It captures the intuition that nothing is added unless adding it contributes to making the revised logic program consistent. (*s*mr*) (*Mixed Relevance*) states that if some rules are removed from the original logic program and some new rules other than those in the revising logic program are added to the original logic program for the revision, then adding back the removed ones and removing the added ones result in inconsistency. Its intuition is a mixture of the two above. Note that putting (*s*rr*) and (*s*er*) together does not guarantee (*s*mr*), nor the reverse. In summary, these three postulates express the necessity of adding and/or removing certain belief for resolving inconsistency and hence to accomplish a revision. In classical belief revision, inconsistency can only be resolved by removing old beliefs; the necessity of removing particular beliefs is captured by the *Relevance* postulate [11].⁷ The three postulates are the counterparts of *Relevance* in our non-monotonic setting, and we need all three of them to deal respectively with addition, removal, and a mixture of addition and removal.

Finally, (*s*u*) (*Uniformity*) states the condition under which two revising logic programs Q and R trigger the same changes to the original logic program P . That is the rules removed from P (i.e., $P \setminus (P * Q)$) and the rules added to P (i.e., $(P * Q) \setminus (P \cup Q)$) for accommodating Q are identical to those for accommodating R . Certainly having Q and R be strongly equivalent (i.e., $SE(Q) = SE(R)$) is a sufficient condition. However, it is too strong a requirement. Suppose $P = \{\leftarrow a.\}$, $Q = \{a.\}$, and $R = \{a \leftarrow b., b.\}$. Then the minimal change to P we have to made to accommodate Q and R are the same, that is we remove $\leftarrow a.$. However Q and R are not strongly equivalent, even though they incur the same change to P . The essential point of this example is that instead of a global condition like strong equivalence, we need a condition that is local to the original logic program P . Unfortunately, it seems there is no existing notion in the logic programming literature that captures this local condition. Thus we use our newly defined notion of *s-compatible* programs and come up with the local but more appropriate condition in (*s*u*).

We can show that these postulates are sufficient to characterise all *slp-revision* functions.

Theorem 1. *A function $*$ is a *slp-revision* function iff it satisfies (*s*s*), (*s*c*), (*s*f*), (*s*rr*), (*s*er*), (*s*mr*), and (*s*u*).*

4 LLP-Revision Functions

Slp-revision functions preserve the syntactic structure of the original logic program as much as possible. Thus is most useful for scenarios in which syntactic information is prioritised over that of logical content. In this section, we provide a revision function called *llp-revision function* that prioritises the preservation of logical content over that of syntactic information.

The main strategy of *llp-revision* functions is the same as that of *slp-revision* functions, which is to first obtain a logic program that differs the least from the original one and that is consistent with the revising one, and then combine it with the revising program. The

distinguishing feature of *llp-revision* is in the interpretation of “differs the least”. *Slp-revision* interprets this notion as symmetric difference between the constituent rules whereas *llp-revision* interprets it as between the logical content. Since the standard approach in characterising the logical content of logic programs is through their SE models, the difference between logical content is represented as the difference between sets of SE models. This brings about the following dual notion of *s-compatible* program which we call *l-compatible program*. Note that although we are using SE models, we concern with the stronger notion of consistency (i.e. P is consistent if it has an answer set).

Definition 7. *The set of l-compatible programs of P with respect to Q , denoted $P \Downarrow Q$, is such that $R \in P \Downarrow Q$ iff*

1. $R \cup Q$ is consistent, and
2. if $SE(P) \ominus SE(R') \subset SE(P) \ominus SE(R)$, then $R' \cup Q$ is inconsistent.

A *l-compatible* program of P with respect to Q is a logic program that is consistent with Q (condition 1) and whose set of SE models differ minimally from that of P (condition 2). We denote by $P \Downarrow Q$ and $P \Uparrow Q$ the set of *l-removal compatible programs* and *l-expansion compatible programs* such that $R \in P \Downarrow Q$ iff $SE(P) \subseteq SE(R)$ and $R \in P \Uparrow Q$; $R \in P \Uparrow Q$ iff $SE(R) \subseteq SE(P)$ and $R \in P \Downarrow Q$. It is easy to see that Proposition 1 and 2 also hold for *l-compatible* programs.

We have given a declarative definition for *l-compatible* programs. The following theorem identifies all the possible constituting SE models of a *l-compatible* program.

Theorem 2. *If $R \in P \Downarrow Q$, then one of the following holds:*

1. $P \cup Q$ is inconsistent. $SE(R) = Cl((SE(P) \cup \{(Y, Y)\}) \setminus M)$ where $(Y, Y) \in SE(Q) \setminus SE(P)$ and $M = \{(X, Z) \in SE(P) \mid (X, Y) \in SE(Q) \text{ and } Z \subseteq Y\}$.⁸
2. $P \cup Q$ is inconsistent but *m-consistent*. $SE(R) = SE(P) \setminus M$ where there is $(Y, Y) \in SE(P) \cap SE(Q)$ such that $(W, W) \in SE(P) \cap SE(Q)$ implies $W \not\subseteq Y$ and $M = \{(X, Z) \in SE(P) \mid (X, Y) \in SE(Q) \cap SE(P), X \neq Y \text{ and } Z \subseteq Y\}$.⁸
3. $P \cup Q$ is consistent. $SE(R) = SE(P)$.

In the AGM setting, a belief set K is inconsistent with another one K' iff the (classical) models of K disjoint with those of K' . So if a belief set K'' differs minimally from K , but is consistent with K' , then the models of K'' consist of the models of K together with a single model of K' . In our non-monotonic setting, P can be inconsistent with Q even though the (SE) models of P intersect with those of Q . In this setting, there are more options for a program to be consistent with Q and such that its models differ minimally from those of P . Theorem 2 makes explicit all such options. Firstly, the set of SE models of the program could contain SE models that are not in $SE(P)$ and it does not contain all SE models of P (i.e., condition 1 of Theorem 2). Secondly, set of SE models of the program could contain only SE models in $SE(P)$ but not all of them (i.e., condition 2 of Theorem 2). In both cases, the exclusion of some SE models of P (i.e., M) is necessary for guaranteeing the program is consistent with Q .

Now we give the definition of *llp-revision* functions. As for *slp-revision* functions, a selection function γ is assumed for choosing the “best” *l-compatible* program. Then the closure of the chosen *l-compatible* program and the revising logic program is returned as the revised logic program.

⁷ If $\psi \in K$ and $\psi \notin K * \phi$, then there is some K' such that $K * \phi \subseteq K' \subseteq K \cup \{\phi\}$, K' is consistent but $K' \cup \{\psi\}$ is inconsistent.

⁸ For $(X, Z) \in M$, X may be identical to Z .

Definition 8. A function $*$: $\mathcal{P} \times \mathcal{P} \mapsto \mathcal{P}$ is a *llp-revision function* iff

$$P * Q = cl(\gamma(P \uparrow Q) \cup Q)$$

where γ is a selection function for P .

For properties of llp-revision functions, consider the following set of postulates for a function $*$: $\mathcal{P} \times \mathcal{P} \mapsto \mathcal{P}$.

- (I*cl) $P * Q = cl(P * Q)$.
- (I*s) $Q \subseteq P * Q$.
- (I*c) If Q is m-consistent, then $P * Q$ is consistent.
- (I*f) If Q is m-inconsistent, then $P * Q = cl(P \cup Q)$.
- (I*rr) If $M \neq \emptyset$ and $M \subseteq (SE(P) \cap SE(Q)) \setminus SE(P * Q)$, then $SE(P * Q) \cup M$ is inconsistent.
- (I*er) If $N \neq \emptyset$ and $N \subseteq SE(P * Q) \setminus SE(P)$, then either $SE(P * Q) \setminus N$ is inconsistent or $Cl(SE(P * Q) \setminus N) = SE(P * Q)$.
- (I*mr) If $M \neq \emptyset$, $M \subseteq (SE(P) \cap SE(Q)) \setminus SE(P * Q)$, $N \neq \emptyset$ and $N \subseteq SE(P * Q) \setminus SE(P)$, then $(SE(P * Q) \cup M) \setminus N$ is inconsistent.
- (I*u) If $P \uparrow Q = P \uparrow R$, then $SE(P * Q) \setminus SE(P) = SE(P * R) \setminus SE(P)$ and $\{\mu \in SE(P) \mid Cl(SE(P * Q) \cup \{\mu\}) \cap SE(Q) \text{ is inconsistent}\} = \{\mu \in SE(P) \mid Cl(SE(P * R) \cup \{\mu\}) \cap SE(R) \text{ is inconsistent}\}$.

(I*cl) (*Closure*) states that the revised logic program is closed. (I*s), (I*c) and (I*f) are the same as their counterparts for slp-revision function, except for (I*f) in which the revised logic program has to be closed.

Since (I*s) requires that $SE(P * Q)$ is a subset of $SE(Q)$, SE models outside of $SE(Q)$ is of no concern. Then the principle of minimal change dictates that as much as possible the intersecting models of P and Q are preserved and as least as possible the models outside of $SE(P)$ are added. (I*rr) states that if a subset M of $SE(P) \cap SE(Q)$ is not in $SE(P * Q)$, then adding M to $SE(P * Q)$ results in inconsistency. The postulate captures the intuition that no intersecting models of P and Q is excluded in the revision unless this contributes to the consistency of $P * Q$. (I*er) states that if a subset N of $SE(P * Q)$ is not in $SE(P)$, then removing N from $SE(P * Q)$ results in inconsistency or its inclusion in $SE(P * Q)$ is required for $SE(P * Q)$ to be closed under completeness. Recall that the set of SE models of any logic program has to be closed under completeness. The postulate captures the intuition that no SE models outside $SE(P)$ is included in those of $P * Q$ unless this contributes to the consistency of $P * Q$ or to the closure of the set of SE models of $P * Q$. (I*mr) is a mixture of (I*rr) and (I*er), capturing the necessity of exclusion of SE models in $SE(P) \cap SE(Q)$ and inclusion of SE models not in $SE(P)$.

Finally, (I*u) states the condition under which the two revising programs Q and R trigger the same changes to the original program in terms of SE models. That is, in accommodating Q , SE models of the revised program that are not those of P (i.e., $SE(P * Q) \setminus SE(P)$) and SE models of P that can not be SE models of the revised program (i.e., $\{\mu \in SE(P) \mid Cl(SE(P * Q) \cup \{\mu\}) \cap SE(Q) \text{ is inconsistent}\}$) are identical to those in accommodating R .

We can show that these postulates are sufficient to characterise all llp-revision functions.

Theorem 3. A function $*$ is a llp-revision function iff $*$ satisfies (I*cl), (I*s), (I*c), (I*f), (I*rr), (I*er), (I*mr), and (I*u).

5 Related Work

There has been much work on belief revision for logic programs. Delgrande et al [7] generalise Satoh's [22] and Dalal's [4] revision operators to logic programs. Significantly, they bring SE model into the picture which has inspired several approaches that are based on SE models. Slota and Leite [24, 25, 26] investigate Katsuno and Mendelzon style [14] update for logic programs. Schwind and Inoue [23] provide a constructive characterisation for the revision operators in [7]. Delgrande et al [5] adapt the model-based revision of Katsuno and Mendelzon [15] to logic programs. Finally, Binnewies et al [3] provide a variant of partial meet revision and contraction for logic programs.

Comparing with our llp-revision function which also makes use of SE models, these approaches assume a weaker notion of consistency, that is m-consistency. For this reason, some contradictions will not be dealt with in these approaches. For instance, the contradictory rule $a \leftarrow \text{not } a$ is m-consistent thus is considered to be an acceptable state of belief. Also in our teaching example, as the program consisting of rules (1) – (3) is m-consistent, no attempt will be made to resolve the contradiction about John's teaching duty by the SE model approaches. Therefore for application scenarios in which such contradictions can not be tolerant, our llp-revision function is clearly a better choice.

Apart from the SE model approaches, Krümpelmann and Kern-Isberner [16] provide a revision function for logic programs that originates from Hansson's *semi-revision* [12]. For reference we call the revision function *base revision function* as the revision they considered is belief base revision. Since they assume the same notion of consistency as ours, all the above mentioned contradictions will be resolved in their approach. As we have noted, classical belief revision is defined for monotonic setting, not for non-monotonic ones. Inconsistency can be caused by wrong assumptions in the non-monotonic setting but not in the monotonic setting. Such causes are not considered in [16]. Consequently, their approach only supports one of the many possible inconsistency-resolution strategies we have developed. Specifically, in [16], inconsistency can be resolved only by removing old beliefs; this strategy is captured by a notion analogous to s-removal compatible programs. The inconsistency-resolution strategies captured by the notion of s-expansion compatible program and s-compatible program in general are not considered.

A group of work under the title of update [2, 6, 9, 17, 20, 21, 28, 19, 29, 30] also deals with changes of logic programs. The update however is different from the Katsuno and Mendelzon style update [14]. Following [2, 17], a typical problem setting is to consider a sequence P_1, P_2, \dots, P_n of programs such that $1 \leq i < j \leq n$ implies P_i has higher priority over P_j . The goal of the update then is to obtain a set of answer sets from such a program sequence that in some sense respects the priority ordering. Clearly, these approaches have very different focus from ours which is to obtain a single new logic program.

6 Conclusion and Future Work

Depending on the application, the logic governing an agent's beliefs could be either monotonic or non-monotonic. Traditional belief revision assumes that an agent reasons monotonically; therefore, by definition, it is applicable to such situations only. Here we have aimed to study belief revision for situations in which the agent reasons non-monotonically. To this end, we defined slp-revision function and llp-revision function for disjunctive logic programs under the answer set

semantics, catering respectively for application scenarios that prioritise the preservation of the syntactic structure and that prioritise the preservation of logical content.

Inconsistency-resolution is an essential task for belief revision. However, the strategies used in traditional belief revision functions are limited to situations when the agent reasons monotonically. With a logic program we have the luxury of making assumptions via lack of contrary evidence, and we can deduce certain facts from such assumptions. Thus if a set of beliefs is inconsistent, then one possible cause is that we made the wrong assumption. In such cases, we can resolve the inconsistency by adding some new rules so that the assumption can no longer be made. Such a cause of inconsistency and the associated inconsistency-resolution strategy is beyond the scope of traditional belief revision, but is crucial for non-monotonic belief revision. We argue that this rationale, which is encoded in our belief revision functions, captures the fundamental difference between monotonic and non-monotonic belief revision.

This paper then has explored AGM-style revision and belief base revision in the non-monotonic setting of disjunctive logic programs; in future work we propose to extend this to a general approach to belief revision in arbitrary non-monotonic settings.

Acknowledgements

We would like to thank Sebastian Binnewies, Yifan Jin, and Kewen Wang for some fruitful discussions that help improve the paper.

Appendix: Proof of Results

Lemma 1. *If $X \in P \uparrow Q$, then $P \cap Q \subseteq X$ and $X \cap (Q \setminus P) = \emptyset$.*

Proof. Let $X \in P \uparrow Q$. Assume $P \cap Q \not\subseteq X$. Then $P \cap Q \neq \emptyset$. By the definition of compatible programs, we have $X \cup Q$ is consistent. Then since $X \cup Q = (X \cup (P \cap Q)) \cup Q$, $(X \cup (P \cap Q)) \cup Q$ is consistent. It then follows from $((X \cup (P \cap Q)) \ominus P) \subset (X \ominus P)$ that $X \notin P \uparrow Q$, a contradiction!

Assume $X \cap (Q \setminus P) \neq \emptyset$. Then $Q \setminus P \neq \emptyset$. Since $X \cup Q = (X \setminus (Q \setminus P)) \cup Q$, $(X \setminus (Q \setminus P)) \cup Q$ is consistent. It then follows from $(X \setminus (Q \setminus P)) \subset (X \ominus P)$ that $X \notin P \uparrow Q$, a contradiction! \square

Proof for Proposition 2

Suppose $P \cup Q$ is m-inconsistent. We need to show $P \uparrow Q = \emptyset$. Since $P \cup Q$ is m-inconsistent, we have $SE(P) \cap SE(Q) = \emptyset$. By the definition of expansion compatible program, any element in $P \uparrow Q$ has to be a superset of P and consistent with Q . However, for any superset R of P , $SE(R) \subseteq SE(P)$. Thus $SE(R) \cap SE(Q) = \emptyset$ which implies $R \cup Q$ is m-inconsistent. \square

Proof for Theorem 1

\Rightarrow : Suppose $*$ is a slp-revision function. Let γ be the selection function for P . We need to show $*$ satisfies (s*s), (s*c), (s*f), (s*rr), (s*er), (s*mr), and (s*u). (s*s), (s*c), and (s*f) follow immediately from the definition of slp-revision functions.

(s*rr): Suppose there is a program R such that $R \neq \emptyset$ and $R \subseteq P \setminus (P * Q)$. By the definition of slp-revision functions, we have $P * Q = \gamma(P \uparrow Q) \cup Q$, hence $\gamma(P \uparrow Q) \neq P$ for otherwise $R = \emptyset$. Then we have by the definition of selection functions that $P \uparrow Q \neq \emptyset$ and $\gamma(P \uparrow Q) \in P \uparrow Q$. Let $\gamma(P \uparrow Q) = X$. Then $(P * Q) \cup R = X \cup Q \cup R$. Since $R \cap X = \emptyset$, $R \neq \emptyset$, and $R \subseteq P$, we have $((X \cup R) \ominus P) \subset (X \ominus P)$. Thus we have by the

definition of compatible programs that $X \cup R \cup Q$ is inconsistent, that is $(P * Q) \cup R$ is inconsistent.

(s*er): Suppose there is a program E such that $E \neq \emptyset$ and $E \subseteq (P * Q) \setminus (P \cup Q)$. By the definition of slp-revision functions, we have $P * Q = \gamma(P \uparrow Q) \cup Q$, hence $\gamma(P \uparrow Q) \neq P$ for otherwise $E = \emptyset$. Then we have by the definition of selection functions that $P \uparrow Q \neq \emptyset$ and $\gamma(P \uparrow Q) \in P \uparrow Q$. Let $\gamma(P \uparrow Q) = X$. Then $(P * Q) \setminus E = (X \cup Q) \setminus E$. Since $E \cap P = \emptyset$, $E \neq \emptyset$, and $E \subseteq X$, we have $((X \setminus E) \ominus P) \subset (X \ominus P)$. Thus we have by the definition of compatible programs that $(X \setminus E) \cup Q$ is inconsistent. Then since $E \cap Q = \emptyset$, we have $(X \setminus E) \cup Q = (X \cup Q) \setminus E = (P * Q) \setminus E$. Thus $(P * Q) \setminus E$ is inconsistent.

(s*mr): Can be proved by combining the proving methods for (s*rr) and (s*er).

(s*u): Suppose $P \uparrow Q = P \uparrow R$. Then $\gamma(P \uparrow Q) = \gamma(P \uparrow R)$. If $P \uparrow Q = P \uparrow R = \emptyset$, then by the definition of slp-revision functions $P * Q = P \cup Q$ and $P * R = P \cup R$. Thus $P \setminus (P * Q) = P \setminus (P * R) = \emptyset$ and $(P * Q) \setminus (P \cup Q) = (P * R) \setminus (P \cup R) = \emptyset$. So suppose $P \uparrow Q \neq \emptyset$ and $P \uparrow R \neq \emptyset$. Let $X = \gamma(P \uparrow Q) = \gamma(P \uparrow R)$.

By the definition of slp-revision functions, we have $P \setminus (P * Q) = P \setminus (X \cup Q)$. By basic set theory we have $P \setminus (X \cup Q) = P \setminus (P \cap (X \cup Q)) = P \setminus ((P \cap X) \cup (P \cap Q))$. Since we have by Lemma 1 that $P \cap Q \subseteq X$, $P \setminus ((P \cap X) \cup (P \cap Q)) = P \setminus (P \cap X)$. Again by bas $P \setminus (P \cap X) = P \setminus X$. Thus $P \setminus (P * Q) = P \setminus X$. It can be shown in the same manner that $P \setminus (P * R) = P \setminus X$. Thus $P \setminus (P * Q) = P \setminus (P * R)$.

Since we have by Lemma 1 that $X \cap (Q \setminus P) = \emptyset$, it follows from basic set theory that $(X \cup Q) \setminus (P \cup Q) = X \setminus P$. This means $(P * Q) \setminus (P \cup Q) = X \setminus P$. It can be shown in the same manner that $(P * R) \setminus (P \cup R) = X \setminus P$. Thus $(P * Q) \setminus (P \cup Q) = (P * R) \setminus (P \cup R)$.

\Leftarrow : Suppose $*$: $\mathcal{P} \times \mathcal{P} \mapsto \mathcal{P}$ is a function that satisfies (s*s), (s*c), (s*f), (s*rr), (s*er), (s*mr), and (s*u). We need to show $*$ is a slp-revision function. Let γ be defined as:

$$\gamma(P \uparrow Q) = ((P * Q) \cap P) \cup ((P * Q) \setminus Q)$$

for all Q . It suffices to show $P * Q = \gamma(P \uparrow Q) \cup Q$ and γ is a selection function for P .

Part 1: By the definition of γ , we have $\gamma(P \uparrow Q) \cup Q = ((P * Q) \cap P) \cup ((P * Q) \setminus Q) \cup Q$. By basic set theory, $((P * Q) \cap P) \cup ((P * Q) \setminus Q) \cup Q = (P * Q) \cup Q$. Since $Q \subseteq P * Q$ follows from (s*s), we have $(P * Q) \cup Q = (P * Q)$. Thus $(P * Q) = \gamma(P \uparrow Q) \cup Q$.

Part 2: We first show γ is well defined. Suppose $P \uparrow Q = P \uparrow R$. Then (s*u) implies $P \setminus (P * Q) = P \setminus (P * R)$ and $(P * Q) \setminus (P \cup Q) = (P * R) \setminus (P \cup R)$. Since $P \setminus (P * Q) = P \setminus (P \cap (P * Q))$ and $P \setminus (P * R) = P \setminus (P \cap (P * R))$, $P \setminus (P * Q) = P \setminus (P * R)$ implies $(P * Q) \cap P = (P * R) \cap P$. Thus $(P * Q) \setminus (P \cup Q) = (P * R) \setminus (P \cup R)$ implies $((P * Q) \cap P) \cup ((P * Q) \setminus (P \cup Q)) = ((P * R) \cap P) \cup ((P * R) \setminus (P \cup R))$. Then by basic set theory, we have $((P * Q) \cap P) \cup ((P * Q) \setminus Q) = ((P * R) \cap P) \cup ((P * R) \setminus R)$. Thus, $\gamma(P \uparrow Q) = \gamma(P \uparrow R)$. So γ is a function.

If $P \uparrow Q = \emptyset$, then we have to show $\gamma(P \uparrow Q) = P$. $P \uparrow Q = \emptyset$ implies Q is m-inconsistent, hence it follows from (s*f) that $P * Q = P \cup Q$. Then by the definition of γ , $\gamma(P \uparrow Q) = ((P * Q) \cap P) \cup ((P * Q) \setminus Q) = ((P \cup Q) \cap P) \cup ((P \cup Q) \setminus Q) = P$.

If $P \uparrow Q \neq \emptyset$, then we have to show $\gamma(P \uparrow Q) \in P \uparrow Q$. Since $P \uparrow Q \neq \emptyset$, Q is m-consistent. Then (s*c) implies $P * Q$ is consistent. Since $P * Q = \gamma(P \uparrow Q) \cup Q$, $\gamma(P \uparrow Q) \cup Q$ is consistent. Assume there is X such that $X \cup Q$ is consistent and $X \ominus P \subset \gamma(P \uparrow Q) \ominus P$. Then we have three cases:

Case 1, $X = \gamma(P \uparrow Q) \cup R$ for $R \neq \emptyset$ and $R \subseteq P \setminus \gamma(P \uparrow Q)$:

If $R \cap Q = \emptyset$, then since $\gamma(P \uparrow Q) \cup Q = P * Q$, $R \cap (P * Q) = \emptyset$. Then it follows from (s*rr) that $(P * Q) \cup R$ is inconsistent. Since $X \cup Q = (P * Q) \cup R$, $X \cup Q$ is inconsistent, a contradiction! If $R \cap Q \neq \emptyset$, then since $R \subseteq P$, $R \cap P \cap Q \neq \emptyset$. Since (s*s) implies $Q \subseteq P * Q$, we have $Q \cap P \subseteq (P * Q) \cap P$, which implies $R \cap ((P * Q) \cap P) \neq \emptyset$. Then since $((P * Q) \cap P) \subseteq \gamma(P \uparrow Q)$, $\gamma(P \uparrow Q) \cap R \neq \emptyset$, a contradiction! Thus $R \cap Q \neq \emptyset$ is an impossible case.

Case 2, $X = \gamma(P \uparrow Q) \setminus E$ for $E \cap P = \emptyset$, $E \neq \emptyset$, and $E \subseteq \gamma(P \uparrow Q)$: Then since $\gamma(P \uparrow Q) \cup Q = P * Q$, $E \subseteq P * Q$. If $E \cap Q = \emptyset$, then (s*er) implies $(P * Q) \setminus E$ is inconsistent. Since $X \cup Q = (\gamma(P \uparrow Q) \setminus E) \cup Q = (P * Q) \setminus E$, $X \cup Q$ is inconsistent, a contradiction! If $E \cap Q \neq \emptyset$, then $E \not\subseteq (P * Q) \setminus Q$. Since $E \cap P = \emptyset$, we have $E \cap (P * Q) \cap P = \emptyset$. Thus $E \not\subseteq ((P * Q) \cap P) \cup ((P * Q) \setminus Q)$. Since $((P * Q) \cap P) \cup ((P * Q) \setminus Q) = \gamma(P \uparrow Q)$, we have $E \not\subseteq \gamma(P \uparrow Q)$, a contradiction! So $E \cap Q \neq \emptyset$ is an impossible case.

Case 3, $X = (\gamma(P \uparrow Q) \cup R) \setminus E$ for $R \neq \emptyset$, $R \subseteq P \setminus \gamma(P \uparrow Q)$, $E \cap P = \emptyset$, $E \neq \emptyset$, $E \subseteq \gamma(P \uparrow Q)$: Then we can show as for Case 1 and 2 that $R \cap P * Q = \emptyset$ and $E \subseteq P * Q$. If $R \cap Q = \emptyset$ and $E \cap Q = \emptyset$, then (s*mr) implies $((P * Q) \cup R) \setminus E$ is inconsistent. Thus $X \cup Q = ((\gamma(P \uparrow Q) \cup R) \setminus E) \cup Q = ((P * Q) \cup R) \setminus E$ is inconsistent, a contradiction! Also we can show as for Case 1 and 2 that that $R \cap Q = \emptyset$ and $E \cap Q = \emptyset$ are impossible cases. \square

Proof for Theorem 2

Suppose $R \in P \uparrow Q$. Then $R \cup Q$ is consistent and $SE(P) \oplus SE(S) \subset SE(P) \oplus SE(R)$ implies $S \cup Q$ is inconsistent. Since $R \cup Q$ is consistent, there is $(Y, Y) \in SE(R) \cap SE(Q)$ and there is no $(X, Y) \in SE(R) \cap SE(Q)$ with $X \subset Y$. We have three cases:

Case 1, $P \cup Q$ is inconsistent and $(Y, Y) \in SE(Q) \setminus SE(P)$: Let $M = \{(X, Z) \in SE(P) \mid (X, Y) \in SE(Q) \text{ and } Z \subset Y\}$ and $SE(S) = Cl((SE(P) \cup \{(Y, Y)\}) \setminus M)$. Then $(Y, Y) \in SE(S) \cap SE(Q)$. Assume there is $(X, Y) \in SE(S) \cap SE(Q)$ such that $X \subset Y$. If $(X, Y) \in SE(P)$, then $(Y, Y) \in SE(P)$, a contradiction! Thus $(X, Y) \in SE(Q) \setminus SE(P)$ which means there is $(X, Z) \in SE(S) \cap SE(P)$ such that $Z \subseteq Y$. But such (X, Z) is in M and thus $(X, Z) \notin SE(S)$, a contradiction. So there is no $(X, Y) \in SE(S) \cap SE(Q)$ such that $X \subset Y$, which implies $S \cup Q$ is consistent.

Assume $SE(R) \neq Cl((SE(P) \cup \{(Y, Y)\}) \setminus M)$. Then $SE(R) = Cl(((SE(P) \cup \{(Y, Y)\}) \setminus M) \setminus G)$, $SE(R) = Cl(((SE(P) \cup \{(Y, Y)\}) \setminus M) \cup N)$, or $SE(R) = Cl(((SE(P) \cup \{(Y, Y)\}) \setminus M) \setminus G) \cup N$ where $G \neq \emptyset$, $G \subseteq SE(P) \setminus M$, $N \neq \emptyset$, and $N \cap SE(P) = \emptyset$. Note that since $(Y, Y) \in SE(R)$, $SE(R) \cap M = \emptyset$ for otherwise $R \cup Q$ is inconsistent. For any of the possibilities, we have by basic set theory that $SE(P) \oplus SE(S) \subset SE(P) \oplus SE(R)$, which means $S \cup Q$ is inconsistent, a contradiction! So $SE(R) = Cl((SE(P) \cup \{(Y, Y)\}) \setminus M)$.

This case corresponds to condition 1.

Case 2, $P \cup Q$ is inconsistent and $(Y, Y) \in SE(Q) \cap SE(P)$: Let $M = \{(X, Z) \in SE(P) \mid (X, Y) \in SE(Q) \cap SE(P), X \neq Y \text{ and } Z \subseteq Y\}$.

Assume $SE(P) \setminus M \neq Cl(SE(P) \setminus M)$. Then there is $(W, Z) \in M$ such that $(X, Z) \in Cl(SE(P) \setminus M)$. If $W = Z$, then there is $(V, Z) \in SE(P) \setminus M$ such that $V \subset Z$. Since $(W, Z) \in M$, there is $(X, Y) \in SE(P) \cap SE(Q)$ such that $X \neq Y$ and $Z \subseteq Y$ which implies $(V, Z) \in M$, a contradiction! If $W \subset Z$, then there are $(W, V), (Z, Z) \in SE(P) \setminus M$ such that $V \subseteq Z$. Again $(W, Z) \in M$ implies there is $(X, Y) \in SE(P) \cap SE(Q)$ such that $X \neq Y$ and $Z \subseteq Y$ which implies $(W, V), (Z, Z) \in M$, a contradiction! So we have $SE(P) \setminus M = Cl(SE(P) \setminus M)$.

Let $SE(S) = SE(P) \setminus M$. $(Y, Y) \in SE(S) \cap SE(Q)$. Assume $S \cup Q$ is inconsistent. Then there is $(X, Y) \in SE(S) \cap SE(Q)$ with $X \subset Y$. But $(X, Y) \in M$, so $(X, Y) \notin SE(S)$, a contradiction! So $S \cup Q$ is consistent.

Assume $SE(R) \neq SE(P) \setminus M$. Then $SE(R) = (SE(P) \setminus M) \setminus G$, $SE(R) = (SE(P) \setminus M) \cup H$, or $SE(R) = ((SE(P) \setminus M) \setminus G) \cup H$ where $G \neq \emptyset$, $G \subseteq SE(P) \setminus M$, $H \neq \emptyset$, and $H \cap SE(P) = \emptyset$. For any of the possibilities, we have $SE(P) \oplus SE(S) \subset SE(P) \oplus SE(R)$, which means $S \cup Q$ is inconsistent, a contradiction! So $SE(R) = SE(P) \setminus M$.

Assume there is $(W, W) \in SE(P) \cap SE(Q)$ such that $W \subset Y$. Let $SE(S) = SE(P) \setminus M'$ where $M' = \{(X, Z) \in SE(P) \mid (X, W) \in SE(Q) \cap SE(P), X \neq W \text{ and } Z \subseteq W\}$. Then we can show as above, $S \cup Q$ is consistent. By the completeness of SE models, it follows from $(X, W) \in SE(Q) \cap SE(P)$ and $W \subset Y$ that $(X, Y) \in SE(Q) \cap SE(P)$. Then it is easy to see that $M' \subset M$. Thus $SE(P) \oplus SE(S) \subset SE(P) \oplus SE(R)$ which implies $S \cup Q$ is inconsistent, a contradiction! So there is no such $(W, W) \in SE(P) \cap SE(Q)$.

This case corresponds to condition 2.

Case 3, $P \cup Q$ is consistent: If $SE(R) \neq SE(P)$, then $SE(P) \oplus SE(S) \subset SE(P) \oplus SE(R)$ which implies $P \cup Q$ is inconsistent, a contradiction! Thus $SE(R) = SE(P)$.

This case corresponds to condition 3. \square

Proof for Theorem 3

\Rightarrow : Suppose $*$ is a llp-revision function. Let γ be the selection function for P . We need to show $*$ satisfies (l*cl), (l*s), (l*c), (l*f), (l*rr), (l*er), (l*mr), and (l*u). (l*cl), (l*s), (l*f), and (l*c) follow immediately from the definition of llp-revision functions.

(l*rr): Let $O \neq \emptyset$ and $O \subseteq (SE(P) \cap SE(Q)) \setminus SE(P * Q)$. We need to show $SE(P * Q) \cup O$ is inconsistent. By the definition of llp-revision functions, we have $P * Q = Cl(\gamma(P \uparrow Q) \cup Q)$. Let $\gamma(P \uparrow Q) = X$. Then we have $SE(P * Q) = SE(X) \cap SE(Q)$. According to Theorem 2, there are three cases:

Case 1, $P \cup Q$ is inconsistent and $SE(X) = Cl((SE(P) \cup \{(Y, Y)\}) \setminus M)$ for $M = \{(X, Z) \in SE(P) \mid (X, Y) \in SE(Q) \text{ and } Z \subseteq Y\}$: Then we have $SE(P * Q) = Cl((SE(P) \cup \{(Y, Y)\}) \setminus M) \cap SE(Q)$. Since $O \subseteq (SE(P) \cap SE(Q)) \setminus SE(P * Q)$, we have $O \subseteq M$. So there is $(X, Y) \in Cl((SE(P * Q) \cup O)$ such that $X \subset Y$ which means (Y, Y) is no longer an answer set in $Cl(SE(P * Q) \cup O)$ and $Cl(SE(P * Q) \cup O)$ is inconsistent.

Case 2, $P \cup Q$ is inconsistent but m-consistent and $SE(X) = SE(P) \setminus M$ for there is $(Y, Y) \in SE(P) \cap SE(Q)$ such that $(W, W) \in SE(P) \cap SE(Q)$ implies $W \not\subset Y$ and $M = \{(X, Z) \in SE(P) \mid (X, Y) \in SE(Q) \cap SE(P), X \neq Y \text{ and } Z \subseteq Y\}$: Then $SE(P * Q) = (SE(P) \cap SE(Q)) \setminus M$. Since $O \subseteq (SE(P) \cap SE(Q)) \setminus SE(P * Q)$, we have $O \subseteq M$. So there is $(X, Y) \in Cl(SE(P * Q) \cup O)$ such that $X \subset Y$ which means (Y, Y) is no longer an answer set in $Cl(SE(P * Q) \cup O)$ and $Cl(SE(P * Q) \cup O)$ is inconsistent.

Case 3, $P \cup Q$ is consistent and $SE(P * Q) = SE(P) \cap SE(Q)$: Then $SE(X) = SE(P)$ and $SE(P * Q) = SE(P) \cap SE(Q)$ which means $O = \emptyset$, contradiction! So this is an impossible case.

(l*er): Let $O \neq \emptyset$ and $O \subseteq SE(P * Q) \setminus SE(P)$. We need to show $SE(P * Q) \setminus O$ is inconsistent or $Cl(SE(P * Q) \setminus O) = SE(P * Q)$. Let $\gamma(P \uparrow Q) = X$. Then we have $SE(P * Q) = SE(X) \cap SE(Q)$. According to Theorem 2, there are three cases:

Case 1: As for (l*rr) we have $SE(P * Q) = Cl((SE(P) \cap SE(Q)) \setminus M \cup \{(Y, Y)\})$. If $(Y, Y) \in O$, then $SE(P * Q) \setminus O$ no longer contains any answer set and $SE(P * Q) \setminus O$ is inconsis-

tent. If $(Y, Y) \notin O$, then since $O \cap SE(P) \cap SE(Q) = \emptyset$, we have $Cl(SE(P * Q) \setminus O) = SE(P * Q)$.

Case 2: As for (I*rr) we have $SE(P * Q) = (SE(P) \cap SE(Q)) \setminus M$ which means $O = \emptyset$, a contradiction! So this is an impossible case.

Case 3: As for (I*rr) we have $SE(P * Q) = SE(P) \cap SE(Q)$ which means $O = \emptyset$, a contradiction! So this is an impossible case.

(I*mr): Can be proved by combining the proving methods for (I*rr) and (I*er).

(I*u): Suppose $P \uparrow Q = P \uparrow R$. We need to show $SE(P * Q) \setminus SE(P) = SE(P * R) \setminus SE(P)$ and $\{\mu \in SE(P) \mid Cl(SE(P * Q) \cup \{\mu\}) \cap SE(Q) \text{ is inconsistent}\} = \{\mu \in SE(P) \mid Cl(SE(P * R) \cup \{\mu\}) \cap SE(Q) \text{ is inconsistent}\}$. Since $P \uparrow Q = P \uparrow R$, we have $\gamma(P \uparrow Q) = \gamma(P \uparrow R)$. Let $\gamma(P \uparrow Q) = \gamma(P \uparrow R) = S$.

Part 1. By the definition of llp-revision functions, we have $SE(P * Q) \setminus SE(P) = (SE(S) \cap SE(Q)) \setminus SE(P)$ and $SE(P * R) \setminus SE(P) = (SE(S) \cap SE(R)) \setminus SE(P)$. If $SE(S) = SE(P)$ or $SE(S) \subseteq SE(P)$, then $SE(P * Q) \setminus SE(P) = SE(P * R) \setminus SE(P) = \emptyset$. So it remains to consider the case when $SE(S) \setminus SE(P) \neq \emptyset$. By Theorem 2, we have $SE(S) = Cl((SE(P) \cup \{(Y, Y)\}) \setminus M)$ where $(Y, Y) \in SE(Q) \setminus SE(P)$ and $M = \{(X, Z) \in SE(P) \mid (X, Y) \in SE(Q) \text{ and } Z \subseteq Y\}$. Thus $(SE(S) \cap SE(Q)) \setminus SE(P) = \{(Y, Y)\}$. Since we also have $SE(S) = Cl((SE(P) \cup \{(Y, Y)\}) \setminus M)$ where $(Y, Y) \in SE(R) \setminus SE(P)$ and $M = \{(X, Z) \in SE(P) \mid (X, Y) \in SE(R) \text{ and } Z \subseteq Y\}$, $(SE(S) \cap SE(R)) \setminus SE(P) = \{(Y, Y)\}$. Thus $SE(P * Q) \setminus SE(P) = SE(P * R) \setminus SE(P)$.

Part 2. By Theorem 2, there are three cases:

Case 1, $SE(S) = Cl((SE(P) \cup \{(Y, Y)\}) \setminus M)$ where $(Y, Y) \in SE(Q) \setminus SE(P)$ and $M = \{(X, Z) \in SE(P) \mid (X, Y) \in SE(Q) \text{ and } Z \subseteq Y\}$: Then we also have $SE(S) = Cl((SE(P) \cup \{(Y, Y)\}) \setminus N)$ where $(Y, Y) \in SE(R) \setminus SE(P)$ and $N = \{(X, Z) \in SE(P) \mid (X, Y) \in SE(R) \text{ and } Z \subseteq Y\}$. Thus $M = N$. Since $\{\mu \in SE(P) \mid Cl(SE(P * Q) \cup \{\mu\}) \cap SE(Q) \text{ is inconsistent}\} = M$ and $\{\mu \in SE(P) \mid Cl(SE(P * R) \cup \{\mu\}) \cap SE(R) \text{ is inconsistent}\} = N$, $\{\mu \in SE(P) \mid Cl(SE(P * Q) \cup \{\mu\}) \cap SE(Q) \text{ is inconsistent}\} = \{\mu \in SE(P) \mid Cl(SE(P * R) \cup \{\mu\}) \cap SE(R) \text{ is inconsistent}\}$.

Case 2, $SE(S) = SE(P) \setminus M$ where there is $(Y, Y) \in SE(P) \cap SE(Q)$ such that $(W, W) \in SE(P) \cap SE(Q)$ implies $W \not\subseteq Y$ and $M = \{(X, Z) \in SE(P) \mid (X, Y) \in SE(Q) \cap SE(P), X \neq Y \text{ and } Z \subseteq Y\}$: As for case 1, we have $\{\mu \in SE(P) \mid Cl(SE(P * Q) \cup \{\mu\}) \cap SE(Q) \text{ is inconsistent}\} = \{\mu \in SE(P) \mid Cl(SE(P * R) \cup \{\mu\}) \cap SE(R) \text{ is inconsistent}\} = M$.

Case 3, $SE(S) = SE(P)$: Then $\{\mu \in SE(P) \mid Cl(SE(P * Q) \cup \{\mu\}) \cap SE(Q) \text{ is inconsistent}\} = \{\mu \in SE(P) \mid Cl(SE(P * R) \cup \{\mu\}) \cap SE(R) \text{ is inconsistent}\} = \emptyset$.

⇐: Suppose $*$ is a function that satisfies (I*cl), (I*s), (I*c), (I*f), (I*rr), (I*er), (I*mr), and (I*u). We need to show $*$ is a llp-revision function.

Let γ be defined as:

$$SE(\gamma(P \uparrow Q)) = Cl((SE(P) \cup SE(P * Q)) \setminus M)$$

for all Q , where $M = \{\mu \in SE(P) \mid Cl(SE(P * Q) \cup \{\mu\}) \cap SE(Q) \text{ is inconsistent}\}$. It remains to show $P * Q = cl(\gamma(P \uparrow Q) \cup Q)$ and γ is a selection function for P .

Part 1: Due to (I*cl), it suffices to show $SE(P * Q) = SE(\gamma(P \uparrow Q) \cup Q)$.

⊆: It follows from (I*s) that $SE(P * Q) \subseteq SE(Q)$, hence it suffices to show $SE(P * Q) \subseteq SE(\gamma(P \uparrow Q))$, that is $SE(P * Q) \subseteq Cl((SE(P) \cup SE(P * Q)) \setminus M)$. Let $\mu \in SE(P * Q)$.

Then $Cl(SE(P * Q) \cup \{\mu\}) \cap SE(Q) = SE(Q) \cap SE(P * Q) = SE(P * Q)$. It follows from (I*c) that $SE(P * Q)$ is consistent, hence $Cl(SE(P * Q) \cup \{\mu\}) \cap SE(Q)$ is consistent. This means $\mu \notin M$, hence $\mu \in Cl((SE(P) \cup SE(P * Q)) \setminus M)$.

⊇: Assume there is $\mu \in SE(\gamma(P \uparrow Q)) \cap SE(Q)$ such that $\mu \notin SE(P * Q)$. Then $\mu \in Cl((SE(P) \cup SE(P * Q)) \setminus M) \cap SE(Q)$. It follows from $\mu \notin SE(P * Q)$ that $\mu \in SE(P) \setminus M$. Then it follows from $\mu \in SE(P) \cap SE(Q)$, $\mu \notin SE(P * Q)$, and (I*rr) that $SE(P * Q) \cup \{\mu\}$ is inconsistent. Since $SE(P * Q) \subseteq SE(Q)$ and $\mu \in SE(Q)$, we have $SE(Q) \cap Cl(SE(P * Q) \cup \{\mu\}) = Cl(SE(P * Q) \cup \{\mu\})$. Thus $SE(Q) \cap Cl(SE(P * Q) \cup \{\mu\})$ is inconsistent, which means $\mu \in M$, a contradiction.

Part 2: For γ to be a selection function, it must be well-defined. Suppose $P \uparrow Q = P \uparrow R$, we need to show $\gamma(P \uparrow Q) = \gamma(P \uparrow R)$, that is $Cl((SE(P) \cup SE(P * Q)) \setminus M) = Cl((SE(P) \cup SE(P * R)) \setminus N)$ for $M = \{\mu \in SE(P) \mid SE(Q) \cap Cl(SE(P * Q) \cup \{\mu\}) \text{ is inconsistent}\}$ and $N = \{\mu \in SE(P) \mid SE(R) \cap Cl(SE(P * R) \cup \{\mu\}) \text{ is inconsistent}\}$. This follows immediately from (I*u). It remains to show $\gamma(P \uparrow Q) \in P \uparrow Q$.

It has been shown in Part 1 that $SE(P * Q) = SE(\gamma(P \uparrow Q)) \cap SE(Q)$. It follows from (I*c) that $SE(P * Q)$ is consistent, which means $SE(\gamma(P \uparrow Q)) \cap SE(Q)$ is consistent. Let $\gamma(P \uparrow Q) = R$ and $SE(P) \ominus SE(X) \subset SE(P) \ominus SE(R)$. We need to show $SE(X) \cap SE(Q)$ is inconsistent. There are three cases:

Case 1, $SE(X) = SE(R) \cup S$ for $S \subseteq SE(P)$ and $S \cap SE(R) = \emptyset$: Since $R = Cl((SE(P) \cup SE(P * Q)) \setminus M)$, we have $S \subseteq M$ which implies $SE(Q) \cap Cl(SE(P * Q) \cup S)$ is inconsistent. Since $SE(P * Q) = SE(R) \cap SE(Q)$, we have $SE(Q) \cap Cl(SE(P * Q) \cup S) = SE(Q) \cap Cl((SE(R) \cap SE(Q)) \cup S) = (SE(Q) \cap SE(R)) \cup (SE(Q) \cap S) = (SE(R) \cup S) \cap SE(Q) = SE(X) \cap SE(Q)$ is inconsistent.

Case 2, $SE(X) = SE(R) \setminus T$ for $T \subseteq SE(R)$ and $T \cap SE(P) = \emptyset$: Since $SE(R) = Cl((SE(P) \cup SE(P * Q)) \setminus M)$, we have $T \subseteq Cl((SE(P) \cup SE(P * Q)) \setminus M)$. Let $T \cap SE(P * Q) = N$. Note that $N \neq \emptyset$ for otherwise $SE(X) \neq Cl(SE(X))$. Then $SE(X) \cap SE(Q) = (SE(R) \setminus T) \cap SE(Q) = (SE(R) \cap SE(Q)) \setminus T = SE(P * Q) \setminus T = SE(P * Q) \setminus N$. It follows from $N \subseteq SE(P * Q)$, $N \cap SE(P) = \emptyset$, and (I*er) that $SE(P * Q) \setminus N$ is inconsistent.

Case 3, $SE(X) = (SE(R) \cup S) \setminus T$ for $S \subseteq SE(P)$, $S \cap SE(R) = \emptyset$, $T \subseteq SE(R)$ and $T \cap SE(P) = \emptyset$: Then from Case 1 we have $S \subseteq M$. Let $N = T \cap SE(P * Q)$. If $N = \emptyset$, then the $SE(X) = SE(R) \cup S$ and this situation has been taken care in Case 1. So suppose $N \neq \emptyset$. Now $SE(X) \cap SE(Q) = ((SE(R) \cup S) \setminus N) \cap SE(Q) = ((SE(R) \cap SE(Q)) \cup (S \cap SE(Q))) \setminus N$. If $S \cap SE(Q) = \emptyset$, then $SE(X) \cap SE(Q) = ((SE(R) \cap SE(Q)) \cup (S \cap SE(Q))) \setminus N = SE(P * Q) \setminus N$ and it follows from (I*er) that $SE(X) \cap SE(Q)$ is inconsistent. If $S \cap SE(Q) \neq \emptyset$, then it follows from (I*mr) that $SE(X) \cap SE(Q)$ is inconsistent. □

REFERENCES

- [1] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson, 'On the logic of theory change: Partial meet contraction and revision functions', *The Journal of Symbolic Logic*, **50**(2), 510–530, (1985).
- [2] José Júlio Alferes, João Alexandre Leite, Luís Moniz Pereira, Halina Przymusińska, and Teodor C. Przymusiński, 'Dynamic updates of non-monotonic knowledge bases', *J. Log. Program.*, **45**(1-3), 43–70, (2000).
- [3] Sebastian Binnewies, Zhiqiang Zhuang, and Kewen Wang, 'Partial meet revision and contraction in logic programs', in *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI-2015)*, pp. 1439–1445, (2015).
- [4] Mukesh Dalal, 'Investigations into a theory of knowledge base revision', in *Proceedings of the 7th National Conference on Artificial Intelligence (AAAI-1988)*, pp. 475–479, (1988).
- [5] James P. Delgrande, Pavlos Peppas, and Stefan Woltran, 'AGM-style belief revision of logic programs under answer set semantics', in *Proceedings of the 12th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-2013)*, pp. 264–276, (2013).
- [6] James P. Delgrande, Torsten Schaub, and Hans Tompits, 'A preference-based framework for updating logic programs', in *Proceedings of the 9th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-2007)*, pp. 71–83, (2007).
- [7] James P. Delgrande, Torsten Schaub, Hans Tompits, and Stefan Woltran, 'A model-theoretic approach to belief change in answer set programming', *ACM Trans. Comput. Log.*, **14**(2), (2013).
- [8] Thomas Eiter, Michael Fink, Jörg Pührer, Hans Tompits, and Stefan Woltran, 'Model-based recasting in answer-set programming', *Journal of Applied Non-Classical Logics*, **23**(1-2), 75–104, (2013).
- [9] Thomas Eiter, Michael Fink, Giuliana Sabbatini, and Hans Tompits, 'On properties of update sequences based on causal rejection', *TPLP*, **2**(6), 711–767, (2002).
- [10] Thomas Eiter, Hans Tompits, and Stefan Woltran, 'On solution correspondences in answer-set programming', in *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-2015)*, pp. 463–498, (2005).
- [11] Sven Ove Hansson, 'Reversing the Levi Identity', *Journal of Philosophical Logic*, **22**(6), 637–669, (1993).
- [12] Sven Ove Hansson, 'Semi-revision', *Journal of Applied Non-Classical Logics*, **7**(1-2), 151–175, (1997).
- [13] Sven Ove Hansson, *A Textbook of Belief Dynamics: Theory Change and Database Updating*, Kluwer, 1999.
- [14] Hirofumi Katsuno and Alberto O. Mendelzon, 'On the difference between updating a knowledge base and revising it', in *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR-1991)*, pp. 387–394, (1991).
- [15] Hirofumi Katsuno and Alberto O. Mendelzon, 'Propositional knowledge base revision and minimal change', *Artificial Intelligence*, **52**(3), 263–294, (1992).
- [16] Patrick Krümpelmann and Gabriele Kern-Isberner, 'Belief base change operations for answer set programming', in *Logics in Artificial Intelligence - 13th European Conference, JELIA 2012, Toulouse, France, September 26-28, 2012. Proceedings*, pp. 294–306, (2012).
- [17] João Alexandre Leite, *Evolving Knowledge Bases: Specifications and Semantics*, IOS Press, 2003.
- [18] Vladimir Lifschitz, David Pearce, and Agustín Valverde, 'Strongly equivalent logic programs', *ACM Trans. Comput. Log.*, **2**(4), 526–541, (2001).
- [19] Mauricio Osorio and Víctor Cuevas, 'Updates in answer set programming: An approach based on basic structural properties', *TPLP*, **7**(4), 451–479, (2007).
- [20] Teodor C. Przymusiński and Hudson Turner, 'Update by means of inference rules', *J. Log. Program.*, **30**(2), 125–143, (1997).
- [21] Chiaki Sakama and Katsumi Inoue, 'An abductive framework for computing knowledge base updates', *TPLP*, **3**(6), 671–713, (2003).
- [22] Ken Satoh, 'Nonmonotonic reasoning by minimal belief revision', in *Proceedings of the International Conference on Fifth Generation Computer Systems*, pp. 455–462, (1988).
- [23] Nicolas Schwind and Katsumi Inoue, 'Characterization theorems for revision of logic programs', in *Proceedings of the 12th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR-2013)*, pp. 485–498, (2013).
- [24] Martin Slota and João Leite, 'On semantic update operators for answer-set programs', in *Proceedings of 19th European Conference on Artificial Intelligence (ECAI-2010)*, pp. 957–962, (2010).
- [25] Martin Slota and João Leite, 'Robust equivalence models for semantic updates of answer-set programs', in *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR-2012)*, pp. 158–168, (2012).
- [26] Martin Slota and João Leite, 'The rise and fall of semantic rule updates based on se-models', *Theory and Practice of Logic Programming*, **14**(6), 869–907, (2014).
- [27] Hudson Turner, 'Strong equivalence made easy: Nested expressions and weight constraints', *Theory Pract. Log. Program.*, **3**(4), 609–622, (2003).
- [28] Fernando Zacañas, Mauricio Osorio, Acosta Guadarrama, and Jürgen Dix, 'Updates in answer set programming based on structural properties', in *Proceedings of the 7th International Symposium on Logical Formalizations of Commonsense Reasoning*, pp. 213–219, (2005).
- [29] Yan Zhang and Norman Y. Foo, 'Towards generalized rule-based updates', in *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI-1997)* 97, pp. 82–88, (1997).
- [30] Yan Zhang and Norman Y. Foo, 'Updating logic programs', in *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-1998)*, pp. 403–407, (1998).

Value Based Reasoning and the Actions of Others

Katie Atkinson and Trevor Bench-Capon¹

Abstract. Practical reasoning, reasoning about what actions should be chosen, is highly dependent both on the individual values of the agent concerned and on what others choose to do. We discuss how value based argumentation about what to do can be performed without making assumptions about the preferences of the other agents. We then show how expected utility calculations relate to the value-based argumentation approach, and express the reasoning as arguments and objections, so that they can be integrated value-based practical reasoning. We illustrate our discussion with examples of value based reasoning in public goods games as used in experimental economics and present an initial evaluation of the approach in terms of these experiments.

1 Introduction

A key difference between theoretical reasoning (reasoning about what is the case) and practical reasoning [35] (reasoning about what to do) is the *direction of fit* [36]. Whereas in theoretical reasoning an agent is trying to fit its beliefs to the world, in practical reasoning an agent is choosing an action intended to fit the world to its desires. For theoretical reasoning, there is only one, shared, world, and so agents should tend to agree, but desires will legitimately differ from agent to agent and so practical reasoning depends on the subjective aspirations and desires of the *individual* agent. Agents may even be in conflict, so that they attempt to bring about different worlds. The conclusions are therefore legitimately subjective, and disagreement is both rational and to be expected. Acceptance of an argument as to what to do depends not only on the argument itself - for it must, of course, be a sound argument - but also on the audience to which it is addressed [33]. This notion of audience was computationally modelled in [21] and made more formal in Value-Based Argumentation Frameworks (VAFs) [9]. VAFs are an extension of the abstract Argumentation Frameworks (AFs) introduced in the seminal paper of Dung [13]. In a VAF arguments are associated with the social (i.e. *not* numeric) values² their acceptance promotes or demotes. Different audiences can now be characterised by the ordering they place on these values. Whereas in an AF an argument is defeated by an attacking argument, in a VAF an argument is *defeated for an audience* by an attacker only if the value associated with the attacking argument is ranked at least as highly by that audience. In this way different audiences will accept different sets of arguments (preferred semantics [13] is used to determine acceptance), and, as is shown in [9], provided the VAF contains no cycles in the same value, there will be a unique non-empty preferred extension. Thus, use of VAFs provides a way of explaining (and computing) the different arguments

accepted by different audiences. Value Based Reasoning has been used as the basis of practical reasoning ([19], [2], [23], [15], [41], [12]) and applied in particular areas such as law ([7], [26], [20]), e-democracy ([11], [45]), policy analysis ([38]), medicine, ([4]), experimental economics ([8]), rule compliance ([10]), decision support ([28]) and even ontology alignment ([39], [32]). Complexity results for VAFs were established in [14] and [29].

1.1 An Argumentation Scheme for Value-Based Practical Reasoning

The application of the preferences of an audience, expressed as an ordering on values, to practical reasoning requires the generation of the arguments and identification of the values associated with them. The proposal made in [3] was to use an argumentation scheme (now included in the compendium of argumentation schemes collected in [44]) justifying an action in terms of the values it promotes. The scheme appears in [3] as:

In the current circumstances R , I should perform action A , to bring about new circumstances S , which will achieve goal G and promote value V .

We will henceforth refer to this scheme as *Practical Reasoning Argumentation Scheme* (PRAS). Like all argumentation schemes, PRAS establishes its conclusion only presumptively [42] and can be challenged using what [42] and [44] call *critical questions*. Thus an argument using PRAS can be challenged by claims against its soundness such as: that the current state is different, that the action is not possible, that the action will reach a different state, fail to achieve its goal or fail to promote its value. It can also be challenged on the basis of the desirability of the action: that it will also demote values and these values are more important, or that alternative actions promote values that are more important. This second group of objections is what gives room for subjectivity arising from different value orderings so that, as Searle puts it in [36]:

Assume universally valid and accepted standards of rationality, assume perfectly rational agents operating with perfect information, and you will find that rational disagreement will still occur; because, for example, the rational agents are likely to have different and inconsistent values and interests, each of which may be rationally acceptable

In [2] seventeen different critical questions were identified that could give rise to objections to, and counter-arguments against, instantiations of PRAS.

1.2 Computational Realisation of this Scheme

In order to make this approach computable, it is necessary to provide an underlying representation of the world and how it can be

¹ Department of Computer Science, University of Liverpool, email: tbc@csc.liv.ac.uk

² *Values* are the aspirations or the purposes an agent might pursue, such as liberty, equality, fraternity, wealth, health and happiness.

affected by the actions of agents. State Transition Diagrams (STDs) are a natural choice for this, since they can represent the world as a set of states, and actions as the transitions between them. In open agent systems, however, the outcome of an action may well depend on what the other agents in the situation choose to do. Thus an individual's choice does not necessarily determine the state that will be reached. To account for this, open agent systems should model transitions as the *joint actions* composed of the the individual actions of all the agents relevant to the situation³. A suitable variant of STDs for use in open agent systems is *Action-based Alternating Transition Systems* (AATS), introduced in [46], which have joint actions as their transitions. AATS are formally based on Alternating-time temporal logic [1]. The basic AATS was augmented in [2] to allow the labelling of the transitions with the values promoted and demoted by that transition (AATS+V) and AATS+Vs were used to provide the underpinning semantical structure for the approach to practical reasoning set out in that paper⁴. Given a representation of the problem situation as an AATS+V, the discovery of arguments, counter arguments and objections can be implemented in the manner of [47]. A database containing tables for the states, joint actions and transitions of the AATS+V is created to hold the problem information and then instantiations of PRAS and challenges to those instantiations can be found by fairly simple queries to that database. For example there will be an instantiation of PRAS if there is a transition from the current state which promotes a value.

Three stages in practical reasoning are identified in [2]:

- **Problem formulation:** essentially the construction of an AATS+V for the particular problem situation;
- **Epistemic stage:** this involves determination of the current state and the joint action that will result from the choice of a particular individual action by the agent concerned;
- **Option selection:** the arguments generated from the AATS+V are formed into a VAF and their acceptability status determined according to the preferences of the agent concerned.

While problem formulation and the identification of the current state can be resolved using normal theoretical reasoning techniques, and the option selection stage is carried out using value-based reasoning base on VAFs, the determination of the joint action is less clear and will be the topic of this paper. The essential problem is that in order to know what it is best to do, it is necessary to anticipate what the other agents will do, since this will critically affect what results from our own actions. But since this reasoning will depend on the beliefs, aspirations and preferences of these other agents, this will require a number of assumptions which are often difficult to justify to be made. For example, agents which adopt the naive approach of assuming that others will be like themselves, tend to perform badly in practice [17].

1.3 Modelling the Values of Others

One approach, common in classical economics, is to see agents as consistently rational and narrowly self-interested agents who usually pursue their subjectively-defined ends optimally. John Stuart Mill [25] put it thus when describing "economic man" (sometimes called *homo economicus*):

³ This is an important difference from classic planning systems such as STRIPS[18].

⁴ To aid readability, formal definitions are collected as an Appendix at the end of the paper. AATS+V are defined in Definitions 1 and 2.

[Economics] is concerned with him solely as a being who desires to possess wealth, and who is capable of judging the comparative efficacy of means for obtaining that end.

Game Theory [27] also takes a single measure of utility expressed as a payoff matrix, which has become a very widespread basis for the design of multi-agent agent systems [31]. This approach has led to some insights, and provided the foundation for much elegant mathematics, but unfortunately does not provide a satisfactory explanation of the way in which humans behave in practice. And of course, if we are deciding what to do, we much cannot expect others to behave as they *should*, so even if this was a good *normative* theory, we would still need an adequate *descriptive* theory.

That others cannot be seen in this way is well demonstrated by a number of experiments carried out in behavioural economics. These experiments are carried out, using a variety of public goods games, to test the theory that behaviour can be predicted using the assumptions of classical economics and game theory. There are valuable meta studies, in particular for the Dictator Game [16] and the Ultimatum Game [30] and [22]. The findings suggest that the canonical model is followed only very rarely. Thus in [22] we read:

in addition to their own material payoffs, many experimental subjects appear to care about fairness and reciprocity, are willing to change the distribution of material outcomes at personal cost, and are willing to reward those who act in a cooperative manner while punishing those who do not even when these actions are costly to the individual

Even in the Prisoner's Dilemma [34], where defection is clearly the dominant strategy, we find a tendency to deviate from it [6]. In [40], the emergence of norms and conventions is discussed in terms of the Prisoner's Dilemma, and some of the other characteristics influencing behaviour, such as empathy, trust and *esprit de corps* are cited as ways in which these norms can be formed. The role of punishment is explored in [24]. What all these meta studies show is

- The canonical model used in classical economics, game theory and many multi agent systems is not adequate to explain the behaviour encountered in experimental studies;
- There is a significant amount of inter-cultural variation, suggesting that the established values of subjects is carried forward into these experiments;
- There is also a significant amount of intra-cultural variation, suggesting that the behaviour of individuals cannot reliably be predicted solely on the basis of their cultural background.

Our view is that by putting the subjective ordering of values to the fore, value based reasoning can provide a fruitful way of exploring these issues. This was borne out by the examination of the Dictator and Ultimatum games in [8]. There, however, like all approaches based on [2], the reasoning about what others would do relied too heavily on unjustifiable assumptions about the values they would use, and how they would order them. Our objectives in this paper are threefold:

- to take account of the actions of others in the framework of value-based practical reasoning without requiring assumptions about the beliefs and preferences of other agents;
- to do so in a manner compatible with the results of game theory and multi-criteria utility (e.g., [37], [41]) while explicitly allowing for subjectivity and altruism;
- to be able to express the reasoning in the form of arguments and objections so as to facilitate integration with value-based practical reasoning.

2 The Games

In this section we describe two games used in experimental economics. We will not consider the Dictator Game here, because although as shown in [16] and [8] it is amenable to analysis in terms of value-based reasoning, there is only one decision maker, and so the need to anticipate the actions of others, which is the aspect in which we are interested here, does not arise. We will therefore only consider the Ultimatum Game and the Prisoner’s Dilemma in this paper.

2.1 The Ultimatum Game

In the Ultimatum Game the first player is given a sum of money and told that he may offer some of it to the second player. Once the proposer has made an offer the respondent may choose to accept the offer, or reject it, in which case both players receive nothing. Whereas traditional game theory would suggest that the proposer would make the smallest offer possible and the respondent would accept it, experiments do not support this. The meta-analysis of 37 papers reported in [30] found that

that on average the proposer offers 40% of the pie to the responder. ... On average 16% of the offers is rejected. ... We find differences in behavior of responders (and not of proposers) across geographical regions.

It may well be that regions (at least at the country or even continent level used in [30]) do not provide the best explanation for different behaviours, being themselves large and often culturally heterogeneous. Another study [22], based on small-scale, homogeneous societies, found the different cultures more predictive:

Among the Achuar, Ache and Tsimane, we observe zero rejections after 16, 51, and 70 proposer offers, respectively. Moreover, while the Ache and Achuar made fairly equitable offers, nearly 50 percent of Tsimane offers were at or below 30 percent, yet all were accepted. Similarly, Machiguenga responders rejected only one offer, despite the fact that over 75 percent of their offers were below 30 percent. At the other end of the rejection scale, Hadza responders rejected 24 percent of all proposer offers and 43 percent of offers at 20 percent and below. Unlike the Hadza, who preferentially rejected low offers, the Au and Gnaou of Papua New Guinea rejected both unfair and hyper-fair (greater than 50 percent)

Two aspects of the societies concerned, namely the amount of cooperation found in the general economic activity of the society and the extent to which market exchanges were a feature of daily life, were found to be explanatory in [22]

the Machiguenga and Tsimane rank the lowest; they are almost entirely economically independent at the family level and engage rarely in productive activities involving more than members of a family. By contrast, the Lamelara whale-hunters go to sea in large canoes manned by a dozen or more individuals. ... The Machiguenga show the lowest cooperation rates in public-good games, reflecting ethnographic descriptions of Machiguenga life, which report little cooperation, exchange, or sharing beyond the family unit.

In contrast, the Lamelara have the highest mean offer (58%) and a zero rejection rate. As shown in [8], this can be explained by differing values and preferences amongst the participants, with the ordering emerging from their everyday activities being applied in the games. The game was analysed in [8], with the following six values:

- Proposer’s Money (M1): Promoted by acceptance of an offer to a degree inversely related to the size of the offer and demoted if the offer is rejected;
- Respondent’s Money (M2): Promoted by acceptance of an offer, to a degree related to the size of the offer;
- Generosity (G): Promoted for the proposer by giving away a reasonable amount of money;
- Equality (E): Promoted by both participants receiving the same amount;
- Proposer’s Contentment (C1): Promoted by the acceptance of a low offer (did not offer too much) and demoted by the rejection of a low offer (did not offer enough), or by the rejection of a good offer, since the respondent would be considered unreasonable;
- Respondent’s Contentment (C2): Promoted by accepting a good offer and demoted by accepting a low offer.

The transition diagram for the Ultimatum Game used in [8] is given in Figure 1. This considers the actions as happening serially, so that the joint actions have two stages. Whilst this makes the interaction, where values are promoted and demoted, more explicit, here we prefer to combine the actions, The proposer may make a very high (vho) offer (more than 50%), an equal (eo) offer (=50%), a fair (fo) offer (40-50%), or a low (lo) offer (less than 40%). The respondent may accept or reject, giving 8 joint actions. j_1 is {vho,accept}, j_2 is {vho,reject} and so on. The AATS state records the money for each participant, and two flags, indicating whether the participants are content. Most important are the values promoted and demoted by the joint actions. These are shown in Table 1.

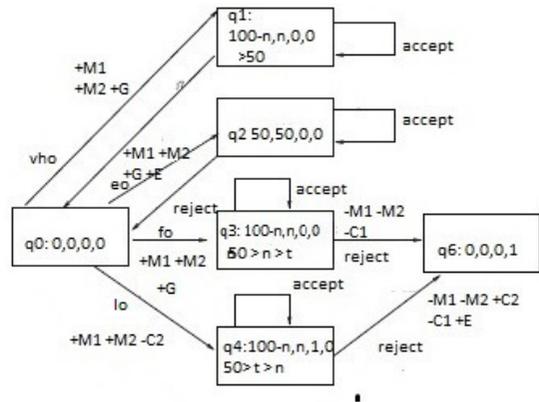


Figure 1. AATS for Ultimatum Game from [8]

Table 1. Value Promotion and Demotion in the Ultimatum Game

Joint Action	Proposal	Response	Promoted	Demoted
j1	vho	accept	M1,M2,G, C2	E
j2	vho	reject	G	M1
j3	eo	accept	M1,M2,G,C2	
j4	eo	reject	G	M1
j5	fo	accept	M1,M2	E
j6	fo	reject		M1
j7	lo	accept	M1,M2,C1	E,C2
j8	lo	reject		M1,C1

2.2 Prisoner's Dilemma

In this very well known game [34], widely used in discussions of norm emergence such as [40] and [6], both players may either cooperate or defect. Mutual cooperation results in a pay off of 3 to each player, mutual defection a payoff of 1 to each player, and if one cooperates and the other defects the defector receives 5 and the cooperator receives 0. The "correct" strategy is to defect since that gives a better payoff whichever move the other makes (is the *dominant* strategy). Also it is not a zero-sum game: collective utility is maximised by mutual cooperation. Here too, experiments find that the game-theoretic choice is not always made in practice. As explained in [40] conventions to encourage mutual cooperation often emerge or are devised. An example used in [40] is a military situation where much effort is made to build up trust and loyalty to create an *esprit de corp* in a regiment so that members will cooperate rather than defect, feeling that they are able to rely on their comrades, and in turn reluctant to let their comrades down. The conventions are often reinforced by punishing defectors [24]. Again there seem to be additional values considered by participants. Here we use the following values:

- *Player Money (M1 and M2)*: promoted if a player's payoff is greater than 1 (which is the least that can be ensured), and demoted if it is less than 1.
- *Player Guilt (G1 and G2)*: demoted if player defects and the other player cooperates
- *Player Self-Esteem (S1 and S2)*: demoted if player 1 (or 2) cooperates and player 2 (or 1) defects: since the player may feel that they should have known better.

In this game there are four joint actions which promote and demote values as shown in Table 2. Note that mutual defection provides a baseline, neither promoting nor demoting any values, since it can always be achieved or bettered.

Table 2. Value Promotion and Demotion in the Prisoner's Dilemma

Joint Action	Player 1	Player 2	Promoted	Demoted
j1	C	C	M1,M2	
j2	C	D	M2	M1,S1,G2
j3	D	C	M1	M2,S2,G1
j4	D	D		

3 Justification of Actions

The current approach to reasoning about the actions of others based on [2] and used in [8] is:

1. Select a desirable transition based on the values it promotes and demotes.
2. Argue for the individual action performed by the agent in the joint action corresponding to that transition.
3. Consider objections based on the other agents choosing different actions and so causing different joint actions to be performed.
4. Attempt to rebut these objections because:
 - (a) The values promoted and demoted by the alternative transition are acceptable.
 - (b) It is considered that the other agents will not act in this way.

Whereas 4a can be resolved on the basis of the preferences of the agent concerned, 4b, which is very often needed, requires more assumptions about the other agents than can be really justified.

In previous treatments based on such transition diagrams and using PRAS (e.g. [8]) we would get arguments such as *we should cooperate to promote M1* which would be challenged with objections such as *but player 2 might defect which would demote M1*. Now if M1 is the most important value for Player 1, then the objection will succeed, unless cooperation can be assumed. If M1 is the only value considered, defection is *dominant*, giving a better outcome whatever the other player chooses. Only if other values are considered will Player 1 choose cooperation. For example, M2 might be rated as highly as M1 (perhaps Player 2 is Player 1's child, or a close colleague), or a clear conscience is regarded as more important than money, in which case Guilt must be considered. The arguments are, however, really for a *particular* transition (joint action), with the agent's own action justified in virtue of its appearance in the transition: the objections are available because other joint actions contain the same individual action. Better would be an argument for the individual action itself, not the joint action and its corresponding transition. This will require us to look at the *set* of transitions containing the action. In the Ultimatum Game suppose that $prob(jointaction)$ is the probability of *jointaction* being performed when the agent concerned chooses some particular individual action. Now the values will be expected to be promoted and demoted according to the probability of the second player's response, as shown in Table 3, and so expected utility can be calculated, obviating the need to assume that the other will perform a particular action.

Table 3. Values Promoted and Demoted in the Ultimatum Game

Proposer Action	Promoted	Demoted
vho	G, prob(j1)M1, prob(j1)M2, prob(j1)C2	prob(j2)C1, prob(j2)M1, prob(j1)E
eo	G, prob(j3)M2, prob(j3)C2, prob(j3)M1	prob(j4)C1, prob(j4)M1
fo	prob(j5)M1, prob(j5)M2	prob(j6)M1, prob(j5)E
lo	prob(j7)M1, prob(j7)M2, prob(j7)C1	prob(j7)C2, prob(j8)M1, prob(j8)C1, prob(j7)E

Now we can base arguments on the complete set of transitions containing an action, rather than having to assume an action on the part of the other and then consider objections based on the potential performance of a different action. Several forms of argument are available (our examples assume the context of a persuasion dialogue with the proposer in the Ultimatum Game [43]):

- Where an action is certain to promote a value. E.g. *You should make a very high offer to promote G.*
- Where an action cannot promote a value. E.g. *You should not make a very high offer as that cannot promote C1.*
- Where an action can promote a value. E.g. *You should make a fair offer as this can promote M1.*
- Where an action can demote a value. E.g. *You should not make a low offer as that will risk demoting C1.*

The third and fourth forms will have variants, if we can say something about the relative probabilities of acceptance and rejection. These variants will replace "can" with an indicator of how probable promotion is, such as "very likely", "more likely than not", "may possibly" etc. For example, we know from [30] that a fair offer is

much more likely to be accepted than rejected, and so we can say *you should make a fair offer as that is likely to promote M1*, or, since low offers are more likely to be rejected, *you should not make a low offer as there is a substantial risk of demoting M1*.

Similar arguments can be generated for Prisoner's Dilemma. Promotions and demotions of the extended set of values for each action are shown in Table 4. From this table we can generate arguments, as given below.

Table 4. Values promoted and demoted in Prisoner's Dilemma

Proposer Action	Promoted	Demoted
C	M2, prob(j1)M1,	prob(j2)M1, probj(2)S1, prob(j2)G2
D	prob(j3)M1	prob(j3)M2, probj(3)S2, prob(j3)G1

- You should cooperate to promote M2
- You should not cooperate as this risks demoting M1, S1 and G2
- You should defect as this might promote M1
- You should not defect as this risks demoting M2, G1 and S2.

The real advance here over previous work such as [2] is that there is no longer any need to make assumptions about the what the other believes and prefers: the agent can now come to a decision using its *own* relative preferences between values, its own beliefs and the degree of risk it is prepared to take, whilst requiring no additional machinery: it uses only the AATS+V as developed in [2]. This fulfils the first of the objectives identified in section 1.

3.1 More than one other agent

The games discussed above have only one other agent. Of course, in practice there will typically be several, or even very many, agents that can have an influence. For example we might extend the Ultimatum Game so that there are several respondents and acceptance or rejection is determined by a majority, or acceptance may require unanimity. Or we might want to look at a problem such as the free-rider problem, whereby defection pays, unless some proportion of the population defects. For example, a small number of tax avoiders will not affect services, but if there are too many, the state infrastructure will collapse. In other situations there may be a number of agents with a range of, perhaps different, choices. This might, at first sight present a problem, since the number of joint actions rises rapidly: n agents each with m actions give rise to m^n joint actions. But we are not especially interested in details of the joint actions: the point of our approach here is to consider the set of joint actions in which the agent of concern performs a particular action. In the standard value-based approach, as proposed in [2] the values promoted and demoted by a transition are determined by the source and target states. Even where the action performed does affect values, as in [5], so that the intrinsic value of an action can be taken into account, what matters for the agent concerned is its own individual action, and so all transitions between the same pair of states containing that action will promote and demote the same values, as far as that agent is concerned. Thus, for our current purposes, we will consider all joint actions with the same action by the agent concerned leading to the same state to be equivalent, so that consideration can be limited to the different outcomes possible for a given action, irrespective of how many joint

actions reach each outcome. Effectively all the other agents can be considered together as a single other. If a majority is required, it does not matter which agents make up that majority; nor does it matter who the other free loaders are provided that there are not too many of them, and so on. Of course, the probabilities may be affected: if we know that only one agent in six will reject a fair offer, then we can be more confident that the larger the number of respondents the more likely is a majority for acceptance, although it is less likely that the offer will be accepted if we require unanimity.

3.2 Preferred Values

If only a single value is recognised as worthy of promotion, the choice is often unproblematic. In the Prisoner's Dilemma, M1 may be promoted and cannot be demoted by defection, M2 is promoted by cooperation, C1 can only be demoted by defection and S1 can only be demoted by cooperation, but in some cases, whether a value is promoted or demoted may depend on what the other agents do. Similarly some combinations of values are unproblematic, but hard choices arise when different values pull us in different directions, because an action may promote one value and demote another, or because values are promoted and demoted to different degrees. In such cases we need to express and quantify our preferences.

3.3 Expected Utilities

We now turn to our second objective. In all value based reasoning it is assumed that an agent is capable of expressing a preference in terms of an ordering on values. However, sometimes quantification of the degree of preference and the degree of promotion is required (e.g. [28]). In PD the payoff matrix gives the degree of promotion e.g. j_1 promotes M1 and M2 to degree 2 etc: (remember that we only count gains in excess of the baseline towards promoting M1 and M2), but to quantify the preference each value must be expressed in terms of a single selected value (M1 is the obvious choice). The valuation is subjective to each agent, but requires reference only to its own preferences. Agent Preferences are defined in Definition 3 in the Appendix. Unlike previous work such as [2] there is no longer any need to make assumptions about the beliefs, domain conceptualisation and preferences of the other: the agent will be able to decide using its *own* relative preferences between values, its own beliefs and, where necessary, the particular degree of risk it is subjectively prepared to accept.

Once the agent preferences have been established, the expected utilities can be calculated as in Definition 4 of the Appendix.

If we apply this to the Prisoner's Dilemma (PD), since there are only two joint actions containing cooperation, $prob(j_2) = 1 - prob(j_1)$. In the traditional PD only the agent's own payoff is recognised as having utility. The utility is the actual payoff minus the guaranteed payoff (i.e. the payoff from mutual defection). For cooperation the utility is 2 when the other cooperates and -1 when the other defects. For defection it is 4 when the other cooperates and 0 when the other defects. The expected utilities for ag cooperating (dark grey) and defecting (light grey) for the various probabilities of the other cooperating are shown in Figure 2.

Suppose, however, that both the values M1 and M2 are recognised in PD, and M2 is weighted at 0.5M1. Now the utility of cooperating when the other also cooperates will be 3M1, and the utility of cooperating when the other defects M1. Similarly we can calculate the expected utility of defecting for the various probabilities of the

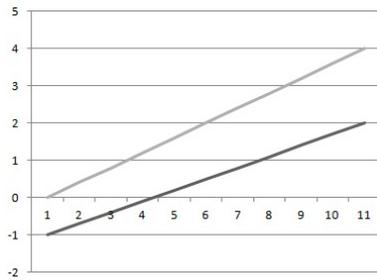


Figure 2. Expected Utilities for M1 only. Dark grey is *ag* cooperates, light grey is *ag* defects.

other cooperating. Defecting when the other cooperates yields a utility of 3.5M1, and mutual defection 0 (since this is the base line case, no values are considered promoted). Again the desired joint action is performed when the other agent cooperates. This gives the graph shown as Figure 3a. The crossover is at $prob(j_0) = 0.67$.

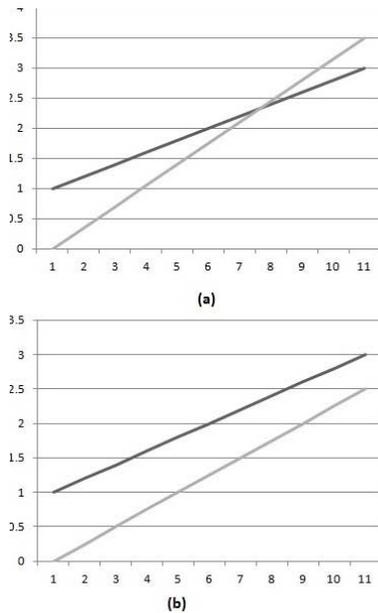


Figure 3. Expected Utilities for (a) $M2 = 0.5M1$ and (b) $M2 = 0.5M1$ and $G = M1$. Dark grey is *ag* cooperates, light grey is *ag* defects.

If we now add in the value of Guilt (with a weight of 1), which gives a negative utility when an agent defects and the other cooperates, we get the expected utilities shown in Figure 3b.

These three figures represent the three possibilities. In Figure 2, which shows the traditional PD, we find that defection *dominates* cooperation: the expected utility is higher for every value of $prob(j_0)$. Therefore defection is the preferred action, whatever the probability of the other cooperating. In Figure 3b the reverse is true: the inclusion of the additional values means that cooperation dominates defection. In Figure 3a, there is a crossover, at $prob(j_0) = 0.7$, so that for high probabilities of cooperation, defection is preferred, but for low levels, the utility afforded to the payoff received by the other makes cooperation preferred.

3.4 Arguments in Prisoner’s Dilemma Using Expected Utilities

Our third objective is addressed by producing arguments based on the expected utilities. These different possibilities mean that several types of argument can be based on the expected utilities. Our examples are expressed in terms suitable for a persuasion dialogue (not between the PD participants, but between a participant and advisor).

1. With your value preferences, you should C (respectively, D) since the expected utility is always greater than any alternative
2. With your value preferences, you should C (respectively, D) since the expected utility is always positive
3. With your value preferences, you should C (respectively, D) since the expected utility is greater than the alternative when the probability of cooperation is greater (less) than P.

Of these (1) is appropriate when the action advocated is dominant, and is the strongest of the three. Argument (2) is rather weak: although the expected utility is always positive, the proposed action can be dominated by the alternative for some (or even all) values of $prob(j_1)$. It may, however, be useful if we wish to reach the target state in order to enable some more beneficial action, since it indicates that no harm is done, and so can be used to rebut objections. The argument shows that we suffer no loss, although there is an opportunity cost. Argument (3) can be effective provided we can give reasons to suppose that probability of cooperation is in the desired range.

A dialogue arising from using (1) for defection might run:

- Since you value M1 and M2 equally, you should C since the expected return is always greater than the alternative.
- this overvalues M2.
- even if M2 is only worth 70% of M1, the expected utility is always greater than the alternative.
- But even 70% overvalues M2
- Even if M2 is only worth half M1, a less than 0.6 probability of cooperation will mean cooperation has the higher expected utility. Moreover the expected utility of cooperation is still always positive.

In the course of the dialogue, the very strong argument of type (1) has become untenable, but a combination of arguments of types (2) and (3) remain potentially persuasive. Here we are producing argumentation dialogues (albeit not yet expressed in a formal dialogue model) which explore the sensitivity to the assessment of the relative valuations, and the sensitivity to the estimates of cooperation. These dialogues do not require any knowledge about the other, but if such information is available these dialogues provide a context in which it can be deployed by constraining the range for the probability of cooperation. For an example based on (2):

- Since you value M2 at 50% of M1, you should C since the expected return is always positive
- But with these values, D gives a better return unless the probability of cooperation is worse than 0.6.

This objection could be reinforced with reasons to suppose it likely that the other will cooperate (family member, team member or similar, or experimental results, if appropriate results are available). Note, however, that these are also reasons to increase the valuation of M2 relative to M1.

The above arguments can, if desired, be presented as argumentation schemes in the manner of [44]. For example the scheme based on (1) above:

- **Values Premise:** V is the set of values considered to be relevant by ag
- **Weighting Premise:** The relative valuation of the members of V given by ag is a set of $\langle value, relativeweight \rangle$ pairs
- **Joint Action Premise:** $\{j_0, j_1, \dots, j_n\}$ is the set of joint actions J in which ag performs α
- **Expected Utility Premise:** $eu_{ag}(\alpha, prob(j_0))$ returns the expected utilities of agent ag performing α for values of $prob(j_0)$ $0 \leq prob(C) \leq 1$ where j_0 is the desired joint action.
- **Dominance Premise:** $eu_{ag}(\alpha, j_0) \geq eu_{ag}(\beta, j_0)$ for any alternative action β available to ag , for all values of $prob(j_0)$; where j_0 is the joint action compliant with the action of ag .
- **Conclusion:** ag should perform α

This scheme would be associated with critical questions such as: *Are all the members of V relevant? Are any other Values relevant? Are any members of V under or over valued?* These critical questions will have their own characteristic rebuttals: For example the third could be met by *even if the value of v is reduced to $n\%$, the expected utility is always greater than its alternatives.*

3.5 Application to the Ultimatum Game

Similar arguments can be produced for the Ultimatum Game. Different weights for the different values will lead to different arguments being dominant. Also the different actions will promote M1 and M2 to varying degrees. M1 will be promoted most (if accepted) by lo , then fo then eo and least by vh , whereas for M2 the reverse will be true. Some examples are given in Table 5.

Table 5. Example value weights and corresponding dominant actions in the Ultimatum Game

M1	M2	G	E	C1	C2	dominant
1	0	0	0	0	0	lo
1	0.3	0	0	0	1	fo
1	0.3	0	1	0	1	eo
1	1	1	0	0.5	0.5	vh
1	0.3	0.7	0	0	0.5	eo/lo
1	0.6	0.4	0	0	0.5	vh/fo

The last two rows give examples of value assignments which produce cross overs. In the penultimate row, at low probabilities of acceptance the best choice is the equal offer: this promotes generosity and avoids angering the other, without sacrificing more money than is necessary to achieve these goals. When the probability of acceptance reaches 0.6 both the fair offer and the low offer take over, with the low offer being slightly preferred. In the final row, the high weight of M2 means that the very high offer is better than the equal offer for low probabilities of acceptance, but the fair offer becomes best for probabilities of acceptance greater than 0.6. When the probability exceeds 0.7, the low offer is also better than the very high offer, but the fair offer remains best.

Finally we have produced some initial results which indicate that the cultural variations encountered in public goods game experiments can be reproduced using suitable value profiles, shown in Table 6. Reproduction of such experimental results will form the basis of our evaluation.

3.6 Evaluation

We offer two aspects of evaluation. Technically, we can ask whether we achieved the objectives set out in section 1. Practically, we can ex-

plore the extent to which our proposed approach is able to reproduce the results of empirical studies such as [22].

Three technical objectives were given in Section 1. Our first objective was to accommodate the need to consider the actions of others, while only considering the values, and preferences of the agent concerned, since modelling of others is inevitably unreliable, given the extent of inter- and intra-cultural variation. We have achieved this, using only the structure of AATS+V of [2], by considering all the joint actions containing a given individual action as a set, obviating the need to consider the specific actions performed by others. The second objective was to do this in a way consistent with existing game and multi-criteria utility theory. We have achieved this by relating the value-based approach to expected utilities. The key notion of a dominant action remains, since, if there is a dominant action, the expected utility of the values promoted by that action will always be greater than any alternative. Moreover where an action is not dominant for all probabilities of the other behaving as required, the bounds can be identified, which allows for the sensitivity to the relative weighting of the relevant values, and, where no action is dominant, to the probability of the other performing the appropriate action, to be quantified. To fulfill the third objective, we have given arguments grounded on the expected utilities. Objections can be based on adding, removing or re-weighting values, which can change the dominant action, or restrict its dominance to a certain range of probabilities of the other agents allowing a particular outcome to be reached. Again the required degree of revaluation can be specified.

Whereas the payoffs of game theory are, as is perfectly correct for games which do require firm rules, fixed and unchanging, here the payoffs are subjective with respect to the individual goals and aspirations of the agent concerned, and so can be individually set and made subject to change, possibly as a result of persuasive argument, or of empirical evidence. This means that we can attempt a more practical evaluation in terms of reproducing the results of studies such as [22].

Recall that that study accounted for differences in terms of the degree of cooperation, and degree of commercial exchange found in daily life. We can relate these characteristics to a value profile. Suppose we associate the value of generosity with the cooperative groups such as the whale hunting Lamelara, and the recognition of C2 (the need not to anger the other) with commercial exchange. Ideally we would produce a value profile for each society, and evaluate both proposers and respondents. Such a full study must await future work, but as an encouraging preliminary we offer the results shown in Table 6.

Table 6. Dominant actions for values relating to cooperation and exchange activities

	M1	M2	G	E	C1	C2	dominant
cooperative	0.3	0	1	0	0	0	eo
exchange	0.3	0	0	0	0	1	eo
neither	1	0	0	0	0	0	lo

These results show that these value profiles do indeed correspond to the action choices typical of corresponding societies. Note that it is the equal offer rather than the very high offer that Table 6 predicts for cooperative societies and those accustomed to commercial exchange. This coheres with the highest offers in [22] being 58% and 51%. Similarly the lowest offer of 26% belonged to groups that did not work cooperatively and rarely engaged in commercial exchange, reflected here by a profile which does not recognise either generosity or the feelings of the respondent.

As well as replicating previous studies, we can also perform our

own experiments in which the value preferences of the subject are established (e.g. through a questionnaire), and then the behaviour in the games compared with what is predicted by the value profile.

4 Concluding Remarks

Previous work on practical reasoning using value-based argumentation has required assumptions about the values and preferences of other agents which can affect the outcome of an action performed by the reasoning agent. Justification of these assumptions is always difficult, particularly when several other agents are involved, multiplying the alternative actions needing consideration. We have described an approach in which no assumptions need be made about the values and preferences of others: all that is required is that the agent concerned can identify the values it recognises and indicate their relative worth to itself. In some cases success may still depend on what the other does, but this can be assessed using bounds on the probabilities of the alternatives available to the other. In this way we are able to achieve our objectives of allowing arguments which consider the actions of others, but which do not require assumptions about the beliefs and preferences of the others, while remaining consistent with multi-criteria utility theories, and the dominant actions of game theory. Thus we have shown how to:

- Remove the need to speculate on the preferences of other agents;
- Relate the value-based argumentation approach to approaches based on multi-criteria utility and game theory.
- Express reasons based on utility and expected returns as arguments, and objections to them, so that the arguments are genuinely for a particular action by the agent concerned rather than participation in a joint action, as was the case in [2].

We believe that this greatly improves the quality of value-based arguments for particular actions. Note also that the dominance of an action is *dominance for that agent*: it depends on the subjective values and aspirations of the individual agent. Which action is considered dominant by a particular agent or audience will depend on the values recognised, and the relative importance assigned to them, rather than fixed payoffs determined by the game, allowing each agent to set its own objectives. In addition to providing some initial results, we have, for future work, set out how the approach can be more broadly empirically tested using both existing and new experimental studies.

Appendix: Formal Definitions

Definition 1: AATS [46]. An *Action-based Alternating Transition System* (AATS) is an $(n + 7)$ -tuple $S = \langle Q, q_0, Ag, Ac_1, \dots, Ac_n, \rho, \tau, \Phi, \pi \rangle$, where:

- Q is a finite, non-empty set of *states*;
- $q_0 \in Q$ is the *initial state*;
- $Ag = \{1, \dots, n\}$ is a finite, non-empty set of *agents*;
- Ac_i is a finite, non-empty set of actions, for each $ag_i \in Ag$ where $Ac_i \cap Ac_j = \emptyset$ for all $ag_i \neq ag_j \in Ag$;
- $\rho : Ac_{ag} \rightarrow 2^Q$ is an *action pre-condition function*, which for each action $\alpha \in Ac_{ag}$ defines the set of states $\rho(\alpha)$ from which α may be executed;
- $\tau : Q \times J_{Ag} \rightarrow Q$ is a partial *system transition function*, which defines the state $\tau(q, j)$ that would result by the performance of j from state q . This function is partial as not all joint actions are possible in all states;

- Φ is a finite, non-empty set of *atomic propositions*; and
- $\pi : Q \rightarrow 2^\Phi$ is an interpretation function, which gives the set of primitive propositions satisfied in each state: if $p \in \pi(q)$, then this means that the propositional variable p is satisfied (equivalently, true) in state q .

AATSs are particularly concerned with the joint actions of the set of agents Ag . j_{Ag} is the joint action of the set of n agents that make up Ag , and is a tuple $\langle \alpha_1, \dots, \alpha_n \rangle$, where for each α_j (where $j \leq n$) there is some $ag_i \in Ag$ such that $\alpha_j \in Ac_i$. Moreover, there are no two different actions α_j and $\alpha_{j'}$ in j_{Ag} that belong to the same Ac_i . The set of all joint actions for the set of agents Ag is denoted by J_{Ag} , so $J_{Ag} = \prod_{i \in Ag} Ac_i$. Given an element j of J_{Ag} and an agent $ag_i \in Ag$, ag_i 's action in j is denoted by j^i . This definition was extended in [2] to allow the transitions to be labelled with the values they promote.

Definition 2: AATS+V. Given an AATS, an AATS+V is defined by adding two additional elements as follows:

- V is a finite, non-empty set of values.
- $\delta : Q \times Q \times V \rightarrow \{+, -, =\}$ is a *valuation function* which defines the status (promoted (+), demoted (-) or neutral (=)) of a value $v_u \in V$ ascribed to the transition between two states: $\delta(q_x, q_y, v_u)$ labels the transition between q_x and q_y with one of $\{+, -, =\}$ with respect to the value $v_u \in V$.

An *Action-based Alternating Transition System with Values* (AATS+V) is thus defined as a $(n + 9)$ tuple $S = \langle Q, q_0, Ag, Ac_1, \dots, Ac_n, \rho, \tau, \Phi, \pi, V, \delta \rangle$. The value may be ascribed on the basis of the source and target states, or in virtue of an action in the joint action, where that action has intrinsic value.

Definition 3: Agent Preferences

The preferences of an agent $ag \in Ag$ is the set $O_{ag} = \{\langle v_0 * w_0 \rangle, \langle v_1 * w_1 \rangle, \dots, \langle v_n * w_n \rangle\}$, where $v_0 \dots v_n$ are values and $w_0 \dots w_n$ are weights with $w_0 \geq w_1 \geq \dots \geq w_n$.

Using these weights we can calculate the expected utility of agent i performing α . We will assume that if the desired joint action (j_0) does not result from the performance of α the worst case alternative joint action (j_w) will be the one that does result (providing a lower bound). Informally the expected utility of performing α will be the utility of j_0 multiplied by the probability of j_0 plus the utility of j_w (which will often be negative) multiplied by (1 minus the probability of j_0).

Definition 4: Expected Utility of ag performing α in state q_s

- Let $J_\alpha = \{j_0, j_1 \dots j_n\}$ be the set of joint actions in which ag performs α (i.e. $j^{ag} = \alpha$) available in the starting state, q_s .
- Let P_{agk} be the values for ag promoted by the performance of $j_k \in J_\alpha$ in q_s . Let D_{agk} be the values of ag demoted by the performance of $j_k \in J_\alpha$ in q_s .
- The positive utility for ag , $pu(ag, j_k)$, of the performance of $j_k \in J_\alpha$ in q_s is $\sum_{i=0}^n (v_i * w_i)$ where $v_i \in P_{agk}$ and the negative utility for ag , $du(ag, j_k)$, of the performance of $j_k \in J_\alpha$ in q_s is $\sum_{i=0}^n (v_i * w_i)$ where $v_i \in D_{agk}$. The utility, $u(ag, j_k)$, for ag of the performance of $j_k \in J_\alpha$ in q_s is $pu(ag, j_k) - du(ag, j_k)$.
- Let U_{ag} be the set of utilities for ag , $\{u_0, u_1 \dots u_n\}$, such that $u_k = u(ag, j_k)$ for $j_k \in J_\alpha$. Let u_w be such that for all $u_i \in U_{ag}$, $u_w \leq u_i$.
- Let $prob(j_0)$ be the probability of j_0 being the joint action performed when ag performs α in q_s .
- Now the expected utility, $eu_{ag}(\alpha)$ for ag of performing α in q_s is $(u(ag, j_0) * prob(j_0)) + (u(ag, j_w) * (1 - prob(j_0)))$.

REFERENCES

- [1] Rajeev Alur, Thomas A Henzinger, and Orna Kupferman, 'Alternating-time temporal logic', *Journal of the ACM (JACM)*, **49**(5), 672–713, (2002).
- [2] Katie Atkinson and Trevor Bench-Capon, 'Practical reasoning as presumptive argumentation using action based alternating transition systems', *Artificial Intelligence*, **171**(10), 855–874, (2007).
- [3] Katie Atkinson, Trevor Bench-Capon, and Peter McBurney, 'Computational representation of practical argument', *Synthese*, **152**(2), 157–206, (2006).
- [4] Katie Atkinson, Trevor Bench-Capon, and Sanjay Modgil, 'Argumentation for decision support', in *Database and expert systems applications*, pp. 822–831. Springer, (2006).
- [5] Katie Atkinson and Trevor J. M. Bench-Capon, 'Taking the long view: Looking ahead in practical reasoning', in *Computational Models of Argument - Proceedings of COMMA 2014*, pp. 109–120.
- [6] Robert Axelrod, 'An evolutionary approach to norms', *American political science review*, **80**(04), 1095–1111, (1986).
- [7] Trevor Bench-Capon, Katie Atkinson, and Alison Chorley, 'Persuasion and value in legal argument', *Journal of Logic and Computation*, **15**(6), 1075–1097, (2005).
- [8] Trevor Bench-Capon, Katie Atkinson, and Peter McBurney, 'Using argumentation to model agent decision making in economic experiments', *Autonomous Agents and Multi-Agent Systems*, **25**(1), 183–208, (2012).
- [9] Trevor JM Bench-Capon, 'Persuasion in practical argument using value-based argumentation frameworks', *Journal of Logic and Computation*, **13**(3), 429–448, (2003).
- [10] Brigitte Burgemeestre, Joris Hulstijn, and Yao-Hua Tan, 'Value-based argumentation for justifying compliance', *Artificial Intelligence and Law*, **19**(2-3), 149–186, (2011).
- [11] Dan Cartwright and Katie Atkinson, 'Using computational argumentation to support e-participation', *Intelligent Systems, IEEE*, **24**(5), 42–52, (2009).
- [12] Günther Charwat, Wolfgang Dvořák, Sarah A Gaggl, Johannes P Wallner, and Stefan Woltran, 'Methods for solving reasoning problems in abstract argumentation—a survey', *Artificial intelligence*, **220**, 28–63, (2015).
- [13] Phan Minh Dung, 'On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games', *Artificial intelligence*, **77**(2), 321–357, (1995).
- [14] Paul E. Dunne, 'Tractability in value-based argumentation', in *Proceedings of COMMA 2010*, pp. 195–206, (2010).
- [15] Uwe Egly, Sarah Alice Gaggl, and Stefan Woltran, 'Answer-set programming encodings for argumentation frameworks', *Argument and Computation*, **1**(2), 147–177, (2010).
- [16] Christoph Engel, 'Dictator games: a meta study', *Experimental Economics*, **14**(4), 583–610, (2011).
- [17] Sevan G Ficici and Avi Pfeffer, 'Simultaneously modeling humans' preferences and their beliefs about others' preferences', in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*, pp. 323–330. International Foundation for Autonomous Agents and Multiagent Systems, (2008).
- [18] Richard E Fikes and Nils J Nilsson, 'Strips: A new approach to the application of theorem proving to problem solving', *Artificial intelligence*, **2**(3-4), 189–208, (1971).
- [19] Artur SD'Avila Garcez, Dov M Gabbay, and Luis C Lamb, 'Value-based argumentation frameworks as neural-symbolic learning systems', *Journal of Logic and Computation*, **15**(6), 1041–1058, (2005).
- [20] Matthias Grabmair and Kevin D Ashley, 'Facilitating case comparison using value judgments and intermediate legal concepts', in *Proceedings of the 13th international conference on Artificial intelligence and law*, pp. 161–170. ACM, (2011).
- [21] Floriana Grasso, Alison Cawsey, and Ray Jones, 'Dialectical argumentation to solve conflicts in advice giving: a case study in the promotion of healthy nutrition', *International Journal of Human-Computer Studies*, **53**(6), 1077–1115, (2000).
- [22] Joseph Henrich, Robert Boyd, Samuel Bowles, Colin Camerer, Ernst Fehr, Herbert Gintis, and Richard McElreath, 'In search of homo economicus: behavioral experiments in 15 small-scale societies', *The American Economic Review*, **91**(2), 73–78, (2001).
- [23] Souhila Kaci and Leendert van der Torre, 'Preference-based argumentation: Arguments supporting multiple values', *International Journal of Approximate Reasoning*, **48**(3), 730–751, (2008).
- [24] Samhar Mahmoud, Nathan Griffiths, Jeroen Keppens, Adel Taweel, Trevor JM Bench-Capon, and Michael Luck, 'Establishing norms with metanorms in distributed computational systems', *Artificial Intelligence and Law*, **23**(4), 367–407, (2015).
- [25] John Stuart Mill, 'On the definition of political economy, and on the method of investigation proper to it', *Essays on some unsettled questions of Political Economy*, 326, (1844).
- [26] Sanjay Modgil and Trevor J. M. Bench-Capon, 'Integrating object and meta-level value based argumentation', in *Proceedings of COMMA 2008*, pp. 240–251, (2008).
- [27] Roger B Myerson, 'Game theory: analysis of conflict', *Harvard University*, (1991).
- [28] Fahd Saud Nawwab, Trevor J. M. Bench-Capon, and Paul E. Dunne, 'A methodology for action-selection using value-based argumentation', in *Computational Models of Argument: Proceedings of COMMA 2008*, pp. 264–275, (2008).
- [29] Samer Nofal, Katie Atkinson, and Paul E Dunne, 'Algorithms for decision problems in argument systems under preferred semantics', *Artificial Intelligence*, **207**, 23–51, (2014).
- [30] Hessel Oosterbeek, Randolph Sloof, and Gijs Van De Kuilen, 'Cultural differences in ultimatum game experiments: Evidence from a meta-analysis', *Experimental Economics*, **7**(2), 171–188, (2004).
- [31] Simon Parsons and Michael Wooldridge, 'Game theory and decision theory in multi-agent systems', *Autonomous Agents and Multi-Agent Systems*, **5**(3), 243–254, (2002).
- [32] Terry R Payne and Valentina Tamma, 'Using preferences in negotiations over ontological correspondences', in *PRIMA 2015: Principles and Practice of Multi-Agent Systems*, 319–334, Springer, (2015).
- [33] Chaim Perelman, *The new rhetoric*, Springer, 1971.
- [34] Anatol Rapoport and Albert M Chammah, *Prisoner's dilemma: A study in conflict and cooperation*, volume 165, University of Michigan press, 1965.
- [35] Joseph Raz, *Practical Reasoning*, Oxford University Press, Oxford, 1979.
- [36] John R Searle, *Rationality in action*, MIT press, 2003.
- [37] Theo J Stewart, 'A critical survey on the status of multiple criteria decision making theory and practice', *Omega*, **20**(5), 569–586, (1992).
- [38] Jasmin Tremblay and Irène Abi-Zeid, 'Value-based argumentation for policy decision analysis: methodology and an exploratory case study of a hydroelectric project in québec', *Annals of Operations Research*, **236**(1), 233–253, (2016).
- [39] Cássia Trojahn, Paulo Quaresma, and Renata Vieira, 'An extended value-based argumentation framework for ontology mapping with confidence degrees', in *Argumentation in Multi-Agent Systems*, 132–144, Springer, (2007).
- [40] E. Ullmann-Margalit, *The emergence of norms*, Clarendon Press Oxford, 1977.
- [41] Thomas L van der Weide, Frank Dignum, J-J Ch Meyer, Henry Prakken, and GAW Vreeswijk, 'Multi-criteria argument selection in persuasion dialogues', in *Argumentation in Multi-Agent Systems*, 136–153, Springer, (2011).
- [42] D.N. Walton, *Argumentation Schemes for Presumptive Reasoning*, Lawrence Erlbaum Associates, Mahwah, NJ, USA, 1996.
- [43] Douglas Walton and Erik CW Krabbe, *Commitment in dialogue: Basic concepts of interpersonal reasoning*, SUNY press, 1995.
- [44] Douglas Walton, Christopher Reed, and Fabrizio Macagno, *Argumentation schemes*, Cambridge University Press, 2008.
- [45] Maya Wardeh, Adam Wyner, Katie Atkinson, and Trevor Bench-Capon, 'Argumentation based tools for policy-making', in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Law*, pp. 249–250. ACM, (2013).
- [46] Michael Wooldridge and Wiebe van der Hoek, 'On obligations and normative ability: Towards a logical analysis of the social contract', *J. Applied Logic*, **3**(3-4), 396–420, (2005).
- [47] Adam Wyner, Maya Wardeh, Trevor Bench-Capon, and Katie Atkinson, 'A model-based critique tool for policy deliberation.', in *JURIX*, pp. 167–176, (2012).

Analogical Classifiers: A Theoretical Perspective

Nicolas Hug¹ and Henri Prade^{1,2} and Gilles Richard^{1,3} and Mathieu Serrurier¹

Abstract. In recent works, analogy-based classifiers have been proved quite successful. They exhibit good accuracy rates when compared with standard classification methods. Nevertheless, a theoretical study of their predictive power has not been done so far. One of the main barriers has been the lack of functional definition: analogical learners have only algorithmic definitions. The aim of our paper is to complement the empirical studies with a theoretical perspective. Using a simplified framework, we first provide a concise functional definition of the output of an analogical learner. Two versions of the definition are considered, a strict and a relaxed one. As far as we know, this is the first definition of this kind for analogical learner. Then, taking inspiration from results in k -NN studies, we examine some analytic properties such as convergence and VC-dimension, which are among the basic markers in terms of machine learning expressiveness. We then look at what could be expected in terms of theoretical accuracy from such a learner, in a Boolean setting. We examine learning curves for artificial domains, providing experimental results that illustrate our formulas, and empirically validate our functional definition of analogical classifiers.

1 Introduction

Analogical reasoning is widely recognized as a powerful ability of human intelligence. It can lead to conclusions for new situations by establishing links between apparently unrelated domains. One well known example is the Bohr's model of atom where electrons circle around the kernel, which is analogically linked to the model of planets running around the sun. It is not surprising that this kind of reasoning has generated a lot of attention from the artificial intelligence community. We can cite for instance [12, 13, 16, 30, 17] where the power of analogical reasoning is emphasized. The interested reader may find in [27] a survey of current trends. More recently, using analogy as a basis for the automatic solving of IQ tests [8, 29] or for machine learning tasks [15, 33] got more attention. In the case of classification, analogical classifiers are mainly based on a particular variant of analogy, namely analogical proportions and they have been proved successful [4, 28, 6], at least from an empirical viewpoint.

But analogy, as an essential ingredient of Artificial Intelligence, has also attracted theoretical investigations. In [10], a thorough investigation of analogical reasoning from a first order logic viewpoint has been done, leading to clearly specify safe conditions of usage of the *analogical jump*. More recently, in [14], an higher order logic framework has been developed, providing another logical theory for analogical reasoning in artificial intelligence and cognitive science. Instead of being described as an inference rule, the analogy-making process is described in terms of generalisation and anti-unification.

On top of this work, a full implementation has been done leading to the so-called Heuristic-Driven Theory Projection (HDTP).

From another viewpoint, we have to mention the work of [18] which is an attempt to consider analogy-making as a particular case of machine learning where very few data are available. In the limit case, only one pair $(a, f(a))$ (a is the source) is available and one has to guess $f(b)$ for another element b , the target. This work describes a model which minimizes the computational cost of producing $(b, f(b))$ from $(a, f(a))$. This computational cost can be estimated via Kolmogorov complexity [7], a measure which is well-known to be hard to compute (but can be estimated via compression).

Finally, we can also recall the work of [2] where the particular case of analogical proportion is investigated in lattices and other algebraic structures, leading to elegant theoretical results and implementations.

Nevertheless, all these theoretical investigations are not directed to provide an analytical view of analogy-based learners. In that sense, they are not really helpful if we want to characterize the behaviour of an analogical classifier for instance. One of the reasons could be that, unlike the k -NN rule, the analogical learning rule is not easily amenable to a functional definition. In fact, each implemented algorithm provides a clean description of *how to compute* but we definitely miss a clean description of *what do we actually compute*. Since such a definition, even a simplified one, is paramount to investigate theoretical properties, we suggest here a concise functional definition and we prove that it fits with the main implementations of analogical classifiers.

Our paper is organized as follows. In Section 2, we recall the fundamentals about analogical proportions as a particular case of analogy. Then, in Section 3, we explain how such proportions underlie analogical classifiers and the principle of their implementations. Then we provide a unified functional view establishing the formal framework allowing to investigate their mathematical properties. In Section 4, we examine some general properties such as convergence and VC-dimension of analogical learners, considering only minimal constraints on the underlying domain. In Section 5, we investigate, from a probabilistic viewpoint, the expected accuracy of an analogical learner in the Boolean case. We empirically validate our formulas in Section 6 with a complete batch of experiments. We provide our final remarks in Section 7, linking the known results about analogical classifiers with their mathematical properties, noting some limitations of our study and suggesting directions for future research.

2 Analogical proportions

Given a set X , an analogical proportion⁴ over X is a quaternary relation A over X satisfying 3 axioms [11, 20]:

1. $\forall a, b, A(a, b, a, b)$

⁴For the remaining of this paper, the term *analogy* always means *analogical proportion*.

¹IRIT Univ. P. Sabatier, Toulouse, France, email: name.surname@irit.fr

²QCIS, University of Technology, Sydney, Australia

³BITE, London, UK

Let A be an analogy relation over X and B an analogy relation over Y , the set of labels. The notion of analogical equation allows us to define the so-called *analogical extension* of S denoted as:

$$A_E^Y(S) = \{x \in X \mid \exists (a, b, c) \in S^3, a : b :: c : x \text{ and} \\ \exists y \in Y, \dot{a} : \dot{b} :: \dot{c} : y\}.$$

An intuitive interpretation of $A_E^Y(S)$ is to see it as the set of all $x \in X$ that are solutions of the analogical equations which can be built over the sample set S , provided that the equation related to the associated labels is also solvable. We have the following properties:

1. $S \subseteq A_E^Y(S)$, since $x : x :: x : x$ always holds ;
2. $A_E^Y(\emptyset) = \emptyset, A_E^Y(X) = X$;
3. $S_1 \subseteq S_2 \implies A_E^Y(S_1) \subseteq A_E^Y(S_2)$.

The dual concept of the analogical extension is the so-called *analogical root* of a given element $x \in X$, denoted $R_S^Y(x)$:

$$R_S^Y(x) = \{(a, b, c) \in S^3 \mid a : b :: c : x \text{ and } \exists y \in Y, \dot{a} : \dot{b} :: \dot{c} : y\}$$

$R_S^Y(x)$ is the set of 3-tuples in S which are analogically linked to x and which provide a prediction for the label. It is clear that $R_S^Y(x)$ may contain more than one 3-tuple: for example in R^m , x may be the summit of more than one parallelogram.

For any element x of $A_E^Y(S)$, we define the *analogical label* of x as:

$$\bar{x} = \begin{cases} \dot{x} & \text{if } x \in S \\ \text{Mode}\{y \mid \dot{a} : \dot{b} :: \dot{c} : y \forall (a, b, c) \in R_S^Y(x)\} & \text{if } x \notin S \end{cases}$$

where $\text{Mode}(\Sigma)$ returns the most frequent element of the multiset Σ . In case of a tie, the returned element is chosen at random between the most frequent elements.

The analogical label will be used to estimate the label of every element. Obviously, in the first case, we do not want to change the label of the elements of S . For elements in $A_E^Y(S) \setminus S$ (i.e. the second case), the analogical label is the most frequent label out of all the labels inferred from the solution of the analogical equations that one can build from $R_S^Y(x)$. It is quite clear that, for these elements, we do not necessarily have $\bar{x} = \dot{x}$. To summarize, for a given element $x \in X$, we may potentially associate 3 labels:

- its true label \dot{x} ;
- in the case where $x \in A_E^Y(S)$, its analogical label \bar{x} ;
- its predicted label \hat{x} .

Conservative classifiers set the prediction of an element $x \in A_E^Y(S)$ as \hat{x} as follows:

$$\text{if } x \in A_E^Y(S), \hat{x} = \bar{x} \text{ else } \hat{x} \text{ is undefined}$$

This kind of classifier cannot predict a label for an element which is not in $A_E^Y(S)$. In Algorithm 1, we provide the corresponding algorithm.

Let us note that $A_E^Y(S)$ is never explicitly computed. Instead, we look for every 3-tuple in S and check if they belong to $R_S^Y(x)$. Clearly, this is a supervised learning setting, where sample instances are stored for future use, without any generalization process. Conservative classifiers are Instance Based Learners as described in [1].

Such a conservative learner cannot generalize to any new input and is restricted to elements in $A_E^Y(S)$. This is not the case for instance-based learner like k -NN. This is why other options have been implemented to overcome this problem and to extend in some sense the generalization ability of analogical learner, as we will see in the next section.

Algorithm 1 Conservative classifier

Input: A sample set S and an element $x \in X$ for which \dot{x} is unknown.

Output: \hat{x} , an estimation of \dot{x}

Init: $C = \emptyset$ // multiset of candidate labels

for all $(a, b, c) \in S^3$ such that $a : b :: c : x$ **do**

if $\exists y \in Y$ such that $\dot{a} : \dot{b} :: \dot{c} : y$ **then**

// we are sure $(a, b, c) \in R_S^Y(x)$

compute the solution y of $\dot{a} : \dot{b} :: \dot{c} : y$

$C = C \cup y$

end if

end for

$\hat{x} = \bar{x} = \text{Mode}(C)$ // undefined if $C = \emptyset$

3.2 Extended classifier

To relax the previous option, we need to be able to predict a label for elements outside $A_E^Y(S)$ i.e. elements which do not constitute a perfect analogy with elements in S . To this end, we can try to measure to what extent such elements are far from building a perfect analogy with those in S . The concept of *analogical dissimilarity*, first defined in [4], will be useful to quantify in some sense how far a relation $a : b :: c : d$ is from being a valid analogy. We keep the initial notation $AD(a, b, c, d)$ to denote the analogical dissimilarity between 4 elements. Some minimal properties have to be satisfied by such a dissimilarity $AD : X^4 \rightarrow \mathbb{R}^+$ to fit with the intuition:

- $\forall a, b, c, d, AD(a, b, c, d) = 0$ iff $a : b :: c : d$
- $\forall a, b, c, d, AD(a, b, c, d) = AD(c, d, a, b) = AD(a, c, b, d)$
- $\forall a, b, c, d, e, f, AD(a, b, e, f) \leq AD(a, b, c, d) + AD(c, d, e, f)$

As the definition of an analogy strongly relies on the structure and operators available on X , we have the same situation for AD : there are a lot of possibilities. For instance:

- When $X = \mathbb{R}^m$ and $a : b :: c : d$ iff $a - b = c - d$, $AD(a, b, c, d) = \|(a - b) - (c - d)\|_p$ is an analogical dissimilarity for any p , where $\|\cdot\|_p$ denotes the standard p norm in \mathbb{R}^m .
- When $X = \mathbb{B}$ and $a : b :: c : d$ iff $(a \wedge b \equiv c \wedge d) \wedge (a \vee b \equiv c \vee d)$, one can define an analogical dissimilarity $AD(a, b, c, d)$ as the number of values that have to be switched to get a proper analogy. For instance, $AD(0, 1, 0, 0) = 1$ and $AD(0, 1, 1, 0) = 2$. The codomain of AD is just $\{0, 1, 2\}$. When extended to $X = \mathbb{B}^m$ with

$$AD(a, b, c, d) = \sum_{i=1}^m AD(a_i, b_i, c_i, d_i),$$

we get an analogical dissimilarity whose co-domain is $[0, 2m]$. In fact, this definition is just the restriction to \mathbb{B}^m of the one coming from \mathbb{R}^m , when considering that $\mathbb{B}^m \subseteq \mathbb{R}^m$ and using the L_1 norm, i.e. $AD(a, b, c, d) = \|(a - b) - (c - d)\|_1$.

As a measure of *how poorly an analogical proportion holds*, the analogical dissimilarity will help to define more flexible classifiers. The main underlying idea is to consider *approximate* analogies which are not valid stricto sensu, but not too far to be valid. In [4], after defining analogical dissimilarity, the authors build an extended classifier allowing classification of elements that do not belong to $A_E^Y(S)$. Algorithm 2 gives a description of their classifier.

This algorithm is similar to the conservative one but, instead of looking for pure analogies, we allow for some analogies not to be perfect when we need to. In their implementation [4], the authors

Algorithm 2 *Extended classifier*

Input: A sample set S , an element $x \in X$ for which \hat{x} is unknown, a constant k .
Output: \hat{x} , an estimation of x
Init: $C = \emptyset$ // multiset of candidate labels
for all $(a, b, c) \in S^3$ such that $\exists y \in Y$ with $\hat{a} : \hat{b} :: \hat{c} : y$ **do**
 compute $AD(a, b, c, x)$ and store it
end for
for all k least values of $AD(a, b, c, x)$ **do**
 compute the solution y of $\hat{a} : \hat{b} :: \hat{c} : y$
 $C = C \cup y$
end for
 $\hat{x} = \text{Mode}(C)$

actually look for all the 3-tuples that have the same analogical dissimilarity as the k th one: this allows them to fit with the previous conservative approach. For the sake of simplicity, we have chosen to ignore this small detail in our explanation.

In [4], the authors evaluated this classifier on a Boolean setting \mathbb{B}^m over 8 benchmarks from the UCI repository. This approach led to remarkable results in terms of accuracy, when compared to off-the-shelf standard classifiers.

Nonetheless, this algorithm does not allow us to grasp its inherent working behaviour and it is difficult to extract theoretical properties. The aim of the next subsection is to give a functional translation of this algorithmic description.

3.3 Analogical classifier: a functional definition

As we have seen in the previous section, in the case of a Boolean setting, $AD(a, b, c, d) = \|(a - b) - (c - d)\|_1$. A simple rewriting leads to:

$$AD(a, b, c, d) = \|d - (c - a + b)\|_1 = \|d - d'\|_1,$$

where $d' = c - a + b$. Actually, d' is nothing but the 4th vertex of the parallelogram $abcd'$ so this means that $AD(a, b, c, d)$ simply is the L_1 distance from d to this 4th vertex. Note that as \mathbb{B}^m is not closed for addition, d' might not belong to \mathbb{B}^m but to \mathbb{R}^m : this happens when one of the terms $AD(a_i, b_i, c_i, d_i)$ is equal to 2, as further discussed later.

As we have seen, for a given $x \in X$, algorithm 2 tries to minimise $AD(a, b, c, x)$ over all the 3-tuples $(a, b, c) \in S^3$. In the light of what has just been explained, we see that this is equivalent to finding the closest vertex $d' = c - a + b$ from x for any $(a, b, c) \in S^3$.

Denoting δ the L_1 distance, $AD(a, b, c, d) = \delta(a - b, c - d) = \delta(d, d')$, it is then natural to consider what we call the *nearest analogical neighbour* (or **nan**) of x from a sample S as the element of $A_E^Y(S)$ defined as:

$$\forall x \in X, \forall S \subseteq X, 1\text{-nan}(x, S) \stackrel{\text{def}}{=} \arg \min_{d' \in A_E^Y(S)} \delta(x, d')$$

When there is more than one nan, one can either proceed to a majority vote procedure among all their analogical labels, or randomly select one of these. This last option is the one we chose in our implementation.

Property 1 *We have the following equality:*

$$1\text{-nan}(x, S) = 1\text{-nn}(x, A_E^Y(S)).$$

The analogical classification rule simply is:

$$\hat{x} = \overline{1\text{-nan}(x, S)}.$$

In words, the predicted label of an element x is the analogical label of its nearest neighbour in $A_E^Y(S)$. In some sense, an analogical classifier behaves as a NN classifier but on an extended sample set.

Obviously if x belongs to $A_E^Y(S)$ then x is its own nearest analogical neighbour: $1\text{-nan}(x, S) = x$ iff $x \in A_E^Y(S)$. Therefore, it is easy to see that this rule is a generalisation of the conservative approach. Instead of using only one nearest analogical neighbour, we can consider the set of the k nearest analogical neighbours, and implement a majority vote as it is done in [21].

The above definition leads to understand the process of analogical classification as follows:

1. First, extend the sample set S to its analogical extension $A_E^Y(S)$. $A_E^Y(S)$ can be viewed as an extended sample set that has **class noise**: the label associated with elements in $A_E^Y(S) \setminus S$ is their analogical label (as defined in 3.1), which may not be correct.
2. Then just apply a classical k -NN strategy over this extended sample set.

Figure 2 gives an illustration of the classification process: the label of $x \in X$ is unknown, and we set it to that of $d' \in A_E^Y(S)$ (a circle), which is its nearest analogical neighbour. To show that the analogical label of d' has itself been inferred, it is depicted as transparent instead of plain black. Let us note that topologically speaking,

Figure 2. A graphical view of $A_E^Y(S)$ and the classification process.

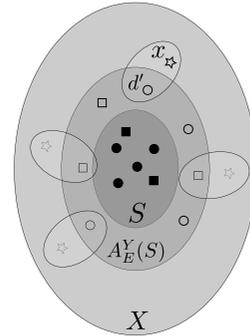


Figure 2 is not representative of a real case: even if we always have $S \subseteq A_E^Y(S) \subseteq X$, this does not mean that these sets are embedded into one another as shown in the drawing. Actually, elements of S (and thus of $A_E^Y(S)$) are usually scattered over the whole universe.

As far as we know, this is the first time a functional definition of analogy-based classifiers is given. This definition clearly fits with the known algorithms but obviously, some implementation details cannot be exactly caught up by such a high level description. It is indeed possible to find a few edge cases where this functional definition may not output the same result as algorithm 2: this is the case for example when the nan of x is not unique. It is also the case when the closest vertex d' does not belong to B^m . However, as we will see in Section 6 these cases are not likely to occur and both approaches produce very similar results, thus empirically validating this functional definition.

Since we now have a clear functional definition of analogical classifiers, we are in position to examine some general properties such as convergence and VC-dimension of analogical learners. This is the purpose of the next section.

4 Some properties in the real case

Let us consider the case where $X = \mathbb{R}^m$, δ any distance issued from a norm, $AD(a, b, c, d) = \delta(a - b, c - d)$ and $x \in X$. In any case, just because $S \subseteq A_E(S)$, we have the following inequality:

$$\delta(x, 1\text{-nan}(x, S)) \leq \delta(x, 1\text{-nn}(x, S))$$

4.1 Study of convergence

Now, let us consider $x^{(i)}$ an i.i.d. sequence of random variables in \mathbb{R}^m , where \mathbb{R}^m is equipped with a probability measure denoted P . As the set $S_n = \{x^{(i)}, i \in [1, n]\}$ is random, then $1\text{-nan}(x, S_n)$ can also be considered as a random element of X . We then are in the exactly same context as the work of Cover & Hart ([9]), and we obtain the same result:

Property 2 $\text{plim}_{n \rightarrow \infty}(1\text{-nan}(x, S_n)) = x$ almost surely,

where plim is the probability limit operator.

Proof. Exactly the same proof as in [9] could be applied. But it is simpler to remember that $\delta(x, 1\text{-nan}(x, S_n)) \leq \delta(x, 1\text{-nn}(x, S_n))$. Then, for a given x , the convergence in probability of $\delta(x, 1\text{-nn}(x, S_n))$ to 0 implies the convergence in probability of $\delta(x, 1\text{-nan}(x, S_n))$ to 0 which exactly means what needs to be proven. The subset of X where $\text{plim}_{n \rightarrow \infty}(1\text{-nan}(x, S_n)) \neq x$ is included into the subset of X where $\text{plim}_{n \rightarrow \infty}(1\text{-nn}(x, S_n)) \neq x$: Cover & Hart lemma tells us that this set has probability 0. Thus the final result. ■

Let us note the following points:

1. The lemma of Cover and Hart is more general than the one above. They have proven the result for any separable metric space, without any additional information. In fact, we cannot follow these lines here just because there is no known way to define an analogical dissimilarity on a metric space, without the help of other structure or operator (see [21] for a detailed discussion on this issue).
2. This result does not say anything regarding the prediction accuracy of 1-nan prediction rule as it is rather different than the 1-nn rule. Such consideration will be investigated in Section 5.
3. We have to be careful about the interpretation of this property in terms of machine learning. Indeed, a stronger property is proved in [9]: for an *integrable* function f over \mathbb{R}^m w.r.t. the probability measure P , the expectation of $f(1\text{-nn}(x, S_n)) - f(x)$ converges to 0 when n goes to infinity. This means that asymptotically, the nearest neighbour of x has the same properties as x , and then the same label. Such a property has not yet been proven for $1\text{-nan}(x, S_n)$.
4. Finally, it is clear that when n goes to infinity, the behavior of an analogical classifier tends to that of a nearest neighbours classifier. Indeed, when S_n is very big, the nearest analogical neighbour of an element x simply is its nearest neighbour, in most cases. Moreover, when the nan and the nn are too close, paying the price of the noise related to the nan may not be worth it. This supports the common acknowledgement that analogical reasoning is mostly useful when very few data are available. In this later case extending a small sample set with its analogical extension may be particularly beneficial.

4.2 VC-dimension

The notion of VC-dimension was originally defined by Vapnik and Chervonenkis [32], and introduced into learnability theory by Blumer et al. [5]. Roughly speaking, the VC-dimension of a class of learners is a numerical measure of their discrimination power. It appears that this number is strongly linked to the confidence interval between the empirical risk (i.e. the error a learner makes on the sample set) and the true risk (the error a learner makes on the whole universe X). As such, the VC-dimension of a class of learners is an essential element of their theoretical study. We consider a universe X (usually a Cartesian product to represent the data) and a family $\mathcal{H} = \{h_i \subseteq X | i \in I\}$ of subsets of X . The elements of \mathcal{H} will be referred as hypothesis or models. Given a subset A of X , we can consider the new family of subsets $\text{tr}(\mathcal{H}, A) = \{h_i \cap A \subseteq X | i \in I\}$: this family is called the *trace* of \mathcal{H} over A . This is obviously a subset of the power set of A , 2^A i.e. $\text{tr}(\mathcal{H}, A) \subseteq 2^A$. We say that \mathcal{H} shatters A iff $\text{tr}(\mathcal{H}, A) = 2^A$. $VC\text{-dim}(\mathcal{H})$ is then the size of the largest finite subset which can be shattered by \mathcal{H} :

Definition 1 $VC\text{-dim}(\mathcal{H}) = \bigsqcup \{|A| \mid \mathcal{H} \text{ shatters } A\}$,

where \bigsqcup is the least upper bound operator. In the case where $\forall n \in \mathbb{N}, \exists A \subset X, |A| = n$ such that \mathcal{H} shatters A , we simply say that:

$$VC\text{-dim}(\mathcal{H}) = \infty.$$

As a binary classifier c over X defines a subset of X with $c^{-1}(1) = \{x \in X | c(x) = 1\}$, we can associate to a class \mathcal{C} of classifiers a family of subsets $\{c^{-1}(1) | c \in \mathcal{C}\}$ and then the VC-dimension of a set of classifiers is as below:

Definition 2 $VC\text{-dim}(\mathcal{C}) = VC\text{-dim}(\{c^{-1}(1) | c \in \mathcal{C}\})$

For instance with $X = \mathbb{R}^n$ and with \mathcal{C} the family of the k -NN classifiers: $\mathcal{C}_{\text{NN}} = \{k\text{-NN classifiers}, k \in \mathbb{N}^*\}$, then $VC\text{-dim}(\mathcal{C}_{\text{NN}}) = \infty$. Let us now consider the family of analogical binary classifiers \mathcal{AC}_k whose classification rule is as below (where a majority vote is implemented):

$$\mathcal{AC}_k(x, S) = \overline{k\text{-nan}(x, S)}$$

In fact, an immediate result comes, derived from the core definition of an analogical proportion:

Property 3 $VC\text{-dim}(\mathcal{AC}_k) = \infty$

Proof. Given any x , the analogical proportion $x : x :: x : x$ always holds so that $1\text{-nan}(x, S) = x$ then the label \hat{x} allocated to x by \mathcal{AC}_1 is just \bar{x} , which by definition equals \hat{x} . It means any set of items can be exactly labelled, thus the infinite $VC\text{-dim}$. ■

Regarding Property 3, the \mathcal{AC}_k class behaves exactly as the k -NN class. Let us note that this is a very general result, which does not rely on any definition of distance. This is directly coming from a core property of analogical proportions.

5 Accuracy analysis in the Boolean case

In this section, we study the accuracy of an analogical classifier, and more particularly that of the 1-nan classifier (NaN). To do so, we restrict our view to a Boolean setting: elements to be classified belong to $X = \mathbb{B}^m$ and the label space is $Y = \mathbb{B}$.

As explained in Section 3.3, for a given $x \in X$ and a sample set $S \subset X$, we have:

$$\hat{x} = \overline{1\text{-nan}(x, S)} = \overline{1\text{-nn}(x, A_E^Y(S))}, \quad (1)$$

where $A_E^Y(S)$ is the analogical extension of S , that we will simply denote by A_E in what follows for notational brevity. We also denote $A_E^* \stackrel{\text{def}}{=} A_E \setminus S$ as the set of elements that belong A_E but not to S .

We now equip the set X with a probability distribution denoted P . The accuracy of the NaN_S classifier⁵ over all the elements of X is defined as:

$$\text{Acc}(\text{NaN}_S, X) \stackrel{\text{def}}{=} P(\hat{x} = \dot{x} \mid x \in X)$$

By observing that for any x , its 1-nan either belongs to S or to A_E^* , the above equation can be split into two distinct parts as follows:

$$\begin{aligned} P(\hat{x} = \dot{x} \mid x \in X) &= P(\overline{[1\text{-nan}(x, S) = \dot{x}]} = \dot{x}) \\ &= P\left(\overline{[1\text{-nan}(x, S) = \dot{x}]} \wedge [1\text{-nan}(x, S) \in S]\right) + \\ &\quad P\left(\overline{[1\text{-nan}(x, S) = \dot{x}]} \wedge [1\text{-nan}(x, S) \in A_E^*]\right) \\ &= P\left(\overline{[1\text{-nn}(x, A_E) = \dot{x}]} \wedge [1\text{-nn}(x, A_E) \in S]\right) + \\ &\quad P\left(\overline{[1\text{-nn}(x, A_E) = \dot{x}]} \wedge [1\text{-nn}(x, A_E) \in A_E^*]\right) \\ &= P\left(\overline{[1\text{-nn}(x, A_E) = \dot{x}]} \mid [1\text{-nn}(x, A_E) \in S]\right) \times \\ &\quad P([1\text{-nn}(x, A_E) \in S]) + \\ &\quad P\left(\overline{[1\text{-nn}(x, A_E) = \dot{x}]} \mid [1\text{-nn}(x, A_E) \in A_E^*]\right) \times \\ &\quad P([1\text{-nn}(x, A_E) \in A_E^*]) \end{aligned}$$

Let us denote $\alpha \stackrel{\text{def}}{=} P(1\text{-nn}(x, A_E) \in S)$ ⁶. The formula becomes:

$$\begin{aligned} \text{Acc}(\text{NaN}_S, X) &= \\ &P\left(\overline{[1\text{-nn}(x, A_E) = \dot{x}]} \mid [1\text{-nn}(x, A_E) \in S]\right) * \alpha + \\ &P\left(\overline{[1\text{-nn}(x, A_E) = \dot{x}]} \mid [1\text{-nn}(x, A_E) \in A_E^*]\right) * (1 - \alpha). \end{aligned}$$

Let us focus on the first term (discarding the factor α):

$$P\left(\overline{[1\text{-nn}(x, A_E) = \dot{x}]} \mid [1\text{-nn}(x, A_E) \in S]\right)$$

It is easy to see that the event $[1\text{-nn}(x, A_E) \in S]$ is equivalent to the event $[1\text{-nn}(x, A_E) = 1\text{-nn}(x, S)]$. As a result, we can transform the first term to get a better grasp of its meaning:

$$\begin{aligned} &P\left(\overline{[1\text{-nn}(x, A_E) = \dot{x}]} \mid [1\text{-nn}(x, A_E) \in S]\right) \\ &= P\left(\overline{[1\text{-nn}(x, A_E) = \dot{x}]} \mid [1\text{-nn}(x, A_E) = 1\text{-nn}(x, S)]\right) \\ &= P\left(\overline{[1\text{-nn}(x, S) = \dot{x}]} \mid [1\text{-nn}(x, A_E) \in S]\right). \end{aligned}$$

In this form, the first term is just the accuracy of the NN_S algorithm over the elements that have their nearest analogical neighbour in S . As for the second term, the same process can be applied by observing that the event $[1\text{-nn}(x, A_E) \in A_E^*]$ is equivalent to the event $[1\text{-nn}(x, A_E) = 1\text{-nn}(x, A_E^*)]$. This leads to

$$\begin{aligned} &P\left(\overline{[1\text{-nn}(x, A_E) = \dot{x}]} \mid [1\text{-nn}(x, A_E) \in A_E^*]\right) \\ &= P\left(\overline{[1\text{-nn}(x, A_E^*) = \dot{x}]} \mid [1\text{-nn}(x, A_E) \in A_E^*]\right). \end{aligned}$$

⁵The S subscript is here to specify that the training set of the NaN algorithm is S . The same notation is used for the *nearest neighbour* algorithm: NN_Σ is the NN algorithm trained on the set Σ .

⁶Obviously, we also have $\alpha = P(1\text{-nan}(x, S) \in S)$.

This second term is then the accuracy of the $\text{NN}_{A_E^*}$ algorithm over the elements that have their nearest analogical neighbour in A_E^* .

In the light of these interpretations, one can rewrite the accuracy formula in a concise form, using a few more definitions:

- $A \stackrel{\text{def}}{=} \{x \in X, 1\text{-nan}(x, S) \in S\}$: the elements that have their nan in S .
- $B \stackrel{\text{def}}{=} \{x \in X, 1\text{-nan}(x, S) \in A_E^*\}$: the elements that have their nan in A_E^* .

Naturally, $A \cup B = X$ and $A \cap B = \emptyset$. Also, $\alpha = P(x \in A)$ and $1 - \alpha = P(x \in B)$. Therefore, the accuracy of NaN_S over X can be understood as the weighted sum of the accuracy of NN over A and B , using a different sample set each time (respectively S and A_E^*):

$$\text{Acc}(\text{NaN}_S, X) = \text{Acc}(\text{NN}_S, A) \cdot \alpha + \text{Acc}(\text{NN}_{A_E^*}, B) \cdot (1 - \alpha). \quad (2)$$

The value $\text{Acc}(\text{NN}_S, A)$ is the accuracy of NN_S over all the elements in A . A theoretical study of this accuracy has been done in [19] when the size of A is known. Regarding $\text{Acc}(\text{NN}_{A_E^*}, B)$, this is the accuracy of 1-nn when the sample set is noisy, and has been studied in [24]. This last formula leads to the consistent facts:

1. The smaller A_E^* (i.e. analogical reasoning does not bring much more labels), the closer α is to 1, the closer A is to X and the more the accuracy of NaN_S tends towards the accuracy of NN_S over X .
2. In return, if A_E is much bigger than S , α is then small, B is close to X and the accuracy of NaN_S greatly depends on the quality of A_E , which can be measured by the value ω defined as:

$$\omega \triangleq P(\bar{x} = \dot{x} \mid x \in A_E^*).$$

Note that the value $1 - \omega$ corresponds to the class noise of A_E . As we will see in the next section, this situation where A_E is big with respect to S is actually extremely likely to occur.

6 Experiments and empirical validation

In order to get an empirical validation of our formulas, we have developed a set of experiments that we describe in the next subsection.

6.1 Validation protocol

Working with Boolean vectors, we have computed the accuracies of the NaN and NN algorithms over $X = \mathbb{B}^m$ for different values of m (namely 8 and 10). The ground truth label of elements of X is defined by different Boolean functions f in such a way that the $\forall x = (x_1, \dots, x_m)$, $\dot{x} = f(x)$. The different functions we have worked with are:

- $f(x) = x_m$: in that case, we can consider the $m - 1$ first parameters are a kind of noise since they have no influence on the final label ;
- $f(x) = 1$ iff at least l components are equal to 1 (this kind of function is usually called l -of- m). We chose to set l to $\frac{m}{2}$;
- $f(x) = x_1 \oplus x_2$ (xor): we here have $m - 2$ useless attributes ;
- $f(x) = 1$ iff $\sum x_i = 2$: all the attributes are relevant in that case ;
- $f(x) = 1$ iff $\sum x_i = m - 1$: an extreme case of the previous one ;

- $f(x) = 1$ iff $x_1 \cdot x_m = 1$: only the first and the last elements are relevant.

Regarding the size of the training set, to be sure to fit with the size of the universe, we have investigated various sizes between 3 and 100. When dealing with a training set of size 100, the cubic complexity of the analogical classifier leads to explore a set of approximately 100^3 elements: as a consequence, we limit our investigation to a maximum of 100 elements in the training set in order to get realistic execution time.

All the accuracy (and other metrics) computations are averaged over a set of 100 experiments. The interested reader may find the Python source code that has generated all our plots and detailed results on Github⁷. For lack of space, we only provide a few examples which are representative of the global behavior. Please note that our implementation of the NaN algorithm is not that of algorithm 2, but is instead that of the functional definition of the analogical classifier developed in Section 3.3: we first construct the analogical extension set of S , and then proceed to a nearest neighbour strategy over this noisy extended training set. We have estimated probabilities by frequencies, thus implicitly assuming a uniform distribution on X .

In addition to these Boolean functions, we have also run the NaN algorithm over the Monk datasets over the UCI repository⁸. They are datasets of 432 binarized elements, among which exactly 169 of them have been used for training.

6.2 Experiments

Figure 3 shows the accuracies (left column) of the NaN and NN over six different Boolean settings with values of $|S|$ varying from 3 to 100. In the right column, we have plotted three different values that will help us analyse and validate the behaviour of the NaN algorithm:

- the theoretical accuracy as defined by equation (2) in Section 5. The probability $\alpha = P(x \in A)$ has been estimated by the frequency: $\frac{|A|}{|X|}$;
- the quality of $A_E^Y(S)$, measured by ω as defined in Section 5 which is estimated by the frequency: $\frac{|\{x \in A_E^* \mid \bar{x} = \hat{x}\}|}{|A_E^*|}$;
- finally, the quantity $\gamma = \frac{|A_E^Y(S)|}{|X|}$: the size of the analogical extension set with respect to that of the whole universe.

Table 1 shows the same metrics for the Monk datasets and also report the results of the Analogical Proportion Classifier (APC) from [21], which corresponds to algorithm 2 with $k = 100$.

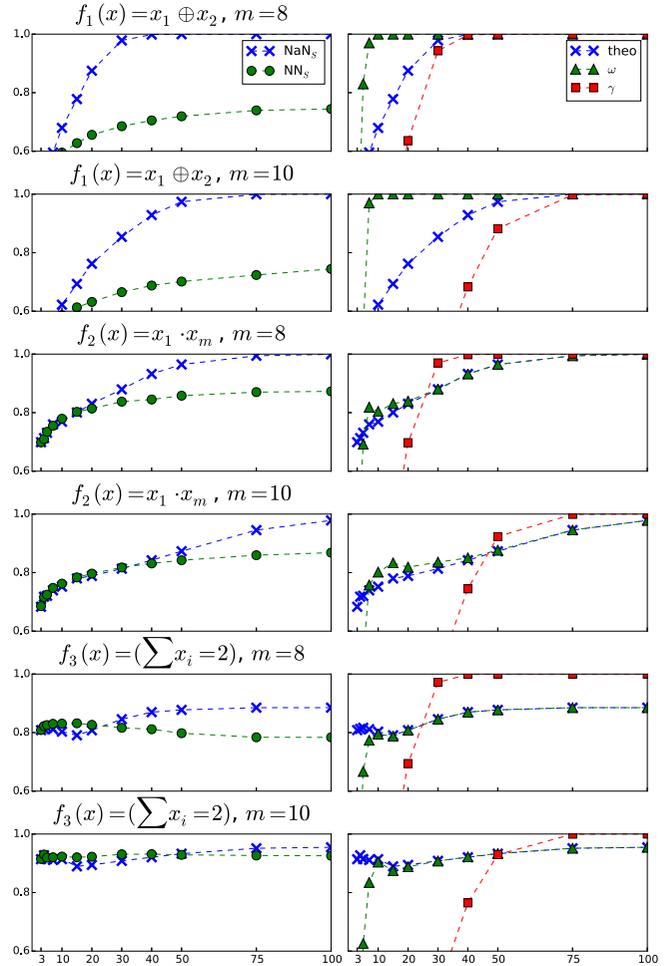
Table 1. Accuracies of the NaN, APC and NN algorithms over the Monk datasets

	NaN	APC	NN	ω	γ
Monk 1	.961	.98	.787	.961	1
Monk 2	.998	1	.738	.996	1
Monk 3	.963	.96	.829	.963	1

6.3 Comments and discussion

The experiments shown in figure 3 allow us to draw interesting conclusions about the behaviour of the NaN algorithm. We can observe one of the two cases:

Figure 3. Accuracies of the NaN and NN algorithms over different Boolean settings and training set sizes, with corresponding values of ω , γ , and theoretical accuracy. The x axis corresponds to the size of the training set.



- either the analogical labels are always correctly predicted⁹ ($\omega = 1$, i.e. there is no class noise) as it is the case for f_1 , $f(x) = x_m$ and (almost) for the Monk datasets ;
- or there is some class noise in $A_E^Y(S)$ ($\omega \neq 1$). In this case, we always observe that the NaN algorithm is outperformed by NN for small values of $|S|$, but eventually takes advantage once analogical prediction becomes more important than the nearest neighbour one, as we are going to see.

The theoretical accuracy seems to fit perfectly with the empirical accuracy of the NaN algorithm, thus validating our theoretical study that led to equation (2)¹⁰.

An interesting observation is that the value of ω always converges to that of the theoretical accuracy (and therefore to the actual accuracy) of NaN. This can be easily explained by paying attention to the value of γ , the proportion of elements of X that belong to $A_E^Y(S)$. We see that in any setting, γ converges to 1 as $|S|$ grows. This means that when $|S|$ is big enough (but not necessarily that big with re-

⁹Note that for small values of $|S|$, it seems that $\omega \neq 1$. This is due to the fact that for such small values, it is sometimes impossible to construct $A_E^Y(S)$, thus leading to a value of $\omega = 0$ (which will be averaged afterwards over the 100 experiments).

¹⁰The maximal difference we observed between the theoretical accuracy and its actual value is of about 10^{-10} .

⁷https://github.com/Niourf/nan_study

⁸<https://archive.ics.uci.edu/ml/datasets/MONK's+Problems>

spect to X), the analogical extension of S covers the whole universe X^{11} : every element x is then its own nearest analogical neighbour and $\hat{x} = \bar{x}$. It is therefore straightforward to see that in this case,

$$\begin{aligned}\omega &= P(\bar{x} = \hat{x} \mid x \in A_E^*) = P(\hat{x} = \hat{x} \mid x \in A_E^*) \\ &= \text{Acc}(\text{NaN}_S, A_E^*)\end{aligned}$$

When $\gamma = 1$, the only elements x we want to classify belong to A_E^* (otherwise they would be in S), so this last term exactly corresponds to the accuracy of the classifier. Another way to see it is to observe that the first term of equation (2) $\text{Acc}(\text{NN}_S, A) \cdot \alpha$ is null because $\alpha = 0$. Only the second term $\text{Acc}(\text{NN}_{A_E^*}, B) \cdot (1 - \alpha)$ is of importance, and its value corresponds to ω . This observation allows us to state that estimating the value of ω is paramount to have a precise idea of the accuracy of an analogical classifier. We will provide in the next subsection a method to accurately estimate this quantity ω with the only help of the training set S .

Regarding the Monk datasets (Table 1), we note that the functional NaN approach (almost) achieves the same results as the somewhat more complex algorithm described in Section 3.2, and that here again the analogical extension set covers the whole universe: this means that a conservative approach would have been sufficient! Actually, this raises the following question: why would we want to look for more than one analogical neighbour when every element of the universe is already in $A_E^*(S)$, and therefore *analogically linked* to those in S ? Our experiments tend to show that this becomes superfluous, provided that the training set is big enough.

6.4 Estimation of the prediction accuracy

We have seen in the previous subsection that the value ω is that of the actual accuracy of an analogical classifier when S is big enough. This leads to the following question: how can we get a precise estimation of this value ω ? Answering this would allow us to have a very precise idea of the accuracy we can expect from our classifier.

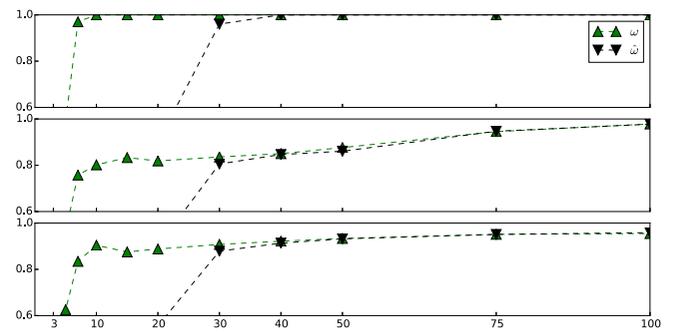
The method we propose for estimating ω only relies on the training set S and is very simple: it consists of applying the conservative algorithm to all the elements of S , and compute the fraction of these elements that have been correctly classified. A small yet important modification to the algorithm needs to be added: we only want to construct analogical proportions of the form $a : b :: c : x$ where a, b, c and x are all distinct elements. Indeed, the proportions $x : x :: x : x$ and $x' : x :: x' : x$ are always true, and the solution label related to these proportions would bias the final majority vote procedure in a significant way towards the real label \hat{x} .

We have applied this estimation protocol to all of the Boolean settings we have considered, and it has shown to be very accurate. Figure 4 illustrates a few of these settings (already considered in Figure 3). We can see that the estimation $\hat{\omega}$ converges to ω when S is big enough. For small values of S , this estimation is indeed imprecise as it is difficult to find a lot of 3-tuples such that an analogical proportion holds for every element.

7 Conclusion

In this paper, we have provided a functional definition of analogical learners. Starting from this definition, we are in a position to prove an analytic convergence result, similar to that of the nearest neighbour algorithm. Obviously, this is not enough to conclude regarding the

Figure 4. Values of ω and its estimation $\hat{\omega}$ for f_1, f_2 and f_3 in \mathbb{B}^{10} .



predictive ability of analogy-based classifiers. We have also shown that their VC-dimension is infinite. It should not come as a surprise, as a very particular case of analogical rule (when the analogical proportion is trivial) is the k -NN rule.

In terms of accuracy in a Boolean setting, we have found a strong link between the accuracy of the NaN_S algorithm and that of the NN_S algorithm. At a first glance, we can consider the NaN algorithm as a NN strategy on an extended and noisy sample set: the analogical extension of S . In the end, we have seen that this extended sample set covers the entire universe provided that S is big enough, simplifying and bringing back the accuracy of the classifier to the value ω which corresponds to the quality of the analogical extension. We have also provided a method to accurately estimate the value of ω that only relies on elements of the S , thus allowing beforehand to have a precise idea of the accuracy of any analogical classifier in a Boolean setting. Some important points remain to be investigated, such as:

- What can we expect in terms of speed convergence from an analogical learner? In other words, what is the minimum size needed from a sample set to get a fixed accuracy threshold?
- If a clever learning strategy can (at least partially) overcome the problem of infinite VC-dimension, can we overcome the issue of the cubic complexity of analogical learners?
- Leaving the field of classification, can we provide a clear strategy for transfer learning with analogy? Indeed, the central goal of transfer learning is to identify and exploit analogies between source and target domains [25].

These points definitely constitute interesting challenges for future works. Nevertheless, we have to remember that analogical reasoning brings its whole power in the case where few data are available. If a lot of data are available, it is very likely that we have elements similar to the one at hand and, in that case, a k -NN style reasoning is natural. In the opposite case, when we only have a few relevant cases at hand, applying analogical proportion-based predictions appears to be a meaningful option.

¹¹Obviously, the bigger the dimension m , the slower the convergence occurs.

REFERENCES

- [1] C. C. Aggarwal, *Data Classification: Algorithms and Applications*, Data Mining and Knowledge Discovery Series, Chapman and Hall/CRC, 2014.
- [2] N. Barbot and L. Miclet, 'La proportion analogique dans les groupes: applications aux permutations et aux matrices', Technical Report 1914, IRISA, (July 2009).
- [3] M. Bayouh, H. Prade, and G. Richard, 'Evaluation of analogical proportions through kolmogorov complexity', *Knowl.-Based Syst.*, **29**, 20–30, (2012).
- [4] S. Bayouh, L. Miclet, and A. Delhay, 'Learning by analogy: A classification rule for binary and nominal data', *Proc. Inter. Joint Conf. on Artificial Intelligence IJCAI07*, 678–683, (2007).
- [5] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth, 'Learnability and the vapnik-chervonenkis dimension', *J. ACM*, **36**(4), 929–965, (1989).
- [6] M. Bounhas, H. Prade, and G. Richard, 'Analogical classification: A new way to deal with examples', in *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pp. 135–140. IOS Press, (2014).
- [7] A. Cornuéjols, 'Analogy as minimization of description length', in *Machine Learning and Statistics: The interface*, eds., G. Nakhaeizadeh and C. Taylor, pp. 321–336. Wiley and Sons, (1996).
- [8] W. Correa, H. Prade, and G. Richard, 'When intelligence is just a matter of copying', in *Proc. 20th Eur. Conf. on Artificial Intelligence, Montpellier, Aug. 27-31*, pp. 276–281. IOS Press, (2012).
- [9] T. M. Cover and P. E. Hart, 'Nearest neighbor pattern classification', in *IEEE Transactions on Information Theory*, volume 13, 21–27, IEEE, (1967).
- [10] T. R. Davies and Russell S. J., 'A logical approach to reasoning by analogy', in *Proc. of the 10th International Joint Conference on Artificial Intelligence (IJCAI'87), Milan*, ed., J. P. McDermott, 264–270, Morgan Kaufmann, (1987).
- [11] M. Dorolle, *Le Raisonnement par Analogie*, PUF, Paris, 1949.
- [12] D. Gentner, 'Structure-mapping: A theoretical framework for analogy', *Cognitive Science*, **7**(2), 155–170, (1983).
- [13] D. Gentner, K. J. Holyoak, and B. N. Kokinov, *The Analogical Mind: Perspectives from Cognitive Science*, Cognitive Science, and Philosophy, MIT Press, Cambridge, MA, 2001.
- [14] H. Gust, K. Kühnberger, and U. Schmid, 'Metaphors and heuristic-driven theory projection (HDTP)', *Theoretical Computer Science*, **354**(1), 98 – 117, (2006).
- [15] T. Hinrichs and K. D. Forbus, 'Transfer learning through analogy in games', *AAAI*, **32**(1), 70–83, (2011).
- [16] D. Hofstadter and M. Mitchell, 'The Copycat project: A model of mental fluidity and analogy-making', in *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*, eds., D. Hofstadter and The Fluid Analogies Research Group, pp. 205–267, New York, NY, (1995). Basic Books, Inc.
- [17] D. Hofstadter and E. Sander, *Surfaces and Essences: Analogy as the Fuel and Fire of Thinking*, Basic Books, 2013. French version: *L'Analogie, Cœur de la Pensée - Odile Jacob*, Paris.
- [18] D. R. Hofstadter and The Fluid Analogy Research Group, *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*, Basic Books, Inc., New York, NY, USA, 1996.
- [19] P. Langley and W. Iba, 'Average-case analysis of a nearest neighbor algorithm', in *Proceedings of the 13th Int. Joint Conf. on Artificial Intelligence. Chambéry, France, August 28 - September 3*, pp. 889–894. Morgan Kaufmann, (1993).
- [20] Y. Lepage, 'De l'analogie rendant compte de la commutation en linguistique', *Habilit. à Diriger des Recher., Univ. J. Fourier, Grenoble*, (2003).
- [21] L. Miclet, S. Bayouh, and A. Delhay, 'Analogical dissimilarity: Definition, algorithms and two experiments in machine learning.', *J. Artif. Intell. Res. (JAIR)*, **32**, 793–824, (2008).
- [22] L. Miclet and A. Delhay, 'Relation d'analogie et distance sur un alphabet défini par des traits', Technical Report 1632, IRISA, (July 2004).
- [23] L. Miclet and H. Prade, 'Handling analogical proportions in classical logic and fuzzy logics settings', in *Proc. 10th Eur. Conf. on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (EQARU'09), Verona*, pp. 638–650. Springer, LNCS 5590, (2009).
- [24] S. Okamoto and N. Yugami, 'An average-case analysis of the k-nearest neighbor classifier for noisy domains', in *Proceedings of the Fifteenth Int. Joint Conf. on Artificial Intelligence, IJCAI 97, Nagoya, Japan, August 23-29*, pp. 238–245. Morgan Kaufmann, (1997).
- [25] S. J. Pan and Q. Yang, 'A survey on transfer learning', *IEEE Trans. on Knowl. and Data Eng.*, **22**(10), 1345–1359, (2010).
- [26] H. Prade and G. Richard, 'From analogical proportion to logical proportions', *Logica Universalis*, **7**(4), 441–505, (2013).
- [27] *Computational Approaches to Analogical Reasoning: Current Trends*, eds., H. Prade and G. Richard, volume 548 of *Studies in Computational Intelligence*, Springer, 2014.
- [28] H. Prade, G. Richard, and B. Yao, 'Enforcing regularity by means of analogy-related proportions-a new approach to classification', *International Journal of Computer Information Systems and Industrial Management Applications*, **4**, 648–658, (2012).
- [29] M. Ragni and S. Neubert, 'Solving Raven's IQ-tests: An AI and cognitive modeling approach', in *Proc. 20th Europ. Conf. on Artificial Intelligence (ECAI'12) Montpellier, Aug. 27-31*, eds., L. De Raedt, C. Bessière, D. Dubois, P. Doherty, P. Frasconi, F. Heintz, and P. J. F. Lucas, pp. 666–671, (2012).
- [30] J. F. Sowa and A. K. Majumdar, 'Analogical reasoning', in *Proc. Inter. Conf. on Conceptual Structures*, pp. 16–36. Springer, LNAI 2746, (2003).
- [31] N. Stroppa and F. Yvon, 'Analogical learning and formal proportions: Definitions and methodological issues', Technical Report D004, ENST-Paris, (2005).
- [32] V. Vapnik, *Statistical Learning Theory*, Wiley, 1998.
- [33] H. Wang and Q. Yang, 'Transfer learning by structural analogy', in *Proc. of the 25 AAAI Conference on Artificial Intelligence*, pp. 513–518. AAAI Press, (2011).

Uncertainty-Sensitive Reasoning for Inferring sameAs Facts in Linked Data

Mustafa Al-Bakri^{1,2} and Manuel Atencia¹ and Jérôme David¹
Steffen Lalande² and Marie-Christine Rousset^{1,3}

Abstract. Discovering whether or not two URIs described in Linked Data — in the same or different RDF datasets — refer to the same real-world entity is crucial for building applications that exploit the cross-referencing of open data. A major challenge in data interlinking is to design tools that effectively deal with incomplete and noisy data, and exploit uncertain knowledge. In this paper, we model data interlinking as a reasoning problem with uncertainty. We introduce a probabilistic framework for modelling and reasoning over uncertain RDF facts and rules that is based on the semantics of probabilistic Datalog. We have designed an algorithm, ProbFR, based on this framework. Experiments on real-world datasets have shown the usefulness and effectiveness of our approach for data linkage and disambiguation.

1 INTRODUCTION

Linked Data provides access to huge, continuously growing amounts of open data in RDF format that describe properties and links on entities referenced by Uniform Resource Identifiers (URIs). Data interlinking consists in deciding whether two URIs refer to the same real-world entity. This is a crucial task for developing innovative applications on top of Linked Data, that exploit the cross-referencing of data [16, 12]. This task is often referred to as data linkage, but it is also known as record linkage and entity resolution, and it has been widely studied for the case of relational data [9]. As regards Linked Data, data interlinking is especially challenging since (1) tools need to scale well with large amounts of data, (2) data is frequently described using heterogeneous vocabularies (ontologies), and (3) tools need to deal with uncertain data as Linked Data contains data which is inherently incomplete, and very often noisy.

In the context of Linked Data and RDF data, different approaches to data interlinking have been proposed. Most of them are based on numerical methods that use linkage rules to compare property values of resources, using similarity measures to handle noisy data. They conclude weighted sameAs links, from which the links with higher weights are expected (but never guaranteed) to be correct [29, 19]. These approaches suffer from two weaknesses. First, rules cannot be chained, as they are thought to be applied only once; and second, weights are combined in a non-formal manner, since there is no formal semantics that captures the combination of weights. A few other works take a logical approach to data interlinking and use logical rules equipped with full reasoning [24, 2]. They make use of uniqueness constraints (such as inverse functional properties and

keys) and other schema constraints, domain knowledge and alignments between different vocabularies which can be modelled as logical rules. They enable rule chaining to infer sameAs links. Logical approaches applying only certain rules over clean and complete data guarantee to provide sound results, i.e., a 100% precision. However, the recall may be low because in Linked Data, data is inherently incomplete and possibly noisy. Input facts may be missing to trigger rules, either because some values for properties involved in rules conditions are absent for some URIs, or because some of these values are noisy with some misspelling that prevents some conditions to be satisfied. In addition, rules may be missing to infer sameAs facts with certainty, although some strong evidence could be obtained from the combination of soft constraints.

This paper introduces a rule-based approach to data interlinking in the context of Linked Data based on uncertain reasoning for inferring sameAs facts. Our contribution is threefold:

- A declarative framework based on probabilistic Datalog [15] in which uncertain facts are modelled as probabilistic facts, and that allows to model in the form of probabilistic rules different kinds of uncertain knowledge useful for inferring sameAs facts.
- An inference algorithm, ProbFR, that takes as input a dataset — possibly including probabilistic facts — and probabilistic rules, and that computes for each of the inferred facts the probability of the fact to be true, as well as the provenance of this computation.
- A series of experiments done with an implementation of ProbFR over three real-world large RDF datasets that show (1) the gain of using uncertain data and knowledge for data interlinking, (2) the gain of using full uncertain reasoning (rule chaining), and (3) the benefits of having probabilities attached to inferred facts for discarding incorrect sameAs links.

There are two main reasons for our choice of probabilistic Datalog as a basis for our approach. First, Datalog rules on top of RDF facts capture in a uniform way most of the OWL and RDFS constraints that are useful for inferring sameAs facts (which includes inverse functional properties and keys), domain knowledge and alignments between different vocabularies. Probabilistic Datalog, in turn, allows to add uncertainty to knowledge simply by attaching probabilistic symbolic events to rules and facts. Uncertain knowledge may be provided by domain experts or may be learnt by specialised automatic tools (as in the case of weighted ontology mappings [11], pseudo keys [6, 28] and complex link specifications [21]).

Second, when compared to other approaches to probabilistic logical reasoning [23, 7], probabilistic Datalog fits better into the setting of Linked Data. These approaches typically make the close-world assumption and perform a supervised learning of probabilistic weights

¹ Univ. Grenoble Alpes, CNRS, Inria, LIG, F-38000 Grenoble, France.

² Institut National de l’Audiovisuel, F-94366 Bry-sur-Marne, France.

³ Institut Universitaire de France, F-75005 Paris, France

that requires full observation of the domain. However, Linked Data makes the open-world assumption and contains very large datasets, which will make these approaches to suffer from scalability issues. In probabilistic Datalog, the uncertain formulas are restricted to Horn rules and ground atoms. Furthermore, probabilities can only be computed for inferred facts. This is a restricted setting compared to Markov Logic or other statistical relational learning. However, the probabilities can be computed more efficiently and more transparently from the provenance expressions that can be obtained for each inferred fact in a forward-chaining manner.

The remainder of the paper is organised as follows. In Section 2 we describe the probabilistic model and the inference algorithm at the core of our approach, and we also make explicit the underlying assumptions for its effectiveness in the setting of Linked Data. In Section 3 we illustrate by example our approach for modelling uncertain data and knowledge useful for data interlinking. Section 4 shows, through experiments conducted on real-world datasets, the feasibility and the added-value of this approach to discover sameAs links. In Section 5, we position our work with respect to existing works, and finally we conclude in Section 6.

2 PROBABILISTIC FRAMEWORK FOR REASONING OVER UNCERTAIN RDF FACTS AND RULES

We have designed a probabilistic framework to model and reason on uncertain RDF facts and rules based on the semantics of probabilistic Datalog [15]. Probabilistic Datalog extends (deterministic) Datalog [1] by associating each ground fact and each instantiated rule with a probabilistic *event* that the corresponding fact or rule is true. Each derived fact is then inferred with its *provenance* in the form of an event expression made of a boolean combination of the events of the ground facts and rules involved in its derivation. It can be written in disjunctive normal form, in which a conjunction of events represents a derivation branch, and disjunctions represent the different derivation branches. Some simplifications can be done before the computation of the resulting probabilities: a conjunction containing disjoint events can be suppressed; events known to be certain can be removed from the conjunctions where they are involved, thus leading to conjunctions with only uncertain events. An extreme case is when a conjunction is made of certain events only, which represents a way to derive a fact with certainty. In this case the whole event expression can be simplified to \top which denotes certain events.

The logical semantics of the (simplified) event expressions is then the basis for computing the probability of the corresponding derived facts in function of the probabilities assigned to the events identifying the input facts and rules taking part in their derivation. In the general case, computing the probability of the disjunction of conjunctions of events requires knowing the probabilities of all the combinations of events in the expression. In practice — and, in particular, in applications dealing with large amounts of data — only the probabilities of single events will be known. We will then make the same default assumptions of independence or disjointness of single events, as it is usually done in most Information Retrieval models [14]. To meet such assumptions, we have to impose some constraints on the rules that will be explained below.

Probabilistic RDF facts extend the standard data model of Linked Data used to state properties on entities referenced by Uniform Resource Identifiers (URIs). Properties are themselves identified by URIs. Data properties relate entities with literals (e.g. numbers, strings or dates), while object properties relate two entities.

A **probabilistic RDF fact** is an RDF triple $t = (s, p, o)$ (in which the subject s is a URI, the predicate p is a URI, and the object o may be either a URI or a literal) associated with an event key e denoting the probabilistic event that t is true.

A **probabilistic RDF rule** is a rule with variables, associated with an event key denoting the probability that any of its instantiations is true. Rules have the form $r : TP_1(v_1) \wedge \dots \wedge TP_k(v_k) \Rightarrow TP(v)$ where $TP_1(v_1), \dots, TP_k(v_k)$ and $TP(v)$ are *triple patterns* of the form (s^v, p, o^v) in which the subject s^v or the object o^v may be variables. We consider **safe** rules, i.e. rules such that all the variables in the conclusion are also in the condition part.

Each probabilistic RDF fact and rule are assigned a distinct event key, except the certain facts and rules that are assigned the special event key \top denoting events that are certain. For a probabilistic fact f , we will denote by $e(f)$ the probabilistic event e associated with the fact f . We will write $e(r)$ in the case of a probabilistic rule r .

In rules, we also allow conditions $B(\bar{x}, \bar{a})$ where B is a built-in predicate (i.e. a function call), \bar{x} a vector of variables appearing in the triple conditions of the same rule, and \bar{a} may be a non empty set of values of parameters for calling B . The following rule is an example of a rule with a built-in predicate (*Similar*):

$$r_0 : (?x \text{ hasName } ?s_1) \wedge (?y \text{ hasName } ?s_2) \wedge \text{Similar}(?s_1, ?s_2, \text{levenshtein}, 0.2) \Rightarrow (?x \text{ sameName } ?y)$$

For each pair of strings (s_1, s_2) for which the two triple conditions are satisfied by the facts $(i_1 \text{ hasName } s_1)$ and $(i_2 \text{ hasName } s_2)$, *Similar* $(s_1, s_2, \text{levenshtein}, 0.2)$ applies normalised Levenshtein distance *levenshtein* (s_1, s_2) on strings s_1 and s_2 . If this distance is less than 0.2, it will return the corresponding probabilistic fact *Similar* $(s_1, s_2, \text{levenshtein}, 0.2)$ with $1 - \text{levenshtein}(s_1, s_2)$ as probability.

The semantics of a knowledge base $F \cup R$ composed of a finite set of facts F and a finite set of rules R can be given based on the least fixed point of immediate consequence operator T_R defined below.

- Definition 1** • $F, R \vdash_1 f$ iff a rule $TP_1(v_1) \wedge \dots \wedge TP_k(v_k) \Rightarrow TP(v)$ is in R and there exists a mapping θ from its variables to constants such that $f = \theta.TP(v)$ and $\theta.TP_i(v_i) \in F$ for every $i \in [1..k]$.
- $F, R \vdash f$ iff there exists i such that $f \in T_R(F_i)$ where $F_0 = F$ and for every $i \geq 0$, $F_{i+1} = T_R(F_i) = F_i \cup \{f | F_i, R \vdash_1 f\}$.

For safe rules, there exists a unique least fixed point F_n , denoted by $SAT(F, R)$, such that for every $k \geq n$, $F_k = T_R(F_n)$, i.e. there exists a step in the iterative application of the immediate consequence operator for which no new fact is inferred. Several forward-chaining algorithms exist to compute $SAT(F, R)$, in particular the semi-naive bottom-up evaluation in Datalog [1], and the RETE algorithm [13] that is implemented in many rule-based reasoners, including in Semantic Web tools such as Jena.⁴

The semantics of inferred probabilistic facts can be obtained based on their *provenance* defined as boolean combinations of all the events associated with the input facts and rules involved in their inference.

Definition 2 For every fact f in $SAT(F, R)$, the *provenance* of f (denoted by $Prov_{R,F}(f)$) is defined as follows:

- if $f \in F$, then $Prov_{R,F}(f) = e(f)$,
- otherwise, let $R(f)$ be the set of instantiated rules (r, θ) having f as conclusion (i.e. rules $TP_1(v_1) \wedge \dots \wedge TP_k(v_k) \Rightarrow TP(v)$ for which θ is a mapping such that $\theta.TP(v) = f$ and $\theta.TP(v_i) \in SAT(F, R)$ for every $i \in [1..k]$). Then:

⁴ <https://jena.apache.org/documentation/inference/>

$Prov_{R,F}(f) = \bigvee_{(r,\theta) \in R(f)} (e(r) \wedge \bigwedge_{i \in [1..k]} Prov_{R,F}(\theta.TP_i(v_i)))$
 For every fact f in $SAT(F, R)$, its probability $P(f)$ is defined as the probability of its provenance: $P(f) = P(Prov_{R,F}(f))$.

Illustrative example. Let us consider the following probabilistic RDF facts and rules (for which we omit to display the event keys) composed of 5 input facts and of 4 rules expressing different ways to infer sameAs facts between individuals (to have the same name, to have the same name and the same birthdate, to be married to the same individual, or by transitivity of the sameAs relation):

- $f_1 : (i_1 \text{ sameName } i_2)$
- $f_2 : (i_1 \text{ sameBirthDate } i_2)$
- $f_3 : (i_1 \text{ marriedTo } i_3)$
- $f_4 : (i_2 \text{ marriedTo } i_3)$
- $f_5 : (i_2 \text{ sameName } i_4)$
- $r_1 : (?x \text{ sameName } ?y) \Rightarrow (?x \text{ sameAs } ?y)$
- $r_2 : (?x \text{ sameName } ?y), (?x \text{ sameBirthDate } ?y) \Rightarrow (?x \text{ sameAs } ?y)$
- $r_3 : (?x \text{ marriedTo } ?z), (?y \text{ marriedTo } ?z) \Rightarrow (?x \text{ sameAs } ?y)$
- $r_4 : (?x \text{ sameAs } ?z), (?z \text{ sameAs } ?y) \Rightarrow (?x \text{ sameAs } ?y)$

Three derived facts are obtained with their provenance:

- $Prov_{R,F}((i_1 \text{ sameAs } i_2)) = (e(r_1) \wedge e(f_1)) \vee (e(r_2) \wedge e(f_1) \wedge e(f_2)) \vee (e(r_3) \wedge e(f_3) \wedge e(f_4))$
- $Prov_{R,F}((i_2 \text{ sameAs } i_4)) = (e(r_1) \wedge e(f_5))$
- $Prov_{R,F}((i_1 \text{ sameAs } i_4)) = e(r_4) \wedge Prov_{R,F}((i_1 \text{ sameAs } i_2)) \wedge Prov_{R,F}((i_2 \text{ sameAs } i_4))$

The fact $(i_1 \text{ sameAs } i_2)$ can be inferred as a result of 3 different derivation branches (one using the rule r_1 and the input fact f_1 , one using r_2 and f_1 and f_2 , and the third one using r_3 and f_3 and f_4). The second fact $(i_2 \text{ sameAs } i_4)$ results from a single derivation branch using the rule r_1 and the fact f_5 . The last one illustrates how the provenances can be built iteratively during the saturation process: the last derivation step leading to the inference of $(i_1 \text{ sameAs } i_4)$ involves the rule r_4 and two facts inferred at a previous iteration (namely, $(i_1 \text{ sameAs } i_2)$ and $(i_2 \text{ sameAs } i_4)$) for which the provenance must be combined with the event key of r_4 .

These provenance expressions can be simplified by exploiting facts and rules that are certain. For instance, if we know that the two facts f_2 and f_3 are certain as well as the rule r_4 , we can suppress $e(f_2)$, $e(f_3)$ and $e(r_4)$ in the conjuncts of the above expressions because they are all equal to the event \top always true. We now obtain for $Prov_{R,F}((i_1 \text{ sameAs } i_2))$:

$$(e(r_1) \wedge e(f_1)) \vee (e(r_2) \wedge e(f_1)) \vee (e(r_3) \wedge e(f_4))$$

When many facts and several rules are certain, such simplifications lead to a drastic reduction of the size of provenance expressions, which is important for the scalability of the approach in practice.

This example illustrates how the construction and simplification of the provenance can be incorporated into the saturation process and how a given forward-reasoning algorithm can be easily extended to compute the provenance during the inference of corresponding facts.

The ProbFR algorithm. Algorithm 1 describes the ProbFR algorithm that we have implemented and used in our experiments. It starts with the set of initial facts and rules and repeats inference steps until saturation. Each inference step (Line (4) to (15)) triggers all the rules whose conditions can be matched with known facts (i.e. input facts or facts inferred at previous steps). At each iteration, the set Δ contains the facts that have been inferred at the previous iteration. The constraint (expressed in Line (6)) that rules are only triggered if at least one of their conditions can be matched with facts in Δ guarantees that instantiated rules are not triggered twice during the

inference process. The algorithm stops as soon as no new fact has been inferred during a given iteration (i.e. Δ_1 remains empty over this iteration). The algorithm returns the set F_{sat} of inferred facts, and computes for each of them an event expression $x(f)$ (Lines (10) and (11)). The function \mathcal{N}_\vee denotes the transformation of a conjunction into its disjunctive normal form. It consists in applying iteratively the distributivity of the conjunction connector (\wedge) over the the disjunction connector (\vee), and in simplifying when possible the (intermediate) results as follows: (1) remove the duplicate events and the certain events \top from each conjunction of events, (2) if a conjunction within a disjunction becomes empty (i.e. if all its events are certain), replace the whole disjunction by \top . Each event expression $x(f)$ is thus \top or of the form $Conj_1 \vee \dots \vee Conj_l$ where $Conj_i$ is a conjunction of event keys tracing the uncertain input facts and rules involved into one of the l branches of uncertain derivation of f .

Algorithm 1: The ProbFR algorithm

ProbFR(F, R)

Input: A set F of input (probabilistic) facts and a set R of (probabilistic) rules

Output: The set F_{sat} of inferred (probabilistic) facts with for each inferred fact f its event expression $x(f)$

- (1) **for each** $f \in F$: $x(f) \leftarrow e(f)$
- (2) $F_{sat} \leftarrow F$
- (3) $\Delta \leftarrow F$
- (4) **repeat**
- (5) $\Delta_1 \leftarrow \emptyset$
- (6) **foreach** rule $r: c_1 \wedge \dots \wedge c_k \Rightarrow c$ for which there exists a substitution θ and facts $f_1, \dots, f_k \in F_{sat}$ (among which atleast one of them belongs to Δ) such that $f_i = \theta.c_i$ for every $i \in [1..k]$:
 - (7) let $f = \theta.c$:
 - (8) **if** $f \notin F_{sat}$
 - (9) **add** f to Δ_1
 - (10) $x(f) \leftarrow \mathcal{N}_\vee(e(r) \wedge \bigwedge_{i \in [1..k]} x(f_i))$
 - (11) **else** $x(f) \leftarrow x(f) \vee$
 - (12) $\mathcal{N}_\vee(e(r) \wedge \bigwedge_{i \in [1..k]} x(f_i))$
 - (13) $F_{sat} \leftarrow F_{sat} \cup \Delta_1$
 - (14) $\Delta \leftarrow \Delta_1$
 - (15) **until** $\Delta_1 = \emptyset$
 - (16) **return** F_{sat}

The termination of the ProbFR algorithm is guaranteed because all the rules are safe. The only facts that can be inferred from safe rules and a set F of ground atoms are instantiations of conclusion atoms by constants appearing in F . Their number is finite. More precisely, since the input facts and conclusion atoms are built on binary predicates, the number of constants appearing in the input facts is less than $2 \times |F|$ (at most two distinct constants per input fact), and the number of inferred facts is then less than $4 \times |R| \times |F|^2$ (at most as many predicates in conclusion as rules, and for each of them, at most as many instantiations as pairs of constants).

The following theorem states the soundness and completeness of the algorithm.

Theorem 1 Let F_{sat} be the result returned by *ProbFR*(F, R):

$$F_{sat} = SAT(F, R).$$

For each $f \in F_{sat}$, let $x(f)$ be the event expression computed by *ProbFR*(F, R):

$$x(f) \equiv Prov_{F,R}(f).$$

For the first point, we prove by induction on i that each iteration $i \geq 1$ of $ProbFR(F, R)$ computes the set of facts $F_i = T_R(F_{i-1})$ (as defined in Definition 1), and thus $SAT(F, R)$ at the last iteration where the least fixed point reached. For the second point, for a derived fact f , we prove, by induction on the number n of iterations of $ProbFR$ after which no new instantiation of rules can infer f , that $x(f)$ is a disjunctive normal form of $Prov_{F,R}(f)$, and therefore is logically equivalent to it.

As a result of Definition 2 and Theorem 1, it is worth to stress that the probability values of inferred facts are independent from the order in which the rules are triggered to derive them.

As a final remark, like in Datalog, we distinguish the predicates that appear in input facts from the predicates involved in inferred facts. Thus, initial uncertain facts cannot be inferred. They can just take part in the provenance derivation of inferred facts.

Data complexity analysis. We are interested in estimating how the worst-case time complexity of the algorithm depends on the size $|F|$ of the input data, which is the most critical parameter in the setting of Linked Data. The number of iterations of ProbFR is at most $|F_{sat}|$, which is less than $4 \times |R| \times |F|^2$ as shown just above. At each iteration, in the worst case, the condition part of each rule must be evaluated against the facts, and the event expressions for the provenance of the inferred facts must be computed. Let c the maximum number of conditions per rule. The evaluation of each condition part of each rule can be performed in polynomial time (in fact, in at most $|R| \times |F_{sat}|^c$ elementary steps). So the computation of F_{sat} can be done in polynomial data complexity.

For the computation of the event expressions, the most costly operation is the transformation \mathcal{N}_\vee into disjunctive normal form of conjunctions of the form $e(r) \wedge \bigwedge_{i \in [1..k]} x(f_i)$. The number k of conjunctions is less than the bound c of conditions per rule, and each $x(f_i)$ is a disjunction of at most l conjunctions of event keys, where l is the maximum number of uncertain derivation branches for inferred facts. This parameter l is bounded by b^d where d is the maximal depth of reasoning to infer a fact from F and R , and b is the maximal branching factor of $ground(F, R)$ (which denotes the set of rules triggered during the execution of $ProbFR(F, R)$). Therefore, each call of \mathcal{N}_\vee performs at most $b^{d \times c}$ distributivity operations on conjunctions of at most $|F| + |R|$ event keys. Since the maximal depth of reasoning is the number of iterations of $ProbFR(F, R)$, d can be equal to $|F_{sat}|$. Then, the data complexity of the provenance computation may be exponential in the worst-case. This meets known results on query evaluation in probabilistic databases [27]. Different solutions are possible to circumvent this worst-case complexity, like restricting the form of rules/queries like in [10] or imposing some constraints on the input facts (such as a bounded treewidth in [3]). In practice, in particular if most of the input facts are certain, the size of the event expressions remains small. If all the input facts are certain, the only event keys that can be involved in the event expressions are the ones attached to the uncertain rules. The complexity of the algorithm can be controlled by imposing a practical bound on the number l of conjunctions produced in Line (11). This solution is justified in our setting since the computed probabilities are used to keep only the most probable inferred facts, i.e., the facts that are inferred with a probability greater than a given high threshold. For our experiments, we have limited this number l to be 8.

Effective computation of probabilities of inferred facts from their provenance. For each inferred fact, given its provenance as an event expression in disjunctive normal form, the following for-

mula is the basic theoretical tool to compute its probability:

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B) \quad (1)$$

The recursive application of the above formula for computing the probability of a disjunction of l conjunctions of events $E_1 \vee \dots \vee E_l$ leads to alternate the subtractions and additions of the probabilities of all the possible conjunctions $E_{j_1} \wedge \dots \wedge E_{j_l}$. This raises two major issues: first, their number is exponential in l ; second, the exact values of all these probabilities is usually not available.

A usual way to circumvent the latter is to make the assumption of independence between events, as it is done in probabilistic databases [27] or in most Information Retrieval models [14]. In our case, however, two rules such that the condition part of one rule is contained in the condition part of the second (like the rules r_1 and r_2 of the example) are obviously not independent. For such rules, we enforce pairwise disjointness by imposing that the more general rule applies only if the more specific rules do not apply. In this way, we are sure that the corresponding dependent events do not appear in any provenance expression computed during the saturation process. To be consistent with the probabilistic setting, we also impose that the probability assigned to the event corresponding to the more specific rule (r_2 in our example) is higher than the one assigned to the event of more general rule (r_1 in our example).

For each pair r, r' with same conclusion (up to variables names), we will write $r \preceq r'$ if $condition(r)$ is contained in $condition(r')$. Checking $r \preceq r'$ can be done by using any conjunctive query containment algorithm [8] with a complexity independent of the data.

To summarise, we make the assumptions of pairwise *disjointness* between events associated with pairs of rules r, r' such that $r \preceq r'$ and *independence* of the events that are not disjoint. For the effective computation of the probability of an inferred fact f , first, the provenance expressions $x(f) = E_1 \vee \dots \vee E_l$ computed by ProbFR are simplified by removing each conjunction of events E_i in which an event $e(r)$ appears if there is a conjunction of events E_j ($j \neq i$) such that $e(r')$ appears in E_j and $r \preceq r'$, and, second, the probability of f is computed by iteratively applying the formula (1) on the resulting provenance expression.

In our example, the rules r_1 and r_2 are such that $r_1 \preceq r_2$. We can thus remove the conjuncts containing $e(r_1)$ and we obtain

$$x((i_1 \text{ sameAs } i_2)) = (e(r_2) \wedge e(f_1)) \vee (e(r_3) \wedge e(f_4))$$

Now, considering the remaining events as independent, we can compute the effective probability of $P((i_1 \text{ sameAs } i_2))$ as follows:

$$(P(e(r_2)) \times P(e(f_1))) + (P(e(r_3)) \times P(e(f_4))) - (P(e(r_2)) \times P(e(f_1)) \times P(e(r_3)) \times P(e(f_4)))$$

Checking, for every two rules r, r' , whether r' is more generic than r is done before launching ProbFR, as it is independent from the facts. Then, within ProbFR, at each update (Lines (11), (12)), it is the function \mathcal{N}_\vee that suppresses the conjuncts of $x(f)$ involving more generic rules than r : a simple scan of $x(f)$ makes it possible.

This simplification has an impact on the practical complexity of the effective computation of the probabilities, even if, in theory and in the worst-case, it remains exponential in the number l of remaining conjunctions within provenance expressions. As we have explained it before, this number l can be bounded in practice in the algorithm.

The assumption of disjointness between events associated with rules r, r' such $r \preceq r'$ is important for the feasibility of the approach but it also fits well with the open-world assumption that holds in Linked Data. In fact, it captures a restricted form of negation since, under this disjointness assumption, the event $e(r)$ models worlds where the condition part of r is satisfied and the additional conditions of r' are not satisfied.

Setting up of the input probabilities. The above approach for probabilistic inference is agnostic with respect to the way the input probabilities are obtained, either given by experts, returned by built-in predicates or tools, or learned by supervised methods. This said, it is important to note that training sets (required by supervised machine learning techniques) that would be big enough to scale to the setting of Linked Data do not exist and are almost impossible to build manually. On the other hand, it is quite easy for domain experts to decide whether a given rule is uncertain, but setting up its probability is tricky. The two-steps computation of a provenance-based approach as ours has the big advantage to possibly re-compute the numerical values of probabilities for the inferred facts from the provenance expressions computed once for all. This enables to start with a rough setting of rules probabilities chosen from a small set of values just for distinguishing rules on a simple scale of uncertainty (for instance set at 0.9 the rules a priori considered as almost always certain, 0.8 the rules judged as highly probable but less than the previous ones, and so on), and to adjust these values a posteriori based on a feedback on a sample of results. The provenance of wrong sameAs links inferred with a high probability provides explicitly the rules involved in the different reasoning branches leading to their derivation. It is a useful information for a domain expert to choose the rules to penalize by decreasing their numerical probabilities.

3 MODELING UNCERTAINTY USING PROBABILISTIC RULES AND FACTS

When used for data interlinking, rules typically translate varied knowledge that combines schema constraints, alignments between different ontologies and general properties on OWL relations such as owl:sameAs. This knowledge may be certain, but, very often, it has some degree of uncertainty. It is the case when a correspondence in an ontology alignment is attached a confidence value lower than 1, or when domain experts provide knowledge they are not 100% sure about, or the case of pseudo-keys that are automatically computed by pseudo-key discovery tools [6, 28]. This uncertain knowledge can be translated by means of probabilistic rules.

Tables 1 and 2 show rules translating, respectively, certain and uncertain knowledge for the task of interlinking person entities in DBpedia and MusicBrainz datasets. These rules are actually part of the rules that we used in our experiments (reported in Section 4). Rule musicalArtist in Table 1, for example, is a certain rule that translates the DBpedia knowledge that the class dbo:musicalArtist is subsumed by dbo:Artist. Rule enrich_dboBand1 translates a certain correspondence in an alignment between Schema.org vocabulary and DBpedia ontology stating that the class schema:MusicGroup is subsumed by dbo:Band. The rule sameAsVIAF is a certain rule that translates the assertion that the VIAF id is a key for persons and, therefore, allows to infer sameAs links between person entities from DBpedia and MusicBrainz. Notice that this rule actually involves the two equivalent properties dbp:vialf and mb:VialfID of DBpedia and MusicBrainz vocabularies. This means that the condition $(?x \text{ dbp:vialf } ?id)$ in the rule will be instantiated by a DBpedia entity, and $(?y \text{ mb:VialfID } ?id)$ by a MusicBrainz entity. This kind of “key across different datasets” is called a link key in the literature [5]. Note also that instead of using owl:sameAs we use our own customised sameAs predicates (:sameAsPerson) which allowed us to easily identify the type of the inferred sameAs links in our experiments. Rule sameAsIsPerson1 is a certain rule that translates transitivity of sameAs.

Rule similarNamesPerson deserves special attention because it contains a built-in predicate (namely MBSolrsimilar) that encapsu-

lates the call to a full-text search tool (namely Solr⁵) to extract strings from MusicBrainz similar to labels of person entities in DBpedia. More precisely, for each string instantiation s of the variable $?l$, obtained by mapping with DBpedia facts the two first conditions $(?x \text{ rdf:type } \text{dbo:Person})$ and $(?x \text{ rdfs:label } ?l)$ of the rule, MBSolrsimilar($s, 0.8, ?z, \text{'person_mb'}$) is a procedure call returning as many probabilistic facts MBSolrsimilar($s, 0.8, s', \text{'person_mb'}$) as labels s' of person entities in MusicBrainz detected by Solr as similar to s with a similarity greater than 0.8. The probability attached to each probabilistic fact MBSolrsimilar($s, 0.8, s', \text{'person_mb'}$) is the calculated string similarity. Thus similarNamesPerson is a certain rule that will infer uncertain facts of the form $(?x \text{ :solrPSimilarName } ?z)$ due to condition MBSolrsimilar($?l, 0.8, ?z, \text{'person_mb'}$), which will be instantiated with built-in uncertain facts. Built-in predicates such as MBSolrsimilar enable to embed standard similarity functions into our rule-based approach to overcome the problem of misspelling errors in names of persons, groups and songs that may occur in DBpedia and MusicBrainz datasets.

Table 2 shows three additional rules allowing to infer sameAs links between person entities from DBpedia and MusicBrainz datasets, but, in contrast with the sameAsVIAF rule explained above, they are not 100% certain. Rule sameAsBirthDate, for example, says that if two persons have similar names and the same birthdate then they are *likely* to be the same person. This rule must be considered uncertain for two reasons. First, it relaxes the strict condition of having exactly the same name by the soft constraint of having similar names as it is specified by $(?x \text{ :solrPSimilarName } ?l)$. Second, strictly speaking the properties “name” and “birthdate” do not constitute a key, even if it is likely that two named entities representing persons that are well-known enough to be described in datasets like DBpedia and MusicBrainz will refer to the same person if they share the same name and birthdate. In fact, sameAsBirthDate translate a *soft* link key, as it combines the equivalent properties dbo:birthDate and mb:beginDateC that are used in DBpedia and MusicBrainz vocabularies to relate a person with her date of birth. The rules sameAsPersonArtistWr and sameAsMemberOfBand are uncertain too. The first one says that, if two persons have similar names and they are artists of songs with similar names, they are the same person, and the second rule says that if two persons have similar names and are members of musical bands with similar names, they are the same person. Again, this may not be always true, but in most cases. The weights in Table 2 correspond to the probabilistic events associated with each of these uncertain rules.

An important point to emphasise is that the (certain or uncertain) rules allowed in our rule-based modelling express pieces of knowledge that can be assembled and combined through several reasoning steps. For instance, the condition $(?u1 \text{ dbo:artist } ?x)$ of the sameAsPersonArtistWr rule may be triggered by facts inferred by the musicalArtist rule. The chaining between rules is not known in advance and is determined by the input datasets which they apply to. In addition, due to recursive rules (such as sameAsIsPerson1 rule), even if the termination of the saturation process is guaranteed, the number of reasoning steps cannot be known in advance and also depends on the input datasets. It is worthwhile to note that recursive rules add an expressive power that is required for data linkage in particular to express sameAs transitivity.

The translation into rules can be semi-automatic, for instance for translating into certain rules schema constraints that have been declared in OWL such as the functionality or transitivity of some re-

⁵ <http://lucene.apache.org/solr/>

ID	Conditions	Conclusion
musicalArtist	(?u dbo:musicalArtist ?x)	(?u dbo:artist ?x)
enrich_dboBand1	(?x rdf:type schema:MusicGroup)	(?x rdf:type dbo:Band)
sameAsVIAF	(?x dbp:viaf ?id), (?y mb:ViafID ?id)	(?x :sameAsPerson ?y)
sameAsIsPerson1	(?x :sameAsPerson ?y), (?z mb:is_person ?y)	(?x :sameAsPerson ?z)
similarNamesPerson	(?x rdf:type dbo:Person), (?x rdfs:label ?l), MBSolrsimilar(?l,0.8,?z,'persons_mb')	(?x :solrPSimilarName ?z)

Table 1. Certain rules for interlinking person entities in DBpedia and MusicBrainz.

ID	Conditions	Conclusion	Weight
sameAsBirthDate	(?x :solrPSimilarName ?l), (?y skos:myLabel ?l), (?x dbo:birthDate ?date), (?y mb:beginDateC ?date)	(?x :sameAsPerson ?y)	w_1
sameAsPersonArtistWr	(?u1 dbo:artist ?x), (?u1 :solrWrSimilarName ?lu), (?y mb:writer ?u2), (?u2 skos:myLabel ?lu), (?x :solrPSimilarName ?lp), (?y skos:myLabel ?lp)	(?x :sameAsPerson ?y)	w_2
sameAsMemberOfBand	(?x :solrPSimilarName ?l), (?y skos:myLabel ?l), (?y mb:member_of_band ?gr2), (?gr2 skos:myLabel ?lg), (?gr1 dbp:members ?x), (?gr1 :solrGrSimilarName ?lg)	(?x :sameAsPerson ?y)	w_3

Table 2. Uncertain rules for interlinking person entities in DBpedia and MusicBrainz.

lations, or for translating into (certain or uncertain) rules alignments discovered by ontology mapping tools [11]. A certain number of uncertain rules useful for data interlinking must however be provided by domain experts to express fine-grained knowledge that may be specific to the datasets concerned by the linkage task. While it is quite easy for domain experts to decide whether a given rule is uncertain, setting up its probability is tricky. The two-steps computation described in Section 2 has the big advantage to allow iterative adjustment of probabilistic weights associated to rules.

In our experiments, such an incremental adjustment for the probabilities of the three uncertain rules of Table 2 resulted into: $w_1 = 0.9$, $w_2 = 0.4$ and $w_3 = 0.6$.

It is worth emphasising that rules with quite low probabilities (such as 0.4 for the sameAsPersonArtistWr rule) can yet significantly contribute to the final probability of a fact inferred by different reasoning branches. For instance, the resulting probability of a fact inferred from (certain facts and) 3 independent rules each with a 0.4 probability is 0.78, and in the case of 4 such rules it raises to 0.87.

4 EVALUATION

We have conducted experiments to evaluate the performance of our method on real datasets. Our main goal was to measure the effectiveness of our method to discover links at large scale, and to assess the expected gain in terms of recall and the loss in precision when using uncertain rules instead of certain rules only. We also wanted to show how the probabilistic weights attached to the links allow to filter out incorrect links. Finally, we aimed at comparing our tool to a state-of-the-art interlinking tool, namely Silk [29].

4.1 Experimental Setting

We used three datasets in our experiments: DBpedia, INA and MusicBrainz. The objective was to find sameAs links between named entities of person, musical band, song and album included in the datasets. Our choice of these datasets was based upon the fact that these are all large datasets (tens of millions of triples), and of a very different nature: DBpedia was built from Wikipedia infoboxes, INA

from catalog records mainly containing plain text, and MusicBrainz from more structured data coming from a relational database.

The DBpedia version we used was DBpedia 2015-04,⁶ the latest version at the time the experiments were conducted. From all available (sub) datasets, we only used the ones including RDF triples with properties appearing in the rules that we used in the experiments (below we give more details about the rules), which make together one single dataset of around 73 million RDF triples. The INA dataset contains around 33 million RDF triples, while the MusicBrainz dataset around 112 million RDF triples. The INA dataset was built from all the records (plain text) in a catalog of French TV musical programs using a specialised RDF extractor. Some RDF facts in the INA dataset have numerical weights between 0 and 1 since their accuracy could not be 100% assessed during the extraction process. The MusicBrainz dataset⁷ was built from the original PostgreSQL table dumps available at the MusicBrainz web site using an RDF converter. This version is richer than the one of the LinkedBrainz project.⁸

Table 3 shows the number of person, musical band, song and album entities in each of the considered datasets, where Person, e.g. symbolises the class union of all the classes that represent persons in each dataset. No bands or albums are declared in INA, written NA (not applicable) in Table 3.

Class	DBpedia	MusicBrainz	INA
Person	1,445,773	385,662	186,704
Band	75,661	197,744	NA
Song	52,565	448,835	67,943
Album	123,374	1,230,731	NA

Table 3. Number of person, musical band, song and album entities in DBpedia, MusicBrainz and INA.

We have designed two sets of rules that we used as inputs for our algorithm to interlink DBpedia and MusicBrainz first and then MusicBrainz and INA. We came up with 86 rules for interlinking DBpedia and MusicBrainz, from which 50 of them are certain and 36 are

⁶ <http://wiki.dbpedia.org/Downloads2015-04>

⁷ Available at <http://exmo-web.inrialpes.fr/MusicBrainz>

⁸ <http://linkedbrainz.org/>

uncertain, and 147 rules for interlinking MusicBrainz and INA, 97 of them certain and 50 uncertain.⁹ By a way of example, Table 1 and Table 2 of Section 3 include some of the certain and uncertain rules that we used for interlinking DBpedia and MusicBrainz.

ProbFR has been implemented on top of Jena RETE and uses SWI-Prolog v6 to compute the disjunctive normal forms for the event expressions during RETE inference. Prolog is also used to implement the second step of ProbFR, i.e. to compute effective probabilities given event expressions. In order to avoid potential combinatorial explosion, the current parameter of ProbFR is tuned to a maximum of 8 derivation branches for each event expression. All ProbFR experiments were run on a Bi-processor intel Xeon 32 x 2.1GHz, 256 GB of RAM, with Linux CentOS 6 as operating system.

4.2 Experimental Results

We ran our algorithm to interlink DBpedia and MusicBrainz first, and then MusicBrainz and INA, using in each case the corresponding rules. Our algorithm discovered 144,467 sameAs links between entities of DBpedia and MusicBrainz and 28,910 sameAs links between entities of MusicBrainz and INA. Additionally, our algorithm found 132,166 sameAs links internal to the INA dataset.

In order to evaluate the quality of the found links, and since no gold standard was available, we estimated precision, recall and F-measure by sampling and manual checking. In order to compute precision, for each of the classes considered we took a sample of 50 links from the links found by our algorithm (i.e. 200 links in total for DBpedia and MusicBrainz, and 100 links for MusicBrainz and INA), and we manually checked whether these links were correct. For computing recall, we randomly selected 50 instances of each of the classes, and we found links manually. Then, we calculated recall based on this make-do gold standard. F-measure was based on the estimations of precision and recall.

In order to assess the gain of using uncertain rules, we also ran our algorithm only with certain rules, and then we compared the results obtained using only certain rules with the ones obtained using all rules (both certain and uncertain rules). This concerned the experiments between DBpedia and MusicBrainz only, as no other certain rule than sameAs transitivity was used for MusicBrainz and INA.

Table 4 shows all the results. Let us focus on the results concerning DBpedia and MusicBrainz. As expected, when certain rules were used only, precision was 100%. This only concerns Person and Band classes because the initial set of rules did not include any certain rule concluding links for Song and Album (written NA in Table 4). However, recall was very low: 0.08 for Person and 0.12 for Band. When both certain and uncertain rules were used, a 100% precision was achieved for Person and Album classes only, since for Band and Song, precision was 0.94 and 0.96, respectively. However, recall increased significantly for Person and Band: 0.80 and 0.84. This shows the gain of using uncertain rules for data linkage. Now, when looking at the samples of Band and Song classes, we realised that all wrong links had a probability value lower than 0.9 and 0.6, respectively. This means that, when limited to those links having a probability value higher or equal to 0.9 and 0.6, the estimated precision for the classes Band and Song was 100% (Table 5). The estimated recall was 0.80 and 0.54. This shows the gain of using weights for interlinking.

Table 6 shows the number of links that are discovered when n sameAs rules¹⁰ are implied in the derivation. For instance, 28,614

links are discovered using two sameAs rules, and among these links 27,692 are new links, i.e. they were not discovered using only one rule. With tools like Silk and LIMES, using the same set of rules, we can expect to find around 115,609 links only.

4.3 Comparison with Silk

Since Silk cannot handle rule chaining, we divided the rules used by ProbFR into sameAs rules (i.e. rules with sameAs in the conclusion), and intermediate rules that are used to trigger antecedents of other rules (including the sameAs rules). We manually translated these intermediate rules into SPARQL Update queries and these updates were performed before the Silk execution. Some sameAs rules could not be translated into Silk because they are recursive (sameAs appears in their antecedent and conclusion). To be able to compare methods on the same basis, we employed the levenshtein normalised distance with a threshold of 0.2, which corresponds to the similarity parameter set up to 0.8 in Solr. The aggregation of different comparisons within a rule was performed using maximum distance to be compliant with the conjunction used in rules. We executed Silk for interlinking DBpedia and MusicBrainz. Silk found 101,778 sameAs links, from which 100,544 were common to the ones found by ProbFR. ProbFR found 43,923 links that were not discovered by Silk and Silk found 1,234 links not discovered by ProbFR. In theory all the links discovered by Silk should have been discovered by ProbFR and Silk should have found up to 115,609 links. These differences can be explained by the way levenshtein distance are implemented in each tools and by a normalisation of URL that is performed by ProbFR and not available in Silk. As a conclusion, ProbFR outperformed Silk because of rule chaining (more links are discovered). Dealing with uncertainty allows to enhance precision without losing much recall.

In terms of time performance, Silk took more than 53 hours (with 16 threads, blocking activated, on a Bi-processor Intel Xeon, 24 x 1.9GHz) while ProbFR achieved the task in 18 hours (on a Bi-processor Intel Xeon, 32 x 2.1GHz). Even if the difference could be partially explained by the difference in hardware, the main reason comes from implementation design. Silk mainly relies on disk indexing and uses few RAM (around 1-2 GB) while ProbFR runs into main memory and uses around 250 GB of RAM for this experiment.

5 RELATED WORK

There exists a considerable number of systems that (semi) automatically perform data linkage [12]. Most of these approaches consists in applying a set of linkage rules that produce links. These linkage rules specify which properties and how their values are compared. The comparison can be strict or based on some similarity measure between property values of two entities. Linkage rules can be defined manually by a domain expert or learned from data. There are numerous works on learning linkage rules. Some methods are supervised like [18, 20], others are unsupervised like [22, 21]. Our work focuses on generating links given a set of rules and how to infer rules from data is out of the scope of this paper.

Tools like Silk [29] and LIMES [19] are designed to efficiently compare similarities between values of all or some of entities properties and aggregate them. They did not consider reasoning with rules. Silk specifications can be translated into logical rules with built-in functions for computing and aggregating similarity degrees between

⁹ All the rules can be found at http://exmo-web.inrialpes.fr/probfr/rules_INA_MB.txt and http://exmo-web.inrialpes.fr/probfr/rules_DBpedia_MB.txt

¹⁰ We only consider rules that conclude to sameAs statements because other

rules can be handled with preprocessing by tools like Silk or LIMES.

	DBpedia and MusicBrainz						MusicBrainz and INA					
	Only certain rules			All rules			Only certain rules			All rules		
	P	R	F	P	R	F	P	R	F	P	R	F
Person	1.00	0.08	0.15	1.00	0.80	0.89	NA	NA	NA	1.00	0.34	0.51
Band	1.00	0.12	0.21	0.94	0.84	0.89	NA	NA	NA	NA	NA	NA
Song	NA	NA	NA	0.96	0.74	0.84	NA	NA	NA	1.00	0.40	0.57
Album	NA	NA	NA	1.00	0.53	0.69	NA	NA	NA	NA	NA	NA

Table 4. Precision (P), recall (R) and F-measure (F) for the task of interlinking DBpedia and MusicBrainz datasets, and MusicBrainz and INA datasets, using certain rules only, and certain and uncertain rules together.

	P	R	F
Band ≥ 0.90	1.00	0.80	0.89
Song ≥ 0.60	1.00	0.54	0.72

Table 5. Gain of using weights for interlinking DBpedia and MusicBrainz.

# rules	# links	# new links
1	115,609	115,609
2	28,614	27,692
3	1,790	1,152
4	59	14

Table 6. Number of links discovered when n rules are implied in the derivation. Results given for interlinking DBpedia and MusicBrainz.

property values. However, these rules are restricted to linkage rules that are applied independently to each other. The possible chaining between rules is not handled by Silk, which makes it incomplete for the task of discovering all the sameAs links that can be logically inferred. Thus, as it has been pointed out in our experiments, neither Silk nor LIMES [19] (similar to Silk in its principles) would discover sameAs links obtained by transitivity.

L2R [24], Hogan et al. [17] and Al-Bakri et al. [2] handle logical rules and provide full reasoning algorithms that guarantee to infer all the links that can be logically entailed from the rules and facts given as input, either based on forward reasoning for L2R [24], Hogan et al. [17] or on backward reasoning like in Al-Bakri et al. [2]. However, the rules considered in these works are considered to be certain. In LN2R [25], the logical rule-based method of L2R is completed by a similarity-based method (called N2R) applied to pairs of entities for which L2R failed to infer links with certainty.

Dedupalog [4] is a Datalog-like language that has been specially designed for handling constraints useful for record linkage. It handles both hard and soft rules that define respectively valid clusterings and their costs. The associated algorithm computes a valid clustering with a minimal cost. Whereas the general problem is NP-complete, they provide a practical algorithm that scales to the ACM database that contains 436,000 records. Even if the algorithmic techniques are very different from ours, the scalability is obtained by similar restrictions on the rule language. However, the goal is to compute a valid clustering and not to compute probabilities of inferred facts.

Probabilistic logical frameworks such as Markov logic [26] and Probabilistic Soft Logic (PSL) [7] have been used for entity resolution. Markov Logic allows for full probabilistic reasoning. The weights attached to formulas are learned either from data or from probabilities arbitrarily given. This learning phase is made under closed-world assumption. Once a Markov Logic Network is learned, the weighted satisfiability of any candidate link has to be computed. This is not scalable in practice. Then, candidate pairs are filtered using a cheap similarity such as TF.IDF: non matching pairs are added as false atoms. Experiments have been conducted on Cora dataset

(1295 instances) and a sample of Bibserv (10,000 instances). PSL allows probabilistic inference based on similarities functions. As Markov Logic, formulas' weights are learned making closed world assumption. Furthermore, it allows to assign weights to facts using the similarity of sets of property values (which assumes that sets are fully known). Like Datalog, it is restricted to conjunctive rules. Experiments have been performed on the task of Wikipedia article classification and ontology matching.

Contrary to aforementioned approaches, in ProbFR, probability computation and inference are separated. All rules are iteratively applied to compute the saturation and the provenances of every deduced facts. Probabilities are then computed from the provenances. This allows to change the probabilities assigned to rules and reevaluated quickly the probabilities of inferred facts without recomputing the saturation. Another difference is that probabilities attached to formulas can be given or learned from data. No further learning is required.

6 CONCLUSIONS

In this paper, we have shown that it is possible to capture and exploit in a uniform rule-based framework knowledge that may be uncertain but very useful for data linkage. For this, we have adapted the formal setting of Probabilistic Datalog [15] in a way well-suited to Linked Data in which data is inherently incomplete and possibly noisy. Our experiments have shown that this approach is feasible in practice and brings important gain in terms of recall compared to purely logical approaches based on rules and facts that are supposed to be 100% true, while keeping good precision.

Decoupling the symbolic computation of provenances from the numerical computation of probabilities makes probabilistic reasoning more modular and more transparent for users. This provides explanations on probabilistic inference for end-users, and useful traces for experts to set up the input probabilistic weights.

Currently, the threshold for filtering the probabilistic sameAs facts that will be retained as being true must be set up and adjusted manually. As future work, we plan to design a method to set up this threshold automatically by, besides inferring sameAs facts, inferring differentFrom facts too, and then exploiting the sameAs and differentFrom facts (and their probabilities) that are inferred for the same pairs of entities. We also plan to design a backward-reasoning algorithm able to deal with probabilistic rules, that could be combined with the ProbFR probabilistic forward-reasoner for importing on demand useful data from external sources.

ACKNOWLEDGEMENTS

This work has been partially supported by the ANR projects Pagoda (12-JS02-007-01) and Qualinca (12-CORD-012), the joint NSFC-ANR Lindicle project (12-IS01-0002), and LabEx PERSYVAL-Lab (11-LABX-0025-01).

REFERENCES

- [1] Serge Abiteboul, Richard Hull, and Victor Vianu, *Foundations of Databases*, Addison-Wesley, 1995.
- [2] Mustafa Al-Bakri, Manuel Atencia, Steffen Lalande, and Marie-Christine Rousset, 'Inferring same-as facts from linked data: an iterative import-by-query approach', in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pp. 9–15. AAAI Press, (2015).
- [3] Antoine Amarilli, Pierre Bourhis, and Pierre Senellart, 'Provenance circuits for trees and treelike instances', in *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II*, volume 9135 of *Lecture Notes in Computer Science*, pp. 56–68. Springer, (2015).
- [4] Arvind Arasu, Christopher Ré, and Dan Suciu, 'Large-scale deduplication with constraints using dedupalog', in *Proceedings of the 25th International Conference on Data Engineering, ICDE 2009, March 29 2009 - April 2 2009, Shanghai, China*, pp. 952–963. IEEE Computer Society, (2009).
- [5] Manuel Atencia, Jérôme David, and Jérôme Euzenat, 'Data interlinking through robust linkkey extraction', in *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pp. 15–20. IOS Press, (2014).
- [6] Manuel Atencia, Jérôme David, and François Scharffe, 'Keys and pseudo-keys detection for web datasets cleansing and interlinking', in *Knowledge Engineering and Knowledge Management - 18th International Conference, EKAW 2012, Galway City, Ireland, October 8-12, 2012. Proceedings*, volume 7603 of *LNCS*, pp. 144–153. Springer, (2012).
- [7] Matthias Bröcheler, Lilyana Mihalkova, and Lise Getoor, 'Probabilistic similarity logic', in *UAI 2010, Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, July 8-11, 2010*, pp. 73–82. AUAI Press, (2010).
- [8] A.K. Chandra and P.M. Merlin, 'Optimal implementation of conjunctive queries in relational databases', in *Proceedings of the 9th ACM Symposium on Theory of Computing*, pp. 77–90, (1975).
- [9] Peter Christen, *Data Matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*, Data-Centric Systems and Applications, Springer, 2012.
- [10] Nilesh Dalvi and Dan Suciu, 'The dichotomy of probabilistic inference for unions of conjunctive queries.', *Journal of the ACM*, **59**(6), 17–37, (2012).
- [11] Jérôme Euzenat and Pavel Shvaiko, *Ontology Matching, Second Edition*, Springer, 2013.
- [12] Alfio Ferrara, Andriy Nikolov, and François Scharffe, 'Data linking for the semantic web', *International Journal on Semantic Web and Information Systems*, **7**(3), 46–76, (2011).
- [13] Charles Forgy, 'Rete: A fast algorithm for the many patterns/many objects match problem', *Artificial Intelligence*, **19**(1), 17–37, (1982).
- [14] Norbert Fuhr, 'Probabilistic models in information retrieval', *The Computer Journal*, **3**(35), 243–255, (1992).
- [15] Norbert Fuhr, 'Probabilistic datalog: implementing logical information retrieval for advanced applications', *Journal of the American Society for Information Science*, **51**(2), 95–110, (2000).
- [16] Tom Heath and Christian Bizer, *Linked Data : Evolving the Web into a Global Data Space*, Morgan and Claypool, 2011.
- [17] Aidan Hogan, Antoine Zimmermann, Jürgen Umbrich, Axel Polleres, and Stefan Decker, 'Scalable and distributed methods for entity matching, consolidation and disambiguation over linked data corpora', *Journal of Web Semantics*, **10**, 76–110, (2012).
- [18] Robert Isele and Christian Bizer, 'Active learning of expressive linkage rules using genetic programming', *Web Semantics: Science, Services and Agents on the World Wide Web*, **23**(0), (2013).
- [19] Axel-Cyrille Ngonga Ngomo and Sören Auer, 'LIMES - A time-efficient approach for large-scale link discovery on the web of data', in *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pp. 2312–2317. IJCAI/AAAI, (2011).
- [20] Axel-Cyrille Ngonga Ngomo and Klaus Lyko, 'EAGLE: efficient active learning of link specifications using genetic programming', in *The Semantic Web: Research and Applications - 9th Extended Semantic Web Conference, ESWC 2012, Heraklion, Crete, Greece, May 27-31, 2012. Proceedings*, volume 7295 of *Lecture Notes in Computer Science*, pp. 149–163. Springer, (2012).
- [21] Axel-Cyrille Ngonga Ngomo and Klaus Lyko, 'Unsupervised learning of link specifications: deterministic vs. non-deterministic', in *Proceedings of the 8th International Workshop on Ontology Matching co-located with the 12th International Semantic Web Conference (ISWC 2013), Sydney, Australia, October 21, 2013.*, volume 1111 of *CEUR Workshop Proceedings*, pp. 25–36. CEUR-WS.org, (2013).
- [22] Andriy Nikolov, Mathieu d'Aquin, and Enrico Motta, 'Unsupervised learning of link discovery configuration', in *The Semantic Web: Research and Applications - 9th Extended Semantic Web Conference, ESWC 2012, Heraklion, Crete, Greece, May 27-31, 2012. Proceedings*, Lecture Notes in Computer Science, pp. 119–133. Springer, (2012).
- [23] Matthew Richardson and Pedro M. Domingos, 'Markov logic networks', *Machine Learning*, **62**(1-2), 107–136, (2006).
- [24] Fatiha Saïs, Nathalie Pernelle, and Marie-Christine Rousset, 'L2R: A logical method for reference reconciliation', in *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pp. 329–334. AAAI Press, (2007).
- [25] Fatiha Saïs, Nathalie Pernelle, and Marie-Christine Rousset, 'Combining a logical and a numerical method for data reconciliation', *Journal on Data Semantics*, **12**, 66–94, (2009).
- [26] Parag Singla and Pedro M. Domingos, 'Entity resolution with markov logic', in *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006), 18-22 December 2006, Hong Kong, China*, pp. 572–582. IEEE Computer Society, (2006).
- [27] Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch, *Probabilistic Databases*, Morgan & Claypool, 1995.
- [28] Danai Symeonidou, Vincent Armant, Nathalie Pernelle, and Fatiha Saïs, 'Sakey: Scalable almost key discovery in RDF data', in *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*, volume 8796 of *LNCS*, pp. 33–49. Springer, (2014).
- [29] Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov, 'Silk - A link discovery framework for the web of data', in *Proceedings of the WWW2009 Workshop on Linked Data on the Web, LDOW 2009, Madrid, Spain, April 20, 2009.*, volume 538 of *CEUR Workshop Proceedings*. CEUR-WS.org, (2009).

Can a Condorcet Rule Have a Low Coalitional Manipulability?

François Durand¹ and Fabien Mathieu² and Ludovic Noirie²

Abstract. We investigate the possibility of designing a voting rule that both meets the Condorcet criterion and has a low vulnerability to coalitional manipulation. For this purpose, we examine the *Condorcification* of a voting rule, where the original rule is altered to elect the Condorcet winner when one exists, and we study its impact on coalitional manipulability. A recent result states that for a large class of voting rules based on strict total orders, their Condorcification is at most as coalitionally manipulable as the original rule. We show that for most of them, the improvement is strict. We extend these results to a broader framework that includes weak orders and cardinal voting rules. These results support the main message of this paper: when searching for a “reasonable” voting rule with minimal coalitional manipulability, investigations can be restricted to Condorcet rules. In other words, in a class of “raisonnable” voting rules, it is possible to have both the Condorcet criterion and a minimal vulnerability to coalitional manipulation.

1 Introduction

Any non-dictatorial voting rule with three eligible candidates or more³ is vulnerable to manipulation by a single manipulator, who may secure an outcome that she prefers to the result of sincere voting by misrepresenting her preferences [12, 32]. Although this result is frequently cited under the form of Gibbard-Satterthwaite theorem, which deals only with *ordinal* voting rules (i.e. whose ballots are orders of preferences), Gibbard’s fundamental theorem applies to any *game form*, where available strategies may be objects of any kind, for example grades [12].

Once this negative result is known, a possible direction consists in trying to mitigate the impact of manipulation. For example, one can investigate to what extent classic voting rules are manipulable, and try to identify ways of designing less manipulable voting rules.

In the case of manipulation by a single voter, assuming a “reasonable” voting rule and a large electorate (like in most political elections), it is unlikely that a voter is pivotal, which is both supported by theory [25, 24, 10, 11, 33] and analysis of real-life elections.

In this paper, we rather focus on coalitional manipulability, where a coalition of voters, by misrepresenting their preferences, may secure an outcome that they all prefer to the result of sincere voting. The very existence of this type of manipulation can have a strong practical impact on voting, even if all voters choose to vote sincerely. Indeed, whereas implementing a full-scale manipulation can be difficult, sincere voters may find out *a posteriori* that a coalitional manipulation was possible. This happened during the 2002 Presidential

election in France, where left-wing voters discovered only after the election that a concerted ballot in favor of the main leftist candidate would probably have avoided the election of the main rightist candidate [5]. Such a scenario may result in a feeling of regret about the ballots cast, questions about the legitimacy of the outcome, and doubts about the voting rule itself, since for some of the voters, sincere ballots did not defend best their opinions.

To quantify the degree of coalitional manipulability of a voting rule, several indicators have been defined [19, 31, 21, 34, 8, 28, 36, 30]. One of the most studied is the *coalitional manipulability rate*, which is the probability that the voting rule is coalitionally manipulable (CM) in a random profile of preferences, under a given assumption on the probabilistic structure of the population (called *culture*).

In this paper, we consider a more detailed indicator: the set of profiles in which a given voting rule is CM [20]. It is closely related to the coalitional manipulability rate: a voting rule f is less CM than another rule g in this sense of set inclusion if and only if, in any culture, f has a lower coalitional manipulability rate than g .

Several authors have used a theoretical approach [19, 20, 17, 21, 16, 9, 22, 8, 29, 18], computer simulations [19, 29, 30, 13, 14, 15] or experimental results [4, 36, 14, 15] to evaluate the coalitional manipulability rates of several voting rules, according to various assumptions about the structure of the population.

Among the studies above, some authors mention the intuition that Condorcet rules have a general trend to be less CM than others [4, 35, 9, 22, 8, 36]. In contrast, some suggest that *Single Transferable Vote* (STV) is one of the least CM among “reasonable” voting rules ever studied [4, 20, 22, 13, 14], despite not being a Condorcet rule.

Recently, Durand et al. [7] gave theoretical insight on this issue. They considered an alteration of any voting rule called its *Condorcification*, by adding the provision that whenever a Condorcet winner exists, she is elected; in other cases, the original rule is used. This idea is a straightforward generalization of Black’s method [1], that was proposed to get a Condorcet-consistent version of the *Borda count* rule. Until recently, Condorcification was hardly studied in general, as it was seen as an inelegant way to produce Condorcet methods. However, Durand et al. [7] showed that for a large class of voting rules, their Condorcification is at most as CM as the original rule. In a recent paper, Green-Armytage et al. [15] proved also this result independently. We will recall this result formally in Theorem 1.

Roadmap The rest of this paper is organized as follows.

Section 2 introduces some basic definitions of voting theory and our notations. Section 3 states some previous work about the coalitional manipulability of Condorcet rules, with the purpose of clearing up some possible misinterpretations of these results.

In Section 4, we give the result mentioned above [7]: for a large

¹ University Paris Dauphine (francois.durand@dauphine.fr).

² Nokia Bell Labs (fabien.mathieu@nokia.com, ludovic.noirie@nokia.com).

³ I.e. where at least three candidates are in the image of the voting rule. Some candidates may exist but not be eligible [32].

class of voting rules, their Condorcification is at most as CM as the original rule. We then introduce a new notion, the *resistant Condorcet winner*, characterized by a form of immunity to coalitional manipulation. We use this to show that for a large class of voting rules, their Condorcification is *strictly* less CM than the original rule. Then, we stress on an important consequence of these two results: among a large class of voting rules that share a natural property, when searching for a voting rule with minimal coalitional manipulability, investigations can be restricted to Condorcet rules. In other words, in a class of “reasonable” voting rules, it is possible to have both the Condorcet criterion and a minimal vulnerability to coalitional manipulation.

Section 5 extends our framework by allowing voters to have weak orders or even more general binary relations of preference over the candidates. We show that all previous results still hold true, provided that the notion of Condorcet winner is replaced by what we call *absolute Condorcet winner*, instead of the usual definition.

Finally, in Section 6, we generalize the results to non-ordinal voting rules, especially cardinal ones, among which are Approval voting and Range Voting.

2 Framework

Consider two non-empty finite sets \mathcal{V} and \mathcal{C} , whose elements are respectively called *voters* and *candidates*. Some or all of the candidates can also be voters themselves, without impact on our results. Let $V = \text{card}(\mathcal{V})$ and $C = \text{card}(\mathcal{C})$.

\mathcal{L} denotes the set of strict total orders over \mathcal{C} , i.e. transitive, ir-reflexive and complete binary relations. We assume for the moment that each voter v has a strict total order of preference over the candidates, denoted $P_v \in \mathcal{L}$; this assumption will be relaxed in Section 5. An element P of $\mathcal{L}^{\mathcal{V}}$ is called a *profile*: for each voter v , it gives her relation of preference P_v . A *voting rule* is a function $f : \mathcal{L}^{\mathcal{V}} \rightarrow \mathcal{C}$ that, to each profile, associates a winning candidate. We say that f is *coalitionally manipulable* (CM) in profile P towards a profile P' if and only if:

$$\begin{cases} f(P') \neq f(P), \\ \forall v \in \mathcal{V}, (P'_v \neq P_v \Rightarrow f(P') P_v f(P)). \end{cases}$$

Denoting $c = f(P')$, we also say that P is CM *in favor of* c .

$M_f \subseteq \mathcal{L}^{\mathcal{V}}$ denotes the set of profiles where f is CM. In this paper, our goal is to diminish coalitional manipulability in the sense of inclusion: so, we will say that a voting rule g is at most as CM as f if and only if (iff) $M_g \subseteq M_f$.

Given a profile P and two distinct candidates c and d , we use $|c P_v d|$ as a short notation for the number of voters who prefer c to d . We say that c has a *victory* against d in P , or equivalently that d has a *defeat* against c in P , iff $|c P_v d| > \frac{V}{2}$. We say that a candidate c is *Condorcet winner* in a profile P iff c has a victory against any other candidate in P . We say that P is a *Condorcet profile* iff there is a Condorcet winner in P . We say that a voting rule f meets the *Condorcet criterion* iff for any Condorcet profile P , the elected candidate $f(P)$ is the Condorcet winner; as a language convenience, we also say that f is a *Condorcet rule*.

In this ordinal framework, we now recall some classic voting rules⁴. The definitions below can lead to ties between several candidates; in all the following, we will consider that an arbitrary tie-breaking rule is used.

Positional scoring rules (PSR) Let $w = (w_1, \dots, w_C)$ be a non-increasing and non-constant vector of real numbers. In the PSR

of weight vector w , the *score* of a candidate c is defined as $\sum_{v \in \mathcal{V}} w_{r(c, P_v)}$, where $r(c, P_v)$ denotes the rank of candidate c in the preference order P_v of voter v . The candidate with highest score is declared the winner. The most studied PSRs are the three following voting rules.

Plurality PSR of weight vector $(1, 0, \dots, 0)$.

Antiplurality PSR of weight vector $(0, \dots, 0, -1)$.

Borda count PSR of weight vector $(C-1, C-2, \dots, 0)$.

Two-round system⁵ Computing the winner involves two steps or *rounds*. Only the two candidates with highest Plurality scores are selected for the second round, during which each voter grants one point to the candidates she prefers among the two. The candidate with highest score in second round is declared the winner.

Single Transferable Vote (STV) There are $C-1$ rounds. At each round, the candidate with the lowest Plurality score is eliminated. Plurality scores are updated each time, depending on the eliminated candidates: each voter gives one point to the highest non-eliminated candidate in her order of preference.

Coombs' method As in STV, there are $C-1$ rounds. At each round, the candidate with the lowest Antiplurality score is eliminated.

Bucklin's method The *median rank* of a candidate c is the median of the list $(r(c, P_v))_{v \in \mathcal{V}}$. The candidate with the best (i.e. lowest) median rank is elected. If several candidates have the same median rank, the winner is the candidate to which a highest number of voters assign this rank or better (i.e. lower).

3 Condorcet Rules and Coalitional Manipulability: Facts and Traps

The following classic result relates Condorcet notions and coalitional manipulability: in a Condorcet rule, a Condorcet profile cannot be CM towards another Condorcet profile [23]. But, despite common belief, a Condorcet profile is not necessarily immune to coalitional manipulation, even in a Condorcet rule. Worse, in any Condorcet rule with 3 candidates and 3 voters, there exists at least one CM Condorcet profile. To prove this assertion, consider first the following non-Condorcet profile P' .

$$P' = \begin{array}{c|c|c} a & b & c \\ b & c & a \\ c & a & b \end{array}$$

We follow the usual convention to represent profiles: for example, the first column above means that voter 1 has the order of preference $a \succ b \succ c$.

If $f(P') = a$, then consider the following profile P , where only the first voter is different from P' .

$$P = \begin{array}{c|c|c} a & b & c \\ c & c & a \\ b & a & b \end{array}$$

Then candidate c is Condorcet winner in P . But the first voter can manipulate towards profile P' , because she prefers a to c .

⁵ We consider an instantaneous version of the Two-round system: voters give an order of preference, and the two rounds are computed automatically. In most actual implementations, the voting rule is slightly different since voters go to the polls once for each round. It is easy to see that the instantaneous version is at most as manipulable (individually or coalitionally) as the version with two actual rounds: in the latter, sincere voting leads to the same outcome, but manipulators have a larger set of available strategies [23].

⁴ For more details, see for example [36].

If $f(P') = b$ or $f(P') = c$, we can exhibit a similar example by using the symmetry of profile P' .

This statement still holds true for more than 3 candidates (by adding candidates at the end of all preferences in P' and on top of the first voter's preferences in P). It also extends to 5 voters or more, by replacing the three voters by three groups of voters of approximately equal size⁶. So, in general, it is not true that a Condorcet profile is immune to coalitional manipulation, even in a Condorcet rule.

Another classic result deals with coalitional manipulability in *single-peaked* contexts [23]. We say that a preference order P_v is single-peaked [1] relatively to an order $P_0 \in \mathcal{L}$ (typically, a left-right political axis) iff for any candidates c, d, e such that $c P_0 d P_0 e$, it is impossible to have simultaneously $c P_v d$ and $e P_v d$. We say that a profile is single-peaked relatively to P_0 iff it is the case for all individual preferences. As made famous by Black [1], in a single-peaked profile with an odd number of voters, there is always a Condorcet winner. Moreover, with an odd number of voters, if a Condorcet rule is restricted to the profiles that are single-peaked relatively to some given order P_0 , then the rule is not CM [23].

Despite common belief, this does not mean that in all single-peaked contexts, coalitional manipulation is not an issue, and that Condorcet rules solve the problem. As discussed by Blin and Satterthwaite [2] for Black's rule, for the non-manipulability result to hold, it is important to assume that *sincere preferences* and *ballots* are both *a priori* restricted to be single-peaked relatively to a given order. More recently, Penn et al. [27] considered a framework where profiles are single-peaked, but relatively to an order that is not known *a priori* when designing the voting rule: in particular, each voter is allowed to use any strict total order as her ballot. They show that in that case, for any non-trivial voting rule, at least one single-peaked profile is manipulable (even by a single manipulator).

Thus, in single-peaked contexts, when the order P_0 is not known in advance, it is not *a priori* obvious that Condorcet rules are less prone to coalitional manipulation than the others.

Given these results, it is not clear that Condorcet rules are less CM than others in general. Actually, as mentioned earlier, some studies suggest that STV is generally less CM than most known Condorcet rules [4, 20, 22, 13, 14]. In the following, we will not support this too optimistic idea but a more nuanced one: in a large class of voting rules, it is possible to combine the Condorcet criterion and a minimal vulnerability to coalitional manipulation (even if the first does not necessarily imply the second).

4 Condorcification

We first study Condorcification in the framework of strict total orders, before expanding these results to arbitrary binary relations of preference (Section 5) and to non-ordinal voting rules (Section 6).

4.1 Weak Theorem of Condorcification

We call *Condorcification* of f the voting rule f^* defined as follows.

- If there is a Condorcet winner in profile P , then she is elected by f^* .
- Otherwise, $f^*(P) = f(P)$.

⁶ This example does not extend to 4 voters, because one of the three groups would consist of half the voters. In fact, with $V = 4$ and $C = 3$, it is easy to check that there exists a Condorcet rule where no Condorcet profile is CM: for each non-Condorcet profile, elect an arbitrary candidate who has no defeat (for example, a Plurality winner).

For example, *Black's method* [1] is defined as the Condorcification of the Borda count.

It is easy to check that Condorcification preserves *anonymity* (symmetry of voters), *neutrality* (symmetry of candidates) and *monotonicity* (if a candidate c wins, then if one voter moves c up in her ballot, then c cannot become a loser). For the latter, it is sufficient to remark that in a Condorcet rule, there cannot be a violation of monotonicity involving a Condorcet profile; so, if there exists a non-monotonicity paradox in f^* , it is between two non-Condorcet profiles, so it also exists in f . Of course, f^* meets the Condorcet criterion and all criteria it implies, for example the *majority criterion* (if a candidate is ranked first by a strict majority, then she is elected).

But our main focus in this paper is its effect on coalitional manipulability: Durand et al. [7] and Green-Armytage et al. [15] showed that for an important class of voting rules, their Condorcification is at most as CM as the original rule. To state this result formally, we call a *coalition* a subset of the voters and a *majority coalition* a coalition whose cardinality is strictly greater than $\frac{V}{2}$. We say that a voting rule f meets the *informed majority coalition criterion* (InfMC) iff for any candidate c , for any majority coalition \mathcal{M} , for any profile P , there exists a profile P' such that:

$$\left\{ \begin{array}{l} \forall v \notin \mathcal{M}, P'_v = P_v, \\ f(P') = c. \end{array} \right.$$

In other words, any majority coalition may ensure the victory of any candidate, provided they know in advance the other voters' ballots. This criterion appears under different names in several sources: InfMC [7], *Conditional Majority Determination* [15] or without explicit name [3]. It is closely related to Peleg's notion of *β -effectivity* [26].

It is easy to check that most usual voting rules meet InfMC (except some exotic positional scoring rules such as Antiplurality, rarely used in actual settings): Plurality, Two-round system, STV, Borda count, Bucklin's and Coombs' methods, and all Condorcet rules. Among common voting rules, it is interesting to see that most meet InfMC, even those whose usual rationale does not rely on the notion of majority (such as Approval voting, as we will see in Section 6). In practice, this gives a wide scope of application for the following theorem. From a theoretical point of view, we can wonder whether there is a deep reason why most common voting systems meet this criterion; we think that this is an interesting question for future work.

If f meets InfMC, it is easy to prove this property: for any profile P' that is a strong Nash equilibrium for the game defined by f and some profile P , the winner $f(P')$ has necessarily no defeat in P (i.e., if V is odd, she must be a Condorcet winner)⁷. This gives a first intuition why choosing the Condorcet winner might be a good idea to prevent coalitional manipulation.

The following theorem is mentioned without proof by Durand et al. [7], and Green-Armytage et al. [15] provides a version of the proof that is only valid for strict total orders of preference, as we will discuss in Section 5. We will give a more general proof in Section 6.

Theorem 1 (Weak Condorcification) *If f meets InfMC, then its Condorcification is at most as CM as f .*

$$M_{f^*} \subseteq M_f.$$

⁷ Actually, the converse is true: this property implies that f meets InfMC [6].

4.2 Strong Theorem of Condorcification

In this section, we give a second Condorcification theorem, stating that for most usual voting rules that do not meet the Condorcet criterion, their Condorcification is not only at most as CM, but *strictly* less CM. In order to prove this, we introduce the notion of *resistant Condorcet winner* (RCW), a candidate that possesses a form of immunity to coalitional manipulation. We say that candidate c is an RCW in profile P iff, for any pair of candidates $d, e \in \mathcal{C} \setminus \{c\}$ (not necessarily distinct from each other):

$$|c P_v d \text{ and } c P_v e| > \frac{V}{2}.$$

We use this notation: given an assertion $\mathcal{A}(v)$ that depends on voter v , we denote $|\mathcal{A}(v)| = \text{card}\{v \in \mathcal{V} \text{ s.t. } \mathcal{A}(v)\}$.

Proposition 1 (Characterization of the RCW) *Given a profile P and a candidate c , the following conditions are equivalent.*

1. *Candidate c is RCW in P .*
2. *For any Condorcet rule f , c is elected by sincere voting, i.e. $f(P) = c$, and f is not CM in P .*

Proof: $1 \Rightarrow 2$. This part being the easiest, we give only a sketch of proof. Assume that c is RCW in P . Let f be a Condorcet rule. Since c is clearly Condorcet winner in P , we have $f(P) = c$. Consider a manipulation attempt in favor of a candidate $d \neq c$, i.e. a profile P' where only voters preferring d to c may change their ballot, whereas those preferring c to d cannot do so. In particular, for any candidate $e \neq c$, voters who simultaneously prefer c to d and c to e in P keep the same ballots in P' ; since c is an RCW in P , they guarantee that c still has a victory against e in P' . So, candidate c still appears as a Condorcet winner in P' , she gets elected and the manipulation fails. Hence, f is not CM in P .

Not $1 \Rightarrow 2$. Assume that condition 1 is false, i.e. c is not an RCW in P . We can assume however that c is Condorcet winner in P , otherwise it is trivial that condition 2 is false (because we can choose a Condorcet rule f such that $f(P) \neq c$).

Let (d, e) be a pair of candidates violating the definition of the RCW. Necessarily, $e \neq d$, otherwise c would not be a Condorcet winner. We will exhibit a profile P' without Condorcet winner and differing from P only by voters preferring d to c . So, it will be possible to choose a Condorcet rule f such that $f(P') = d$. From this, we will deduce that f is CM in profile P towards P' , in favor of d .

So, let us exhibit such a profile P' . Up to switching roles between d and e , we can assume that e has no victory against d in profile P . Let p be a strict total order of the form: $(d \succ e \succ c \succ \text{other candidates})$. For each voter v preferring d to c in P (“manipulator”), let $P'_v = p$. For each other voter v (“sincere voter”), let $P'_v = P_v$. In the new profile P' , candidate c is not a Condorcet winner because she does not have a victory against e : indeed, the only voters who claim preferring c to e in P' are those of the sincere voters who already preferred c to e in P , which leads to $|c P'_v e| = |c P_v d \text{ and } c P_v e| \leq \frac{V}{2}$. Candidate d is not a Condorcet winner (it is an easy and classic result that her duel against c cannot have been improved by manipulation [23]). Neither can candidate e because she still has no victory against d . And neither can other candidates, because the number of voters who claim preferring c to them has not diminished from P to P' . \square

We say that a voting rule meets the *resistant-Condorcet criterion*⁸ iff, whenever there is an RCW, she is elected. Clearly, this criterion is

⁸ We use a dash to stress on the fact that the adjective *resistant* applies to the word Condorcet, not criterion.

weaker than the Condorcet criterion because it constrains the result in a smaller set of profiles.

In practice, all the usual voting rules violating the Condorcet criterion also violate the resistant-Condorcet criterion (for some values of V and C). Indeed, consider a profile P of the following type, with $V = 100$ voters and $C = 17$ candidates.

	17	13	14	14	14	14	14
	a	c	d_1	d_2	d_4	d_7	d_{11}
		a	c	d_3	d_5	d_8	d_{12}
			a	c	d_6	d_9	d_{13}
$P =$				a	c	d_{10}	d_{14}
					a	c	d_{15}
						a	c
	Others	Others	Others	Others	Others	Others	Others
	c	d_1	d_2	d_4	d_7	d_{11}	a

In the above notation, each column gathers identical voters and its top cell indicates the corresponding number of voters. For each column, the respective positions of candidates denoted “others” is not important for this example. We let the reader check the following. Candidate c is an RCW. However, in any PSR (including Plurality, Borda count and Antiplurality), candidate a has a better score than c . It is also true in Bucklin’s method. In the Two-round system, STV or Coombs’ method, c is eliminated during the first round. Hence, none of these voting rules meet the resistant-Condorcet criterion.

It is easy to define an artificial example of a voting rule that meets the resistant-Condorcet criterion but not the Condorcet criterion: for example, consider a rule electing the RCW when she exists, and a constant candidate otherwise. But the observation above tends to show that a voting rule that was not designed to elect all Condorcet winners has no “natural” reason to elect the resistant ones.

Now, we have the necessary tools to state and prove the strong theorem of Condorcification.

Theorem 2 (Strong Condorcification) *If f meets InfMC but not the resistant-Condorcet criterion, then its Condorcification f^* is strictly less CM than f :*

$$M_{f^*} \subsetneq M_f.$$

Proof: The weak theorem of Condorcification (Th. 1) ensures the inclusion. Since f does not meet the resistant-Condorcet criterion, there exists a profile P , a candidate c who is RCW in P , such that $f(P) \neq c$. Since c is a Condorcet winner, a strict majority of voters prefer c to $f(P)$; by InfMC, it implies that f is CM in P in favor of c . In contrast, Proposition 1 ensures that f^* is not CM in P . Hence, the inclusion is strict. \square

In particular, Theorem 2 proves that for Plurality, Two-round system, STV, Borda count, Bucklin’s and Coombs’ methods, their Condorcification is strictly less CM than the original rule.

The reader may have noticed that the implication $2 \Rightarrow 1$ in Proposition 1 is not necessary to prove Theorem 2. We mentioned it to show the deep connection between the property of being an RCW and the immunity to coalitional manipulation in the Condorcet rules.

4.3 Optimality Corollary

Up to now, we have considered a given voting rule f and compared the set of CM profiles for f and for its Condorcification f^* . At first look, these results may suggest to use voting rules such as the Condorcification of Plurality, STV, etc. However, we think that it is not the main consequence of the Condorcification theorems. Indeed, they imply the following corollary. As a notational convenience, the set of voting rules meeting InfMC is also denoted by InfMC.

Corollary 1 (Optimality) *Let us consider the function:*

$$M : \begin{cases} \text{InfMC} & \rightarrow \mathcal{P}(\mathcal{L}^V) \\ f & \rightarrow M_f \end{cases}$$

returning, for each voting rule f meeting InfMC, the set M_f of its CM profiles.

Let $A \in \mathcal{P}(\mathcal{L}^V)$ be a minimal value of M , i.e. a set of profiles such that at least one voting rule $f \in \text{InfMC}$ meets $M_f = A$, but no rule $f \in \text{InfMC}$ meets $M_f \subsetneq A$. Then:

- Any rule $f \in \text{InfMC}$ meeting $M_f = A$ meets the resistant-Condorcet criterion.
- There exists a Condorcet rule f such that $M_f = A$.

In order to understand the scope of this theorem, let us notice that the function M may have several minima that are not comparable, because the inclusion relation over $\mathcal{P}(\mathcal{L}^V)$ is not a total order. In other words, there may be different rules f and g such that no voting rule is less CM than f or g , but whose sets of CM profiles, M_f and M_g respectively, are not comparable.

This corollary can be summed up this way: when looking for a voting rule meeting InfMC with minimal coalitional manipulability, then investigations *must* be restricted to rules meeting the resistant-Condorcet criterion and *can* be restricted to Condorcet rules. In other words, this corollary answers the main question of this paper: when restricting to “reasonable” voting rules, in the sense that they meet InfMC, it is possible to have both the Condorcet criterion and a minimal vulnerability to coalitional manipulation.

5 Arbitrary Binary Relations

Now, let \mathcal{P} be a subset of the binary relations over the candidates. \mathcal{P} will represent the set of relations we assume possible for each voter. The relation $P_v \in \mathcal{P}$ of a voter v is interpreted in the following way: for any pair of distinct candidates (c, d) , the assertion $c P_v d$ means that when d is the winner of sincere voting, v may be interested in taking part in a coalitional manipulation in favor of c .

In most usual models, this relation is identified with the voter’s binary relation of strict preference over the candidates. With this interpretation, it is natural to assume that it is antisymmetric: v cannot strictly prefer c to d and d to c in the same time. However, with the general interpretation of P_v as an inclination to manipulate, it is conceivable to have a “crazy manipulator” who wants to manipulate for c when d would win by sincere voting, and vice-versa. Moreover, the antisymmetry assumption is not needed for the proofs of our results. So, for the sake of generality, we will not make this assumption in the rest of this paper. That being said, should the reader be confused with the absence of antisymmetry assumption, she can read all the following with this additional assumption in mind and the usual interpretation of P_v as a strict preference.

Since it is common to identify the inclination to manipulate with strict preferences, we will use the following language shortcut: when $c P_v d$, we will go on saying that voter v prefers c to d .

Typically, \mathcal{P} can be the set of strict total orders like in previous sections, or the set of strict weak orders (negatively transitive, irreflexive and antisymmetric relations), or the set of preferences that are single-peaked relatively to a given order, etc. But in the general case, absolutely no assumption is made about \mathcal{P} . A relation $P_v \in \mathcal{P}$ may not be complete (e.g. strict weak orders). It may not be transitive either: voter v may prefer candidate a to b , b to c and c to a .

In this first extension of the framework, a voting rule is a function $f : \mathcal{P}^V \rightarrow \mathcal{C}$. In this case, there are at least two natural generalizations of the Condorcet winner.

1. We say that a candidate c is an *absolute Condorcet winner* iff for any other candidate d , she has an *absolute victory* against d , in the sense that $|c P_v d| > \frac{V}{2}$ and $|d P_v c| \leq \frac{V}{2}$. The main motivation for the second condition is to ensure the uniqueness of the absolute Condorcet winner in the unusual models where non-antisymmetric relations are allowed. For antisymmetric relations, it can safely be omitted, because it becomes redundant with the first condition.
2. We say that candidate c is a *relative Condorcet winner* iff for any other candidate d , she has a *relative victory* against d , in the sense that $|c P_v d| > |d P_v c|$.

When preferences are strict total orders, these two notions are obviously equivalent, and both amount to the notion of the Condorcet winner that we have used up to now.

Similarly, there are also two natural notions that generalize Condorcification: *absolute Condorcification* and *relative Condorcification*, which respectively add a preliminary test about the existence of an absolute or a relative Condorcet winner and elect her if she exists.

In Section 6, we will prove that the weak theorem of Condorcification (Th. 1) still holds in the general case when considering the absolute Condorcification. For this reason, in the following, we will use the terms *Condorcet winner* and *Condorcification* for the absolute version of these notions.

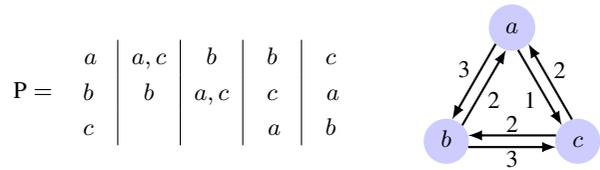
In contrast, we will now show that the weak theorem of Condorcification (Th. 1) is not true when replacing f^* by the relative Condorcification of f , denoted by f^{rel} . In other words, some profiles may be CM in f^{rel} whereas they are not CM in f .

Let us start with a voting rule that is a bit artificial but makes it possible to prove this concisely. Assume that preferences are strict weak orders. Let f be the voting rule that we call *Condorcet-dean*:

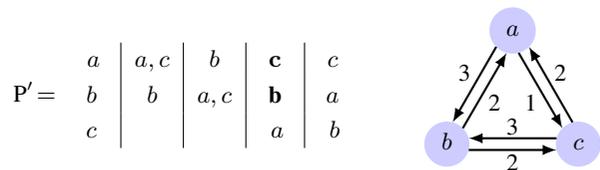
1. If there is an absolute Condorcet winner, then she is elected.
2. Otherwise, a constant candidate called the *dean* (say, candidate a) is elected.

Obviously, this rule meets InfMC.

Consider $V = 5, C = 3$ and the following profile P .



The above notation on the left means for example that the second voter is indifferent between candidates a and c , which she prefers to b . On the right is the *weighted majority graph*: nodes are the candidates, and for each pair of distinct candidates (x, y) , there is a directed edge from x to y whose weight is $|x P_v y|$. It is easy to check that $f(P) = f^{\text{rel}}(P) = a$ and that f is not CM in P . Now, consider the following profile P' , where only the fourth voter changes her ballot.



Since candidate c is the relative Condorcet winner, we have $f^{\text{rel}}(P') = c$. In conclusion of this example, f^{rel} is CM in P towards P' in favor of candidate c (it is even *individually* manipulable), whereas f is not CM in P . Hence, Theorem 1 does not generalize when considering relative Condorcification.

While the voting rule used above is exotic, we can produce a similar counter-example with f being the Single Transferable Vote (STV), with weak orders allowed. If a voter has two candidates or more tied on top of her ballot, her vote is equally shared between these candidates. Let us consider the following profile P with $V = 100$ voters.

$P =$	12	11	25	12	12	28
	a	a	b	b	c	d
	b	d	a, c	c	a	a, c
	c	c	d	a	d	b
	d	b		d	b	

In f , candidates c , then d , then b are successively eliminated hence $f(P) = a$. First, let us show that f is not manipulable in P .

- In favor of candidate b : even if she reached the last round, she would lose against any other candidate (by 63 or 51 votes).
- In favor of candidate c : the fourth and fifth groups of voters are interested (12 + 12 voters). For candidate c not being eliminated during the first round, it is necessary that 23 or 24 manipulators put her on top of her ballot and that candidate a is eliminated (it is not possible to eliminate candidate b or d). In the second round, since candidates b and d have more than one third of the votes each (37 and 39, respectively), candidate c is eliminated.
- In favor of candidate d : only the sixth group (28 voters) is interested. During the first round, they cannot simultaneously save candidates c and d from elimination: indeed, since candidate a receives 23 votes, manipulators and voters sincerely casting a ballot for candidates c or d would need to have at least $2 \times 23 = 46$ votes, but they have only $12 + 28 = 40$. Hence, since candidate d must stay, candidate c must be eliminated in the first round. In the second round, candidates a and b have more than one third of the votes each (35 and 37 respectively), so candidate d is eliminated.

Since candidate c is the relative Condorcet winner, we have $f^{\text{rel}}(P) = c$. Let us consider the following profile P' , an attempt of manipulation in favor of candidate a .

$P' =$	12	11	25	12	12	28
	a	a	b	b	c	d
	b	d	a, c	c	a	a, c
	d	b	d	a	d	b
	c	c		d	b	

For the point of view of STV, the counting resolve the same way and we have $f(P') = a$. And since there is no relative Condorcet winner, we also have $f^{\text{rel}}(P') = a$. In conclusion, f^{rel} is CM in P towards P' in favor of a , whereas f is not CM in P .

Remark that the two examples are not constructed in the same way. In Condorcet-dean, f and f^{rel} give the same output in the initial profile but different outcomes in the manipulated profile. In contrast, in the example of STV, f and f^{rel} return different outputs in the initial profile but the same output in the manipulated profile. Both types of problems can occur with the relative Condorcification.

We insist again on the importance of distinguishing the notions of relative and absolute Condorcet winner when dealing with more

general preferences than strict total orders. For example, the proof of the weak Condorcification Theorem presented by Green-Armytage et al. [15] relies on relative Condorcification, so it cannot be adapted to weak orders. In details, at the end of point 2 of their proof, it is established that no candidate B is preferred by a strict majority to some candidate A . In point 3 of the proof, it is deduced from this that no candidate B can be a (relative) Condorcet winner. This implication fails with weak orders: indeed, a candidate B can be a relative Condorcet winner and be preferred to A by only 45% of the voters, whereas A is preferred to B by 40% of the voters.

6 Generalization

Up to now, we considered only *ordinal* voting rules: we use this term in an extended sense, meaning that the winner depends only on the binary relations of preference (whether they are orders or not, strictly speaking). In this section, we generalize the previous results to non-ordinal voting rules, where the ballot of a voter may contain information that is not included in her order of preference, for example *Range Voting* (where each voter assigns a grade to each candidate in a set of authorized values, and the candidate with highest average grade is elected).

An *electoral space* is defined by:

- Two non-empty finite sets \mathcal{V} and \mathcal{C} ;
- For each voter $v \in \mathcal{V}$, a non-empty set Ω_v of her possible *states*;
- For each voter $v \in \mathcal{V}$, a function $P_v : \Omega_v \rightarrow \mathcal{R}$, where \mathcal{R} denote the set of binary relations over the candidates.

Denote $\Omega = \prod_{v \in \mathcal{V}} \Omega_v$. An element $\omega \in \Omega$ is called a *configuration*: for each voter v , it gives her state ω_v . Such an electoral space is denoted by $(\mathcal{V}, \mathcal{C}, \Omega, P)$, or just Ω in short. A *voting rule* (over an electoral space Ω) is a function $f : \Omega \rightarrow \mathcal{C}$.

As an example, for Range Voting, we can consider the following model: for each voter v , her state ω_v is a vector of grades, one for each candidate. Her order of preference $P_v(\omega_v)$ is the one induced by ω_v , in the sense that she prefers a candidate c to a candidate d iff she assigns a strictly higher grade to c than d . This model is especially relevant if there is a great number of authorized grades: in that case, it is reasonable to consider that if a voter sincerely assigns the same grade to two candidates, then she is indifferent between them.

But this assumption is not reasonable when there is a small number of authorized grades (the extreme case being Approval Voting, which can be seen as Range Voting with only grades 0 and 1). In any case, the following model can also be considered. For each voter v , her state ω_v is a pair (p_v, g_v) , where p_v is a strict weak order of preference over the candidates and g_v is a vector of C grades that is coherent with p_v , in the sense that for any two candidates c and d , if $c p_v d$, then $g_v(c) \geq g_v(d)$. The function P_v is then defined by $P_v(p_v, g_v) = p_v$.

The framework of electoral spaces is a generalization of the ordinal framework. Indeed, consider the model of Section 5, where \mathcal{P} is the set of binary relations that are possible for any voter. This can be modeled by an electoral space where for each voter v , $\Omega_v = \mathcal{P}$ and P_v is the identity function.

We say that f is *coalitionally manipulable* (CM) in configuration ω towards a configuration ψ iff:

$$\begin{cases} f(\psi) \neq f(\omega), \\ \forall v \in \mathcal{V}, (\psi_v \neq \omega_v \Rightarrow f(\psi) P_v(\omega_v) f(\omega)). \end{cases}$$

The notions of InfMC, Condorcet winner and Condorcification extend easily to this new framework. We denote by $M_f \subseteq \Omega$ the set of configurations where f is CM.

6.1 Weak Theorem of Condorcification

Theorem 3 (Weak Condorcification) *If f meets InfMC, then its Condorcification is at most as CM as f .*

$$M_{f^*} \subseteq M_f.$$

Remark that if ψ is a Condorcet configuration, then changing its result to the Condorcet winner cannot worsen manipulability in ψ (i.e. make it manipulable if it was not in the original rule). Indeed, if $f(\psi)$ is not the Condorcet winner, then f is CM in ψ anyway, because f meets InfMC; so, the modified voting rule cannot do worse. However, this simple remark is not sufficient to prove the theorem: it does not exclude the possibility that changing the result in ψ make another configuration ω manipulable towards ψ .

Proof: Suppose that f^* is CM in a configuration ω towards a configuration ψ , but f is not CM in ω .

Let $c = f(\omega)$. For any $d \in \mathcal{C} \setminus \{c\}$, we have $|d P_v(\omega_v) c| \leq \frac{V}{2}$: otherwise, since f meets InfMC, f would be CM in ω in favor of d .

As a consequence, no other candidate than c is an absolute Condorcet winner in ω . By definition of the Condorcification f^* , this leads to $f^*(\omega) = c$.

Now, let $d = f^*(\psi)$. We already know that $|d P_v(\omega_v) c| \leq \frac{V}{2}$. Voters who do not prefer d to c do not modify their ballots from ω to ψ , hence $|d P_v(\psi_v) c| \leq |d P_v(\omega_v) c| \leq \frac{V}{2}$. As a consequence, d is not an absolute Condorcet winner in ψ . So, by definition of the Condorcification f^* , there is no absolute Condorcet winner in ψ and we have $f(\psi) = d$.

Hence, $f(\omega) = f^*(\omega)$ and $f(\psi) = f^*(\psi)$ so f is CM in ω towards ψ : this is a contradiction. \square

6.2 Strong Theorem of Condorcification

For the strong theorem of Condorcification (Th. 2), the key point is to generalize correctly the central notion of RCW. In the most general case, we say that candidate c is an RCW in configuration ω iff, for any pair of candidates $d, e \in \mathcal{C} \setminus \{c\}$ (not necessarily distinct from each other):

$$\left\{ \begin{array}{l} |\text{not}(d P_v(\omega_v) c) \text{ and } c P_v(\omega_v) e| > \frac{V}{2}, \\ |\text{not}(d P_v(\omega_v) c) \text{ and } \text{not}(e P_v(\omega_v) c)| \geq \frac{V}{2}, \end{array} \right. \quad (1)$$

With the (usual) assumption that preferences are antisymmetric, Eq. (2) becomes redundant and the definition amounts only to:

$$|\text{not}(d P_v(\omega_v) c) \text{ and } c P_v(\omega_v) e| > \frac{V}{2}.$$

Proposition 1, characterizing the RCW, generalizes as follows.

Proposition 2 (Characterization of the RCW) *Given a configuration ω and a candidate c , consider the following conditions.*

1. Candidate c is RCW in ω .
2. For any Condorcet rule f , c is elected by sincere voting, i.e. $f(\omega) = c$, and f is not CM in ω .

We have: $1 \Rightarrow 2$. If all strict total orders are authorized for any voter, i.e. if $\forall v \in \mathcal{V}, \mathcal{L} \subseteq P_v(\Omega_v)$, then the converse $2 \Rightarrow 1$ is true.

This theorem states that the converse implication $2 \Rightarrow 1$ is true, for example, if for each voter v , her set $P_v(\Omega_v)$ of possible binary relations of preferences is the set of strict weak orders, since it includes the set of strict total orders.

In order to have the converse implication $2 \Rightarrow 1$, it is not possible to omit, in condition 2, the assumption that c is elected in any Condorcet rule (or, equivalently, that c is a Condorcet winner). Otherwise, one may consider a configuration ω where all voters are indifferent between all candidates, i.e. all their binary relations of preference are empty. In that case, obviously, no voting rule is manipulable in ω , but no candidate is RCW.

Proof: $1 \Rightarrow 2$. The proof is essentially the same as in proposition 1.

Not $1 \Rightarrow 2$. Assume that condition 1 is false, i.e. c is not an RCW. As in the proof of proposition 1, we can assume however that c is a Condorcet winner, otherwise it is trivial that condition 2 is false. We will prove that there exists a Condorcet rule f that is CM in ω .

Since c is not RCW, at least one of equations (1) or (2) from the definition is not met. We distinguish three cases: **A.** Eq. (2) is not met; **B.** Eq. (1) is not met for some $e = d$; or **C.** Eq. (1) is not met with $e \neq d$.

In each case, the principle is the same as in the proof of proposition 1: exhibit a configuration ψ with no Condorcet winner, differing from ω only for some voters who prefer d to c . As a consequence, it is possible to choose a Condorcet rule f such that $f(\psi) = d$. Finally, f is CM in ω towards ψ in favor of d .

Case A. If there exists some candidates d and e such that Eq. (2) is not met, it means that $|\text{not}(d P_v(\omega_v) c) \text{ and } \text{not}(e P_v(\omega_v) c)| < \frac{V}{2}$. Remark that $e \neq d$, otherwise we would have $|d P_v(\omega_v) c| > \frac{V}{2}$, implying that c is not Condorcet winner. Up to switching roles between d and e , we can assume that e does not have an absolute victory against d in ω . Let p be a strict total order of the form: $(d \succ e \succ c \succ \text{other candidates})$. For each voter v preferring d to c in ω ("manipulator"), we can choose ψ_v such that $P_v(\psi_v) = p$, thanks to the assumption that all strict total orders are authorized. For each other voter v ("sincere voter"), let $\psi_v = \omega_v$. In the new configuration ψ , candidate c is not a Condorcet winner, because she is defeated by e : indeed, the only voters that claim not preferring e to c in ψ are those of the sincere voters who already did so in ω ; formally, $|\text{not}(e P_v(\psi_v) c)| = |\text{not}(d P_v(\omega_v) c) \text{ and } \text{not}(e P_v(\omega_v) c)| < \frac{V}{2}$, which translates to $|e P_v(\psi_v) c| > \frac{V}{2}$. Candidate d cannot appear as a Condorcet winner (because her duel against c cannot have been improved by manipulation [23]). Neither can candidate e because she still has no absolute victory against d . And neither can other candidates, because the number of voters who claim preferring c to them has not decreased.

Case B. If Eq. (1) is not met for some $e = d$, it means that $|\text{not}(d P_v(\omega_v) c) \text{ and } c P_v(\omega_v) d| \leq \frac{V}{2}$. Let p be a strict total order of the form: $(d \succ c \succ \text{other candidates})$. For each voter v preferring d to c in ω ("manipulator"), we can choose ψ_v such that $P_v(\psi_v) = p$, thanks to the assumption that all strict total orders are authorized. For each other voter v ("sincere voter"), let $\psi_v = \omega_v$. In the new configuration ψ , candidate c is not a Condorcet winner, because she does not have a victory against d : indeed, the only voters that claim preferring c to d in ψ are those of the sincere voters who already did so in ω ; formally, $|c P_v(\psi_v) d| = |\text{not}(d P_v(\omega_v) c) \text{ and } c P_v(\omega_v) d| \leq \frac{V}{2}$. Candidate d cannot appear as a Condorcet winner (because her duel against c cannot have been improved by manipulation [23]). And nei-

ther can other candidates, because the number of voters who claim preferring c to them has not decreased.

Case C. Remains the case where Eq. (1) is not met, with $e \neq d$. For any real number X , we will denote by $\lfloor X \rfloor$ (resp. $\lceil X \rceil$) the floor (resp. ceiling) function applied to X .

For any pair of candidates x and y , we will write:

- $x \text{ I}_v(\omega_v) y$ iff not $x \text{ P}_v(\omega_v) y$ and not $y \text{ P}_v(\omega_v) x$ (indifference).
- $x \text{ PP}_v(\omega_v) y$ iff $x \text{ P}_v(\omega_v) y$ and not $y \text{ P}_v(\omega_v) x$ (antisymmetric part of preferences: this is equivalent to $x \text{ P}_v y$ when the usual assumption is made that preferences are antisymmetric).
- $x \text{ MP}_v(\omega_v) y$ iff $x \text{ P}_v(\omega_v) y$ and $y \text{ P}_v(\omega_v) x$ (mutual preference: this cannot happen when the usual assumption is made that preferences are antisymmetric).

As a notational convenience, we will omit the configuration when it is ω (and not when it is ψ): for example, $x \text{ P}_v y$ means $x \text{ P}_v(\omega_v) y$.

We denote $A_{cd}(\omega) = |c \text{ P}_v(\omega_v) d|$: it is the number of voter who prefer c to d in ω .

In this third case, Eq. (1) is not met, with $e \neq d$. Denoting $B = |\text{not}(d \text{ P}_v c) \text{ and } c \text{ P}_v e|$, it means that $B \leq \frac{V}{2}$. Using case A, we can assume, however, that Eq. (2) is met.

We will see that in the final configuration ψ , we can ensure that there is a victory neither for c against e , nor for e against c .

Let p be a strict total order of the form: ($d \succ e \succ c \succ$ other candidates).

Let p' be a strict total order of the form: ($d \succ c \succ e \succ$ other candidates).

Since c is Condorcet winner, we have $A_{ce}(\omega) > \frac{V}{2}$, so:

$$|d \text{ P}_v c \text{ and } c \text{ P}_v e| > \frac{V}{2} - B \geq 0.$$

As a consequence, we can choose $\lfloor \frac{V}{2} \rfloor - B$ voters among the manipulators (voters preferring d to c in ω); for each of them, denoted v , choose ψ_v such that $\text{P}_v(\psi_v) = p'$. For each other manipulator v , choose ψ_v such that $\text{P}_v(\psi_v) = p$. Finally, for each voter who prefers c to d in ω ("sincere voter"), let $\psi_v = \omega_v$.

Then, we have:

$$A_{ce}(\psi) = B + \left(\left\lfloor \frac{V}{2} \right\rfloor - B \right) = \left\lfloor \frac{V}{2} \right\rfloor, \quad (3)$$

so c has no victory against e .

By the way, Eq. (1) is not met for this pair (d, e) but Eq. (2) is met, which respectively translate to the first and second following equations:

$$\begin{cases} \left| \text{not}(d \text{ P}_v c) \text{ and } c \text{ PP}_v e \right| + \left| \text{not}(d \text{ P}_v c) \text{ and } c \text{ MP}_v e \right| \leq \left\lfloor \frac{V}{2} \right\rfloor, \\ \left| \text{not}(d \text{ P}_v c) \text{ and } c \text{ PP}_v e \right| + \left| \text{not}(d \text{ P}_v c) \text{ and } c \text{ I}_v e \right| \geq \left\lceil \frac{V}{2} \right\rceil, \end{cases}$$

hence, by subtraction:

$$\left| \text{not}(d \text{ P}_v c) \text{ and } c \text{ MP}_v e \right| - \left| \text{not}(d \text{ P}_v c) \text{ and } c \text{ I}_v e \right| \leq \left\lfloor \frac{V}{2} \right\rfloor - \left\lceil \frac{V}{2} \right\rceil.$$

Thanks to our assumptions on the manipulators' ballots in ψ , the only voters who claim preferring mutually c to e or be indifferent between these two candidates in ψ are those of the sincere voters who did so in ω . Formally:

$$\begin{cases} \left| c \text{ MP}_v(\psi_v) e \right| = \left| \text{not}(d \text{ P}_v c) \text{ and } c \text{ MP}_v e \right|, \\ \left| c \text{ I}_v(\psi_v) e \right| = \left| \text{not}(d \text{ P}_v c) \text{ and } c \text{ I}_v e \right|. \end{cases}$$

By substitution in the previous equation, this leads to:

$$\left| c \text{ MP}_v(\psi_v) e \right| - \left| c \text{ I}_v(\psi_v) e \right| \leq \left\lfloor \frac{V}{2} \right\rfloor - \left\lceil \frac{V}{2} \right\rceil. \quad (4)$$

As a general remark, it is easy to prove that:

$$A_{ec}(\psi) + A_{ce}(\psi) = V + \left| c \text{ MP}_v(\psi_v) e \right| - \left| c \text{ I}_v(\psi_v) e \right|. \quad (5)$$

Substituting equations (3) and (4) in equation (5), we deduce:

$$A_{ec}(\psi) \leq V + \left\lfloor \frac{V}{2} \right\rfloor - \left\lceil \frac{V}{2} \right\rceil - \left\lfloor \frac{V}{2} \right\rfloor = \left\lfloor \frac{V}{2} \right\rfloor,$$

so e has no victory against c .

To sum up, neither c nor e can be Condorcet winner. For the same reasons as in previous cases, neither can d nor any other candidate. \square

As a corollary, the strong theorem of Condorcification (Th. 2) still holds true in the general case (remind that f^* designates the absolute Condorcification). This also implies the optimality corollary (Cor. 1) in this more general framework.

Consequently, even in a broader framework when non-ordinal voting are authorized, our main message still holds. In the class InfMC, when searching for a voting rule with minimal coalitional manipulability, investigations can be restricted to Condorcet rules. In other words, it is possible to have both the Condorcet criterion and a minimal vulnerability to coalitional manipulation.

7 Conclusion

We recalled the weak theorem of Condorcification, initially stated by Durand et al. [7] and Grenn-Armytage et al. [15]: for all voting rules that meet the informed majority coalition criterion, their Condorcification is at most as CM as the original rule (Th. 1). Then we introduced the notion of resistant Condorcet winner and we used it to prove the strong theorem of Condorcification (Th. 2): for a large class of voting systems, the improvement provided by Condorcification is strict. We think that the most important consequence of these results is the optimality corollary (Cor. 1): when searching for a "reasonable" voting rule (i.e. meeting InfMC) with minimal manipulability, investigations *must* be restricted to voting rules meeting the resistant-Condorcet criterion and *can* be restricted to Condorcet rules.

When preferences are not limited to strict total orders, and in particular when they are strict weak orders, we showed that all previous results hold, provided that the notions of Condorcet winner and resistant Condorcet winner are generalized adequately. In particular, we showed that the weak theorem of Condorcification (and, as a consequence, the strong theorem) becomes false when considering the usual notion of relative Condorcet winner, but holds true when using the absolute Condorcet winner.

Finally, we showed that all our results extend to non-ordinal voting rules, and in particular cardinal voting rules such as Approval voting and Range voting. In particular, we presented a new proof of the weak theorem of Condorcification (Th. 3) that covers this most general model.

For future work, it would be interesting to evaluate quantitatively the difference of manipulability between a voting rule and its Condorcification: that could be done using a theoretical approach or computer simulations.

ACKNOWLEDGEMENTS

The work presented in this paper has been carried out at LINC (www.lincs.fr).

REFERENCES

- [1] Duncan Black, *The theory of committees and elections*, Cambridge University Press, 1958.
- [2] Jean-Marie Blin and Mark Satterthwaite, 'Strategy-proofness and single-peakedness', *Public Choice*, **26**, 51–58, (1976).
- [3] Markus Brill and Vincent Conitzer, 'Strategic voting and strategic candidacy', in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, USA*, (2015).
- [4] John Chamberlin, Jerry Cohen, and Clyde Coombs, 'Social choice observed: Five presidential elections of the american psychological association', *The Journal of Politics*, **46**(02), 479–502, (1984).
- [5] Alistair Cole, 'A strange affair: The 2002 presidential and parliamentary elections in france', *Government and Opposition*, **37**(03), 317–342, (2002).
- [6] François Durand, *Vers des modes de scrutin moins manipulables*, Ph.D. dissertation, Université Pierre et Marie Curie, 2015.
- [7] François Durand, Fabien Mathieu, and Ludovic Noirie. Reducing manipulability. Poster presented in the 5th International Workshop on Computational Social Choice (COMSOC), 2014.
- [8] Pierre Favardin and Dominique Lepelley, 'Some further results on the manipulability of social choice rules', *Social Choice and Welfare*, **26**(3), 485–509, (2006).
- [9] Pierre Favardin, Dominique Lepelley, and Jérôme Serais, 'Borda rule, Copeland method and strategic manipulation', *Review of Economic Design*, **7**(2), 213–228, (2002).
- [10] Peter Fristrup and Hans Kleiding, 'A note on asymptotical strategy-proofness', *Economics Letters*, **31**(4), 307–312, (1989).
- [11] Peter Fristrup and Hans Kleiding, 'Erratum: A note on asymptotical strategy-proofness', *Economics Letters*, **105**(32), (1990).
- [12] Allan Gibbard, 'Manipulation of voting schemes: A general result', *Econometrica*, **41**(4), 587–601, (1973).
- [13] James Green-Armytage, 'Four Condorcet-Hare hybrid methods for single-winner elections', *Voting matters*, **29**, 1–14, (2011).
- [14] James Green-Armytage, 'Strategic voting and nomination', *Social Choice and Welfare*, **42**(1), 111–138, (2014).
- [15] James Green-Armytage, Nicolaus Tideman, and Rafael Cosman, 'Statistical evaluation of voting rules', *Social Choice and Welfare*, **46**(1), 183–212, (2016).
- [16] HC Huang and Vincent CH Chua, 'Analytical representation of probabilities under the IAC condition', *Social Choice and Welfare*, **17**(1), 143–155, (2000).
- [17] K.H. Kim and F.W. Roush, 'Statistical manipulability of social choice functions', *Group Decision and Negotiation*, **5**(3), 263–282, (1996).
- [18] Dominique Lepelley, Ahmed Louichi, and Hatem Smaoui, 'On Ehrhart polynomials and probability calculations in voting theory', *Social Choice and Welfare*, **30**(3), 363–383, (2008).
- [19] Dominique Lepelley and Boniface Mbih, 'The proportion of coalitionally unstable situations under the plurality rule', *Economics Letters*, **24**(4), 311–315, (1987).
- [20] Dominique Lepelley and Boniface Mbih, 'The vulnerability of four social choice functions to coalitional manipulation of preferences', *Social Choice and Welfare*, **11**, 253–265, (1994).
- [21] Dominique Lepelley and Fabrice Valognes, 'On the Kim and Roush voting procedure', *Group Decision and Negotiation*, **8**, 109–123, (1999).
- [22] Dominique Lepelley and Fabrice Valognes, 'Voting rules, manipulability and social homogeneity', *Public Choice*, **116**, 165–184, (2003).
- [23] Hervé Moulin, *La Stratégie du vote*, Cahiers du CEREMADE, Université Paris IX-Dauphine, Centre de recherche de mathématiques de la décision, 1978.
- [24] Shmuel Nitzan, 'The vulnerability of point-voting schemes to preference variation and strategic manipulation', *Public Choice*, **47**, 349–370, (1985).
- [25] Bezalel Peleg, 'A note on manipulability of large voting schemes', *Theory and Decision*, **11**, 401–412, (1979).
- [26] Bezalel Peleg, *Game theoretic analysis of voting in committees*, number 7 in *Econometric Society monographs in pure theory*, Cambridge Univ. Press, 1984.
- [27] Elizabeth Penn, John Patty, and Sean Gailmard, 'Manipulation and single-peakedness: A general result', *American Journal of Political Science*, **55**(2), 436–449, (2011).
- [28] Geoffrey Pritchard and Arkadii Slinko, 'On the average minimum size of a manipulating coalition', *Social Choice and Welfare*, **27**, 263–277, (2006).
- [29] Geoffrey Pritchard and Mark Wilson, 'Exact results on manipulability of positional voting rules', *Social Choice and Welfare*, **29**(3), 487–513, (2007).
- [30] Reyhaneh Reyhani, Geoffrey Pritchard, and Mark Wilson. A new measure of the difficulty of manipulation of voting rules, 2009.
- [31] Donald Saari, 'Susceptibility to manipulation', *Public Choice*, **64**(1), 21–41, (1990).
- [32] Mark Satterthwaite, 'Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions', *Journal of Economic Theory*, **10**(2), 187–217, (1975).
- [33] Arkadii Slinko, 'On asymptotic strategy-proofness of classical social choice rules', *Theory and Decision*, **52**, 389–398, (2002).
- [34] Arkadii Slinko, 'How large should a coalition be to manipulate an election?', *Mathematical Social Sciences*, **47**(3), 289–293, (2004).
- [35] David Smith, 'Manipulability measures of common social choice functions', *Social Choice and Welfare*, **16**, 639–661, (1999).
- [36] Nicolaus Tideman, *Collective Decisions And Voting: The Potential for Public Choice*, Ashgate, 2006.

Upper and Lower Time and Space Bounds for Planning

Christer Bäckström and Peter Jonsson¹

Abstract. There is an extensive literature on the complexity of planning, but explicit bounds on time and space complexity are very rare. On the other hand, problems like the constraint satisfaction problem have been thoroughly analysed in this respect. We provide a number of upper and lower bound results for both plan satisfiability (PSAT) and length-optimal planning (LOP), with an emphasis on monotone planning (where actions have only positive effects) which is used in, for instance, h^+ and similar heuristics. Let v and a be the number of variables and actions, respectively. We consider both restrictions on the number and polarity of preconditions and effects of actions and the PUBS restrictions in SAS⁺. For all such classes, we show that PSAT and LOP is either tractable or cannot be solved in subexponential time $2^{o(v)}$ or time $2^{o(a)}$, unless the so-called *Exponential Time Hypothesis (ETH)* is false. There is also a sharp transition: monotone LOP can be solved in time $2^{o(v)}$ if $a \in o(\frac{v}{\log v})$ but not if $a \in \Omega(v)$. We also study upper bounds and discuss the trade-off between time and space, providing a polynomial-space algorithm for monotone LOP that beats depth-first search in most cases. This raises the important question how lower bounds are affected by polynomial space restrictions.

1 INTRODUCTION

The computational complexity of the plan satisfiability problem (PSAT) and the plan-length optimisation problem (LOP) is well-studied in the literature. Bylander [8] analysed subclasses of both problems based on restricting the preconditions and effects of actions in the STRIPS language. Bäckström and Nebel [6] made a similar study for the PUBS restrictions in the SAS⁺ language. The complexity of cost-optimal planning (COP) has also been studied (cf. Katz and Domshlak [28]). More recently, parameterised complexity analysis has been used. Bäckström et al. [5] analysed LOP for a number of subclasses using plan length as parameter, Kronegger et al. [30] used many parameters, including plan length, analysing the complexity for combinations of these parameters, and Aghighi and Bäckström [3] analysed COP using plan cost as parameter. Many more examples can be found in the literature. However, they all have in common that they classify problems into complexity classes, rather than providing any explicit time bounds.

While explicit upper bounds can be provided by demonstrating algorithms, it is more difficult to prove non-trivial lower bounds. An important step forward was the Exponential Time Hypothesis (ETH) [22], which conjectures that k -SAT cannot be solved in subexponential time $2^{o(n)}$, where n is the number of variables. This has proven a very useful hypothesis since there is a large number of NP-complete problems that are related in the sense that either all of them can be solved in subexponential time or none of them can. Hence, proving

that a problem cannot be solved in subexponential time under the assumption that the ETH holds is a very strong indication of hardness. While all NP-complete problems are equivalent under the theory of NP-completeness, it is known that they differ widely in hardness in practice. The ETH has enabled to separate the NP-complete problems with respect to concrete time bounds, which is more fine grained and better related to practice than the usual classifications into complexity classes. The ETH is nowadays a standard assumption in complexity theory [31].

The ETH, and similar assumptions, was recently used to analyze the constraint satisfaction problem (CSP) [13, 27]. This showed that if the ETH is true, CSP cannot be solved in subexponential time even for a large number of common restrictions, although some special cases were identified where subexponential algorithms exist. CSP is a very important NP-complete problem. Apart from its widespread use in AI and elsewhere, it is also an archetypical NP-complete problem in the sense that many other NP-complete problems can easily be modelled as CSP classes. Planning is similarly a good and natural modelling language for many problems in PSPACE, and in NP, but no similar analysis of lower bounds exists for planning.

We address the issue of explicit upper and lower bounds for planning, with an emphasis on monotone planning (where actions have only positive effects). One reason for the latter is that monotone planning is NP-complete, while general planning is PSPACE-complete, so it is a stronger result to prove that not even monotone planning can be solved in subexponential time. Another reason is that monotone planning is important for many heuristics, like h^+ [21].

First, we derive some straightforward upper bounds in Sec. 3, both for monotone and non-monotone planning, in order to put the forthcoming lower-bound results into a perspective. We then turn to lower bounds, using restrictions on the number and polarity of preconditions and effects of actions (cf. Bylander [8]). Let v and a be the number of variables and actions, respectively. In Sec. 4, we give a complete classification of PSAT for these restrictions in the sense that each such class is either tractable or not solvable in subexponential time in the number of variables or actions, i.e. in time $2^{o(v)}$ or $2^{o(a)}$, (unless the ETH is false). We then do similar analyses of the LOP problem for the same type of restrictions. In Sec. 5, we focus on monotone planning and show that not even severely restricted classes can be solved in time $2^{o(v)}$ or $2^{o(a)}$ (unless the ETH is false). For the general non-monotone case, we show a sharper result in Sec. 6 based on graph colouring. If LOP can be solved faster than time $2^{\frac{3}{2}} \cdot \text{poly}(v)$, then there would be a faster algorithm for graph colouring than currently known. We then show in Sec. 7 that there exists a sharp transition: Monotone LOP cannot be solved in time $2^{o(v)}$ if $a \in \Omega(v)$, i.e. the number of actions is at least linear in the number of variables, but it can be solved in time $2^{o(v)}$ if the number of actions is sublinear in the number of variables. This resembles a similar transition result for CSP [13]. Then we consider lower bounds for other

¹ Department of Computer and Information Science, Linköping University, Sweden. Email: christer.backstrom@liu.se, peter.jonsson@liu.se

types of restrictions. In Sec. 8 we consider the PUBS restrictions (cf. Bäckström and Nebel [6]) and provide complete classifications for both PSAT and LOP, in the sense that all combinations of restrictions are either tractable or cannot be solved in time $2^{o(v)}$ (unless the ETH is false). We also settle an open question and prove NP-hardness for those combinations that have remained unclassified in the literature. For the last type of lower bound results, we consider planning classes defined by the structure of the causal graph (cf. Katz and Domshlak [29] and Giménez and Jonsson [19]). We show that even if restricted to a number of simple types of causal graphs, including several types studied in the literature, planning cannot be solved in time $2^{o(v)}$ (unless the ETH is false). After that, we once again consider upper bounds in Sec. 10, and, in particular, we consider the trade-off between time and space. While the previously derived upper bounds are based on algorithms requiring exponential space, we ask how fast we can plan using only polynomial space. For monotone planning, depth-first search satisfies this criterion, since the depth is limited, but the time bound increases with the branching factor. We provide an alternative polynomial-space algorithm which outperforms depth-first search except for instances with small branching factor. This also raises an important open question: How would lower-bound results be affected by a restriction to polynomial space?

2 PRELIMINARIES

For a set or sequence X of objects, we write $|X|$ to denote the cardinality (the number of objects) of X and we write $\|X\|$ to denote the size (the number of bits of the representation) of X .

2.1 Planning

In the general case, we will use the SAS⁺ planning framework [6], which uses variables with arbitrary finite domain. Let $V = \{v_1, \dots, v_n\}$ be a finite set of *variables*, with an implicit order v_1, \dots, v_n , each with a finite *domain* $D(v_i)$. This defines the *state space* $S(V) = D(v_1) \times \dots \times D(v_n)$. A member $s \in S(V)$ is called a (*total*) *state* and can be viewed as a total function that specifies a value in D for each $v_i \in V$. A *partial state* may leave the value undefined for some (or all) variables, and is thus a partial function. The value of a defined variable v_i in a (total or partial) state s is called the *projection* of s onto v_i and is denoted $s[v_i]$. If s is a partial state, then $\text{vars}(s)$ is the set of variables with a defined value in s .

A *planning instance* $\mathbb{P} = \langle V, A, I, G \rangle$ has a set of variables V over the domain D , a set of *actions* A , a total *initial state* I and a partial *goal state* G . Each action $a \in A$ has a *precondition* $\text{pre}(a)$ and an *effect* $\text{eff}(a)$, both partial states. Let $a \in A$ and $s \in S(V)$. Then a is *valid in* s if $\text{pre}(a)[v] = s[v]$ for all $v \in \text{vars}(\text{pre}(a))$, and the *result of* a in s is a state $t \in S(V)$ such that for all $v \in V$, $t[v] = \text{eff}(a)[v]$ if $v \in \text{vars}(\text{eff}(a))$ and $t[v] = s[v]$ otherwise. Let $s_0, s_\ell \in S(V)$ and let $\omega = a_1, \dots, a_\ell$ be a sequence of actions. Then ω is a *plan from* s_0 to s_ℓ if either (1) $\omega = \langle \rangle$ and $\ell = 0$ or (2) there are states $s_1, \dots, s_{\ell-1} \in S(V)$ such that for all i ($1 \leq i \leq \ell$), a_i is valid in s_{i-1} and s_i is the result of a_i in s_{i-1} . Furthermore, ω is a *plan* (i.e. a *solution*) for \mathbb{P} if it is a plan from I to G .

For every class C of SAS⁺ instances, we define the following two problems.

PLAN SATISFIABILITY (PSAT(C))

Instance: An instance $\mathbb{P} = \langle V, A, I, G \rangle$ in C .

Question: Does \mathbb{P} have a plan?

LENGTH-OPTIMAL PLANNING (LOP(C))

Instance: An instance $\mathbb{P} = \langle V, A, I, G \rangle$ in C and a non-negative integer k .

Question: Does \mathbb{P} have a plan ω of length $|\omega| \leq k$?

Most of our results, in particular the lower-bound results, only make use of binary variables. In these cases, it is often clearer and more convenient to use Propositional STRIPS with Negative goals (PSN) [8], which can be viewed as a different way to define SAS⁺ with binary variables. Let V be a set of binary *variables*, i.e. propositional atoms. For any set $V' \subseteq V$, the set of *literals* over V' is $L(V) = \{v, \bar{v} \mid v \in V'\}$. A *total state* s over V is a subset $s \subseteq V$, where a variable v is true in s if and only if $v \in s$. The *space of total states* over V is $S(V) = 2^V$. A *partial state* p over V is a consistent subset $p \subseteq L(V)$, i.e. it does not contain both v and \bar{v} for any $v \in V$. A variable v is true in p if $v \in p$, false if $\bar{v} \in p$ and undefined if $p \cap \{v, \bar{v}\} = \emptyset$. We also define $p^+ = \{v \in V \mid v \in p\}$ and $p^- = \{v \in V \mid \bar{v} \in p\}$. Let p be a partial state and s a total state. Then p is *satisfied* in s , denoted $p \sqsubseteq s$ if both $p^+ \subseteq s$ and $p^- \cap s = \emptyset$. The \times *operator* is defined as $s \times p = (s \setminus p^-) \cup p^+$. Finally, $\text{vars}(p) = \{v \mid v \in p \text{ or } \bar{v} \in p\}$.

A PSN *instance* is a tuple $\mathbb{P} = \langle V, A, I, G \rangle$ where V is a set of variables, A is a set of *actions*, the *initial state* I is a total state over V and the *goal* G is a partial state over V . Each action a in A has a precondition $\text{pre}(a)$ and an effect $\text{eff}(a)$, which are both partial states. For all total states s, t over V and all $a \in A$, a is *from* s to t if both (1) $\text{pre}(a) \sqsubseteq s$ and (2) $t = s \times \text{eff}(a)$. A sequence $\omega = a_1, \dots, a_\ell$ of actions in A is a *plan* from a state s_0 to a state s_ℓ if either (1) $s_0 = s_\ell$ and ω is the empty sequence or (2) there are total states $s_1, \dots, s_{\ell-1}$ such that a_i is from s_{i-1} to s_i for all i ($1 \leq i \leq \ell$). The sequence s_0, \dots, s_ℓ is the *state sequence* of ω . A solution for \mathbb{P} is a plan from I to some total state s such that $G \sqsubseteq s$. A solution for \mathbb{P} is called a *plan* for \mathbb{P} .

We write $a : P \Rightarrow E$ to define an action a with precondition P and effect E . We define PSN (and later SAS⁺) subclasses based on the number and polarity of the preconditions and effects, e.g. PSN₁²⁺ denotes the class of PSN instances where the actions have at most two positive literals in the precondition and one literal in the effect. We use $*$ to denote an unrestricted number of literals, i.e. PSN₁^{*} allows any number of literals in the preconditions but only one literal in the effects. A PSN instance is *monotone* if no action has any negative effects, i.e. all monotone instances belong to PSN₊^{*}. Note that PSN₊^{*} = PSN.

2.2 Satisfiability and the ETH

The k -SAT problem is defined as follows and it is known to be NP-complete for $k \geq 3$ [18].

k -SAT

Instance: A CNF formula \mathbb{F} where each clause has at most k literals.

Question: Does \mathbb{F} have a satisfying assignment?

We will use n for the number of variables and m for the number of clauses of k -SAT instances. A more precise complexity characterization is possible by using the *Exponential Time Hypothesis* (ETH) [22].

Definition 1 For all constant integers $k \geq 3$, let s_k be the infimum of all real numbers δ such that k -SAT can be solved in time $O(2^{\delta n})$. The ETH says that $s_k > 0$ for all $k \geq 3$.

Somewhat informally, the ETH says that k -SAT cannot be solved in subexponential time $2^{o(n)}$.

If the ETH holds, then for every $k \geq 3$, there is some constant c_k such that k -SAT cannot be solved in time $2^{c_k n}$. The ETH is a quite strong assumption that allows for defining a theory similar to the one of NP-completeness. There is a large number of NP-complete problems that form a completeness class in the sense that either all of them can be solved in subexponential time, or none of them can [24]. There is also strong variant of the hypothesis (SETH) that additionally conjectures that $\lim_{k \rightarrow \infty} s_k = 1$.

We will frequently make use of the following result, that it is sufficient to assume a linear number of clauses in the instances.

Lemma 2 (de Haan et al. [13, Lemma 1]) k -SAT ($k \geq 3$) is solvable in time $2^{o(n)}$ if and only if k -SAT with a linear number of clauses and in which the number of occurrences of each variable is at most 3 is solvable in time $2^{o(n)}$.

3 UPPER BOUNDS

In order to set the forthcoming lower-bound results into a perspective we first derive some straightforward upper bounds for planning. Consider the general case, a SAS⁺ instance with v variables, each with a domain of size d . Then the state space consists of d^v states. Each state can have an arc to every state, including itself, so the maximum number of arcs in the state-transition graph is $d^v \cdot d^v = d^{2v}$. A straightforward way to solve LOP is to use Dijkstra's algorithm. Fredman and Tarjan's [16] variant of Dijkstra's algorithm runs in time $O(|E| + |V| \log |V|)$. A planning instance \mathbb{P} can be viewed as a compact representation (in the sense of Galperin and Wigderson [17]) of its state-transition graph, where we can check in time $\text{poly}(|\mathbb{P}|)$ if an arc exists. It follows that we can solve LOP(SAS⁺) in time $O((d^{2v} + d^v \log d^v) \cdot \text{poly}(|\mathbb{P}|)) = O(d^{2v} \cdot \text{poly}(|\mathbb{P}|))$. Heuristic search algorithms may be preferable in many practical cases, but they give no advantage in the worst case, e.g. Dijkstra's algorithm is essentially equivalent to the A* algorithm [15].

For PSN we have $d = 2$, so the state space is of size 2^v and can have up to $2^{2v} = 4^v$ arcs, i.e. we can solve LOP(PSN) in time $O(4^v \cdot \text{poly}(|\mathbb{P}|))$. We then proceed to the monotone case, when the actions have no negative effects, i.e. PSN_{*+}. Then there can only be an arc from s to t if $s \subseteq t$. For each i , there are $\binom{v}{i}$ states of size i . For each such state, there are 2^i subsets. Hence, the maximum number of arcs is

$$e(v) = \sum_{i=1}^v \binom{v}{i} 2^i = \sum_{i=0}^v \binom{v}{i} 2^i - 1.$$

Using the binomial formula

$$(a+b)^v = \sum_{i=0}^v \binom{v}{i} a^{v-i} b^i$$

and setting $a = 1$ and $b = 2$ we get

$$\sum_{i=0}^v \binom{v}{i} 2^i = \sum_{i=0}^v \binom{v}{i} 1^{v-i} 2^i = (1+2)^v = 3^v$$

i.e. $e(v) < 3^v = 2^{v \log 3}$.

Observation 3 LOP(PSN_{*+}) can be solved in time $O(3^v \cdot \text{poly}(|\mathbb{P}|))$ using Dijkstra's algorithm.

4 SUBEXPONENTIAL LOWER BOUNDS FOR PSAT

Both problems PSAT(PSN₁₊¹⁺) and PSAT(PSN₂₊¹⁺) are known to be NP-hard [8]. We will strengthen these results below by also proving that neither can be solved in subexponential time (in either the number of variables or the number of actions), unless the ETH is false.

Construction 4 Let \mathbb{F} be a 3-SAT instance with variables x_1, \dots, x_n and clauses c_1, \dots, c_m , where each clause c_j is of the form $\{\bar{l}_j^1, \bar{l}_j^2, \bar{l}_j^3\}$. Construct a corresponding PSAT(PSN₁₊¹⁺) instance $\mathbb{P} = \langle V, A, I, G \rangle$ as follows:

- $V = \{f_i, t_i \mid 1 \leq i \leq n\} \cup \{y_j \mid 1 \leq j \leq m\}$;
- A contains the actions
 - $\text{set } f_i : \{\bar{t}_i\} \Rightarrow \{f_i\}$, for all i ($1 \leq i \leq n$),
 - $\text{set } t_i : \{\bar{f}_i\} \Rightarrow \{t_i\}$, for all i ($1 \leq i \leq n$) and
 - $\text{vfy}_j^k : \{\bar{l}_j^k\} \Rightarrow \{y_j\}$, for all j, k ($1 \leq j \leq m, 1 \leq k \leq 3$), where $\bar{l}_j^k = f_i$ if $l_j^k = \bar{x}_i$ and $\bar{l}_j^k = t_i$ if $l_j^k = x_i$;
- $I = \emptyset$ and $G = \{y_1, \dots, y_m\}$.

Theorem 5 If PSAT(PSN₁₊¹⁺) can be solved in time $2^{o(v)}$ or time $2^{o(a)}$, then the ETH is false.

Proof. Proof by reduction from 3-SAT to PSAT(PSN₁₊¹⁺). Let \mathbb{F} be a 3-SAT instance and let \mathbb{P} be the corresponding PSN instance according to Construction 4. For each variable x_i in \mathbb{F} , a plan for \mathbb{P} can set either f_i or t_i to true, but not both. It follows that \mathbb{P} has a plan if and only if \mathbb{F} is satisfiable, so Construction 4 is a polynomial reduction from 3-SAT to PSAT(PSN₁₊¹⁺).

Let S_d denote the class of 3-SAT instances where $m \leq dn$ and choose d such that S_d cannot be solved in time $2^{o(n)}$ if the ETH is true. Such a d exists according to Lemma 2. Let P_d be the class of PSN₁₊¹⁺ instances we can get by applying Construction 4 to S_d .

Suppose PSAT(PSN₁₊¹⁺) can be solved in time $2^{o(v)}$. Choose an arbitrary $c > 0$. Then PSAT(PSN₁₊¹⁺) can be solved in time in 2^{cv} for large v . Let \mathbb{F} be a 3-SAT instance in S_d with n variables. Then \mathbb{P} has $m \leq dn$ clauses. Let \mathbb{P} be the corresponding PSN instance. We have $v = 2n + m \leq 2n + dn = (2+d)n$ and $|\mathbb{P}|$ is polynomial in n , so it follows from our assumption that we can solve satisfiability for S_d in time $\text{poly}(n) + 2^{c(2+d)n} \leq 2^{(c(2+d)+\epsilon)n}$, for all $\epsilon > 0$ and large n . However, c is arbitrary so we can choose arbitrary $c', \epsilon > 0$ such that $c(2+d) + \epsilon \leq c'$ and S_d can be solved in time $2^{c'n}$ for large n . Unless the ETH is false, this contradicts our assumptions and it follows that PSAT(PSN₁₊¹⁺) cannot be solved in time $2^{o(v)}$.

We further have $a = 2n + 3m$, i.e. $v \in O(a)$, so PSAT(PSN₁₊¹⁺) cannot be solved in time $2^{o(a)}$ unless the ETH is false. \square

Construction 6 Let \mathbb{F} be a 3-SAT instance with variables x_1, \dots, x_n and clauses c_1, \dots, c_m . Construct a corresponding PSAT(PSN₂₊¹⁺) instance $\mathbb{P} = \langle V, A, I, G \rangle$ as follows:

- $V = \{e_i, f_i, t_i \mid 1 \leq i \leq n\} \cup \{y_j \mid 1 \leq j \leq m\}$;
- A contains the actions
 - $\text{set } f_i : \{e_i\} \Rightarrow \{\bar{e}_i, f_i\}$, for all i ($1 \leq i \leq n$),
 - $\text{set } t_i : \{e_i\} \Rightarrow \{\bar{e}_i, t_i\}$, for all i ($1 \leq i \leq n$) and
 - $\text{vfy}_j^k : \{\bar{l}_j^k\} \Rightarrow \{y_j\}$, for all j, k ($1 \leq j \leq m, 1 \leq k \leq 3$), where $\bar{l}_j^k = f_i$ if $l_j^k = \bar{x}_i$ and $\bar{l}_j^k = t_i$ if $l_j^k = x_i$;
- $I = \{e_1, \dots, e_n\}$ and $G = \{y_1, \dots, y_m\}$.

Theorem 7 If $\text{PSAT}(\text{PSN}_2^{1+})$ can be solved in time $2^{o(v)}$ or time $2^{o(a)}$, then the ETH is false.

Proof. Analogous to the proof of Theorem 5, but using Construction 6 instead. For each variable x_i in \mathbb{F} , a plan for \mathbb{P} can set either f_i or t_i to true, but not both since we can never set e_i to true again once it is reset. Hence, this is a polynomial reduction from 3-SAT to $\text{PSAT}(\text{PSN}_2^{1+})$ such that $v = 3n + m$ and $a = 2n + 3m$. \square

It is further known that the problems $\text{PSAT}(\text{PSN}_{**}^{+})$, $\text{PSAT}(\text{PSN}_*^0)$ and $\text{PSAT}(\text{PSN}_1^{*+})$ can be solved in polynomial time [8]. Hence, we have a complete classification of all PSN classes defined by the number and polarity of preconditions and effects of actions, each being classified as either tractable or not solvable in subexponential time.

5 LOWER BOUNDS FOR MONOTONE LOP

We start with LOP for monotone instances. Finding a length-optimal plan is hard even when the actions are restricted to only one positive precondition and one positive effect.

Construction 8 Let \mathbb{F} be a 3-SAT instance with variables x_1, \dots, x_n and clauses c_1, \dots, c_m , where each clause c_j is on the form $\{l_j^1, l_j^2, l_j^3\}$. Construct a corresponding PSN_{1+}^{1+} instance $\mathbb{P} = \langle V, A, I, G \rangle$ as follows:

- $V = \{f_i, t_i, s_i \mid 1 \leq i \leq n\} \cup \{y_j \mid 1 \leq j \leq m\}$;
- A contains the actions
 $\text{set}_i^f : \emptyset \Rightarrow \{f_i\}$, $\text{set}_i^t : \emptyset \Rightarrow \{t_i\}$, $\text{set}_i^s : \{f_i\} \Rightarrow \{s_i\}$ and
 $\text{set}_i^s : \{t_i\} \Rightarrow \{s_i\}$, for all i ($1 \leq i \leq n$), and
 $\text{vf}_j^k : \{l_j^k\} \Rightarrow \{c_j\}$, for all j, k ($1 \leq j \leq m, 1 \leq k \leq 3$), where
 $\hat{l}_j^k = x_i^f$ if $l_j^k = \bar{x}_i$ and $\hat{l}_j^k = x_i^t$ if $l_j^k = x_i$;
- $I = \emptyset$ and $G = \{s_1, \dots, s_n\} \cup \{y_1, \dots, y_m\}$.

Theorem 9 If $\text{LOP}(\text{PSN}_{1+}^{1+})$ can be solved in time $2^{o(v)}$ or time $2^{o(a)}$, then the ETH is false.

Proof. Proof by reduction from 3-SAT to $\text{LOP}(\text{PSN}_{1+}^{1+})$. Let \mathbb{F} be a 3-SAT instance with variables x_1, \dots, x_n and clauses c_1, \dots, c_m , where each clause c_j is of the form $\{l_j^1, l_j^2, l_j^3\}$. Let \mathbb{P} be the corresponding PSN instance according to Construction 8. We claim that \mathbb{F} is satisfiable if and only if \mathbb{P} has a plan of length $2n + m$.

\Rightarrow : Suppose α is a satisfying assignment for \mathbb{F} . Construct a plan ω as follows. For each x_i , let ω contain set_i^f and set_i^s if $\alpha(x_i) = 0$ and otherwise let ω contain set_i^t and set_i^s . Then, for each clause $c_j = \{l_j^1, l_j^2, l_j^3\}$, there is at least one k such that α makes l_j^k true. Choose such a k and add action vf_j^k at the end of ω . Clearly, ω is a plan for \mathbb{P} of length $2n + m$.

\Leftarrow : Suppose ω is a plan for \mathbb{P} of length $2n + m$. It must contain n actions setting the s_i variables and m actions setting the y_j variables. In order to set the s_i variables, it must also set either of f_i and t_i for each i , but it cannot set both since there can only be n such actions in total. Hence, ω corresponds to a satisfying assignment.

It follows that the construction is a polynomial reduction.

Let S_d denote the class of 3-SAT instances where $m \leq dn$ and choose d such that S_d cannot be solved in time $2^{o(n)}$ if the ETH is true. Such a d must exist according to Lemma 2. Let P_d be the class of PSN_{1+}^{1+} instances we get by applying the reduction above to S_d .

Suppose $\text{LOP}(\text{PSN}_{1+}^{1+})$ can be solved in time in $2^{o(v)}$. Choose an arbitrary $c > 0$. Then $\text{LOP}(\text{PSN}_{1+}^{1+})$ can be solved in time 2^{cv} for large v . Let \mathbb{F} be a 3-SAT instance in S_d with n variables. Then \mathbb{F}

has $m \leq dn$ clauses. Let \mathbb{P} be the corresponding PSN instance. We have $v = 3n + m \leq 3n + dn = (3 + d)n$ and $|\mathbb{P}|$ is polynomial in n , so it follows from our assumption that we can solve the class S_d in time $\text{poly}(n) + 2^{c(3+d)n} \leq 2^{(c(3+d)+\epsilon)n}$, for all $\epsilon > 0$ and large n . However, c is arbitrary so we can also choose arbitrary $c', \epsilon > 0$ such that $c(3 + d) + \epsilon \leq c'$ and S_d can be solved in time $2^{c'n}$ for large n . Unless the ETH is false, this contradicts our assumptions and it follows that $\text{LOP}(\text{PSN}_{1+}^{1+})$ cannot be solved in time $2^{o(v)}$.

We further have $a = 4n + 3m$, i.e. $v \in O(a)$, so $\text{LOP}(\text{PSN}_{1+}^{1+})$ cannot be solved in time $2^{o(a)}$ unless the ETH is false. \square

This lower bound for $\text{LOP}(\text{PSN}_{1+}^{1+})$ holds even if the number of actions is linear in the number of variables.

Theorem 10 If $a \in \Omega(v)$, then $\text{LOP}(\text{PSN}_{1+}^{1+})$ cannot be solved in time $2^{o(v)}$ unless the ETH is false.

Proof. In the proof of Theorem 9 we have $|V| = 3n + m \leq 3n + dn = (3 + d)n$ and $|A| = 4n + 3m \leq 4n + 3dn = (4 + 3d)n$. Hence, the class P_d satisfies that $a \in \Omega(v)$ and the proof works also in this case. \square

We will see in Sec. 7 that this no longer holds if the number of actions is sublinear in the number of variables.

The previous theorem covers all cases of monotone planning, except when actions have no preconditions at all. In the case of no preconditions, we have to rely on a conjecture about the SET COVER problem, which is defined as follows:

k -SET COVER

Instance: A set S and a set C of subsets of S .

Question: Does S have a cover of size k , i.e. is there a subset $C' \subseteq C$ such that $\bigcup_{X \in C'} X = S$ and $|C'| \leq k$?

Cygan et al. [11] conjectured that k -SET COVER cannot be solved in time $2^{o(n)}$ unless the SETH is false, where $n = |S|$.

Theorem 11 $\text{LOP}(\text{PSN}_{k+}^0)$ cannot be solved in time $2^{o(v)}$ unless k -SET COVER can be solved in time $2^{o(n)}$.

Proof. Polynomial reduction from k -SET COVER. Given an instance $\mathbb{I} = \langle S, C \rangle$ of k -SET COVER, construct a $\text{LOP}(\text{PSN}_{k+}^0)$ instance $\mathbb{P} = \langle V, A, I, G \rangle$, where $V = S \cup \{y_c \mid c \in C\}$, A contains the action $a_c : \emptyset \Rightarrow \{x \mid x \in c\}$ for all $c \in C$ and all $x \in c$, $I = \emptyset$ and $G = S$. Clearly, \mathbb{P} has a plan of length k if and only if \mathbb{I} has a cover of size k . \square

6 LOWER BOUNDS FOR GENERAL LOP

For the general case, we prove a sharper bound than for monotone LOP based on results about graph colouring.

GRAPH COLOURABILITY

Instance: A graph $G = \langle V, E \rangle$ and a positive integer $k \leq |V|$.

Question: Is G k -colourable, i.e. is there a function $f : V \rightarrow \{1, \dots, k\}$ such that $f(u) \neq f(v)$ whenever $\{u, v\} \in E$?

The best known upper bound for GRAPH COLOURABILITY is time $2^n \cdot \text{poly}(n)$, where $n = |V|$, [7, Prop. 1], and it is considered an important open question whether a faster algorithm can exist [23]. Hence, we can use GRAPH COLOURABILITY instead of the ETH to prove sharper bounds, but under somewhat different assumptions. In particular, we show a sharper limit for planning with negative effects.

Theorem 12 If LOP(PSN) can be solved in time $2^{\frac{cv}{2}} \cdot \text{poly}(v)$ for some $c > 0$, then GRAPH COLOURABILITY can be solved in time $2^{cn} \cdot \text{poly}(n)$.

Proof. Proof by reduction from GRAPH COLOURABILITY to LOP(PSN). Let $\mathbb{I} = \langle G, k \rangle$ be an instance of GRAPH COLOURABILITY, where $G = \langle V, E \rangle$ is a graph and $k > 0$ is an integer. Assume $V = \{v_1, \dots, v_n\}$. Construct a corresponding LOP(PSN) instance $\mathbb{I}' = \langle \mathbb{P}', k' \rangle$ as follows. Let $\mathbb{P}' = \langle V', A', I', G' \rangle$, where

- $V' = \{v_1, \dots, v_n, b_1, \dots, b_n\}$;
- A' contains the actions $a_s : \emptyset \Rightarrow \{\overline{b_1}, \dots, \overline{b_n}\}$ and $a_i : \{\overline{b_i}\} \Rightarrow \{v_i\} \cup \{b_j \mid \{v_i, v_j\} \in E\}$ for all $v_i \in V$;
- $I' = \{\overline{b_1}, \dots, \overline{b_n}, \overline{v_1}, \dots, \overline{v_n}\}$ and $G' = \{v_1, \dots, v_n\}$.

Let $k' = n + k - 1$.

A plan colours the vertices in phases, one colour in each phase. The phases are separated by occurrences of action a_s , which switches to the next colour. Variable v_i is true if vertex v_i has been coloured and variable b_i is a blocking variable, preventing v_i from being coloured for the moment. The actual colour of a vertex is only implicit in the plan, and not explicitly represented. At the start of each phase, any node can be (re)coloured and colouring a node immediately blocks its neighbours from being coloured in the same phase.

We now claim that \mathbb{I} is k -colourable if and only if \mathbb{I}' has a plan of length k' .

\Rightarrow : Suppose G has a k -colouring. Then there is a partition C_1, \dots, C_k of V such that C_i is an independent set for all i . Create the action sequence $\omega = \omega_1, a_s, \omega_2, a_s, \dots, a_s, \omega_k$, where ω_i contains action a_j for each $v_j \in C_i$ in arbitrary order. The initial state guarantees that all b_j variables are false at the start of sequence ω_1 and the a_s actions guarantee that all b_j variables are false at the start of sequence ω_i for each $i > 1$. Since all vertices in C_i have the same colour, there are no two $v_j, v_h \in C_i$ such that $\{v_j, v_h\} \in E$. Hence, no action in ω_i will set b_j for any $v_j \in C_i$. It follows that all actions in ω_i are valid. Furthermore, the a_s actions are always valid. Since each $v_j \in V$ occurs in some C_i , it follows that action a_j occurs somewhere in ω for each $v_j \in V$. Hence, the resulting state satisfies G .

\Leftarrow : Suppose ω is a plan for \mathbb{P}' of length k' or less. Without losing generality, assume ω is a shortest such plan. Then ω does not contain any successive occurrences of action a_s , so it is of the form $\omega = \omega_1, a_s, \omega_2, a_s, \dots, a_s, \omega_m$, for some m , where the subplans ω_i do not contain any occurrences of action a_s . Since ω must contain at least one occurrence of action a_j for each $v_j \in V$, it follows that $|\omega| \geq n + m - 1$, i.e. $m \leq k$. Suppose there is some i and two actions a_j, a_h in ω_i such that $\{v_j, v_h\} \in E$. Without losing generality, assume v_j occurs before v_h . Then a_j sets b_h , but this blocks the execution of a_h . Hence, the assumption must be false and $\{v_j, v_h\} \notin E$ for all $a_j, a_h \in \omega_i$. It follows that G must have an m -colouring, and, thus, a k -colouring.

It follows that the construction is a polynomial reduction from GRAPH COLOURABILITY to LOP(PSN).

Now, suppose there is some $c > 0$ such that LOP(PSN) can be solved in time $2^{\frac{cv}{2}} \cdot \text{poly}(v)$. Since $|V'| = 2|V|$, we can solve GRAPH COLOURABILITY in time $2^{cn} \cdot \text{poly}(n)$. \square

This is a sharper result than the previous ones for monotone planning in the following sense. If LOP(PSN) can be solved faster than time $2^{\frac{v}{2}} \cdot \text{poly}(v)$, then there is a faster algorithm for GRAPH COLOURABILITY than previously known, i.e. this result is based on an assumption about a specific fixed value for the constant in the ex-

ponent. Note that Theorem 9 still applies, i.e. LOP(PSN) cannot be solved in time $2^{o(v)}$ unless the ETH is false.

7 SUBEXPONENTIAL SOLVABILITY

We will now demonstrate three PSN classes that can be solved in subexponential time in the number of variables, if the number of actions is subexponential in the number of variables.

For the first class, we need the following lemma.

Lemma 13 LOP(PSN $_{*+}^{*+}$) can be solved in time $O(2^{|A|})$.

Proof. Enumerate all subsets of A . For each such subset A' , we can apply the actions greedily until we either have a plan, or no more action is applicable. Since we try to find plan for each subset of A , we must find an optimal plan, so it is sufficient to keep track of the shortest plan found. Since there are $2^{|A|}$ subsets of A and each subset can be checked in polynomial time by the greedy strategy, it follows that we can solve LOP(PSN $_{*+}^{*+}$) in time $O(2^{|A|})$. \square

Theorem 14 LOP(PSN $_{*+}^{*+}$) can be solved in time $2^{o(v)}$ if $a \in o(v)$.

Proof. Immediate from Lemma 13. \square

There is a similarly sharp bound for actions with arbitrary preconditions, if we limit their effects to a constant number of variables.

Theorem 15 LOP(PSN $_{k+}^*$) can be solved in time $2^{o(v)}$ if $a \in o(v)$.

Proof. With a actions, we can set at most ka variables, so $v - ka$ variables are redundant and can be removed from the instance before solving it. Since $a \in o(v)$, we get $o(v)$ remaining variables. \square

If allowing also arbitrary preconditions, we get a somewhat less sharp bound.

Lemma 16 Generating all plans of length ℓ , or less, can be done in time $O(\ell^{|A|+2}|V|^2)$.

Proof. For $|A| \geq 2$ and $\ell \geq 2$, there are at most

$$0^{|A|} + 1^{|A|} + \dots + \ell^{|A|} = 1 + \sum_{i=1}^{\ell} i^{|A|} \leq \ell \cdot \ell^{|A|} = \ell^{|A|+1}$$

action sequences of length ℓ , or less. Each plan of length ℓ , or less, can be verified in time $O(\ell|V|^2)$. Hence, we can generate all plans of length ℓ , or less, in time $O(\ell^{|A|+1} \cdot \ell|V|^2) = O(\ell^{|A|+2} \cdot |V|^2)$. \square

Theorem 17 If $a \in o(\frac{v}{\log v})$, then LOP(PSN $_{*+}^*$) can be solved in time $2^{o(v)}$.

Proof. No action need to occur more than once in a plan for a monotone instance, so the maximum plan length is a . Hence, we know from Lemma 16 that we can generate all plans of length a , or less, in time $O(a^{a+2}v^2)$. Hence, we can solve LOP(PSN $_{*+}^*$) by keeping track of the shortest plan found.

It remains to prove that $a^{a+2}v^2 \in 2^{o(v)}$, but $a^{a+2}v^2 = a^a \cdot a^2v^2$ so we can show separately that $a^a \in 2^{o(v)}$ and $a^2v^2 \in 2^{o(v)}$.

We first prove that $a^a \in 2^{o(v)}$. Since $a \in o(\frac{v}{\log v})$, it holds for all $c > 0$ that $a < c\frac{v}{\log v}$, for large v . We get

$$a^a = 2^{a \log a} < 2^{(c\frac{v}{\log v}) \log \frac{c\frac{v}{\log v}}{\log v}} = 2^{c\frac{v}{\log v}}$$

Choose an arbitrary $c' > 0$. We want to prove that there is a $c > 0$ such that

$$2^{cv \frac{\log \frac{cv}{\log v}}{\log v}} \leq 2^{c'v},$$

that is,

$$cv \frac{\log \frac{cv}{\log v}}{\log v} \leq c'v.$$

We rewrite to

$$cv \frac{\log c + \log v - \log \log v}{\log v} \leq c'v,$$

but

$$\lim_{v \rightarrow \infty} \frac{\log c + \log v - \log \log v}{\log v} = 1$$

so it is sufficient to choose $c = c'$, which is allowed since we only require that $c > 0$. It follows that $a^a \in 2^{o(v)}$, since c' was chosen arbitrarily. We must next prove that also $a^2 v^2 \in 2^{o(v)}$. We know that $a \in o(v)$ so it holds for all $c > 0$ that $a < cv$, for large v . Choose $c = 1$. We get $a^2 v^2 \leq v^2 v^2 = v^4$ and it is straightforward that $v^4 \in 2^{o(v)}$. We have now shown that $a^a \in 2^{o(v)}$ and that $a^2 v^2 \in 2^{o(v)}$, so it follows that $a^a \cdot a^2 v^2 = a^{a+2} v^2 \in 2^{o(v)}$. \square

8 THE PUBS RESTRICTIONS

For the SAS^+ language, it is common to consider classes defined by combinations of the following four restrictions on instances [6].

P (post-unique): For all $v \in V$ and $x \in D(v)$, $\text{eff}(a)[v] = x$ for at most one $a \in A$.

U (unary): For each $a \in A$, $|\text{vars}(\text{eff}(a))| = 1$.

B (binary): $|D(v)| = 2$ for all $v \in V$.

S (single-valued): For all $a, b \in A$ and $v \in V$, if $v \in \text{vars}(\text{pre}(a)) \cap \text{vars}(\text{pre}(b))$ and $v \notin \text{vars}(\text{eff}(a)) \cup \text{vars}(\text{eff}(b))$ then $\text{pre}(a)[v] = \text{pre}(b)[v]$.

Combinations of these restrictions are written by juxtaposing the corresponding letters, eg. SAS^+ -PUB is the class of all SAS^+ instances that are post-unique, unary and binary. We will now prove subexponential lower-bound results for all non-tractable combinations of such restrictions. The following construction can be used to implement disjunctions in SAS^+ -PUB.

Construction 18 (Bäckström et al. [5], proof of Lemma 2) An OR gate g with two inputs x_1, x_2 and output y can be encoded as a SAS^+ -PUB instance $\mathbb{P} = \langle V, A, I, G \rangle$ as follows:

- $V = \{x_1, x_2, y, y_1, y_2, i_1, i_2\}$, all with domain $\{0, 1\}$,
- A contains the following actions:
 - $a_y : \{y_1 = 1, y_2 = 1\} \Rightarrow \{y = 1\}$,
 - $a_{y_1} : \{i_1 = 1, i_2 = 0\} \Rightarrow \{y_1 = 1\}$,
 - $a_{y_2} : \{i_1 = 0, i_2 = 1\} \Rightarrow \{y_2 = 1\}$,
 - $a_{i_1} : \{\emptyset\} \Rightarrow \{i_1 = 1\}$,
 - $a_{i_2} : \{\emptyset\} \Rightarrow \{i_2 = 1\}$,
 - $a_{v_1} : \{x_1 = 1\} \Rightarrow \{i_1 = 0\}$,
 - $a_{v_2} : \{x_2 = 1\} \Rightarrow \{i_2 = 0\}$.
- $I[x_1]$ and $I[x_2]$ are arbitrary and $I[v] = 0$ for all other $v \in V$.
- $G[y] = 1$ and G is otherwise undefined.

Theorem 19 If $\text{PSAT}(SAS^+\text{-PUB})$ can be solved in time $2^{o(v)}$ or time $2^{o(a)}$, then the ETH is false.

Proof sketch. Instance \mathbb{P} in Construction 4 is a SAS^+ -UB instance, but it is not post-unique since there are three actions with the same effect for each clause variable y_j . The three actions together simulate the disjunction in the clause. Construction 18 computes the logical OR of two variables in a PUB instance, using seven actions and four additional variables. This gadget has the property that it has no plan if both input variables are false, and otherwise it always has a plan of length six that sets the output variable. We can then compute the logical OR of four variables by using three such gadgets, using the outputs of the first two as inputs to the third. Since we only need three variables as input, we can use one of the variables for two inputs. We need three OR gadgets for each clause, and the outputs of the first two must be new variables. That is, we need $3 \cdot 4 + 2 = 14$ new variables for each clause, which yields a total of $2n + 15m$ variables for the instance. We also need $3 \cdot 7 = 21$ new actions for each clause, but the original three ones are not needed, so there are 18 actions per clause, which yields a total of $4n + 18m$ actions. The proof of Theorem 5 can easily be modified to this case. \square

Corollary 20 If $\text{PSAT}(SAS^+\text{-PBS})$ can be solved in time $2^{o(v)}$ or time $2^{o(a)}$, then the ETH is false.

Proof. Immediate from Theorem 19 since there is a polynomial reduction from $\text{LOP}(SAS^+\text{-PUB})$ to $\text{LOP}(SAS^+\text{-PBS})$ that increases the number of variables by a factor 2 [6, Proof of Thm. 4.16]. \square

It has remained an open question in the literature whether $\text{PSAT}(SAS^+\text{-PUB})$ and $\text{PSAT}(SAS^+\text{-PBS})$ are **NP**-hard, while the corresponding LOP problems are known to be **NP**-hard. Since the two preceding proofs use polynomial reduction from 3-SAT we can settle this question affirmatively as a spin-off result.

Corollary 21 $\text{PSAT}(SAS^+\text{-PUB})$ and $\text{PSAT}(SAS^+\text{-PBS})$ are **NP**-hard.

It is further known that $\text{PSAT}(SAS^+\text{-US})$ is in **P** [6] and corresponding results for all other combinations of the PUBS restrictions follow trivially, so this is a complete classification for PSAT for all combinations of the PUBS restrictions.

All the hardness results for PSAT above immediately apply also to LOP, but the tractable cases are fewer for LOP, it is only known that $\text{LOP}(SAS^+\text{-PUS})$ is in **P** [6]. It is sufficient to add the following result to get a complete classification also for LOP.

Corollary 22 (To Theorem 9) If $\text{LOP}(SAS^+\text{-UBS})$ can be solved in time $2^{o(v)}$ or time $2^{o(a)}$, then the ETH is false.

9 CAUSAL GRAPHS

The *causal graph* of a planning instance describes certain types of variable dependencies of a planning instance, and has frequently been exploited for identifying easy subclasses or for classifying the complexity of planning classes [19, 20, 26, 29, 38].

Definition 23 The causal graph for a SAS^+ instance $\mathbb{P} = \langle V, A, I, G \rangle$ is the directed graph $CG(\mathbb{P}) = \langle V, E \rangle$ where for all $u, v \in V$, $\langle u, v \rangle \in E$ if and only if both $u \neq v$ and there is some $a \in A$ such that $u \in \text{vars}(\text{pre}(a)) \cup \text{vars}(\text{eff}(a))$ and $v \in \text{vars}(\text{eff}(a))$.

We first show a general hardness result for instances with quite restricted causal graphs.

Theorem 24 If $\text{LOP}(\text{PSN}_{1+}^{1+})$ can be solved in time $2^{o(v)}$ or time $2^{o(a)}$ for instances where the causal graph is acyclic, bipartite and has degree 3 and depth 2, then the ETH is false.

Proof. Consider the construction in the proof of Theorem 9. The causal graph contains the following arcs:

- $\langle f_i, s_i \rangle$ and $\langle t_i, s_i \rangle$ for all i ($1 \leq i \leq n$);
- $\langle f_i, y_j \rangle$ for all i, j ($1 \leq i \leq n, 1 \leq j \leq m$) such that $\bar{x}_i \in c_j$ and $\langle t_i, y_j \rangle$ for all i, j ($1 \leq i \leq n, 1 \leq j \leq m$) such that $x_i \in c_j$.

This graph is bipartite and each y_j variable has at most 3 incoming arcs and no outgoing arcs. Similarly, each s_i variable has two incoming arcs and no outgoing arc. It also follows from Lemma 2 that we can restrict the class S_d in the proof of Theorem 5 to instances where each variable occurs at most 3 times. We can assume that the SAT instance is preprocessed so variables which occur with only one polarity are removed. Then each variable of type f_i or t_i has at most two outgoing arcs to variables of type y_j and one arc to variable s_i and no incoming arcs. It follows that the graph has degree 3. \square

It can be analogously shown that also Theorems 5 and 7 hold when restricted to instances where the causal graph is acyclic bipartite of degree 3 and depth 2.

We then continue to some special types of causal graphs that have been studied in the literature: out-stars (aka. forks), in-stars (aka. inverted forks), directed-path graphs (aka. chains) and fences. It is known that PSAT remains NP-hard when restricted to instances having a causal graph of either of these types [4, 14, 19]. We can sharpen these results by the following explicit lower bounds.

Theorem 25 If $\text{PSAT}(\text{SAS}_1^{+2})$ can be solved in time $2^{o(v)}$ or time $2^{o(a)}$ for instances where the causal graph is an out-star, then the ETH is false.

Proof sketch. There is a polynomial reduction from 3-SAT to $\text{PSAT}(\text{SAS}_1^{+2})$ with out-star causal graphs and $m + 1$ variables [4, Lemma 5]. This can be used to make a proof analogous to the one for Theorem 5. \square

Theorem 26 If $\text{PSAT}(\text{SAS}_1^{+1})$ can be solved in time $2^{o(v)}$ or time $2^{o(a)}$ for instances where the causal graph is an in-star, then the ETH is false.

Proof sketch. There is a polynomial reduction from 3-SAT to $\text{PSAT}(\text{SAS}_1^{+1})$ with in-star causal graphs and n variables [4, Lemma 4]. \square

Theorem 27 If $\text{PSAT}(\text{SAS}_1^{+2})$ can be solved in time $2^{o(v)}$ or time $2^{o(a)}$ for instances where the causal graph is a directed-path graph, then the ETH is false.

Proof sketch. There is a polynomial reduction from 3-SAT to $\text{PSAT}(\text{SAS}_1^{+2})$ with directed-path causal graphs and $(2m + 4)n$ variables [19, Proposition 5.5]. \square

Theorem 28 If $\text{PSAT}(\text{SAS}_1^{+1})$ can be solved in time $2^{o(v)}$ or time $2^{o(a)}$ for instances where the causal graph is a fence graph, then the ETH is false.

Proof sketch. There is a polynomial reduction from 3-SAT to $\text{PSAT}(\text{SAS}_1^{+1})$ with fence causal graphs and $2m + 1$ variables [4, Lemma 7]. \square

10 TIME VS. SPACE

The best upper bounds for hard problems usually assume algorithms that do not run in polynomial space. For instance, the result of Björklund et al. [7] that GRAPH COLOURABILITY can be solved in time $2^n \text{poly}(n)$ also requires using space $2^n \text{poly}(n)$. They also show an upper bound of time $2.2461^n \text{poly}(n)$ under the additional restriction of polynomial space [7, Proposition 7].

Our Observation 3 gives an upper bound of time $O(3^v)$ for monotone planning, but this result also requires space $O(3^v)$, since it is based on Dijkstra's algorithm. Almost all heuristic search algorithms also require exponential space. However, Depth-first search (DFS) runs in time $O(b^d)$ and space $O(bd)$ for implicitly represented graphs [32], where b is the branching factor and d is the search depth. Since the shortest plans are of length v at most, we can solve $\text{LOP}(\text{PSN}_{*+}^*)$ in time $O(b^v)$ and polynomial space. This is still heavily dependent on the branching factor, so we will present an algorithm that also runs in polynomial space and beats DFS for larger branching factors. To do so, we first need to recapitulate some theory on ordered partitions.

The Stirling number $\left\{ \begin{smallmatrix} n \\ k \end{smallmatrix} \right\}$ of the second kind denotes the number of ways we can partition a set of size n into k parts. Each partition of size k can be ordered in $k!$ different ways, so the total number of ordered partitions of all sizes of a set with n elements is $F(n) = \sum_{k=0}^n k! \left\{ \begin{smallmatrix} n \\ k \end{smallmatrix} \right\} = \frac{1}{2} \sum_{m=0}^{\infty} \frac{m^n}{2^m}$, which is known as the n th *Fubini number* (or the n th *ordered Bell number*).

Theorem 29 $\text{LOP}(\text{PSN}_{*+}^*)$ can be solved in time $O(F(v) \cdot \text{poly}(|\mathbb{P}|))$ using polynomial space.

Proof. Let $\mathbb{P} = \langle V, A, I, G \rangle$ be a PSN_{*+}^* instance. Let a_1, a_2, \dots, a_ℓ be a plan from I to some state s_ℓ , and let s_0, s_1, \dots, s_ℓ be its state sequence. Then $I = s_0 \subseteq s_1 \subseteq \dots \subseteq s_\ell$, since all action effects are positive. Furthermore, if the plan is optimal, then all subset relations are strict, since a_i is redundant if $s_{i-1} = s_i$. It follows that for each optimal plan $\omega = a_1, a_2, \dots, a_\ell$, there is some sequence $I = s_0 \subset s_1 \subset \dots \subset s_k = V$ of states such that s_0, s_1, \dots, s_ℓ is the state sequence of ω for some ℓ ($1 \leq \ell \leq k$). Hence, we can find all plans for \mathbb{P} by enumerating all such state sequences and check which ones have prefixes that correspond to a plan. Also define the sequence $\delta = d_1, d_2, \dots, d_k$ such that $d_i = s_i \setminus s_{i-1}$ for all i ($1 \leq i \leq k$). We note that δ is a partition on V , and it is furthermore an ordered partition since different orders on its parts generate different state sequences. That is, there is a one-to-one correspondence between the monotone state sequences and the ordered partitions, so it is sufficient to enumerate the latter.

For each ordered partition d_1, d_2, \dots, d_k , generate the corresponding state sequence s_0, s_1, \dots, s_k , where $s_0 = I$ and $s_i = s_{i-1} \cup d_i$ for all i . For all i from 0 to k do the following: If there is no $a \in A$ such that $\text{pre}(a) \subseteq s_{i-1}$ and $d_i \subseteq \text{eff}(a)$, then break and continue with the next partition. Otherwise, choose any such action as action a_i . If $G \subseteq s_i$, then break and remember i if it is the shortest plan length so far.

Generating all partitions of the set $\{1, \dots, n\}$ takes $O(1)$ amortized time per partition [35] and generating all permutations of $\{1, \dots, n\}$ takes $O(1)$ time per permutation [34], both in polynomial space. Hence, all $F(n)$ ordered partitions of $\{1, \dots, n\}$ can be generated in time $O(F(n))$. Checking each partition takes polynomial time in the instance size, so our algorithm runs in time $O(F(v) \cdot \text{poly}(|\mathbb{P}|))$ and uses only polynomial space, since it considers only one partition at a time. \square

It is known that $F(n) \simeq \frac{n!}{2(\ln 2)^{n+1}}$ [36] and that $n! < (\frac{n}{2})^n$,

so this algorithm will beat DFS for branching factors approximately greater than $\frac{v}{2}$. Furthermore, our algorithm does not even depend on the branching factor.

For the general, non-monotone case, however, we probably cannot hope for nearly as efficient algorithms under the polynomial-space constraint. It may even be difficult to find any algorithm running in polynomial space since the shortest plans may themselves be of exponential length in the general case. Exceptions exist in restricted cases, though. Jonsson and Bäckström [26] report a class of planning problems where the shortest solutions can be of exponential length, but it is always possible to decide in polynomial time if there is a solution or not. Jonsson [25] has further shown that it is even possible to generate a polynomial-size macro representation of a solution in this case. However, deciding if there is a solution of a specified length is **NP**-complete.

We do know, of course, that there must exist an algorithm for LOP(SAS⁺) that runs in polynomial space, since the problem is in **PSPACE**, but this does not tell us much about the actual time bounds. Even if using an implicit representation of the state-transition graph, most search algorithms may still use an exponential amount of memory. This applies even to depth-first search (since there are planning instances with exponentially long shortest solutions). By using Savitch's theorem [33], the amount of memory can be lowered. Savitch showed that there exists an algorithm \mathcal{A}_S that takes a graph $G = \langle U, E \rangle$ as input and checks whether there exists a path from $u \in U$ to $v \in U$ of length k or less using space $O(\log^2(|U|))$ and time $|U|^{O(\log k)}$. The time bound did not appear in Savitch's article, but it is a well-known folklore result. The only thing one has to keep in mind when using this time bound is that we must be able to check whether two vertices are connected or not in polynomial time (in the size of the graph). Problem PSAT can thus be solved by asking if there is a plan of length $k = |S| = 2^{|V|}$, i.e. by solving LOP for this value of k . If we assume an implicit graph representation (where vertex adjacency can be checked in $p(|\mathbb{P}|)$ time for some polynomial p) we can thus solve PSAT in time

$$\begin{aligned} & |S|^{O(\log k)} \cdot p(|\mathbb{P}|) \\ &= |S|^{O(\log |S|)} \cdot p(|\mathbb{P}|) = (d^{|V|})^{O(\log d^{|V|})} \cdot p(|\mathbb{P}|) \\ &= (d^{|V|})^{O(|V|)} \cdot p(|\mathbb{P}|) = d^{O(|V|^2)} \cdot p(|\mathbb{P}|) \end{aligned}$$

using space

$$O(\log^2 |S|) = O(\log^2 d^{|V|}) = O(|V|^2)$$

Savitch's theorem is clearly also useful for problem LOP, since checking whether there exists a plan of length k or less takes time $2^{O(|V| \log k)}$ and uses space $O(|V|^2)$, which can be substantially better than solving PSAT when k is moderately large.

We conclude by noting that the polynomial factor $p(|\mathbb{P}|)$ that we have used to cover the time for checking the action set of an instance \mathbb{P} is sufficient also for verifying an action.

Note that Savitch's theorem has been repeatedly applied to planning in the literature for proving membership in **PSPACE**. However, it has never been used to derive explicit bounds on time and space in the way we do.

11 DISCUSSION

Most of the planning classes that we prove not solvable in subexponential time (unless the ETH is false) are already known to be **NP**-hard. Our results are stronger in the following sense: even if it is

the case that $\mathbf{P} \neq \mathbf{NP}$, it is possible that an **NP**-hard problem can be solved in subexponential time (there are superpolynomial subexponential functions). Our results rule out that possibility (assuming the ETH holds).

For problems that are not solvable in polynomial time, one usually resorts to alternative methods, for instance, polynomial-time approximation algorithms or heuristic search, in the latter case hoping that this will perform satisfactorily in practice. However, with modern computers it is becoming increasingly popular to consider also algorithms running in superpolynomial time. Preferably, such an algorithm should still run in subexponential time. It is then interesting to know whether such an algorithm can exist or not, thus asking for the type of lower-bound results we derive in this paper. There are even cases where one considers algorithms, and even approximation algorithms, that require low-order exponential time [12]. In such cases, the performance is very sensitive to the constant in the exponent, requiring results in the style of our Theorem 12.

Our analysis of LOP is similar in spirit to recent analyses of lower bounds for CSP [27, 13]. It is interesting to note that they prove a case where CSP can be solved in time $2^{o(n)}$ if $m \in o(n)$, but cannot be solved in time $2^{o(n)}$ if $m \in \Omega(n)$ and the ETH holds, where n is the number of variables and m the number of constraint tuples. Although there are no immediate connections, this is a sharp easy-hard transition of the same type indicated by Theorem 10 contrasted with the results in Section 7.

Obviously the ratio a/v is crucial here. This has similarities to the phenomenon of phase transitions for **NP**-complete problems, which was pioneered by Cheeseman et al. [10] and has remained an active research area ever since. For instance, in the case of k -SAT, the phase transition occurs at a particular value of the ratio m/n for each k such that instances around this ratio are likely to be hard and the probability of hard instances is very low for other values of the ratio. While the vast majority of work in this area has been empirical, the exact values of the phase transitions for k -SAT have been determined analytically [2]. There is also a previous result on this type of phase transitions for planning, but with a very broad transition region rather than a sharp transition [9]. However, these are all transitions of the type easy-hard-easy. Our transition is of the type easy-hard and is, thus, more similar to the type of transitions for resolution proofs studied by Achlioptas et al. [1].

The upper bounds of time 3^v for monotone planning and time 4^v for the general case might, perhaps, suggest that the latter case is much harder, especially since monotone planning is **NP**-complete but the general case is **PSPACE**-complete. However, it is dangerous to draw any such conclusions, as Stearns [37] has pointed out:

Although **PSPACE**-completeness is stronger evidence of hardness than **NP**-completeness, there is no reason to believe that **PSPACE**-complete problems are harder in the sense that they require more time.

While upper-bound results sometimes take space into account, lower-bound results generally refer to time only, making no additional restrictions on space. Having seen in Sec. 10 how additional space restrictions can affect the upper bound, it is a valid question to ask if additional space bounds could also strengthen the lower-bound results upwards.

Acknowledgements

Bäckström is partially supported by the Swedish Research Council (VR) under grant 621-2014-4086.

REFERENCES

- [1] Dimitris Achlioptas, Paul Beame, and Michael S. O. Molloy, 'A sharp threshold in proof complexity yields lower bounds for satisfiability search', *J. Comput. Syst. Sci.*, **68**(2), 238–268, (2004).
- [2] Dimitris Achlioptas and Cristopher Moore, 'Random k -SAT: Two moments suffice to cross a sharp threshold', *SIAM J. Comput.*, **36**(3), 740–762, (2006).
- [3] Meysam Aghighi and Christer Bäckström, 'Cost-optimal and net-benefit planning - A parameterised complexity view', in *Proc. 24th Int'l Joint Conf. Artif. Intell. (IJCAI 2015), Buenos Aires, Argentina*, pp. 1487–1493, (2015).
- [4] Christer Bäckström and Peter Jonsson, 'A refined view of causal graphs and component sizes: SP-closed graph classes and beyond', *J. Artif. Intell. Res.*, **47**, 575–611, (2013).
- [5] Christer Bäckström, Peter Jonsson, Sebastian Ordyniak, and Stefan Szeider, 'A complete parameterized complexity analysis of bounded planning', *J. Comput. Syst. Sci.*, **81**(7), 1311–1332, (2015).
- [6] Christer Bäckström and Bernhard Nebel, 'Complexity results for SAS⁺ planning', *Comput. Intell.*, **11**, 625–656, (1995).
- [7] Andreas Björklund, Thore Husfeldt, and Mikko Koivisto, 'Set partitioning via inclusion-exclusion', *SIAM J. Comput.*, **39**(2), 546–563, (2009).
- [8] Tom Bylander, 'The computational complexity of propositional STRIPS planning', *Artif. Intell.*, **69**(1-2), 165–204, (1994).
- [9] Tom Bylander, 'A probabilistic analysis of propositional STRIPS planning', *Artif. Intell.*, **81**(1-2), 241–271, (1996).
- [10] Peter Cheeseman, Bob Kanefsky, and William M. Taylor, 'Where the really hard problems are', in *Proc. 12th Int'l Joint Conf. Artif. Intell. (IJCAI 1991) Sydney, Australia*, pp. 331–340, (1991).
- [11] Marek Cygan, Holger Dell, Daniel Lokshtanov, Dániel Marx, Jesper Nederlof, Yoshio Okamoto, Ramamohan Paturi, Saket Saurabh, and Magnus Wahlström, 'On problems as hard as CNF-SAT', in *Proc. 27th Conf. Computational Complexity, (CCC 2012), Porto, Portugal*, pp. 74–84, (2012).
- [12] Marek Cygan, Lukasz Kowalik, and Mateusz Wykurz, 'Exponential-time approximation of weighted set cover', *Inf. Process. Lett.*, **109**(16), 957–961, (2009).
- [13] Ronald de Haan, Iyad A. Kanj, and Stefan Szeider, 'On the subexponential-time complexity of CSP', *J. Artif. Intell. Res.*, **52**, 203–234, (2015).
- [14] Carmel Domshlak and Yefim Dinitz, 'Multi-agent off-line coordination: Structure and complexity', in *Proceedings of the 6th European Conference on Planning (ECP'01), Toledo, Spain*, (2001).
- [15] Ariel Felner, 'Position paper: Dijkstra's algorithm versus uniform cost search or a case against Dijkstra's algorithm', in *Proc. 4th Ann'l Symp. Combinatorial Search, (SoCS 2011), Castell de Cardona, Barcelona, Spain*, (2011).
- [16] Michael L. Fredman and Robert E. Tarjan, 'Fibonacci heaps and their uses in improved network optimization algorithms', in *Proc. 25th Ann. Symp. Foundations Comput. Sci. (FOCS'84), West Palm Beach, FL, USA*, pp. 338–346, (1984).
- [17] Hana Galperin and Avi Wigderson, 'Succinct representations of graphs', *Inform. Control*, **56**(3), 183–198, (1983).
- [18] Michael R. Garey and David S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, 1979.
- [19] Omer Giménez and Anders Jonsson, 'Planning over chain causal graphs for variables with domains of size 5 is NP-hard', *J. Artif. Intell. Res.*, **34**, 675–706, (2009).
- [20] Malte Helmert, 'The fast downward planning system', *J. Artif. Intell. Res.*, **26**, 191–246, (2006).
- [21] Jörg Hoffmann, 'Where 'ignoring delete lists' works: Local search topology in planning benchmarks', *J. Artif. Intell. Res.*, **24**, 685–758, (2005).
- [22] Russell Impagliazzo and Ramamohan Paturi, 'On the complexity of k -SAT', *J. Comput. Syst. Sci.*, **62**(2), 367–375, (2001).
- [23] Russell Impagliazzo and Ramamohan Paturi, 'Exact complexity and satisfiability - (invited talk)', in *Parameterized and Exact Computation - 8th Int'l Symp., (IPEC 2013), Sophia Antipolis, France, Revised Selected Papers*, pp. 1–3, (2013).
- [24] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane, 'Which problems have strongly exponential complexity?', *J. Comput. Syst. Sci.*, **63**(4), 512–530, (2001).
- [25] Anders Jonsson, 'The role of macros in tractable planning', *J. Artif. Intell. Res.*, **36**, 471–511, (2009).
- [26] Peter Jonsson and Christer Bäckström, 'Tractable plan existence does not imply tractable plan generation', *Annals Math. Artif. Intell.*, **22**(3-4), 281–296, (1998).
- [27] Iyad A. Kanj and Stefan Szeider, 'On the subexponential time complexity of CSP', in *Proc. 27th AAAI Conf. Artif. Intell. (AAAI 2013), Bellevue, WA., USA*, (2013).
- [28] Michael Katz and Carmel Domshlak, 'New islands of tractability of cost-optimal planning', *J. Artif. Intell. Res.*, **32**, 203–288, (2008).
- [29] Michael Katz and Carmel Domshlak, 'Optimal admissible composition of abstraction heuristics', *Artif. Intell.*, **174**(12-13), 767–798, (2010).
- [30] Martin Kronegger, Andreas Pfandler, and Reinhard Pichler, 'Parameterized complexity of optimal planning: A detailed map', in *Proc. 23rd Int'l Joint Conf. Artif. Intell. (IJCAI-13), Beijing, China*, pp. 954–961, (2013).
- [31] Daniel Lokshtanov, Dániel Marx, and Saket Saurabh, 'Lower bounds based on the exponential time hypothesis', *Bulletin of the EATCS*, **105**, 41–72, (2011).
- [32] Stuart J. Russell and Peter Norvig, *Artificial intelligence - a modern approach: the intelligent agent book*, Prentice Hall series in artificial intelligence, Prentice Hall, 1995.
- [33] W. Savitch, 'Relationships between nondeterministic and deterministic tape complexities', *J. Comput. Syst. Sci.*, **4**(2), 177–192, (1970).
- [34] Robert Sedgewick, 'Permutation generation methods', *ACM Comput. Surv.*, **9**(2), 137–164, (1977).
- [35] Ichiro Semba, 'An efficient algorithm for generating all partitions of the set $\{1, 2, \dots, n\}$ ', *J. Inf. Process.*, **7**(1), 41–42, (1984).
- [36] Renzo Sprugnoli, 'Riordan arrays and combinatorial sums', *Disc. Math.*, **132**(1-3), 267–290, (1994).
- [37] Richard Edwin Stearns, 'Turing award lecture: It's time to reconsider time', *Commun. ACM*, **37**(11), 95–99, (1994).
- [38] Brian C. Williams and P. Pandurang Nayak, 'A reactive planner for a model-based executive', in *Proc. 15th Int'l Joint Conf. Artif. Intell. (IJCAI'97), Nagoya, Japan*, pp. 1178–1185, (1997).

Abstraction-Based Verification of Infinite-State Reactive Modules

Francesco Belardinelli¹ and Alessio Lomuscio²

Abstract. We introduce the formalism of infinite-state reactive modules to reason about the strategic behaviour of autonomous agents in a setting where data are explicitly exhibited in the systems description and in the specification language. Technically, we endow reactive modules with an infinite domain of interpretation for individual variables, and introduce FO-ATL, a first-order version of alternating time temporal logic, for the specification of properties of interest. We show that their verification is decidable for classes of data types of interest. This result is proved by defining a first-order version of alternating bisimulations and finite bisimilar abstractions. We illustrate the formal machinery by applying it to English and sealed bid auctions. In particular, we show that strategic properties of agents in auctions, including manipulability and collusion, can be expressed and verified in this framework.

1 Introduction

The formalism of alternating-time temporal logic (ATL) [5] has been widely used to reason about the strategic behaviour of agents in multi-agent systems (MAS) [2, 13]. Specifications in ATL can express the ability of agents to bring about particular states of affairs (represented as temporal formulas in linear-time temporal logic) in the system. Models for ATL are traditionally given in terms of concurrent game structures, alternating transition systems, or in variants of interpreted systems. An attractive feature of some of these semantics is that models can be given via compact representations [26]. For instance, programs in the interpreted system programming language (ISPL), supported by the MCMAS model checker [30], generate interpreted systems upon which ATL formulas can be evaluated. Alternatively, *reactive modules* have been put forward as a flexible framework to model relevant behaviours of distributed systems, including pure and observable asynchronicity, atomic and non-atomic synchronicity [4]. Moreover, this compact representation constitutes the basis of the programming language for the jMOCHA model checker [3]. Both ISPL and simple reactive modules denote finite-state systems.

Yet, it is crucial to be able to reason about multi-agent systems that are intrinsically associated with infinite-state models. These arise naturally in MAS programming, when, for example, the data type of a given variable is not finite. These requirements may appear also in the modelling phase, when one is unable to assign a bound to a particular modelling concept, e.g., a queue. In this paper we focus on rational real variables [20]. Programs normally generate infinite-state models, but their verification is prone to undecidability, at least in the most general case.

In this paper we introduce *infinite-state reactive modules*, an extension of reactive modules to reason about the strategic behaviour of autonomous agents in a setting where data are explicitly exhibited in the systems description and in the specification language. We show that while their execution model is infinite, their model checking problem is decidable. Specifically, we endow reactive modules with an infinite domain of interpretation for individual variables, as well as relational symbols for total orders. Then, to express strategic behaviours of modules, we introduce FO-ATL, a first-order version of alternating time temporal logic, suitable for representing explicitly the data content of modules. We prove the decidability of the model checking problem for this setting by introducing a novel, first-order version of alternating bisimulation. We show that although reactive modules are infinite-state systems in general, for specific classes of data types we can construct a finite abstraction, which is bisimilar to the concrete, infinite-state system. As a result, the verification procedure can be conducted on the finite abstraction, and the result transferred to the original system. We illustrate the interest and workings of the formal machinery through an application to two auction mechanisms: English ascending bid auctions and repeated sealed bid auctions. In particular, we show that strategic properties of agents in auctions, including manipulability and collusion, can be expressed and verified in this framework.

Related Work. The area of logics for reasoning about strategies has witness a steady growth in recent years [5, 15, 31]. Here we consider only the contributions most closely related to the present setting. The inspiration for this work comes from the original paper on reactive modules [4], even though here we consider their *simple* version introduced in [33], where the *implicit* model checking problem is analysed, even though in a purely propositional setting.

This paper builds on a stream of results on the verification of *data-aware systems* [18, 11, 7, 10, 22], i.e., systems whose execution depends crucially on their data content. In these works a data model is coupled with an update mechanism that determine the system's evolution. For instance, in the line of [7, 14] the data model is represented by means of description logics, while the system evolves in response to conjunctive queries. In [22, 17] the formalism of situation calculus is exploited to represent the data model and update mechanisms. In this line of research, including [10, 23], the model checking problem is proved to be decidable by using two key features of these systems: *boundedness* (only a bounded number of individuals is active at each state in the system's execution) and *uniformity* (the system's execution is determined only by the elements that are named explicitly in the system's description). These properties are shared also by reactive modules. However, our contribution differs from previous works in several aspects. First, we adopt a modular, agent-based approach to data-aware systems, while in [18, 11, 7, 22]

¹ Imperial College London, UK, email: f.belardinelli@imperial.ac.uk

² Imperial College London, UK, email: a.lomuscio@imperial.ac.uk

systems are described monolithically. This feature is key to model distributed scenario, such as auctions, where agents have only imperfect knowledge of the system's global state. This is also reflected in the specification language: here we build upon ATL, a logic for strategies of individuals and coalitions, while previous works have focused on CTL, LTL [19], and μ -calculus, that account only for the system's global behaviour. Aspects of agency and individual knowledge have been analysed in [10], but protocols and actions are given completely abstractedly therein, while here we provide a computational semantics grounded on reactive modules.

Furthermore, this paper contributes towards the formal verification of auction-based mechanisms, which is a topic of growing interest in the AI community [24, 35, 34]. However, with some notable exceptions, most of the research in this area has focused on the design of auctioning mechanisms and the analysis of their formal properties, while the automated verification of these designs has only partially been addressed, and only for specific classes of auctions, by using purpose-built formalisms [6]. Here we put forward a principled approach to the verification of infinite-state systems that can handle general classes of auctions as well.

Scheme of the Paper. In Section 2 we introduce infinite-state (agent) modules, the first-order specification language FO-ATL, define the semantics of infinite-state reactive modules systems (IRMS) and the corresponding model checking problem. We exemplify and motivate the technical notions in Section 3, where we briefly describe modules for English auctions and repeated sealed bid auctions. Section 4 is devoted to the main result of this paper: decidability through finite bisimilar abstractions, which is then applied in Section 5 to our auctioning mechanisms. We conclude in Section 6 with discussion and future work.

2 Infinite-state Reactive Modules

In this section we introduce a generalisation of reactive modules [4, 33], that admits variables with an infinite domain of interpretation, possibly totally ordered (e.g., natural, rational, and real numbers). The specifications for these systems will be given in an expressive first-order extension of alternating-time temporal logic [5], also defined here.

In the following we assume a finite set $\mathcal{T} = \{T_1, \dots, T_k\}$ of types (e.g. booleans, integers, rationals, etc.), each endowed with a (possibly ordered) interpretation domain D_T . Also, for each type T we consider a set $V_T = \{v_0, v_1, \dots\}$ of variables and a set $P_T = \{x_0, x_1, \dots\}$ of parameters. We use V (resp. P) to denote $\bigcup_{T \in \mathcal{T}} V$ (resp. $\bigcup_{T \in \mathcal{T}} P$). Intuitively, variables are used to describe the data model, while parameters appear in formulas.

Further, we introduce a set Ag of agent modules (or simply agents), each comprising of a set L of local states, a set Act of actions, and a protocol function Pr , according to the formal account of agents in the literature on interpreted systems [21]. In line with reactive systems [4], we assume that each agent module $m \in Ag$ controls a finite set $cnt_m \subseteq V$ of variables. Specifically, $\{cnt_1, \dots, cnt_{|Ag|}\}$ form a partition of V , that is, every variable in V is controlled by exactly one agent. Hence, the set V can be assumed to be finite as well. Next, the set cnt_m of variables controlled by module m is partitioned into the sets $priv_m$ and $intf_m$ of private and interface variables respectively: private variables are only accessible to owner m , while interface variables are readable, but not writable, by any other agent. Given private and interface variables for a set Ag of agent modules, the variables in obs_m observable by agent module m are comprised of her controlled variables and the interface variables

of all agents, i.e., $obs_m = cnt_m \cup \bigcup_{j \neq m} intf_j$, or equivalently, $obs_m = priv_m \cup \bigcup_{j \in Ag} intf_j$. Observe that by considering controlled, private, and interface variables, in [4] the authors are able to model a number of different behaviours for reactive systems, including pure asynchronicity (interleaving), observable asynchronicity, atomic and non-atomic synchronicity. For our purposes, we will use private and interface variables to model partial observability and imperfect information of agents in auctions.

To provide a formal account of the local state of an agent module, we introduce local interpretations as functions $\theta_m : cnt_m \rightarrow D$, i.e., (finite, type-consistent) interpretations of the variables in cnt_m with values in D . For simplicity, in the following we often identify an interpretation θ_m with its range $\theta_m(cnt_m) \subseteq D$, whenever domain cnt_m is clear by the context. Then, provided interpretation $\theta_1, \dots, \theta_{Ag}$ for all agent modules, the local states $l \in L$ of agent module m is comprised of the values for her observed variables in obs_m , i.e., $l = \theta_m \cup \bigcup_{j \neq m} \theta_j(intf_j)$ by definition. Any local state l_m is assumed to characterise the knowledge of agent module m . Notice that l is well-defined as $\{cnt_i, \dots, cnt_{|Ag|}\}$ is a partition. Moreover, since the domain D is infinite in general, the set L of local states is infinite as well.

To define the individual actions in Act and the protocol Pr for each agent module, we introduce a typed first-order language built on variables, parameters and relational symbols $=$ and \leq when appropriate.

Definition 1 (FO-formulas) First-order formulas over types \mathcal{T} are defined according to the following BNF:

$$\phi ::= z = z' \mid z \leq z' \mid \neg \phi \mid \phi \rightarrow \phi \mid \forall x \phi$$

where $z, z' \in V \cup P$ have the same type, and $x \in P$.

The symbols $\neq, <, \geq, \top, \perp$, connectives \wedge, \vee , quantifier \exists , and free and bound variables and parameters are defined as standard [25]. Notice that quantification applies to parameters only, this is in accordance with the intuition above on the use of variables and parameters. Throughout the paper we assume that types are manipulated consistently, without explicitly mentioning this fact each time.

Following [33], we now introduce a particular notion of action.

Definition 2 (Guarded Command) A guarded command γ over V and P is an expression

$$g(x_1, \dots, x_k) \rightsquigarrow v_1 := x_1; \dots; v_k := x_k$$

where (i) guard g is an FO-formula with free parameters among x_1, \dots, x_k ; (ii) all v_i are variables in V ; and (iii) in each assignment $v_i := x_i$, v_i and x_i have the same type.

As customary [33], we require that no variable v_i appears on the left-hand-side of two assignments in the same guarded command (hence no issue on the ordering of updates arises). Also, the sets of parameters appearing in the various commands are assumed to be disjoint. The intuitive meaning of a guarded command is that if guard g evaluates to true for some interpretation $\sigma : P \rightarrow D$ of parameters, then the command is enabled for execution. By executing the command we set each variable v_i to value $\sigma(x_i) \in D$. We say that v_1, \dots, v_k are the variables controlled by γ , and denote this set by $ctr(\gamma)$, while the variables in g are the observable variables $obs(\gamma)$. A set of guarded commands is disjoint if their controlled variables are mutually disjoint. In particular, the skip command can be represented as $\top \rightsquigarrow \epsilon$, where ϵ is the empty sequence.

We now have all preliminary notions necessary to introduce agent modules.

Definition 3 (Agent Module) An agent module is a tuple $m = \langle \text{ctr}, \text{init}, \text{update} \rangle$ where

- $\text{ctr} \subseteq V$ is the (finite) set of variables controlled by m , partitioned into sets priv and intf ;
- init is a (finite) set of initialisation guarded commands s.t. for all $\gamma \in \text{init}$, $\text{ctr}(\gamma) \subseteq \text{ctr}$ and $\text{obs}(\gamma) \subseteq \text{obs}$;
- update is a (finite) set of update guarded commands s.t. for all $\gamma \in \text{update}$, $\text{ctr}(\gamma) \subseteq \text{ctr}$ and $\text{obs}(\gamma) \subseteq \text{obs}$.

According to Def. 3, an agent module initialises the variables she controls according to her guarded commands in init , then the same variables are updated following the commands in update . In particular, controllability and observability of variables in guards and assignments has to be respected. Given an agent module m , we denote the initialisation and update commands of m by init_m and update_m respectively.

Next, we define the *global state* of a reactive system as a tuple $s = \langle \theta_1, \dots, \theta_{|Ag|} \rangle$, where each θ_m is an interpretation for agent m . Equivalently, global states can be represented as functions $s : V \rightarrow D$, i.e., (finite, type-consistent) interpretations of the variables in V with values in D such that for every $v \in V$, $s(v) = \theta_m(v)$, where m is the agent controlling v . As anticipated above, any state s is well-defined as V is partitioned among the agents in Ag . Further, given a global state s , we denote as $l_1, \dots, l_{|Ag|}$ the corresponding local states for all agents in Ag . Observe that $\langle \theta_1, \dots, \theta_{|Ag|} \rangle$ and $\langle l_1, \dots, l_{|Ag|} \rangle$ are equivalent representation of a global state s , in terms of controlled, respectively observable, variables. So, we will use the two notations interchangeably. We remarked that agent modules have only partial observability of the global state of the system. Specifically, two states s and s' are *indistinguishable* for agent module m , or $s \sim_m s'$, iff $l_m = l'_m$, that is, iff s and s' coincide on the interpretation of obs_m . We denote the set of all global states as \mathcal{G} .

To introduce the semantics of guarded commands formally, we define the satisfaction relation \models for FO-formulas. An FO-formula ϕ is given meaning by a *finite interpretation* $\sigma : \text{fr}(\phi) \rightarrow D$ that assigns values in D to the free parameters in ϕ . A *reinterpretation* σ_u^x coincides with σ , but assigns value $u \in D$ to parameter $x \in \text{fr}(\phi)$. By Σ we denote the set of all interpretations σ for parameters. Further, given $z \in V \cup P$, $(s, \sigma)(z) = s(z)$ for $z \in V$, and $(s, \sigma)(z) = \sigma(z)$ for $z \in P$, that is, variables are interpreted according to s , while parameters according to σ .

Definition 4 (Satisfaction) A state s satisfies an FO-formula ϕ for a finite interpretation σ , or $(s, \sigma) \models \phi$, iff (clauses for propositional connectives are immediate and thus omitted)

$$\begin{aligned} (s, \sigma) \models z = z' & \quad \text{iff} & \quad (s, \sigma)(z) = (s, \sigma)(z') \\ (s, \sigma) \models z \leq z' & \quad \text{iff} & \quad (s, \sigma)(z) \leq (s, \sigma)(z') \\ (s, \sigma) \models \forall x \phi & \quad \text{iff} & \quad \text{for all } u \in s(V), \sigma_u^x \models \phi \end{aligned}$$

The interpretation of FO-formulas is completely standard, but for quantification that takes values from the finite set $s(V) = \{u \in D \mid u = s(v) \text{ for some } v \in V\}$ of images of variables in V . This is consistent with the interpretation of quantification on *active domains* in database theory [1]. Indeed, at this stage quantification can be considered syntactic sugar, as $s(V)$ is finite. However, once the temporal evolution of reactive modules is taken into account – as we shall see shortly – quantification makes the specification language strictly more expressive than its propositional counterpart.

Since the set V of variables and the set P of parameters appearing free in any guarded command are both finite and defined at design-time, we deem them fixed. Hence, hereafter we will always consider

suitable states s and finite interpretations σ . Further, when evaluating guards for commands of an agent module m , it is sufficient to look at the interpretation of observable variables provided by m 's local state l_m . Therefore, we can introduce the satisfaction relation $(l_m, \sigma) \models \phi$, for $\text{fr}(\phi) \subseteq \text{obs}_m$, in analogy with Def. 4.

Finally, an *infinite-state reactive module system* (IRMS) is defined as a set $M = \{m_1, \dots, m_{|Ag|}\}$ of agent modules. Given an IRMS M , the sets Act and ACT of *individual* and *joint actions*, the set I of *initial states*, the *protocol* Pr , the *transition function* τ , and the set S of *reachable states* are defined as follows:

- for every agent module m , $Act = \text{update}$; while ACT is the set of tuples of update commands $(\gamma_1, \dots, \gamma_{|Ag|})$, for $\gamma_m \in \text{update}_m$;
- I is the set of states s' such that for every $v \in V$, $s'(v) = \sigma(x)$ for some state s , interpretation σ , and initialisation command $\gamma = (\gamma_1, \dots, \gamma_{|Ag|})$, such that each local state satisfies the corresponding guard, i.e., $(l_m, \sigma) \models g_m$;
- $Pr_m : L \rightarrow ((2^{Act_m} \setminus \emptyset) \times \Sigma)$ such that $Pr_m(l) = \{(\gamma, \sigma) \mid (l, \sigma) \models g_\gamma\}$;
- $\tau : \mathcal{G} \times ACT \times \Sigma \rightarrow \mathcal{G}$ such that $\tau(s, \gamma, \sigma) = s'$ iff (i) for all $m \in M$, $(\gamma_m, \sigma_m) \in Pr_m(l_m)$; and (ii) $s'(v_i) = \sigma(x_i)$. Often we write $s \xrightarrow{\gamma, \sigma} s'$ for $\tau(s, \gamma, \sigma) = s'$;
- S is the closure of I according to the transition function τ .

The definitions above provide the computational counterpart to the notions of action and protocol introduced earlier. Specifically, an agent module m can update the variables she controls by means of actions in Act , according to protocol Pr , which returns the actions whose guard is satisfied in her local state l_m for some interpretation σ of parameters. Overall, an IRMS M describes the evolution of a reactive system from an initial state $s \in I$, according to the transition function τ , which returns the successive global state provided the current state and an enabled joint action (including its data content σ). Again, since the domain D is infinite in general, IRMS are infinite-state systems, differently from [33].

Since we assumed that the sets of parameters in the various commands are disjoint, in the definition of τ the restriction σ_m of interpretation σ to the parameters in update_m is well-defined. Hereafter we make use of a notion of *extension* $\sigma \subseteq \sigma'$ between interpretations, viewed as functions. Hence, above we have that for every $m \in M$, $\sigma_m \subseteq \sigma$.

To specify the behaviour of IRMS and to reason about the strategic abilities of agent modules, we introduce a first-order version of alternating-time temporal logic.

Definition 5 (FO-ATL) Formulas in first-order ATL are defined in BNF as follows:

$$\psi ::= \phi \mid \neg\psi \mid \psi \wedge \psi \mid \forall x \psi \mid \langle\langle C \rangle\rangle X \psi \mid \langle\langle C \rangle\rangle (\psi U \psi) \mid \langle\langle C \rangle\rangle G \psi$$

where ϕ is an FO-formula and $C \subseteq M$ is coalition of agents.

The meaning of ATL operators is standard: a formula $\langle\langle C \rangle\rangle \Phi$ says that *coalition C has a (collective) strategy to achieve Φ* . Again, quantification is defined on parameters only. Notice that in FO-ATL we can have arbitrary alternations of quantifiers and ATL operators. A consequence of this, as we shall see, is that quantification in FO-ATL is not syntactic sugar. Also, this is in contrast with previous works [7, 10, 22] that consider modalities (e.g., CTL, LTL, μ -calculus) capable of expressing only the temporal evolution of the system, but do not support naturally the specification of strategic abilities of agents.

To interpret FO-ATL formulas, we introduce a suitable notion of *local strategy*.

Definition 6 (Strategy) An imperfect information, memoryless strategy (henceforth simply a strategy) for an agent module $m \in M$ is a function $f_m : L \rightarrow Act_m$ such that for every local state $l \in L$, $(f_m(l), \sigma) \in Pr_m(l)$ for some interpretation $\sigma \in \Sigma$.

We can check that strategies, as introduced in Def. 6, are *uniform* in the sense of [28], as they only depend on the local state of agents. In particular, if $s \sim_m s'$, then $l_m = l'_m$ by definition, and therefore $f_m(l_m) = f_m(l'_m)$.

Given an IRMS M , a path λ is an infinite sequence $s_0 s_1 \dots$ of states, in which $\lambda(i)$ denotes the $i+1$ -th element s_i of λ . Further, for a set $F_C = \{f_m \mid m \in C\}$ of strategies, a path λ is F_C -compatible iff for every $j \geq 0$, $\lambda(j+1) = \tau(\lambda(j), \gamma, \sigma)$ for some joint action γ and interpretation σ such that (i) for $m \in C$, $\gamma_m = f_m(\lambda(j)_m)$; and (ii) for $m \notin C$, $(\gamma_m, \sigma_m) \in Pr_m(\lambda(j)_m)$. We denote the set of F_C -compatible paths from state s as $out(s, F_C)$.

In the following definition we assume that the sets of parameters appearing in commands and in formula ϕ are disjoint. This can be done without loss of generality, as both sets are finite and defined at design-time.

Definition 7 (Satisfaction) Given an IRMS M , a state s satisfies an FO-ATL formula ψ for interpretation σ , or $(M, s, \sigma) \models \psi$, iff (clauses for propositional connectives are immediate and thus omitted).

$$\begin{aligned} (M, s, \sigma) \models \phi & \quad \text{iff } (s, \sigma) \models \phi, \text{ where } \phi \text{ is an FO-formula} \\ (M, s, \sigma) \models \forall x \psi & \quad \text{iff for every } u \in s(V), (M, s, \sigma_u^x) \models \psi \\ (M, s, \sigma) \models \langle\langle C \rangle\rangle X \psi & \quad \text{iff for some strategy } F_C, \text{ for all } \lambda \in out(s, F_C), \\ & \quad (M, \lambda(1), \sigma) \models \psi \\ (M, s, \sigma) \models \langle\langle C \rangle\rangle G \psi & \quad \text{iff for some strategy } F_C, \text{ for all } \lambda \in out(s, F_C), \\ & \quad \text{for all } i \geq 0, (M, \lambda(i), \sigma) \models \psi \\ (M, s, \sigma) \models \langle\langle C \rangle\rangle (\psi U \psi') & \quad \text{iff for some strategy } F_C, \text{ for all } \lambda \in out(s, F_C), \\ & \quad \text{for some } i \geq 0, (M, \lambda(i), \sigma) \models \psi', \text{ and} \\ & \quad \text{for all } j, 0 \leq j < i \text{ implies } (M, \lambda(j), \sigma) \models \psi \end{aligned}$$

We remark that the semantics of ATL operators in Def. 7 is standard, while quantification ranges on the active domain $s(V)$. However, differently from Def. 4, quantification is not syntactic sugar: transitions might take us to a successor state s' , in which an individual $u \in s(V)$ is no longer active, i.e., $u \notin s'(V)$. As a consequence, quantification in FO-ATL gives us a language that is strictly more expressive than propositional ATL, as it allows to refer to individuals across states. This feature of FO-ATL will become apparent in Section 3.

An FO-ATL formula ψ is *true* in state s , or $(M, s) \models \psi$, iff for all interpretations σ , $(M, s, \sigma) \models \psi$; ψ is *true* in M , or $M \models \psi$, iff for all initial states $s \in I$, $(M, s) \models \psi$. We can now state the model checking problem for infinite-state reactive module systems against FO-ATL.

Definition 8 (Model Checking) Given an IRMS M and an FO-ATL formula ψ , the model checking problem concerns determining whether $M \models \psi$.

Notice that M is an infinite-state system, and the model checking problem for infinite-state data-aware systems is normally undecidable [18]. However, in what follows we define an abstraction-based technique to obtain decidability. First we present some instances of IRMS.

3 Auctions

In this section we illustrate the formal machinery introduced in Section 2 with examples from the literature on auctions. Specifically, we

model English ascending bid auctions and repeated sealed auctions as infinite-state reactive module systems, and specify the behaviour of agents participating in the corresponding IRMS by means of FO-ATL formulas. We provide an informal description of these auctioning mechanisms and refer to [20] for further details.

In English auctions several bidders bid for an item auctioned by the auctioneer. All the participating agents can be represented as modules, beginning with the auctioneer.

Definition 9 (Auctioneer) The auctioneer module $m_a = \langle ctr_a, init_a, update_a \rangle$ is such that

- $ctr_a = \{base, t_out\} = inf_{t_a}$, while $priv_a = \emptyset$. Variable t_out has type boolean, while $base$ ranges over the rational numbers.
- $init_a$ contains guarded commands:

$$\top \rightsquigarrow base := x_1; t_out := \perp$$

- $update_a$ contains guarded commands skip and

$$t_out = \perp \rightsquigarrow t_out := \top$$

$$t_out = \top \rightsquigarrow base := x_2; t_out := \perp$$

Intuitively, the auctioneer module keeps track of the base price $base$ for the auctioned item (given as a rational), and owns a boolean variable t_out to terminate non-deterministically the bidding round (both are public). At the start of the execution the auctioneer initialises the base price $base$ to a random rational number x_1 and t_out to false (\perp). Then, by using the updates, she can either do nothing or terminate the bidding round, and then start a new one, with a different base price x_2 for a possibly different item.

The modules for bidders can be given as follows.

Definition 10 (Bidder) The bidder module $m_i = \langle ctr_i, init_i, update_i \rangle$ is such that

- $ctr_i = \{tvalue_i, bid_i\}$ with $inf_{t_i} = \{bid_i\}$ and $priv_i = \{tvalue_i\}$. Both $tvalue_i$ and bid_i range over rational numbers.
- $init_i$ contains guarded commands:

$$\top \rightsquigarrow bid_i := uu; tvalue_i := x_3$$

- $update_i$ contains guarded commands skip and

$$(t_out = \perp) \wedge \bigwedge_{j \in M} (bid_j = uu) \wedge (x_4 \leq tvalue_i) \rightsquigarrow bid_i := x_4$$

$$(t_out = \perp) \wedge \bigvee_{j \neq i} (bid_i < bid_j) \wedge$$

$$\bigwedge_{j \neq i} (bid_j \neq uu \rightarrow bid_j < x_5) \wedge (x_5 \leq tvalue_i) \rightsquigarrow bid_i := x_5$$

$$t_out = \top \rightsquigarrow bid_i := uu; tvalue_i := x_6$$

By Def. 10 every bidder i has a *private* true value $tvalue_i \in \mathbb{Q}$, up to which she is happy to bid, and a *public* $bid_i \in \mathbb{Q}$. At the beginning she initialises her true value, while her bid is set to ‘undefined’. Thereafter, she might choose to bid and then update it according to the other bidders’ offers. At the end of the bidding round, she reinitialises her true value for a new round.

Given the auctioneer and bidder modules as defined above, an IRMS for an English auction is a set $M = \{m_a, m_1, \dots, m_n\}$ of modules for the auctioneer a and bidders b_1, \dots, b_n . Since base prices, true values, and bids all take rationals as values, M is actually an infinite-state system. Notice that IRMS M is non-terminating, as

agents can skip indefinitely. We can eliminate such behaviours by introducing fairness constraints. Also, new items are put on auction, thus bidders can take part in successive auctions.

As a further example of the expressivity of IRMS we present modules for a repeated sealed auction, in which, differently from above, the winning bid is used to provide feedback on the value of the base price and true values for the next bidding round. This scenario is inspired to real-time bidding, where this feedback is provided by complex algorithms [32]. Given the limited expressivity of our specification language, here we consider a much simpler mechanism.

We start with a new module for the auctioneer.

Definition 11 (Auctioneer) *The auctioneer module $m_a = \langle ctr_a, init_a, update_a \rangle$ is such that*

- $ctr_a = \{base, t_out, w_bid\} = intf_a$, while $priv_a = \emptyset$, where w_bid has type rational;
- $init_a$ contains guarded commands:

$$\top \rightsquigarrow base := x_7; t_out := \perp; w_bid := uu;$$

- $update$ contains guarded commands

$$(t_out = \perp) \wedge \bigvee_{i \in M} (bid_i = x_8) \wedge \bigwedge_{j \neq i} (bid_j \neq uu \rightarrow bid_j \leq bid_i) \rightsquigarrow w_bid = x_8 \wedge t_out := \top$$

$$(t_out = \top) \wedge (x_9 \leq w_bid) \rightsquigarrow base := x_9; t_out := \perp$$

By Def. 11 bidding rounds are only one-step long (given by toggling t_out from false to true), in line with sealed auctions. Further, the auctioneer keeps track of the highest (winning) bid w_bid , and makes use of this value as upper bound when setting the new base price for the next bidding round.

As regards bidders, the corresponding module is as follows:

Definition 12 (Bidder) *The bidder module $m_i = \langle ctr_i, init_i, update_i \rangle$ is such that component $init_i$ is given as above, while*

- $ctr_i = \{tvalue_i, bid_i\} = priv_i$ and $intf_i = \emptyset$;
- $update_i$ contains guarded commands skip and

$$(t_out = \perp) \wedge (x_9 \leq tvalue_i) \rightsquigarrow bid_i := x_9$$

$$(t_out = \top) \wedge (w_bid \leq x_{10}) \rightsquigarrow bid_i := uu; tvalue_i := x_{10}$$

Notice that, differently from Def. 10, now bids are private, all bidders submit them at the same time and cannot raise them, as it is customary in sealed bid auctions. Moreover, bidders use the winning bid as lower bound when setting the new true value for the next bidding round. We observe that the feedback provided by the winning bid is rather crude; more sophisticated mechanisms can be considered. Note that no quantification appears in the guards of commands for the auctioneer and bidders. In fact, we remarked above that quantification in guards is purely syntactic sugar. However, this is no longer the case when quantification is combined with ATL operators, as we now show.

Once we modelled auctions as IRMS, we might want to verify the behaviour of the auctioneer and bidders against properties written in FO-ATL. For instance, we might want to check that there is always one base price $base$ and each bidder i is associated with *at most* one defined true value $tvalue_i$ (possibly none). This can be expressed in FO-ATL as follows:

$$AG(\exists!x(base = x) \wedge \exists^{\leq 1}y(y \neq uu \wedge tvalue_i = y))$$

where the CTL operator AG is tantamount to $\langle\langle\emptyset\rangle\rangle G$, and quantifiers $\exists!$ and $\exists^{\leq 1}$ are defined as standard in first-order logic with identity.

Further, each bidder b_i can (has a strategy to) bid less or as much as her true value:

$$\langle\langle b_i \rangle\rangle G (bid_i \leq tvalue_i)$$

More interestingly, we can specify elaborate strategic abilities of agents. For instance, each bidder b_i can raise her bid unless she has already hit her true value:

$$AG \forall x(x = bid_i \rightarrow (x = tvalue_i \vee \langle\langle b_i \rangle\rangle F \exists y(y > x \wedge y = bid_i))) \quad (1)$$

Observe that in (1) the use of quantification on parameters allows us to compare values of bid_i at different moments of the system's execution. Such features are not expressible in a purely propositional language.

A crucial notion to analyse in auctions is *collusion*: does a certain coalition C of bidders have a strategy to win the auction, possibly by bidding a lower amount than 'normally necessary'? In order to express variants of this property in FO-ATL, we introduce a formula $win_i = t_out \wedge \bigwedge_{j \neq i} (bid_j \leq bid_i)$, which intuitively says that bidder b_i is among the winners of the current auction (we may have multiple winner, but this issue is not relevant for the present discussion). The simplest form of collusion we can specify in FO-ATL states that coalition C can act so as to enforce that one of its members eventually wins:

$$\langle\langle C \rangle\rangle F \bigvee_{b_i \in C} win_i$$

Yet another key property analysed on auctions is *manipulability*: certain agents might exploit the auction design to their benefit. For instance, we might want to check that if bidder b_i has a strategy to win by bidding x , then, no matter what the other bidders do, b_i has a strategy to win in which x is strictly less than her true value:

$$AG \forall x(\langle\langle b_i \rangle\rangle (win_i \wedge bid_i = x) \rightarrow \llbracket Ag \setminus \{b_i\} \rrbracket (win_i \wedge x < tvalue_i))$$

We conclude this section by observing that IRMS, together with FO-ATL, are a sound framework to represent various relevant types of auctions, as well as to specify interesting properties thereof. In the following section we study the problem of verifying these properties on infinite-state reactive module systems.

4 Decidability by Finite Abstraction

In this section we introduce abstractions of infinite-state reactive module systems, and show that, under specific assumptions, these abstractions are finite. Moreover, we prove a preservation result for FO-ATL specifications that allows us to verify an IRSM by model checking its finite abstraction. Here we use ideas from [10, 9], but contextualise them to ATL specifications. In the rest of the section we consider a *finite* abstract interpretation domain D_T^A for every type T . We start by considering a notion of abstract (global) state.

Definition 13 (Abstract Global State) *Given an IRMS $M = \{m_1, \dots, m_{|Ag|}\}$, an abstract global state is a tuple $s = \langle \theta_1, \dots, \theta_{|Ag|} \rangle$ of interpretations $\theta_m : cnt_m \rightarrow D^A$, together with a total order \leq_s defined on each D_T^A (when appropriate).*

Notice that, differently from the concrete global states in Section 2, the total order \leq_s depends on the particular abstract state s , rather than being defined on D_T .

Next, we introduce a notion of *isomorphism* between concrete and abstract states.

Definition 14 (Isomorphism) *A global state s and an abstract state s' are isomorphic, or $s \simeq s'$, iff for some type-consistent bijection $\iota : s(V) \mapsto s'(V)$, we have*

- (i) *for every $m \in Ag$, $\theta'_m = \iota \circ \theta_m$;*
- (ii) *ι preserves \leq , that is, for every $v, v' \in V$, $s(v) \leq s(v')$ iff $s'(v) \leq_{s'} s'(v')$.*

Any function ι as above is a witness for $s \simeq s'$, or $s \stackrel{\iota}{\simeq} s'$ for short.

Intuitively, isomorphic states share the same relational structure. Observe that, as regards order $\leq_{s'}$, witness ι preserves the interpretation of individuals in the active domain $s'(V)$ only. This feature of isomorphisms is key to obtain finite abstractions.

Now we show that isomorphic states satisfy the same FO-formulas ϕ . However, ϕ might contain free variables interpreted outside the active domain. This remark motivates the following definition.

Definition 15 (Equivalent Interpretations) *Given a state s , an isomorphic abstract state s' , and a FO-formula ϕ , the finite interpretations $\sigma : fr(\phi) \mapsto D$ and $\sigma' : fr(\phi) \mapsto D^A$ are equivalent for ϕ w.r.t. s and s' iff for some bijection $\chi : s(V) \cup \sigma(fr(\phi)) \mapsto s'(V) \cup \sigma'(fr(\phi))$, we have*

- (i) *the restriction $\chi|_{s(V)}$ is a witness for $s \simeq s'$;*
- (ii) $\sigma' = \chi \circ \sigma$;
- (iii) *for every $u, u' \in s(V) \cup \sigma(fr(\phi))$, $u \leq u'$ iff $\chi(u) \leq_{s'} \chi(u')$.*

Again, notice that for order $\leq_{s'}$, a witness χ preserves only the interpretation of individuals in $s'(V) \cup \sigma(fr(\phi))$, which are in finite number. As customary in first-order logic, we can prove that equivalent assignments on isomorphic states preserve FO-formulas. We report this result for our particular setting.

Lemma 1 *Given a state s , and an isomorphic abstract state s' , if interpretations σ and σ' are equivalent for FO-formula ϕ w.r.t. s and s' , then*

$$(s, \sigma) \models \phi \quad \text{iff} \quad (s', \sigma') \models \phi$$

Proof. The proof is by induction on the structure of ϕ . If $\phi \equiv (z = z')$, then $(s, \sigma) \models \phi$ iff $(s, \sigma)(z) = (s, \sigma)(z')$. In particular, for some bijection $\chi : s(V) \cup \sigma(fr(\phi)) \mapsto s'(V) \cup \sigma'(fr(\phi))$, we have $s \stackrel{\chi}{\simeq} s'$ and $\sigma' = \chi \circ \sigma$. Hence, $(s', \sigma')(z) = (\chi \circ s, \chi \circ \sigma)(z) = \chi((s, \sigma)(z)) = \chi((s, \sigma)(z')) = (s', \sigma')(z')$, as required.

If $\phi \equiv (z \leq z')$, then $(s, \sigma) \models \phi$ iff $(s, \sigma)(z) \leq (s, \sigma)(z')$. Again, this is the case iff $(s', \sigma')(z) \leq_{s'} (s', \sigma')(z')$, as $s \stackrel{\chi}{\simeq} s'$ and for every $u, u' \in s(V) \cup \sigma(fr(\phi))$, $u \leq u'$ iff $\chi(u) \leq_{s'} \chi(u')$.

The cases for propositional connectives are immediate.

If $\phi \equiv \forall x \psi$, then $(s, \sigma) \models \phi$ iff for all $u \in s(V)$, $(s, \sigma_u^x) \models \psi$. Observe that interpretations σ_u^x and $\sigma_{\chi(u)}^x$ are equivalent for ψ w.r.t. s and s' . Hence, by induction hypothesis, $(s', \sigma_{\chi(u)}^x) \models \psi$. Since u is arbitrary and $\chi : s(V) \rightarrow s'(V)$ is a bijection, we have that $(s', \sigma') \models \phi$. \square

Lemma 1 also applies to local states. Hence, if s, s' are isomorphic, and σ, σ' are equivalent for all guards g_γ of module m 's commands, then ‘isomorphic’ actions are available to module m in s and s' , that is, $(\gamma, \sigma) \in Pr_m(s_m)$ iff $(\gamma, \sigma') \in Pr'_m(s'_m)$, for the same command γ . This remark will be frequently used in the following, without explicitly mentioning it.

We now define the execution of an infinite-state reactive module system on the abstract domain D^A . Given an IRMS M defined on infinite domain D , the abstraction M^A of M is the same IRMS, executed on abstract domain D^A with a different semantics. Specifically, the abstract components Act_m^A , ACT^A , Pr_m^A , and S^A are defined as for M (in particular, $Act_m^A = Act_m$ and $ACT^A = ACT$), while components I^A and τ^A are given as follows:

- I^A is the set of abstract states s' such that for every $v \in V$, $s'(v) = \sigma(x)$ for some state s , interpretation σ , and initialisation command γ , such that $(s, \sigma) \models g_\gamma$. Moreover, $\leq_{s'}$ is any total linear extension on D^A of the partial order $\leq_s|_{s'(V)}$;
- $\tau^A : \mathcal{G}^A \times ACT^A \times \Sigma^A \rightarrow 2^{\mathcal{G}^A}$ is such that $s' \in \tau^A(s, \gamma, \sigma)$ iff (i) for all $m \in M$, $(\gamma_m, \sigma_m) \in Pr_m^A(l_m)$; and (ii) $s'(v_i) = \sigma(x_i)$. Moreover, $\leq_{s'}$ is any total linear extension of the partial order $\leq_s|_{s'(V)}$.

The main difference between the concrete and abstract execution of an IRMS is that in the latter the total orders are updated at each transition. Specifically, in a transition from a state s to s' we preserve the order \leq_s on the elements in s that appear also in the active domain of s' . As to the remaining elements in $D^A \setminus (s(V) \cap s'(V))$, we extend the restriction $\leq_s|_{s'(V)}$ to the whole D^A arbitrarily. Notice that this can be done in polynomial time. By doing so, the abstract transition τ^A is non-deterministic, differently from the concrete τ . In particular, all abstract states $s' \in \tau^A(s, \gamma, \sigma)$ are total completions of the partial order $\leq_s|_{s'(V)}$.

We now prove that Lemma 1 can be lifted to the full language FO-ATL. To do so, we need a few more definitions. First, an IRMS is *dense (with no end points)* iff for each type, the total linear order \leq is. Also, for a type T , $par_T(\psi)$ is the set of all parameters (free and bound) appearing in ψ , while N_T denotes the set of parameters appearing in all guarded command, for all agents. For instance, in the IRMS for English auctions in Section 3, for the type of rational numbers $N = 2 + 4n$, where n is the number of bidders. Finally, for a coalition $C \subseteq Ag$, a C -action $\gamma_C \in Act_C$ is a tuple of commands for all agents in C . We say that a joint action $\gamma' \in ACT$ extends γ_C , or $\gamma' \supseteq \gamma_C$, iff for all $i \in C$, $\gamma'_i = \gamma_i$.

We now prove the following auxiliary lemma, which states that, provided a sufficient number of abstract values, transitions in a dense IRMS M (with no endpoints) can be replicated in the abstraction M^A , and viceversa.

Lemma 2 *Consider a dense IRMS M with abstraction M^A , a state $s \in S$, an isomorphic abstract state $s' \in S'$, and an FO-ATL formula ψ . If for every type T , $|D_T^A| \geq |V_T| + |par_T(\psi)| + |N_T|$, then for every assignments σ and σ' equivalent for ψ w.r.t. s and s' , we have*

1. *for every C -action $\gamma_C \in Act_C$ and finite interpretation ρ_C , if for every $m \in C$, $(\gamma_m, \rho_m) \in Pr_m(l_m)$, then there exists a finite interpretation ρ'_C such that for every $m \in C$, $(\gamma_m, \rho'_m) \in Pr_m^A(l'_m)$, and for every extension $\gamma \supseteq \gamma_C$ and $\rho \supseteq \rho'_C$, if $s' \xrightarrow{\gamma, \rho'} t'$, then for some extension $\rho \supseteq \rho_C$, $s \xrightarrow{\gamma, \rho} t$ and σ and σ' are equivalent for ψ w.r.t. t and t' .*
2. *for every C -action $\gamma'_C \in Act_C^A$ and finite interpretation ρ'_C , if for every $m \in C$, $(\gamma'_m, \rho'_m) \in Pr_m^A(l'_m)$, then there exists a finite interpretation ρ_C such that for every $m \in C$, $(\gamma'_m, \rho_m) \in Pr_m(l_m)$, and for every extension $\gamma' \supseteq \gamma'_C$ and $\rho \supseteq \rho_C$, if $s \xrightarrow{\gamma', \rho} t$, then for some extension $\rho' \supseteq \rho'_C$, $s' \xrightarrow{\gamma', \rho'} t'$ and σ and σ' are equivalent for ψ w.r.t. t and t' .*

Proof. To prove (1), let $\chi : s(V) \cup \sigma(fr(\psi)) \mapsto s'(V) \cup \sigma'(fr(\psi))$ be a bijection witnessing that σ and σ' are equivalent for ψ w.r.t. s

and s' , i.e., $\sigma' = \chi \circ \sigma$. Also, consider C -action γ_C and finite interpretation ρ_C such that for every $m \in C$, $(\gamma_m, \rho_m) \in Pr_m(l_m)$. Since for every type T , $|D_T^A| \geq |V_T| + |par_T(\varphi)| + |N_T|$, we can extend χ to an injective function $\bar{\chi} : s(V) \cup \sigma(fr(\psi)) \cup \rho_C(N) \rightarrow D^A$ satisfying the condition: $u \leq u'$ iff $\bar{\chi}(u) \leq_{s'} \bar{\chi}(u')$. Then define $\rho'_C = \bar{\chi} \circ \rho_C$. By construction and Lemma 1, for every $m \in C$ and action γ_m , $(l'_m, \rho'_m) \models g_m$ iff $(l_m, \rho_m) \models g_m$. Hence, action γ_C is enabled in state s' as well. Then, consider the execution of any joint action $\gamma \sqsupseteq \gamma_C$ with parameters $\rho' \supseteq \rho'_C$, thus giving $s' \xrightarrow{\gamma, \rho'} t'$ for some $t' \in S^A$. Since the IRMS M is infinite, dense, and with no end points, we can always find a further extension $\bar{\bar{\chi}} : s'(V) \cup \sigma'(fr(\psi)) \cup \rho'(N) \rightarrow D$, that agrees with the converse $\bar{\chi}^{-1}$ of $\bar{\chi}$ on $s'(V) \cup \sigma'(fr(\psi)) \cup \rho'_C(N)$ and also satisfies the condition: $u \leq_{s'} u'$ iff $\bar{\bar{\chi}}(u) \leq \bar{\bar{\chi}}(u')$. Then define $\rho = \bar{\bar{\chi}} \circ \rho'$. Again, by construction and Lemma 1, for every $m \in Ag$ and action γ_m , $(l_m, \rho_m) \models g_m$ iff $(l'_m, \rho'_m) \models g_m$. Hence, joint action γ is also enabled in state s , and the execution of γ in s with parameters ρ gives $s \xrightarrow{\gamma, \rho} t$, where in particular t is isomorphic to t' with witness $\bar{\bar{\chi}}$. Also, by the construction above, σ and σ' are equivalent for ψ w.r.t. t and t' .

The proof for (2) follows a similar line of reasoning: given a C -action $\gamma'_C \in Act_C^A$ and finite interpretation ρ'_C such that for every $m \in C$, $(\gamma'_m, \rho'_m) \in Pr_m^A(l'_m)$, by exploiting the density of M (and the lack of end points), we can construct a finite interpretation ρ_C such that for every $m \in C$, $(\gamma'_m, \rho_m) \in Pr_m(l_m)$. Moreover, for every joint action $\gamma' \sqsupseteq \gamma'_C$ and extension $\rho \supseteq \rho_C$, if $s \xrightarrow{\gamma', \rho} t$, then by using the constraint on the cardinality of D^A in M^A , we can construct an extension $\rho' \supseteq \rho'_C$ such that $s' \xrightarrow{\gamma', \rho'} t'$ and σ and σ' are equivalent for ψ w.r.t. t and t' . \square

Lemma 2 states that, by the constraint on the cardinality of D^A , in abstraction M^A we have ‘enough’ elements to simulate the transitions in M . On the other hand, by using density (and the lack of endpoints), in M we can simulate the transitions in abstraction M^A . Actually, Lemma 2 provides a notion of alternating bisimulation for first-order ATL, which is an original contribution of the paper to our knowledge. Also, this result is applied in the proof of the following key lemma.

Lemma 3 Consider a dense IRMS M with abstraction M^A , state $s \in S$ and isomorphic abstract state $s' \in S'$, and an FO-ATL formula ψ . If for every type T , $|D_T^A| \geq |V_T| + |par_T(\varphi)| + |N_T|$, then for every assignments σ and σ' equivalent for ψ w.r.t. s and s' , we have

1. for every joint strategy F_C , there exists F'_C such that for every $\lambda' \in out(s', F'_C)$, there exists some $\lambda \in out(s, F_C)$ such that for all $i \geq 0$, σ and σ' are equivalent for ψ w.r.t. $\lambda(i)$ and $\lambda'(i)$.
2. for every joint strategy F'_C , there exists F_C such that for every $\lambda \in out(s, F_C)$, there exists some $\lambda' \in out(s', F'_C)$ such that for all $i \geq 0$, σ and σ' are equivalent for ψ w.r.t. $\lambda(i)$ and $\lambda'(i)$.

Proof. We begin by proving (1). We build the strategy F'_C and prove the statement of the lemma by induction on length n of paths. For $n = 0$, we have that σ and σ' are equivalent for ψ w.r.t. $\lambda(0) = s$ and $\lambda'(0) = s'$. As to the inductive step, suppose that σ and σ' are equivalent for φ w.r.t. $\lambda(i)$ and $\lambda'(i)$, and consider C -action γ_C such that for every $m \in C$, $\gamma_m \in F_C(\lambda(i)_m)$ and finite interpretation ρ_C such that for every $m \in C$, $(\gamma_m, \rho_m) \in Pr_m(l_m)$. We set $F'_C(\lambda'(i)_m) = \gamma_m$ for every $m \in C$. In particular, by Lemma 2.1 there exists a finite interpretation ρ'_C such that for every $m \in C$, $(\gamma_m, \rho'_m) \in Pr_m^A(l'_m)$. Now consider the joint action $\gamma \sqsupseteq \gamma_C$

and extension $\rho' \supseteq \rho'_C$ such that $\lambda'(i) \xrightarrow{\gamma, \rho'} \lambda'(i+1)$. Again by Lemma 2.1, there exist some extension $\rho \supseteq \rho_C$ and $t \in S$ such that $\lambda(i) \xrightarrow{\gamma, \rho} t$. Then, set $\lambda(i+1) = t$. In particular, σ and σ' are equivalent for ψ w.r.t. $\lambda(i+1)$ and $\lambda'(i+1)$, and therefore the statement of the lemma is satisfied.

Item (2) is proved similarly, by using Lemma 2.2 instead. \square

Intuitively, Lemma 3 states that joint strategies can be simulated between M and its abstraction M^A , provided that the relevant constraint are met. By this lemma we can prove the main result of this section.

Theorem 4 Consider a dense IRMS M , its abstraction M^A , a state $s \in S$, an isomorphic abstract state $s' \in S'$, and an FO-ATL formula ψ . If for every type T , $|D_T^A| \geq |V_T| + |par_T(\varphi)| + |N_T|$, then for every assignments σ and σ' equivalent for ψ w.r.t. s and s' , we have that

$$(M, s, \sigma) \models \psi \quad \text{iff} \quad (M^A, s', \sigma') \models \psi.$$

Proof. The proof is by induction on the structure of ψ . The base case for first-order formulas follows by Lemma 1; the inductive cases for propositional connectives are immediate. In particular, notice that, since for every type T , $|D_T^A| \geq |V_T| + |par_T(\neg\psi)| + |N_T| = |V_T| + |par_T(\psi)| + |N_T|$ and $|D_T^A| \geq |V_T| + |par_T(\psi_1 \wedge \psi_2)| + |N_T| \geq |V_T| + |par_T(\psi_i)| + |N_T|$, for $i = 1, 2$, the induction hypothesis holds.

For $\psi \equiv \forall x \phi$, $(M, s, \sigma) \models \psi$ iff for all $u \in s(V)$, $(M, s, \sigma_u^x) \models \phi$. If χ is a witness to the fact that σ and σ' are equivalent for ψ w.r.t. s and s' , then interpretations σ_u^x and $\sigma'_{\chi(u)^x}$ are equivalent for ϕ (also w.r.t. s and s'). Moreover, $|D_T^A| \geq |V_T| + |par_T(\psi)| + |N_T| \geq |V_T| + |par_T(\phi)| + |N_T|$. Hence, the induction hypothesis holds and it follows that $(M^A, s', \sigma'_{\chi(u)^x}) \models \phi$. Since χ is a bijection, we obtain that $(M^A, s', \sigma') \models \psi$.

Suppose that $\psi \equiv \langle\langle C \rangle\rangle X \phi$. As regards the \Rightarrow direction, $(M, s, \sigma) \models \psi$ iff for some joint strategy F_C , for every $\lambda \in out(s, F_C)$, $(M, \lambda(1), \sigma) \models \phi$. By Lemma 3.1 there exists a strategy F'_C depending on F_C , such that for every $\lambda' \in out(s', F'_C)$, there exists some $\lambda \in out(s, F_C)$ such that σ and σ' are equivalent for φ w.r.t. $\lambda(1)$ and $\lambda'(1)$. Since $|D_T^A| \geq |V_T| + |par_T(\psi)| + |N_T| = |V_T| + |par_T(\phi)| + |N_T|$, by induction hypothesis $(M^A, \lambda'(1), \sigma') \models \psi$ for every $\lambda' \in out(s', F'_C)$, that is, $(M^A, s', \sigma') \models \psi$. The \Leftarrow direction is proved similarly, by using Lemma 3.2. The proof for the other ATL operators follows an analogous line of reasoning. \square

By Theorem 4 a dense IRMS M and its abstraction M^A satisfy the same formulas in FO-ATL, whenever the abstract domain D^A contains enough elements to replicate transitions in M . Most importantly, M^A can be assumed to be finite. As a consequence, we can verify an FO-ATL formula ψ on M by model checking M^A . We state this last result formally in the following corollary.

Corollary 5 Consider a dense IRMS M , its abstraction M^A , and an FO-ATL formula φ . If for every type T , $|D_T^A| \geq |V_T| + |par_T(\varphi)| + |N_T|$, then

$$M \models \varphi \quad \text{iff} \quad M^A \models \varphi$$

To conclude, we have identified a significant class of infinite-state reactive module systems for which the model checking problem is decidable. Specifically, whenever, the orders on the domains of interpretations are assumed to be dense, with no end points, as it is

the case for the rational number in the auction IRMS, by choosing a domain of abstract values of appropriate cardinality, we can construct a finite abstraction that preserves the interpretation of FO-ATL formulas.

5 Discussion: Complexity and Abstract Auctions

In Section 4 we provided an abstraction-based technique for the verification of IRMS. Here we elaborate more on the model checking procedure.

By previous contributions [18, 10] we know that the model checking problem for *finite* general data-aware systems, against a first-order extension of CTL, is EXPSPACE-complete in the combined size $|D| + \|\varphi\|$ of data and the formula. Since ATL subsumes CTL, we immediately obtain that model checking finite reactive module systems against FO-ATL is EXPSPACE-hard. Moreover, by combining the procedures for model checking ATL under imperfect information and first-order logic, we obtain an algorithm in EXPSPACE for FO-ATL. Hence, we can state the complexity of model checking, for instance, the finite abstractions in Section 4.

Theorem 6 *The model checking problem for finite reactive module systems with respect to FO-ATL is EXPSPACE-complete in the combined size $|D| + \|\varphi\|$ of data and the formula.*

It should be noted that the complexity is the same as that of model checking similar structures against first-order CTL [10]. Thus, the enhanced expressiveness of ATL comes at no extra computational cost. This is in contrast with the propositional case, where complexity jumps from PTIME to Δ_2^P under imperfect information [27]. Here the situation is different as the complexity of model checking data trumps that of the modal fragment. Moreover, we discussed above that reactive module can be thought of as compact representations of transition systems, where states and transitions are given implicitly in the form of agent programs. Hence, Def. 8 is really an instance of *implicit* model checking, whose complexity is typically higher than the explicit counterparts [12].

Additionally, we anticipate to be able to find cases of interest, whose complexity is amenable to practical model checking. For example, consider the IRMS for English auctions described in Section 3 for n bidders. The only infinite type are the rationals used to represent bids, true values, and base prices, for which we have $|V| = 1 + 2n$ variables, $|N| = 2 + 4n$ parameters, and $par(\phi) = 5n$, where ϕ is the conjunction of all specifications appearing in Section 3. Thus, to simulate rational values in the IRMS for English auctions, it is sufficient to consider a domain D of abstract elements, whose size is $|D| \geq (1 + 2n) + (2 + 4n) + 5n = 3 + 11n$, but finite and linear in the number n of bidders. As a result, to verify English auctions against the strategic behaviours formalised in Section 3, it is enough to consider the finite abstraction built on such finite domain D , and whose execution is described in Section 4. Moreover, to alleviate further the verification burden, we can exploit the symmetries of IRMS, both at the level of data and of the behaviours of agents (e.g., all bidders share the same actions and protocol). Thus, we envisage to deploy data-symmetry reductions [16] as well as further abstraction methodologies (e.g., multi-valued abstraction [8, 29]) on the finite, abstract reactive module system. We leave this for future work.

6 Conclusions

In this paper we have put forward a technique for the verification of infinite-state MAS against first-order modal specifications expressing strategic abilities of agents. The contributions of the paper are as

follows. Firstly, we have introduced infinite-state reactive modules as an extension of simple reactive modules [33], which are suitable to model MAS with variables ranging over infinite domains. Secondly, we have defined a first-order ATL to specify strategic interactions of reactive modules in IRMS. However, the execution of IRMS generates infinite-state systems; these normally admit an undecidable verification problem. The third contribution consisted in observing that IRMS admit a decidable verification problem under specific conditions, namely a dense total order with no endpoints. Additional validation of the formalism here studied came from the modelling of auctions: IRMS were used to describe formally English and repeated sealed bid auctions, while FO-ATL was employed to capture specifications accounting for infinite domains. Purely in terms of modelling, we are not aware of other formalisms able to capture the strategic interaction of agents in a first-order setting. Moreover, to achieve the decidability results, specifically to show that a finite abstract model can be used to reason about an IRMS, we introduced a novel notion of alternating bisimulation at the first order. This notion is likely to be applicable in similar forms to show decidability for other first-order logics for strategic reasoning.

Several extensions of the proposed framework appear promising. Firstly, we envisage to introduce epistemic operators in FO-ATL to represent individual and group knowledge explicitly, and be able to express secrecy properties in auctioning scenarios. Secondly, IRMS can be extended with richer specification languages supporting arithmetic operations for instance [19].

Acknowledgements

This research was funded by the EPSRC under grant EP/I00520X.

REFERENCES

- [1] S. Abiteboul, R. Hull, and V. Vianu, *Foundations of Databases*, Addison-Wesley, 1995.
- [2] T. Ågotnes, V. Goranko, W. Jamroga, and M. Wooldridge, ‘Knowledge and ability’, in *Handbook of Logics for Knowledge and Belief*, College Publications, (2015).
- [3] R. Alur, L. de Alfaro, R. Grosu, T. Henzinger, A. Thomas, M. Kang, C. Kirsch, R. Majumdar, F. Mang, and B.-Y. Wang, ‘jMocha: A model checking tool that exploits design structure’, in *Proceedings of the 23rd International Conference on Software Engineering (ICSE01)*, pp. 835–836. IEEE, (2001).
- [4] R. Alur and T. Henzinger, ‘Reactive modules’, *Formal Methods in System Design*, **15**(1), 7–48, (1999).
- [5] R. Alur, T. A. Henzinger, and O. Kupferman, ‘Alternating-time temporal logic’, *Journal of the ACM*, **49**(5), 672–713, (2002).
- [6] A. Badica and C. Badica, ‘Specification and verification of an agent-based auction service’, in *Information Systems Development*, 239–248, Springer US, (2010).
- [7] B. Bagheri, D. Calvanese, M. Montali, G. Giacomo, and A. Deutsch, ‘Verification of relational data-centric dynamic systems with external services’, in *Proceedings of the 32nd Symposium on Principles of Database Systems (PODS13)*, pp. 163–174. ACM, (2013).
- [8] T. Ball and O. Kupferman, ‘An abstraction-refinement framework for multi-agent systems’, in *Proceedings of the 21st Annual IEEE Symposium on Logic in Computer Science (LICS06)*, pp. 379–388. IEEE, (2006).
- [9] F. Belardinelli, ‘Model checking auctions as artifact systems: Decidability via finite abstraction’, in *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI14)*, pp. 81–86, (2014).
- [10] F. Belardinelli, A. Lomuscio, and F. Patrizi, ‘Verification of agent-based artifact systems’, *Journal of Artificial Intelligence Research*, **51**, 333–376, (2014).
- [11] K. Bhattacharya, C. Gerede, R. Hull, R. Liu, and J. Su, ‘Towards formal analysis of artifact-centric business process models’, in *Business Process Management: Proceedings of the 5th International Conference (BPM07)*, pp. 288–304. Springer, (2007).

- [12] N. Bulling, J. Dix, and W. Jamroga, ‘Model checking logics of strategic ability: Complexity’, in *Specification and Verification of Multi-agent Systems*, 125–159, Springer, (2010).
- [13] N. Bulling and W. Jamroga, ‘Comparing variants of strategic ability: how uncertainty and memory influence general properties of games’, *Autonomous Agents and Multi-Agent Systems*, **28**(3), 474–518, (2014).
- [14] D. Calvanese, G. Delzanno, and M. Montali, ‘Verification of relational multiagent systems with data types’, in *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI15)*, pp. 2031–2037. AAAI Press, (2015).
- [15] K. Chatterjee, T. Henzinger, and N. Piterman, ‘Strategy logic’, in *Proceedings of the 18th International Conference on Concurrency Theory (CONCUR07)*, volume 4703, pp. 59–73, (2007).
- [16] M. Cohen, M. Dam, A. Lomuscio, and H. Qu, ‘A symmetry reduction technique for model checking temporal-epistemic logic’, in *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI09)*, pp. 721–726, (2009).
- [17] G. de Giacomo, Y. Lesperance, F. Patrizi, and S. Vassos, ‘Progression and verification of situation calculus agents with bounded beliefs’, in *Proceedings of the International conference on Autonomous Agents and Multi-Agent Systems (AAMAS14)*, pp. 141–148. IFAAMAS, (2014).
- [18] A. Deutsch, R. Hull, F. Patrizi, and V. Vianu, ‘Automatic verification of data-centric business processes’, in *Proceedings of the 12th International Conference on Database Theory (ICDT09)*, pp. 252–267. ACM, (2009).
- [19] A. Deutsch, Y. Li, and V. Vianu, ‘Verification of hierarchical artifact systems’.
- [20] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*, Cambridge University Press, New York, NY, USA, 2010.
- [21] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi, *Reasoning about Knowledge*, MIT Press, Cambridge, 1995.
- [22] G. De Giacomo, Y. Lesperance, and F. Patrizi, ‘Bounded epistemic situation calculus theories’, in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI13)*, (2013).
- [23] P. Gonzalez, A. Griesmayer, and A. Lomuscio, ‘Verification of GSM-based artifact-centric systems by predicate abstraction’, in *Proceedings of the 13th International Conference on Service Oriented Computing (ICSOC15)*, volume 9435 of *Lecture Notes in Computer Science*, pp. 253–268. Springer, (2015).
- [24] E. M. Tadjouddine F. Guerin and W. Vasconcelos, ‘Abstractions for model-checking game-theoretic properties of auctions’, in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems (AAMAS08)*, pp. 1613–1616. IFAAMAS, (2008).
- [25] A. G. Hamilton, *Logic for Mathematicians*, Cambridge University Press, 1978.
- [26] W. Jamroga and T. Ågotnes, ‘Modular interpreted systems’, in *Proceedings of the 6th International Conference on Autonomous Agents and Multi-Agent systems (AAMAS07)*, pp. 131–138. IFAAMAS, (2007).
- [27] W. Jamroga and J. Dix, ‘Model checking abilities under incomplete information is indeed δ_p^2 -complete’, in *Proceedings of the 4th European Workshop on Multi-Agent Systems (EUMAS’06)*, pp. 14–15, (2006).
- [28] W. Jamroga and W. van der Hoek, ‘Agents that know how to play’, *Fundamenta Informaticae*, **62**, 1–35, (2004).
- [29] A. Lomuscio and J. Michaliszyn, ‘Verification of multi-agent systems via predicate abstraction against ATLK specifications’, in *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS16)*, (2016).
- [30] A. Lomuscio, H. Qu, and F. Raimondi, ‘MCMAS: A model checker for the verification of multi-agent systems’, *Software Tools for Technology Transfer*, (2015). <http://dx.doi.org/10.1007/s10009-015-0378-x>.
- [31] F. Mogavero, A. Murano, G. Perelli, and M. Y. Vardi, ‘Reasoning about strategies: On the model-checking problem’, *ACM Transactions in Computational Logic*, **15**(4), 34:1–34:47, (2014).
- [32] N. Nisan, J. Bayer, D. Chandra, Tal Franji, R. Gardner, Y. Matias, N. Rhodes, M. Seltzer, D. Tom, Hal Varian, and D. Zigmund, ‘Google’s auction for TV ads’, in *Automata, Languages and Programming, 36th International Colloquium (ICALP09), Proceedings*, pp. 309–327. Springer, (2009).
- [33] W. van der Hoek, A. Lomuscio, and M. Wooldridge, ‘On the complexity of practical ATL model checking knowledge, strategies, and games in multi-agent systems’, in *Proceedings of the 5th international joint conference on Autonomous agents and multiagent systems (AAMAS06)*, pp. 201–208. ACM Press, (2006).
- [34] H. Xu, C. K. Bates, and S. M. Shatz, ‘Real-time model checking for skill detection in live online auctions’, in *Software Engineering Research and Practice*, pp. 134–140, (2009).
- [35] H. Xu and Y. Cheng, ‘Model checking bidding behaviors in internet concurrent auctions.’, *Computer System Science & Engineering*, **22**(4), (2007).

Translation-Based Revision and Merging for Minimal Horn Reasoning

Gerhard Brewka¹ and Jean-Guy Mailly² and Stefan Woltran²

Abstract. In this paper we introduce a new approach for revising and merging consistent Horn formulae under minimal model semantics. Our approach is translation-based in the following sense: we generate a propositional encoding capturing both the syntax of the original Horn formulae (the clauses which appear or not in them) and their semantics (their minimal models). We can then use any classical revision or merging operator to perform belief change on the encoding. The resulting propositional theory is then translated back into a Horn formula. We identify some specific operators which guarantee a particular kind of minimal change. A unique feature of our approach is that it allows us to control whether minimality of change primarily relates to the syntax or to the minimal model semantics of the Horn formula. We give an axiomatic characterization of minimal change on the minimal model for this new setting, and we show that some specific translation-based revision and merging operators satisfy our postulates.

1 Introduction

Belief revision is a highly active area of research in the field of knowledge representation and reasoning. Whereas the initial focus was on revising propositional belief sets, as in the famous AGM theory [1], respectively propositional knowledge bases, as in the KM theory [20], later on also revision operators for various more expressive logics have been studied. Examples are description logics, e.g. [24], modal and multi-valued logics [16], and others. Interestingly, in recent years there has as well been a steadily growing interest in the opposite direction, that is, in fragments of propositional logic which are less expressive but interesting for some specific reason.

Belief change operators for fragments of propositional logic have been thoroughly studied in [9, 7, 8]. Horn formulae, *i.e.* conjunctions (or equivalently sets) of clauses which contain at most one positive literal [10, 19], play a special role in this context. Horn formulae are particularly interesting for computational reasons, as they allow for linear inference methods.

Revising Horn formulae is also the topic of this paper. However, contrary to the existing work cited above which considers classical reasoning based on *all* models of a Horn formula, we are interested here in reasoning under the *minimal model semantics*, or minimal reasoning, for short. This form of reasoning is based on the assumption that an atom should be considered *false* whenever it is not provably *true*. It is well-known that consistent Horn formulae have a unique minimal (and thus least) model. Since new information may

modify the least model in arbitrary ways, minimal reasoning is non-monotonic, and the revision operators we are looking for are actually operators for a nonmonotonic formalism.

Although the relationship between nonmonotonic reasoning and revision was already investigated by Gärdenfors [18], there is relatively little work on revising specific nonmonotonic formalisms. Notable exceptions are revision of default logic [3], revision of logic programs under answer set semantics [2, 15, 12, 11, 25, 26] and, rather recently, revision of argumentation frameworks [5, 4, 13].

The revision of logic programs is of special interest here. Since sets of Horn clauses under minimal model semantics are a special case of logic programs under stable semantics, one might say the problem is already solved. However, the work presented here is very different, as we will see. We will analyze these issues in detail in the discussion section.

The approach we are going to introduce in this paper is translation-based. The idea underlying translation-based revision has been pioneered in [17]. Basically, a revision operator for a formula F in a logic L is defined by representing F , possibly together with relevant meta-information, in a logic L^* for which a revision operator already exists. The encoding of F is then revised in L^* , and the result of this revision is translated back into logic L . Mailly [23] has shown how to apply this form of revision to a nonmonotonic formalism, namely to Dung-style argumentation frameworks [14].

In a nutshell, the goal of this paper is to demonstrate that translation-based revision can be also used for other, less simple nonmonotonic formalisms, in our case Horn formulae under minimal model semantics. In addition, we will also show how to use the translation-based approach for merging, that is, the process of integrating several Horn formulae into a single one.

More specifically, we encode the relation between the syntax of a Horn formula (its set of Horn clauses) and its semantics (its minimal model) in propositional logic. Revision or merging are then performed on this logical encoding with classical operators [20, 21], followed by a decoding step which gives the result of the Horn revision or merging. Our approach permits to revise a consistent Horn formula by an expressive piece of information: our revision formulae concern the set of clauses and the minimal model *at the same time*. Similarly, we use such an expressive logical language to express integrity constraints in the merging process. In contrast to other works, we thus do *not* restrict the revision formulae to the Horn fragment.

Let us schematically explain our approach to revision using Figure 1. Here, φ is a Horn formula, and $enc(\varphi)$ is the encoding of φ . We want to define a revision operator op for minimal model reasoning. We define our new revision operator op through three steps: encoding φ in propositional logic, resulting in $enc(\varphi)$, revising the encoding using an existing propositional operator op' , and finally de-

¹ Computer Science Institute, Leipzig University, Germany, brewka@informatik.uni-leipzig.de

² Institute of Information Systems, TU Wien, Austria, {jmailly,woltran}@dbai.tuwien.ac.at

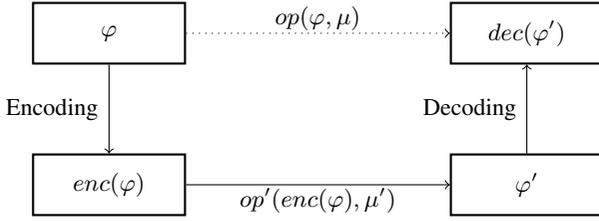


Figure 1. Schematic view of translation-based revision

coding the result of this revision back into a Horn formula. Note that the minimal Horn revision with μ requires revision with a substantially stronger formula μ' at the level of the encoding.

The schema for merging is similar to the one for revision. However, rather than a single Horn formula φ , a collection of such formulae (usually called a profile in the context of merging) is the starting point of the process, formulae μ and μ' can be seen as constraints for IC-merging [21] or are obsolete in simpler forms of merging.

The main contributions of this paper are the following. We introduce a propositional encoding of a consistent Horn formula, or equivalently set of clauses, φ such that the minimal model of φ is contained in the single model of the encoding. We use this encoding to define translation-based revision and merging of Horn formulae under minimal model semantics. For both operations,

- we define concrete translation-based operators which exhibit different minimal change properties;
- we adapt rationality postulates from the classical setting to express minimal change on the minimal model;
- we prove that some specific translation-based operators satisfy these postulates.

The remainder of the paper is organized as follows. Sect. 2 recalls the relevant background the paper builds upon. Sect. 3 provides the encoding of Horn formulae under minimal model semantics used throughout the paper. Sect. 4 and Sect. 5 introduce and analyze the translation-based approach to revision and merging, respectively. Sect. 6 shows how to limit the size of the encoding – which is exponential at the conceptual level – in practical settings. Sect. 7 discusses related work and concludes the paper with an outlook on future work. Some of the proofs are given in an appendix.

2 Background Notions

Let us first recall the basic notions of propositional logic and introduce some notations. We consider a set of Boolean variables V . We use \mathcal{L} to denote the set of all propositional formulae built on V with usual connectives (\neg, \vee, \wedge). Satisfiability of formulae is defined as usual, and $\text{mod}(\varphi)$ gives the set of models of a formula φ . Interpretations and models are represented by sets containing those variables which are assigned true. Each variable $x \in V$ is associated with a positive literal x and a negative literal $\neg x$. A clause is a disjunction of literals $l_1 \vee l_2 \vee \dots \vee l_n$. A Horn clause is a clause which contains at most one positive literal. $\text{hcl}(V)$ is the set of all Horn clauses built on V . A Horn formula φ is a conjunction of Horn clauses, or equivalently a set of Horn clauses, denoted $\text{hcl}(\varphi)$. $\mathcal{L}_H(V)$ denotes the set of all Horn formulae built on V , $\mathcal{L}_H^{\text{cons}}(V)$ the subset of consistent formulae in $\mathcal{L}_H(V)$. Given the set of models $\text{mod}(\varphi)$ of a formula $\varphi \in \mathcal{L}_H^{\text{cons}}(V)$, the minimal model of φ , denoted $\text{mod}_{\min}(\varphi)$,

is the unique \subseteq -minimal element of $\text{mod}(\varphi)$. $\text{mod}_{\min}(\varphi)$ models skepticism regarding positive atomic information, since each variable which is assigned *true* in this model is also assigned *true* in each other model of φ .

Given two sets S_1, S_2 , the Hamming distance between them is $d_H = |(S_1 \setminus S_2) \cup (S_2 \setminus S_1)|$. When S_1 and S_2 are propositional interpretations, the Hamming distance counts the number of Boolean variables which are assigned different values in these interpretations. Given a set of Boolean variables S , d_H^S is the Hamming distance between interpretations restricted to S .

Belief revision incorporates a new piece of information in an agent's beliefs. One of the most prominent characterizations of belief revision considers the beliefs and the new piece of information as formulae from propositional logic [20]. An axiomatic characterization is given by a set of postulates which express logical relations between formulae. We give here a reformulation of these postulates as set-theoretical relations between the sets of models of the formulae. A KM revision operator \circ is a mapping from $\mathcal{L} \times \mathcal{L}$ to \mathcal{L} which satisfies the postulates:

- (R1) $\text{mod}(\varphi \circ \mu) \subseteq \text{mod}(\mu)$.
- (R2) If $\text{mod}(\varphi) \cap \text{mod}(\mu) \neq \emptyset$, then $\text{mod}(\varphi \circ \mu) = \text{mod}(\varphi) \cap \text{mod}(\mu)$.
- (R3) If $\text{mod}(\mu) \neq \emptyset$, then $\text{mod}(\varphi \circ \mu) \neq \emptyset$.
- (R4) If $\text{mod}(\varphi_1) = \text{mod}(\varphi_2)$ and $\text{mod}(\mu_1) = \text{mod}(\mu_2)$, then $\text{mod}(\varphi_1 \circ \mu_1) = \text{mod}(\varphi_2 \circ \mu_2)$.
- (R5) $\text{mod}(\varphi \circ \mu_1) \cap \text{mod}(\mu_2) \subseteq \text{mod}(\varphi \circ (\mu_1 \wedge \mu_2))$.
- (R6) If $\text{mod}(\varphi \circ \mu_1) \cap \text{mod}(\mu_2) \neq \emptyset$, then $\text{mod}(\varphi \circ (\mu_1 \wedge \mu_2)) \subseteq \text{mod}(\varphi \circ \mu_1) \cap \text{mod}(\mu_2)$.

These postulates express logical constraints which must be satisfied by the models of the revised formula $\varphi \circ \mu$, depending on the models of the initial formula φ and the revision formula μ . A representation theorem associates this axiomatic characterization with a constructive one: a revision operator \circ satisfies the postulates iff it can be expressed as

$$\text{mod}(\varphi \circ \alpha) = \min(\text{mod}(\alpha), \leq_{\varphi}) \quad (1)$$

where \leq_{φ} is a total pre-order expressing the relative plausibility of interpretations; this pre-order has to satisfy some conditions. We only exhibit a specific family of pre-orders based on distances. Given a distance d , we overload the notation and define $d(\omega, \varphi) = \min_{\omega' \in \text{mod}(\varphi)} d(\omega, \omega')$. The pre-order \leq_{φ}^d is then defined by $\omega_1 \leq_{\varphi}^d \omega_2$ iff $d(\omega_1, \varphi) \leq d(\omega_2, \varphi)$. Instantiating equation (1) with $\leq_{\varphi} = \leq_{\varphi}^d$ defines a revision operator which satisfies all rationality postulates.

Belief merging operations obtain the beliefs of a group from the beliefs of each member of the group. In particular, belief merging with integrity constraints (IC-merging) is a generalization of belief revision [21]. In this scenario, we want to merge a tuple of formulae $E = \langle \varphi_1, \dots, \varphi_n \rangle$ called a *profile*, where each formula represents an agent's belief. It is expected that the result of the merging satisfies an integrity constraint μ . The result of such an operation is denoted $\Delta_{\mu}(E)$. Revision and IC-merging are strongly connected; indeed when IC-merging is performed on a single formula, the result yields a revision operation, i.e. there is a KM revision operator \circ such that $\Delta_{\mu}(\langle \varphi \rangle) \equiv \varphi \circ \mu$. Rationality postulates (which generalize the postulates for revision from above) and a representation theorem have also been stated for IC-merging. Similarly to revision operators, an IC-merging operator can be defined thanks to a total pre-order which represents the relative plausibility of interpretations: $\text{mod}(\Delta_{\mu}(E)) = \min(\text{mod}(\mu), \leq_E)$. The pre-order should satisfy

additional properties. We focus here on the method to define a pre-order from a distance. We need first the notion of aggregation functions. An *aggregation function* is a function \otimes which associates a non-negative real number to every finite tuple of non-negative numbers, and which satisfies:

- if $y \leq z$ then $\otimes(x_1, \dots, y, \dots, x_n) \leq \otimes(x_1, \dots, z, \dots, x_n)$;
- $\otimes(x_1, \dots, x_n) = 0$ iff $x_1 = \dots = x_n = 0$;
- $\forall x \in \mathbb{R}^+, \otimes(x) = x$.

For any distance d and any profile $E = \langle \varphi_1, \dots, \varphi_n \rangle$, we overload the notation of distances: $d(\omega, E) = \otimes(d(\omega_1, \varphi_1), \dots, d(\omega_n, \varphi_n))$. Now, given a distance d and an aggregation function \otimes , the pre-order $\leq_E^{d, \otimes}$ is defined by $\omega_1 \leq_E^{d, \otimes} \omega_2$ iff $d(\omega_1, E) \leq d(\omega_2, E)$.

3 Encoding Horn Formulae

The principle of our encoding is to define a propositional formula Ξ which establishes the links between the syntax of a consistent Horn formula, represented by the set of clauses which appear in it, and the semantics of the formula, as given by its minimal model. Then we need a way to encode the syntax of a Horn formula φ , such that the conjunction of this encoding with Ξ permits to deduce the minimal model of φ .

Let us introduce the propositional encoding of the syntax of a formula.

Definition 1. Let $\varphi \in \mathcal{L}_H^{cons}(V)$. For each Horn clause c built on V , a fresh variable cl_c is introduced with the intended meaning that the clause c appears in φ . Then the syntax of φ is encoded by

$$CL(\varphi) = \left(\bigwedge_{c \in \text{hcl}(\varphi)} cl_c \right) \wedge \left(\bigwedge_{c \in \text{hcl}(V) \setminus \text{hcl}(\varphi)} \neg cl_c \right)$$

Let us notice that for each variable $x \in V$, cl_x is true iff the unit clause x appears in the formula.

Example 1. We describe the encoding of the formula $\varphi = (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3) \wedge x_1$. Here, $\text{hcl}(\varphi) = \{x_1 \vee \neg x_2 \vee \neg x_3, \neg x_1 \vee x_2 \vee \neg x_3, x_1\}$.

$$CL(\varphi) = cl_{x_1 \vee \neg x_2 \vee \neg x_3} \wedge cl_{\neg x_1 \vee x_2 \vee \neg x_3} \wedge cl_{x_1} \\ \wedge \left(\bigwedge_{c \in \text{hcl}(V) \setminus \text{hcl}(\varphi)} \neg cl_c \right)$$

Now we need a formula which expresses the link between the syntax of a consistent Horn formula and its minimal model. Note that below we do ignore purely negative clauses since they do not influence the computation of the minimal model for consistent formulae.

Definition 2. We consider Horn formulae in $\mathcal{L}_H^{cons}(V)$. Let $|V| = n$. The encoding of Horn minimal model semantics is given by the formula $H(V)$ defined as follows:

$$\begin{aligned} H^{(x,0)} &= x^0 \Leftrightarrow cl_x \\ H^{(x,i)} &= x^i \Leftrightarrow (x^{i-1} \vee ded^{(x,i)}), \text{ for } 1 \leq i < n, \text{ where} \\ ded^{(x,i)} &= \bigvee_{c = \neg y_1 \vee \dots \vee \neg y_k \vee x} (cl_c \wedge y_1^{i-1} \wedge \dots \wedge y_k^{i-1}) \\ H^{(x,n)} &= x^n \Leftrightarrow x \\ H(V) &= \bigwedge_{x \in V, 0 \leq i \leq n} H^{(x,i)} \end{aligned}$$

This formula mimics the well-known linear-time marking algorithm for computing the minimal model of a Horn formula. The variables $\{x^i \mid x \in V, 0 \leq i \leq n\}$ are used to represent the state of the variable x at the i^{th} step of the algorithm. The algorithm is guaranteed to terminate after at most $n = |V|$ steps. For this reason it suffices to consider subformulae $H^{(x,j)}$ for $j \leq n$.

They are initialized by $H^{(x,0)}$, which states that the variables x is true at the beginning of the algorithm iff it appears as a unit clause in φ . Then, from $H^{(x,i)}$ we obtain that the variable x is true at the i^{th} step iff either it was true at the previous step, or it can be deduced from the Horn clauses which appear in φ and the variables which were true at the $i-1^{\text{th}}$ step. This part is represented by $ded^{(x,i)}$.

Finally, $H^{(x,n)}$ states that the variables which are true at the n^{th} step are those which are true in the minimal model of φ .

Example 2. Let $\varphi = (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_2 \vee \neg x_4) \wedge (x_3 \vee \neg x_4) \wedge x_4$, and consider an arbitrary model M of $CL(\varphi) \wedge H(V)$.

- $H^{(x,0)}$: for each $x \neq x_4$ we have $M(x^0) = 0$ since there is no unit clause x in φ (so cl_x is false);
- $H^{(x_4,0)}$: $M(x_4^0) = 1$ since there is a unit clause x_4 in φ (so cl_{x_4} is true);
- $H^{(x_2,1)}, H^{(x_3,1)}$: at step 1, two rules from $ded^{(x,1)}$ can be applied (since x_4 allows to trigger them). So, for each $x \in \{x_2, x_3\}$, $M(x^1) = 1$.
- $H^{(x_1,1)}, H^{(x_4,1)}$: x_4^1 receives the value 1 because $M(x_4^0) = 1$; x_1^1 receives the value 0 because no rule concerning it can be triggered.
- $H^{(x,2)}$: at step 2, since x_2 and x_3 are true from the previous step, a rule is triggered and $M(x_2^2) = 1$. For other variables $x \in \{x_2, x_3, x_4\}$, $M(x^2) = 1$ because $M(x^1) = 1$.
- The same scheme is repeated for each i : $M(x^i) = 1$ because $M(x^{i-1}) = 1$. Finally, from $H^{(x,n)}$, each variable x_1, \dots, x_4 receives the value 1.

$\text{mod}_{\min}(\varphi) = \{x_1, x_2, x_3, x_4\}$ can thus be deduced from $CL(\varphi) \wedge H(V)$.

We can show the following result:

Proposition 1. Let $\varphi \in \mathcal{L}_H^{cons}(V)$. The propositional formula $CL(\varphi) \wedge H(V)$ has a unique model M such that $M \cap V = \text{mod}_{\min}(\varphi)$.

The reader will have noticed that the size of our encoding is actually exponential in the number of propositional variables in V . We are fully aware that this is far from tolerable from a practical point of view. For the time being we will stick to this exponential encoding, as this makes it easier to introduce our approach at the conceptual level. However, we will discuss in Sect. 6 how to deal with this issue in practical settings.

4 Revising Horn Formulae

We focus here on belief revision in a situation where the relevant information is carried by the minimal model of formulae. We call this operation mod_{\min} -revision.

4.1 Translation-based Revision

Our approach benefits from the encoding presented in the previous section. Indeed, the formula $CL(\varphi) \wedge H(V)$ expresses information about the minimal model of φ , which is given by the value of the variables $V = \{x_1, \dots, x_p\}$. It is thus possible to revise $CL(\varphi) \wedge H(V)$ by a propositional formula μ built on V which expresses the new piece of information to incorporate in the minimal model of the agent's beliefs. On the other hand, $CL(\varphi) \wedge H(V)$ also expresses information about the structure of the formula φ , since $CL(\varphi)$ is built using the cl_x variables. So we can include

structural information as well and use revision formulae μ in \mathcal{L}' , the propositional language built on $V' = V \cup \{\text{cl}_c \mid c \in \text{hcl}(V)\}$.

One important issue needs to be addressed: performing the revision $(CL(\varphi) \wedge H(V)) \circ \mu$, with \circ any KM revision operator, does not guarantee to give a result which is compatible with the Horn minimal model semantics. As a simple example, if we revise a formula φ by $\mu = x_1 \wedge \neg x_2 \wedge \text{cl}_{\neg x_1 \vee x_2}$, then only x_1 should belong to the minimal model of the result, and the clause $\neg x_1 \vee x_2$ should appear in this result. This is obviously incompatible. For this reason, we define revision operators as follows.

Definition 3. Let \circ be an arbitrary KM revision operator, $\varphi \in \mathcal{L}_H^{\text{cons}}(V)$ and $\mu \in \mathcal{L}'$. The translation-based revision operator based on \circ , denoted \star_\circ , is a mapping from $\mathcal{L}_H^{\text{cons}}(V) \times \mathcal{L}'$ to $2^{\mathcal{L}_H^{\text{cons}}(V)}$, defined as:

$$\varphi \star_\circ \mu = \text{dec}((CL(\varphi) \wedge H(V)) \circ (\mu \wedge H(V)))$$

with dec the decoding of the clause variables cl_x , defined as follows. Let ω be a propositional interpretation built on $V' = V \cup \{\text{cl}_c \mid c \in \text{hcl}(V)\}$. Let Ω be a set of such interpretations. Let Φ be a propositional formula built on V' .

- $\text{dec}(\omega)$ is the Horn formula $\{c \in \text{hcl}(V) \mid \omega(\text{cl}_c) = 1\}$;
- $\text{dec}(\Omega) = \{\text{dec}(\omega) \mid \omega \in \Omega\}$;
- $\text{dec}(\Phi) = \text{dec}(\text{mod}(\Phi))$.

Since \circ satisfies the KM rationality postulates, the result of the revision is obviously inconsistent whenever μ is not consistent with $H(V)$, as intended.

Now we define different revision operators \star . In particular, we show that – depending on the underlying operator \circ – we can actually choose between minimizing change on the semantic level of a formula (its minimal model) or on the syntactic level (the set of clauses of a formula).

Definition 4. Given $W_1 > 0, W_2 > 0$, we define the (W_1, W_2) -weighted distance between interpretations $d_{(W_1, W_2)}$ as follows:

$$d_{(W_1, W_2)}(\omega_1, \omega_2) = W_1 \times d_H^V(\omega_1, \omega_2) + W_2 \times d_H^{\text{Syn}}(\omega_1, \omega_2)$$

with ω_1, ω_2 interpretations on the set of variables $V \cup \{\text{cl}_c \mid c \in \text{hcl}(V)\}$, and $\text{Syn} = \{\text{cl}_c \mid c \in \text{hcl}(V)\}$ the set of variables related to syntax of formulae.

Here we consider only the variables which correspond to the syntax (cl_c variables) and to the minimal model semantics (x variables). But our encoding also uses additional variables x^i . For this reason, we cannot directly define a distance-based revision operator from $d_{(W_1, W_2)}$ as explained in Section 2, since $d_{(W_1, W_2)}$ is not strictly speaking a distance between interpretations of our encoding. However, we prove that we can use $d_{(W_1, W_2)}$ to define a KM revision operator.

Proposition 2. The pre-order between interpretations corresponding to $d_{(W_1, W_2)}$ satisfies the properties of faithful assignments, and yields a KM revision operator denoted $\circ_{(W_1, W_2)}$.

This family of weighted distances is a generalization of the Hamming distance ($d_{(1,1)}$ yields Hamming distance). It is possible to assign particular weights to obtain some properties about minimal change; if the value of W_1 is high enough, then it is more expensive in the revision process to change the value of a x variable than to

change the values of all cl_c variables. Then the revision operator will ensure the minimal change on the x variables (which represent the minimal model of the Horn formula); minimal change of the syntax will be applied as a secondary criterion.

Definition 5. Let $W_1 = |\text{hcl}(V)| + 1$, and $W_2 = 1$. The semantic minimal change revision operator is the translation-based revision operator \star_{sem} based on the KM revision operator $\circ_{(W_1, W_2)}$.

Example 3. Given $V = \{x_1, x_2, x_3, x_4, x_5\}$, we revise $\varphi = (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3) \wedge x_1$ from Example 1 by $\mu = x_4 \vee x_5$. We can see that $\text{mod}_{\min}(\varphi) = \{x_1\}$. Minimal change of the minimal model, in this situation, leads to $\text{mod}_{\min}(\varphi \star \mu) = \{\{x_1, x_4\}, \{x_1, x_5\}\}$. For each of these possible minimal models there is a possible Horn formula which corresponds to it and which is minimal with respect to the secondary criterion: $\varphi_1 = \varphi \wedge x_4$, $\text{mod}_{\min}(\varphi_1) = \{x_1, x_4\}$; and $\varphi_2 = \varphi \wedge x_5$, $\text{mod}_{\min}(\varphi_2) = \{x_1, x_5\}$. For each of these formulae, there is only one new variable in the minimal model, and one new clause in the formula. So the result of the revision is $\varphi \star_{sem} \mu = \{\varphi_1, \varphi_2\}$.

In Example 3, all the solutions are optimal with respect to minimal change of minimal model and minimal change of the syntax, but it is not the case in general. If it is more expensive to change the value of a single cl_c variable than to change the values of all x variables, then minimal change of the syntax is the main minimality criterion, and minimal change of the minimal model is applied as a secondary criterion.

Definition 6. Let $W_1 = 1$, and $W_2 = |V| + 1$. The syntactic minimal change revision operator is the translation-based revision operator \star_{syn} based on the KM revision operator $\circ_{(W_1, W_2)}$.

Even when the formula μ concerns only one kind of information (the minimal model, or the syntax), both operators lead to a different result in general.

Example 4. We exemplify the difference between both kinds of minimal change. We consider the set of Boolean variables $V = \{x_1, x_2, x_3, x_4\}$ and the formula $\varphi = (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_2 \vee \neg x_4) \wedge (x_3 \vee \neg x_4) \wedge x_4$ from Example 2. Its minimal model is $\text{mod}_{\min}(\varphi) = \{x_1, x_2, x_3, x_4\}$. We want to revise it by the formula $\mu = \neg x_2$, which means that x_2 should not belong to the minimal model.

The encoding of φ is

$$CL(\varphi) = \text{cl}_{x_1 \vee \neg x_2 \vee \neg x_3} \wedge \text{cl}_{x_2 \vee \neg x_4} \wedge \text{cl}_{x_3 \vee \neg x_4} \wedge \text{cl}_{x_4} \\ \wedge \bigwedge_{c \in \text{hcl}(V) \setminus \text{hcl}(\varphi)} \neg \text{cl}_c$$

Horn minimal model semantics is encoded by $H(V) = \bigwedge_{x \in V, 0 \leq i \leq 4} H^{(x,i)}$.

We perform first a revision with the semantic minimal change operator \star_{sem} . Obviously, giving priority to minimal change on the minimal model leads to removing x_2 from it, which means that $\text{mod}_{\min}(\varphi \star_{sem} \mu) = \{x_1, x_3, x_4\}$. There are four possible Horn formulae, corresponding to this minimal model, which are minimal w.r.t. syntax change: $\varphi \star_{sem} \mu = \{(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_3 \vee \neg x_4) \wedge x_4 \wedge X \mid X \in \{x_1, x_1 \vee \neg x_4, x_1 \vee \neg x_3, x_1 \vee \neg x_3 \vee \neg x_4\}\}$. In this case, there is a single change in the minimal model, and two changes in the syntax of the formula (one clause is removed, one clause is added). Alternatively, if we first consider minimal change of the syntax, then the result is the set of formulae $\varphi \star_{syn} \mu = \{(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_3 \vee \neg x_4) \wedge x_4\}$, and $\text{mod}_{\min}(\varphi \star_{syn} \mu) = \{x_3, x_4\}$. Here, there is a single change in the syntax (the removal of one clause), and there are two changes in the minimal model (x_1 and x_2 are removed).

4.2 Axiomatization of mod_{\min} -Revision

If the minimal model of a formula is the most important information for an agent, then she can revise her beliefs by a formula μ which expresses what the new minimal model of her beliefs should be. The result of the revision is then a set of Horn formulae³, the minimal model of each of them being a model of μ . For a set of formulae Φ , we use $\text{mod}_{\min}(\Phi)$ as a notation for $\{\text{mod}_{\min}(\varphi) \mid \varphi \in \Phi\}$. This kind of revision should satisfy the following postulates:

- (★1) $\text{mod}_{\min}(\varphi \star \mu) \subseteq \text{mod}(\mu)$.
- (★2) If $\{\text{mod}_{\min}(\varphi)\} \cap \text{mod}(\mu) \neq \emptyset$, then $\text{mod}_{\min}(\varphi \star \mu) = \{\text{mod}_{\min}(\varphi)\} \cap \text{mod}(\mu)$.
- (★3) If $\text{mod}(\mu) \neq \emptyset$, then $\text{mod}_{\min}(\varphi \star \mu) \neq \emptyset$.
- (★4) If $\text{mod}_{\min}(\varphi_1) = \text{mod}_{\min}(\varphi_2)$ and $\text{mod}(\mu_1) = \text{mod}(\mu_2)$, then $\text{mod}_{\min}(\varphi_1 \star \mu_1) = \text{mod}_{\min}(\varphi_2 \star \mu_2)$.
- (★5) $\text{mod}_{\min}(\varphi \star \mu_1) \cap \text{mod}(\mu_2) \subseteq \text{mod}_{\min}(\varphi \star (\mu_1 \wedge \mu_2))$.
- (★6) If $\text{mod}_{\min}(\varphi \star \mu_1) \cap \text{mod}(\mu_2) \neq \emptyset$, then $\text{mod}_{\min}(\varphi \star (\mu_1 \wedge \mu_2)) \subseteq \text{mod}_{\min}(\varphi \star \mu_1) \cap \text{mod}(\mu_2)$.

The notion of faithful assignments in classical revision aims at sorting interpretations to ensure that the models of the agent's beliefs (which are the main information of the agent) are the minimal elements of a pre-order. In our case, we can relax this assumption, as soon as the minimal model of the formula is the "preferred" interpretation.

Definition 7. A min-faithful assignment is a mapping from a Horn formula φ to a total pre-order between interpretations \leq_{φ} such that:

1. $\forall \omega \neq \text{mod}_{\min}(\varphi), \text{mod}_{\min}(\varphi) <_{\varphi} \omega$;
2. if $\text{mod}_{\min}(\varphi_1) = \text{mod}_{\min}(\varphi_2)$, then $\leq_{\varphi_1} = \leq_{\varphi_2}$.

Theorem 3. Given a min-faithful assignment which maps each Horn formula φ to a total pre-order \leq_{φ} , if the revision operator \star satisfies $\text{mod}_{\min}(\varphi \star \mu) = \min(\text{mod}(\mu), \leq_{\varphi})$ then \star satisfies the postulates (★1)–(★6).

We notice that contrary to the classical theorem from Katsuno and Mendelzon, this result does not lead directly to the result for revision. Indeed, while the classical faithful assignment theorem characterizes the set of models of a formula (which is then unique, modulo logical equivalence), here we only know that the result of the revision should be a set of Horn formulae such that the minimal model of each of them belongs to $\min(\text{mod}(\mu), \leq_{\varphi})$. In general, for each minimal model $\omega \in \min(\text{mod}(\mu), \leq_{\varphi})$, there is not a single Horn formula which corresponds to ω . So, for a given min-faithful assignment, there may be several options to define the result of the revision.

We now focus on a particular family of revision operators, based on distances between interpretations.

Definition 8. Let d be a distance between interpretations. For each Horn formula φ , we define the total pre-order between interpretations \leq_{φ} by $\omega_1 \leq_{\varphi}^d \omega_2$ iff $d(\omega_1, \text{mod}_{\min}(\varphi)) \leq d(\omega_2, \text{mod}_{\min}(\varphi))$. The distance-based revision operator \star_d is defined by

$$\text{mod}_{\min}(\varphi \star_d \mu) = \min(\text{mod}(\mu), \leq_{\varphi}^d)$$

Proposition 4. Every distance-based revision operator satisfies the postulates (★1)–(★6).

³ If a specific application requires us to have a single Horn formula, a tie-break rule can be used on the result of the revision.

To illustrate distance-based revision, we look at a specific operator from this family.

Example 5. Let \star_H the distance based revision operator defined from the Hamming distance. We consider the Horn formula φ from Example 3 built on the set of variables $V = \{x_1, x_2, x_3, x_4, x_5\}$. Recall that $\text{mod}_{\min}(\varphi) = \{x_1\}$.

Now we revise φ by $\mu = x_4 \vee x_5$, which expresses that the minimal model of the agent's beliefs should contain at least one of the variables x_4, x_5 . The models of μ which are minimal w.r.t. the pre-order associated with φ and the Hamming distance are $M = \{\{x_1, x_4\}, \{x_1, x_5\}\}$. So the result of the revision should be a set of Horn formulae such that their minimal model belongs to M .

A possible solution to obtain the result of the revision is to consider the set of Horn formulae $R = \{\psi \mid \text{mod}_{\min}(\psi) \in \min(\text{mod}(\mu), \leq_{\varphi})\}$. This set represents all the possible alternatives for the agent's revised beliefs. Let us notice that this result can be refined thanks to the translation-based operators defined in Section 4.1.

Proposition 5. The semantic minimal change revision operator \star_{sem} of Definition 5 satisfies the postulates (★1)–(★6), provided the revision formula μ is built on V .

This translation-based revision operator satisfies the postulates, and is a concrete method to obtain the result of the revision (the set of revised Horn formulae), while the distance-based revision given in Definition 8 only gives the revised minimal models, but not the actual formulae corresponding to these models.

5 Merging Horn Formulae

Now we turn our attention to merging operators. Similarly to what we have proposed for revision, we propose operators such that not all models of formulae are considered in the merging process, but only their minimal model. We call this operation mod_{\min} -merging. To avoid confusion with classical IC-merging operators, we use Θ to denote mod_{\min} -merging operators. We focus on Horn profiles, meaning $\langle \varphi_1, \dots, \varphi_n \rangle$ such that $\varphi_1, \dots, \varphi_n \in \mathcal{L}_H^{cons}(V)$.

5.1 Translation-based Merging

We can also benefit from the logical encoding of syntax and mod_{\min} -semantics of Horn formulae to define merging operators. The idea is the same as for translation-based revision: we propose to translate all Horn formulae in the profile into propositional formulae dealing with the syntax and the minimal model, and to merge them with a classical IC-merging operator. The formula μ gives a constraint on the set of clauses and on the minimal model.

We first define the encoding of a Horn profile.

Definition 9. Given a Horn profile $E = \langle \varphi_1, \dots, \varphi_m \rangle$, the encoding of E is $H(E) = \langle CL(\varphi_1) \wedge H(V), \dots, CL(\varphi_m) \wedge H(V) \rangle$.

Definition 10. Given Δ any IC-merging operator, the translation-based merging operator based on Δ is Θ^{Δ} , a mapping from a Horn profile E and a formula $\mu \in \mathcal{L}'$ to a set of consistent Horn formulae, such that $\Theta_{\mu}^{\Delta}(E) = \text{dec}(\Delta_{(\mu \wedge H(V))}(H(E)))$ with dec as given in Definition 3.

We use the well-known distance-based IC-merging operators to define mod_{\min} -merging operators. To exhibit specific operators, we use the sum as aggregation function.

Definition 11. Given the (W_1, W_2) -weighted distance between interpretations $d_{(W_1, W_2)}$, the (W_1, W_2) -weighted IC-merging operator $\Delta^{(W_1, W_2), \Sigma}$ is the distance based IC-merging operator defined from the distance $d_{(W_1, W_2)}$ and the sum aggregation function.

Now we define translation-based merging operators which are generalizations of the revision operators \star_{sem} and \star_{syn} .

Definition 12. Let $W_1 = |\text{hcl}(V)| + 1$, and $W_2 = 1$. The semantic minimal change merging operator is the translation-based merging operator Θ^{sem} based on the IC-merging operator $\Delta^{(W_1, W_2), \Sigma}$.

Definition 13. Let $W_1 = 1$, and $W_2 = |V| + 1$. The syntactic minimal change merging operator is the translation-based merging operator Θ^{syn} based on the IC-merging operator $\Delta^{(W_1, W_2), \Sigma}$.

Example 6. We consider $V = \{x_1, x_2, x_3, x_4\}$, and $E = \langle \varphi_1, \varphi_2 \rangle$, with $\varphi_1 = (x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_1 \vee \neg x_4) \wedge x_4$, and $\varphi_2 = (x_1 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3) \wedge x_3$. We have $\text{mod}_{\min}(\varphi_1) = \{x_1, x_4\}$ and $\text{mod}_{\min}(\varphi_2) = \{x_1, x_3\}$.

We will compare the behaviour of Θ^{sem} and Θ^{syn} when merging E . Both operators require to determine $H(E) = \langle CL(\varphi_1) \wedge H(V), CL(\varphi_2) \wedge H(V) \rangle$. The translation of φ_1 and φ_2 are formulae with a single model. We use ω_i as a shortcut for $\text{mod}(CL(\varphi_i) \wedge H(V))$, respectively:

$$\begin{aligned} \omega_1 &= \{x_1, x_4, \text{cl}_{x_1 \vee \neg x_2 \vee \neg x_3}, \text{cl}_{x_1 \vee \neg x_4}, \text{cl}_{x_4}\} \\ \omega_2 &= \{x_1, x_3, \text{cl}_{x_1 \vee \neg x_3}, \text{cl}_{\neg x_1 \vee \neg x_2 \vee \neg x_3}, \text{cl}_{x_3}\} \end{aligned}$$

We consider a constraint μ such that $\text{mod}(\mu \wedge H(V)) = \{\omega'_1, \omega'_2\}$ as follows:

$$\begin{aligned} \omega'_1 &= \{x_1, x_3, x_4, \text{cl}_{x_1}, \text{cl}_{x_3}, \text{cl}_{x_4}\} \\ \omega'_2 &= \{\text{cl}_{x_1 \vee \neg x_2 \vee \neg x_3}, \text{cl}_{x_1 \vee \neg x_4}, \text{cl}_{x_1 \vee \neg x_3}, \text{cl}_{\neg x_1 \vee \neg x_2 \vee \neg x_3}\} \end{aligned}$$

The result of the merging, at the level of the propositional translation, is a subset of models of $\mu \wedge H(V)$, which are minimal w.r.t. the distance-based pre-orders corresponding to $\Delta^{(|\text{hcl}(V)|+1, 1), \Sigma}$ (for Θ^{sem}) and $\Delta^{(1, |V|+1), \Sigma}$ (for Θ^{syn}). Table 1 represents the distances between models of $\mu \wedge H(V)$ and the models of the Horn formulae from E , and finally the sum aggregation, expressed with the weights (W_1, W_2) which underly the distances.

$\text{mod}(\mu \wedge H(V))$	ω_1	ω_2	Σ
ω'_1	$W_1 + 4W_2$	$W_1 + 4W_2$	$2W_1 + 8W_2$
ω'_2	$2W_1 + 3W_2$	$2W_1 + 3W_2$	$4W_1 + 6W_2$

Table 1. Distances between interpretation w.r.t. weights (W_1, W_2)

To obtain the result of merging for Θ^{sem} and Θ^{syn} , we need to instantiate the weights in Table 1 with the values corresponding to the actual distances (see Definition 12 and Definition 13). We observe that Θ^{sem} (resp. Θ^{syn}) selects as minimal model ω'_1 (resp. ω'_2). So, $\Theta^{sem}(E) = \{x_1 \wedge x_3 \wedge x_4\}$ and $\Theta^{syn} = \{(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_1 \vee \neg x_4) \wedge (x_1 \vee \neg x_3) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)\}$.

5.2 Axiomatization of mod_{\min} -Merging

Before stating the mod_{\min} adaptation of IC-merging postulates, let us introduce some notations. For any E , $\bigwedge \text{mod}_{\min}(E) = \{\omega \mid \forall \varphi \in E, \omega = \text{mod}_{\min}(\varphi)\}$, which means that $\bigwedge \text{mod}_{\min}(E)$ is

empty if all formulae in E do not have the same minimal model. We define the union of profiles E_1 and E_2 as their concatenation. Finally, two Horn profiles E_1 and E_2 are called equivalent, denoted $E_1 \equiv E_2$, if there is a bijective function f from E_1 to E_2 such that $\forall \varphi \in E_1, \text{mod}_{\min}(f(\varphi)) = \text{mod}_{\min}(\varphi)$. Now, a mod_{\min} -merging operator is a mapping from a Horn profile E and a formula μ to a Horn formula $\Theta_\mu(E)$ which satisfies

- (Θ0) $\text{mod}_{\min}(\Theta_\mu(E)) \subseteq \text{mod}(\mu)$.
- (Θ1) If $\text{mod}(\mu) \neq \emptyset$, then $\text{mod}_{\min}(\Theta_\mu(E)) \neq \emptyset$.
- (Θ2) If $\bigwedge \text{mod}_{\min}(E) \cap \text{mod}(\mu) \neq \emptyset$, then $\text{mod}_{\min}(\Theta_\mu(E)) = \bigwedge \text{mod}_{\min}(E) \cap \text{mod}(\mu)$.
- (Θ3) If $E_1 \equiv E_2$ and $\text{mod}(\mu_1) = \text{mod}(\mu_2)$, then $\Theta_{\mu_1}(E_1) = \Theta_{\mu_2}(E_2)$.
- (Θ4) If $\{\text{mod}_{\min}(\varphi_1)\} \subseteq \text{mod}(\mu)$ and $\{\text{mod}_{\min}(\varphi_2)\} \subseteq \text{mod}(\mu)$, then $\text{mod}_{\min}(\Theta_\mu(\langle \varphi_1, \varphi_2 \rangle)) \cap \{\text{mod}_{\min}(\varphi_1)\} \neq \emptyset$ iff $\text{mod}_{\min}(\Theta_\mu(\langle \varphi_1, \varphi_2 \rangle)) \cap \{\text{mod}_{\min}(\varphi_2)\} \neq \emptyset$.
- (Θ5) $\text{mod}_{\min}(\Theta_\mu(E_1)) \cap \text{mod}_{\min}(\Theta_\mu(E_2)) \subseteq \text{mod}_{\min}(\Theta_\mu(E_1 \cup E_2))$.
- (Θ6) If $\text{mod}_{\min}(\Theta_\mu(E_1)) \cap \text{mod}_{\min}(\Theta_\mu(E_2)) \neq \emptyset$, then $\text{mod}_{\min}(\Theta_\mu(E_1 \cup E_2)) \subseteq \text{mod}_{\min}(\Theta_\mu(E_1)) \cap \text{mod}_{\min}(\Theta_\mu(E_2))$.
- (Θ7) $\text{mod}_{\min}(\Theta_{\mu_1}(E)) \cap \text{mod}(\mu_2) \subseteq \text{mod}_{\min}(\Theta_{\mu_1 \wedge \mu_2}(E))$.
- (Θ8) If $\text{mod}_{\min}(\Theta_{\mu_1}(E)) \cap \text{mod}(\mu_2) \neq \emptyset$, then $\text{mod}_{\min}(\Theta_{\mu_1 \wedge \mu_2}(E)) \subseteq \text{mod}_{\min}(\Theta_{\mu_1}(E)) \cap \text{mod}(\mu_2)$.

Similarly to the case of classical propositional belief merging, we can extend the notion of min-faithful assignment, to define pre-orders suitable to define merging operators.

Definition 14. A min-syncretic assignment is a mapping from a Horn profile E to a total pre-order between interpretations \leq_φ such that:

1. $\forall \omega_1 \notin \bigwedge \text{mod}_{\min}(E)$, if $\omega_2 \in \bigwedge \text{mod}_{\min}(E)$ then $\omega_2 <_E \omega_1$;
2. if $E_1 \equiv E_2$, then $\leq_{E_1} = \leq_{E_2}$;
3. if $\omega_1 = \text{mod}_{\min}(\varphi_1)$ and $\omega_2 = \text{mod}_{\min}(\varphi_2)$, then $\omega_1 \simeq_{\langle \varphi_1, \varphi_2 \rangle} \omega_2$;
4. If $\omega_1 \leq_{E_1} \omega_2$ and $\omega_1 \leq_{E_2} \omega_2$, then $\omega_1 \leq_{E_1 \cup E_2} \omega_2$;
5. If $\omega_1 <_{E_1} \omega_2$ and $\omega_1 \leq_{E_2} \omega_2$, then $\omega_1 <_{E_1 \cup E_2} \omega_2$.

Theorem 6. Given a min-syncretic assignment which maps each Horn profile E to a total pre-order between interpretations \leq_E , if the merging operator Θ satisfies

$$\text{mod}_{\min}(\Theta_\mu(E)) = \min(\text{mod}(\mu), \leq_E)$$

then Θ satisfies the postulates (Θ0)–(Θ8).

Definition 15. Let d be a distance between interpretations. For each Horn profile $E = \langle \varphi_1, \dots, \varphi_n \rangle$, we define a total pre-order between interpretations \leq_E as follows:

$$\omega_1 \leq_E^d \omega_2 \text{ iff } \sum_{\varphi \in E} d(\omega_1, \text{mod}_{\min}(\varphi)) \leq \sum_{\varphi \in E} d(\omega_2, \text{mod}_{\min}(\varphi))$$

The distance-based merging operator Θ^d is defined by

$$\text{mod}_{\min}(\Theta_\mu^d(E)) = \min(\text{mod}(\mu), \leq_E^d)$$

Proposition 7. Every distance-based merging operator satisfies the postulates (Θ0)–(Θ8).

Proposition 8. The semantic minimal change merging operator Θ^{sem} of Definition 12 satisfies the postulates (Θ0)–(Θ8), provided the integrity constraint μ is built on V .

6 Reducing the Size of the Encoding

The encoding we propose to express the syntax and semantics of a Horn formula φ is exponentially larger than φ . This stems from the fact that we have to specify, for each possible Horn clause $c \in \text{hcl}(V)$, whether c appears in φ or not. Similarly, we have to specify how c influences the computation of the minimal model. So both $CL(\varphi)$ and $H(V)$ exhibit an exponential blowup. This is the price we have to pay, at least at the conceptual level, for the expressiveness of our approach.

However, for practical applications, we can reduce the size of the encoding. Indeed, it seems reasonable to consider that some clauses are not relevant for the agent (resp. the group of agents) which revises (resp. merges) beliefs, and should not be involved in the revision (resp. merging). For instance, it may be reasonable to assume that only clauses up to a restricted size s , where s is some not too large constant, are relevant. For this reason, $CL(\varphi)$ and $H(V)$ can be adapted to take into account a specific pool of available clauses.

Definition 16. Given $\varphi \in \mathcal{L}_H^{\text{cons}}(V)$ and P a set of Horn clauses such that $\text{hcl}(\varphi) \subseteq P$, we define

$$CL_P(\varphi) = \left(\bigwedge_{c \in \text{hcl}(\varphi)} \text{cl}_c \right) \wedge \left(\bigwedge_{c \in P \setminus \text{hcl}(\varphi)} \neg \text{cl}_c \right)$$

Similarly, we define $H_P(V)$ following Definition 2, but re-writing the deduction part to take into account the available clauses:

$$\text{ded}_P^{(x,i)} = \bigvee_{c \in P, c = \neg y_1 \vee \dots \vee \neg y_k \vee x} (\text{cl}_c \wedge y_1^{i-1} \wedge \dots \wedge y_k^{i-1})$$

Given a translation-based revision (resp. merging) operator \star (resp. Θ), we use \star^P (resp. Θ^P) for the corresponding P -based operator.

The minimal pool of clauses P which can be used must contain at least each possible unit clause (to be able to obtain every minimal model), and each clause which is directly involved in the revision (resp. merging) process. These are the clauses c such that cl_c appears explicitly in μ , and the clauses $c \in \text{hcl}(\varphi)$ (resp. $c \in \text{hcl}(\varphi_i)$, for some $\varphi_i \in E$). This minimal pool is polynomially larger than the initial formula (resp. profile). Let us still notice that it is useful to keep a larger pool of clauses in some situations. Especially, with the minimal change of syntax criterion, revision (resp. merging) can lead to a different result if more clauses are available.

Example 7. Let $V = \{x_1, x_2, x_3\}$, and $\varphi = x_1$. We want to revise φ by $\mu = x_3 \wedge \neg \text{cl}_{x_3}$, which means that x_3 must appear in the minimal model, but must not appear as a unit clause. We use the revision operator \star_{syn} . If the revision uses the pool $P = \{\text{cl}_{x_1}, \text{cl}_{x_2}, \text{cl}_{x_3}, \text{cl}_{\neg x_1 \vee x_2}, \text{cl}_{\neg x_2 \vee x_3}\}$, the result of the revision is $\varphi \star_{\text{syn}}^P \mu = \{x_1 \wedge (\neg x_1 \vee x_2) \wedge (\neg x_2 \vee x_3)\}$. We observe that two clauses have been added to the initial formula φ . Now, if we consider the pool $P' = P \cup \{\neg x_1 \vee x_3\}$, we notice that it is possible to obtain a better result (w.r.t. minimal change of the syntax): $\varphi \star^{P'} \mu = \{x_1 \wedge (\neg x_1 \vee x_3)\}$.

Of course, all occurrences of $\text{hcl}(V)$ must be replaced by P to define P -based operators (e.g. in Definition 4 to define $d_{(W_1, W_2)}$ on the adequate set of variables).

7 Discussion

Related Work As far as we know, this is the first work which considers belief revision or merging in a situation where the relevant information is the minimal model of a formula. In particular, previous

works on Horn revision and merging consider the whole set of classical models. These approaches are classified along two lines. First, [7, 8] proposes to refine existing revision operators to guarantee that the result belongs to a given fragment. These refined operators coincide with the original ones in the case when the result of the original operator belongs to the target fragment. The second research direction is a modification of the revision and merging postulates [10, 19] and the definition of new operators corresponding to these postulates to ensure that the result belongs to the Horn fragment.

Also belief change for logic programs is related to our work, since sets of Horn clauses under minimal model semantics are a special case of logic programs under stable semantics. In fact, there is a huge body of work which addresses (usually a syntactic approach for) update, see e.g. [2, 15]. Hereby, dedicated properties different to AGM-style postulates have been proposed; hence, this research branch significantly differs from our approach. However, there is also work which is using the notion of SE-models [22, 27] as the objects revision [11, 25] or merging [12] operators are defined on. SE-models capture classical models of programs together with models of the respective program reducts. For the special case of Horn programs, reduct models and program models coincide. Thus, SE-models amount to classical models. Hence, for Horn programs these approaches are equivalent to the ones for Horn revision and merging based on classical models [10, 19]. Given the wide range of work in revision or update of logic programs, we are not aware of any approach that allows for simultaneous control (via the revision formula) whether minimality of change primarily relates to the syntax or to semantics of the program, in the way our method does.

The first work on translation-based revision is [17] which uses translations in classical logic to revise theories from other settings, like modal logic K , Łukasiewicz's finitely many-valued logic \mathcal{L}_n , algebraic logic and Belnap's four-valued logic. These translations allow to revise a theory by a formula from the same formalism, but do not provide any means to revise the information about the syntax of formulae. The work described in [6] is closer to our contribution. Here, argumentation frameworks are translated into propositional formulae which can be revised by a piece of information about the semantics (arguments statuses) and the syntax (attack relation) of the framework.

Conclusion and Future Research In this paper, we have defined an original translation-based approach to revise and merge consistent Horn formulae, in the special scenario of agents reasoning based on the minimal model of the formulae. We have defined specific revision and merging operators, proposed rationality postulates for both operations, and we have shown that some of our operators satisfy the postulates. Relaxing this restriction to consistent formulae is not straightforward, and would unnecessarily complexify the whole definition of the process.

With our adaptation of KM revision postulates and IC-merging postulates, we have proved that some of our operators are related to purely model-based revision and merging. In future work, we want to investigate the relation between our translation-based approach and syntax-based revision and merging. Indeed, we think that this kind of translation-based approach is a perfect way to explore the "middle ground" between syntactic and model-based approaches to belief change, combining the importance of both aspects.

There are various interesting open questions which we want to address in future work. First of all, our approach is nondeterministic in the sense that it may produce a collection of Horn formulae, each of them representing an acceptable result of revision or merging. This

kind of nondeterminism is quite common, especially in nonmonotonic reasoning where, say, a default theory can produce multiple extensions, or a logic program may have multiple stable models. Nevertheless, it is interesting to further investigate this issue. On one hand, conditions under which a unique Horn formula is generated are of interest. This is the case in particular when propositional operators are used, at the level of the encoding, which always yield a formula with a single model. On the other hand, we want to study selection functions for determining a unique outcome. Such functions may be based on further criteria, like additional preference information, or on some kind of a tie-break rule. A selection function must be used whenever a specific application requires a single Horn formula as the result of the process.

Finally, we aim for generalizing our logical encoding to take all models of a Horn formula into account, not only the minimal one. Such a generalization might pave the way to apply our translation-based method to other formalisms of interest, such as full propositional logic and logic programs.

Acknowledgements. G. Brewka was supported by DFG research unit FOR 1513 (HYBRIS). J.-G. Mailly and S. Woltran were supported by the Austrian Science Fund (FWF) under grant P25521.

A Proofs

Some proofs are omitted due to space reasons. In particular, proofs about translation-based merging are similar to proofs about translation based revision, with special care to the aggregation function.

Proof of Proposition 2. First, let us notice that if two interpretations ω_1, ω_2 coincide on the variables from $V \cup \{\text{cl}_c \mid c \in \text{hcl}(V)\}$, then $\omega_1 = \omega_2$, since the variables x^i can be deduced from the other variables. We call M the set of models of $CL(\varphi) \wedge H(V)$.

$\leq_{\varphi}^{(W_1, W_2)}$ denotes the pre-order defined by $\omega_1 \leq_{\varphi}^{(W_1, W_2)} \omega_2$ iff $\min_{\omega' \in M} d_{(W_1, W_2)}(\omega_1, \omega') \leq \min_{\omega' \in M} d_{(W_1, W_2)}(\omega_2, \omega')$.

Let $\omega_1 \in M$. Obviously, $\min_{\omega' \in M} d_{(W_1, W_2)}(\omega_1, \omega') = 0$. Now, given another interpretation ω_2 , $\min_{\omega' \in M} d_{(W_1, W_2)}(\omega_2, \omega') = 0$ holds iff $\omega_2 \in M$. In this case, $\omega_1 \simeq_{\varphi}^{(W_1, W_2)} \omega_2$, otherwise $\omega_1 <_{\varphi}^{(W_1, W_2)} \omega_2$. If we consider a formula $\varphi' \equiv \varphi$, the distance do not change (since the set of models are identical), and the pre-order $\leq_{\varphi'}^{(W_1, W_2)}$ is equal to $\leq_{\varphi}^{(W_1, W_2)}$.

The properties of faithful assignments are satisfied. This means that the revision operator induced by $\leq_{\varphi}^{(W_1, W_2)}$ satisfies the KM rationality postulates. \square

Proof of Theorem 3. Let \star be a revision operator based on a min-faithful assignment. \star satisfies $(\star 1)$ by definition.

$\{\text{mod}_{\min}(\varphi)\} \cap \text{mod}(\mu) \neq \emptyset$ is equivalent to $\text{mod}_{\min}(\varphi) \in \text{mod}(\mu)$. Since $\text{mod}_{\min}(\varphi)$ is the minimal element w.r.t. \leq_{φ} , it is obviously the minimal model of μ w.r.t. \leq_{φ} , so $(\star 2)$ is satisfied.

If $\text{mod}(\mu) \neq \emptyset$, then $\min(\text{mod}(\mu), \leq_{\varphi}) \neq \emptyset$ is ensured since \leq_{φ} is a total relation, so $(\star 3)$ is satisfied.

$(\star 4)$ follows from the definition of \leq_{φ} and \star . Indeed, if $\text{mod}_{\min}(\varphi_1) = \text{mod}_{\min}(\varphi_2)$ and $\text{mod}(\mu_1) = \text{mod}(\mu_2)$,

$$\begin{aligned} \text{mod}_{\min}(\varphi_1 \star \mu_1) &= \min(\text{mod}(\mu_1), \leq_{\varphi_1}) \\ &= \min(\text{mod}(\mu_2), \leq_{\varphi_1}) \text{ since } \mu_1 \equiv \mu_2 \\ &= \min(\text{mod}(\mu_2), \leq_{\varphi_2}) \text{ from Def.7} \\ &= \text{mod}_{\min}(\varphi_2 \star \mu_2) \end{aligned}$$

If $\text{mod}_{\min}(\varphi \circ \mu_1) \cap \text{mod}(\mu_2) = \emptyset$, then the last postulates are satisfied. So now we suppose that $\text{mod}_{\min}(\varphi \circ \mu_1) \cap \text{mod}(\mu_2) \neq \emptyset$.

We have $\text{mod}_{\min}(\varphi \circ \mu_1) \cap \text{mod}(\mu_2) = \min(\text{mod}(\mu_1), \leq_{\varphi}) \cap \text{mod}(\mu_2)$, and $\text{mod}_{\min}(\varphi \circ (\mu_1 \wedge \mu_2)) = \min(\text{mod}(\mu_1 \wedge \mu_2), \leq_{\varphi})$.

Using *reductio ad absurdum*, we suppose that there exists some interpretation ω_1 such that $\omega_1 \in \min(\text{mod}(\mu_1), \leq_{\varphi}) \cap \text{mod}(\mu_2)$ and $\omega_1 \notin \min(\text{mod}(\mu_1 \wedge \mu_2), \leq_{\varphi})$. From the first part, we know that $\omega_1 \in \text{mod}(\mu_1)$ and $\omega_1 \in \text{mod}(\mu_2)$, i.e. $\omega_1 \in \text{mod}(\mu_1 \wedge \mu_2)$. Now we deduce $\exists \omega_2 \in \text{mod}(\mu_1 \wedge \mu_2)$ such that $\omega_2 <_{\varphi} \omega_1$; but $\omega_2 \in \text{mod}(\mu_1)$, so $\omega_1 \notin \min(\text{mod}(\mu_1), \leq_{\varphi})$. This is a contradiction so \star satisfies $(\star 5)$.

To prove the opposite inclusion, let us also reason with *reduction ad absurdum*. We suppose that there is an interpretation ω_1 such that $\omega_1 \in \min(\text{mod}(\mu_1 \wedge \mu_2), \leq_{\varphi})$ and $\omega_1 \notin \min(\text{mod}(\mu_1), \leq_{\varphi}) \cap \text{mod}(\mu_2)$. From the first part, $\omega_1 \in \text{mod}(\mu_1 \wedge \mu_2)$, stated otherwise: $\omega_1 \in \text{mod}(\mu_1)$ and $\omega_1 \in \text{mod}(\mu_2)$. Since we have suppose that $\text{mod}_{\min}(\varphi \circ \mu_1) \cap \text{mod}(\mu_2) \neq \emptyset$, there is an interpretation ω_2 such that $\omega_2 \in \min(\text{mod}(\mu_1), \leq_{\varphi}) \cap \text{mod}(\mu_2)$. From these, we deduce $\omega_2 <_{\varphi} \omega_1$ and $\omega_2 \in \text{mod}(\mu_1) \cap \text{mod}(\mu_2) = \text{mod}(\mu_1 \wedge \mu_2)$, and so $\omega_1 \notin \min(\text{mod}(\mu_1 \wedge \mu_2), \leq_{\varphi})$. This is a contradiction, so \star satisfies $(\star 6)$. \square

Proof of Proposition 4. We just need to prove that mapping a Horn formula to a distance-based pre-order defines a min-faithful assignment. Let d be a distance between interpretations. $\forall \omega \neq \text{mod}_{\min}(\varphi)$, $d(\text{mod}_{\min}(\varphi), \text{mod}_{\min}(\varphi)) = 0$ is strictly less than $d(\omega, \text{mod}_{\min}(\varphi))$, so $\text{mod}_{\min}(\varphi) <_{\varphi}^d \omega$. $\forall \omega_1, \omega_2, \omega_1 \leq_{\varphi_1}^d \omega_2$ iff $d(\omega_1, \text{mod}_{\min}(\varphi_1)) \leq d(\omega_2, \text{mod}_{\min}(\varphi_1))$. Under the assumption $\text{mod}_{\min}(\varphi_1) = \text{mod}_{\min}(\varphi_2)$, this is equivalent to $d(\omega_1, \text{mod}_{\min}(\varphi_2)) \leq d(\omega_2, \text{mod}_{\min}(\varphi_2))$, and so $\omega_1 \leq_{\varphi_2}^d \omega_2$. This means that the pre-orders \leq_{φ_1} and \leq_{φ_2} are equals. \square

Proof of Proposition 5. We want to prove that the semantic minimal change revision operator is a particular distance based revision operator. $\star_{(W_1, 1)}$, with $W_1 = |\text{hcl}(V)| + 1$, is built from a distance between interpretations on the set of variables $V' = V \cup \{\text{cl}_c \mid c \in \text{hcl}(V)\}$. To be able to apply Proposition 4, we need to reformulate the definition of the operator to represent it with a distance between propositional interpretations on the set of variables V .

First of all, we observe from the definition of the translation-based revision operators that $\text{mod}_{\min}(\text{dec}(\varphi \star_{(W_1, 1)} \mu)) \subseteq \text{mod}(\mu)$. We need to identify a distance d on V such that $\text{mod}_{\min}(\text{dec}(\varphi \star_{(W_1, 1)} \mu)) = \min(\text{mod}(\mu), \leq_d)$. In our case, the Hamming distance on V proves enough to obtain the result.

Given $\omega_1 \in \min(\text{mod}(\mu \wedge H(V)), \leq_{\varphi}^{(W_1, 1)})$, the distance $d_{(W_1, 1)}(\omega_1, \text{mod}(CL(\varphi) \wedge H(V)))$ is minimal. In particular, since $CL(\varphi) \wedge H(V)$ has a single model ω_{φ} , $d_{(W_1, 1)}(\omega_1, \omega_{\varphi})$ is minimal. Stated otherwise, $W_1 d_H^V(\omega_1, \omega_{\varphi}) + d_H^{Syn}(\omega_1, \omega_{\varphi})$ is the minimal distance between ω_{φ} and another interpretation. Let us suppose that the x -part is not minimal, i.e. there exists ω_2 such that $W_1 d_H^V(\omega_2, \omega_{\varphi}) < W_1 d_H^V(\omega_1, \omega_{\varphi})$. We consider the extreme case when the cl_c -part is nul for ω_1 , i.e. $d_H^{Syn}(\omega_1, \omega_{\varphi}) = 0$, and it is maximal for ω_2 , i.e. $d_H^{Syn}(\omega_2, \omega_{\varphi}) = |\text{hcl}(V)|$. Then we obtain

$$\begin{aligned} W_1 d_H^V(\omega_2, \omega_{\varphi}) + d_H^{Syn}(\omega_2, \omega_{\varphi}) \\ < W_1 d_H^V(\omega_1, \omega_{\varphi}) + d_H^{Syn}(\omega_1, \omega_{\varphi}) \end{aligned}$$

because $W_1 = |\text{hcl}(V) + 1|$. This means that $d_{(W_1, 1)}(\omega_2, \omega_{\varphi}) < d_{(W_1, 1)}(\omega_1, \omega_{\varphi})$, which is a contradiction. This means that for each $\omega_1 \in \min(\text{mod}(\mu \wedge H(V)), \leq_{\varphi}^{(W_1, 1)})$, and for each ω_2 , $W_1 d_H^V(\omega_1, \omega_{\varphi}) \leq W_1 d_H^V(\omega_2, \omega_{\varphi})$, which is equivalent to $d_H^V(\omega_1, \omega_{\varphi}) \leq d_H^V(\omega_2, \omega_{\varphi})$. We notice that is means that the x -part of ω_1 is a minimal interpretation on V with respect to the Hamming distance. \square

REFERENCES

- [1] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson, 'On the logic of theory change: Partial meet contraction and revision functions', *J. Symb. Log.*, **50**(2), 510–530, (1985).
- [2] José Júlio Alferes, João Alexandre Leite, Luís Moniz Pereira, Halina Przymusińska, and Teodor C. Przymusiński, 'Dynamic updates of non-monotonic knowledge bases', *Journal of Logic Programming*, **45**(1–3), 43–70, (2000).
- [3] Grigoris Antoniou and Mary-Anne Williams, 'Revising default theories', in *Proceedings of the Tenth IEEE International Conference on Tools with Artificial Intelligence (ICTAI'98)*, pp. 423–430, (1998).
- [4] Ringo Baumann and Gerhard Brewka, 'AGM meets abstract argumentation: Expansion and revision for Dung frameworks', in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI'15)*, pp. 2734–2740, (2015).
- [5] Sylvie Coste-Marquis, Sébastien Konieczny, Jean-Guy Mailly, and Pierre Marquis, 'On the revision of argumentation systems: Minimal change of arguments statuses', in *Proceedings 14th International Conference on Principles of Knowledge Representation and Reasoning (KR'14)*, pp. 52–61, (2014).
- [6] Sylvie Coste-Marquis, Sébastien Konieczny, Jean-Guy Mailly, and Pierre Marquis, 'A translation-based approach for revision of argumentation frameworks', in *Proceedings of the Fourteenth European Conference on Logics in Artificial Intelligence (JELIA'14)*, pp. 397–411, (2014).
- [7] Nadia Creignou, Odile Papini, Reinhard Pichler, and Stefan Woltran, 'Belief revision within fragments of propositional logic', *J. Comput. Syst. Sci.*, **80**(2), 427–449, (2014).
- [8] Nadia Creignou, Odile Papini, Stefan Rümmele, and Stefan Woltran, 'Belief merging within fragments of propositional logic', in *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI'14)*, pp. 231–236, (2014).
- [9] Nadia Creignou, Reinhard Pichler, and Stefan Woltran, 'Do hard SAT-related reasoning tasks become easier in the Krom fragment?', in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI'13)*, pp. 824–831, (2013).
- [10] James P. Delgrande and Pavlos Peppas, 'Belief revision in Horn theories', *Artif. Intell.*, **218**, 1–22, (2015).
- [11] James P. Delgrande, Pavlos Peppas, and Stefan Woltran, 'AGM-style belief revision of logic programs under answer set semantics', in *Proceedings of the Twelfth International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR'13)*, pp. 264–276, (2013).
- [12] James P. Delgrande, Torsten Schaub, Hans Tompits, and Stefan Woltran, 'A model-theoretic approach to belief change in answer set programming', *ACM Trans. Comput. Log.*, **14**(2), 14, (2013).
- [13] Martin Diller, Adrian Haret, Thomas Linsbichler, Stefan Rümmele, and Stefan Woltran, 'An extension-based approach to belief revision in abstract argumentation', in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI'15)*, pp. 2926–2932, (2015).
- [14] Phan Minh Dung, 'On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and n-person games', *Artif. Intell.*, **77**(2), 321–357, (1995).
- [15] Thomas Eiter, Michael Fink, Giuliana Sabbatini, and Hans Tompits, 'On properties of update sequences based on causal rejection', *TPLP*, **2**(6), 711–767, (2002).
- [16] Dov M. Gabbay, Odinaldo Rodrigues, and Alessandra Russo, *Revision, Acceptability and Context - Theoretical and Algorithmic Aspects*, Cognitive Technologies, Springer, 2010.
- [17] Dov M. Gabbay, Odinaldo Rodrigues, and Alessandra Russo, 'Revision by translation', in *Revision, Acceptability and Context*, 223–270, Springer, (2010).
- [18] Peter Gärdenfors, 'Belief revision and nonmonotonic logic: Two sides of the same coin?', in *Proceedings of the Ninth European Conference on Artificial Intelligence (ECAI'90)*, pp. 768–773, (1990).
- [19] Adrian Haret, Stefan Rümmele, and Stefan Woltran, 'Merging in the Horn fragment', in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI'15)*, pp. 3041–3047, (2015).
- [20] Hirofumi Katsuno and Alberto O. Mendelzon, 'Propositional knowledge base revision and minimal change', *Artificial Intelligence*, **52**, 263–294, (1991).
- [21] Sébastien Konieczny and Ramón Pino Pérez, 'Merging information under constraints: A logical framework', *J. Log. Comput.*, **12**(5), 773–808, (2002).
- [22] Vladimir Lifschitz, David Pearce, and Agustin Valverde, 'Strongly equivalent logic programs', *ACM Trans. Comput. Log.*, **2**(4), 526–541, (2001).
- [23] Jean-Guy Mailly, *Dynamics of Argumentation Frameworks*, Ph.D. dissertation, Université d'Artois, 2015.
- [24] Guilin Qi and Jianfeng Du, 'Model-based revision operators for terminologies in description logics', in *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI'09)*, pp. 891–897, (2009).
- [25] Nicolas Schwind and Katsumi Inoue, 'Characterization of logic program revision as an extension of propositional revision', *TPLP*, **16**(1), 111–138, (2016).
- [26] Martin Slota and João Leite, 'The rise and fall of semantic rule updates based on se-models', *TPLP*, **14**(6), 869–907, (2014).
- [27] Hudson Turner, 'Strong equivalence made easy: nested expressions and weight constraints', *TPLP*, **3**(4–5), 609–622, (2003).

Parallel Filter-Based Feature Selection Based on Balanced Incomplete Block Designs

Antonio Salmerón¹ and Anders L. Madsen² and Frank Jensen³ and Helge Langseth⁴ and Thomas D. Nielsen⁵ and Darío Ramos-López⁶ and Ana M. Martínez⁷ and Andrés R. Masegosa⁸

Abstract. In this paper we propose a method for scaling up filter-based feature selection in classification problems. We use the conditional mutual information as filter measure and show how the required statistics can be computed in parallel avoiding unnecessary calculations. The distribution of the calculations between the available computing units is determined based on balanced incomplete block designs, a strategy first developed within the area of statistical design of experiments. We show the scalability of our method through a series of experiments on synthetic and real-world datasets.

1 INTRODUCTION

For datasets with a high number of variables, a key step in developing classification models is to determine the most informative variables for the model and exclude the less informative ones. Furthermore, some of the variables may be redundant in the context of others. Learning classifiers with an excessive number of variables may yield models overfitting irrelevant aspects of the data, and therefore showing a poor predictive power. Also, the complexity of learning usually grows with the number of variables, and hence the computational cost of learning with all the available variables may become infeasible.

Feature selection methods can be grouped into three main blocks, *wrapper*, *filter*, and *embedded* methods [17, 19, 1]. There are also mixed filter-wrapper approaches, some of which are able to operate in high dimensional domains [5, 4].

Wrapper methods [15] explore the space of possible feature subsets, optimizing some model-dependent metric like accuracy. It means that for each explored feature subset, a model is learnt and evaluated. Traditional wrapper approaches for feature selection are often time consuming, and therefore not viable within the context of learning from large amounts of data or in high dimensionality domains. Embedded methods [18] are also model-dependent. They use some specific property of the target model to guide the search procedure. On the other hand, filter methods [10] are independent of the underlying model, and can be seen as a pre-processing step

performed before model learning. Univariate filters use some score function to produce a ranking of the features. Sometimes a threshold is defined and the features that score below it are discarded, other times a maximum number of features could be retained. There are also methods based on multivariate filters, which attempt to assess the goodness of a feature subset rather than a single feature [24].

In this paper we focus on univariate filter methods, as their low computational complexity, compared to wrapper schemes, make them appropriate for problems involving a high number of variables. We propose a method for scaling up filter-based feature selection in classification problems. We use the conditional mutual information as filter measure and show how the required statistics can be computed in parallel avoiding unnecessary calculations. The distribution of the calculations between the available computing units is determined based on balanced incomplete block designs, a strategy first developed within the area of statistical design of experiments.

2 FILTER-BASED FEATURE SELECTION

Some of the most outstanding filter methods are based on the use of information-theoretic score functions, all of them related to the concept of *entropy* [8]. Throughout this paper we shall assume a set of discrete variables $\mathbf{X} = \{X_1, \dots, X_n\}$ and a class variable C . The entropy of a discrete variable $X \in \mathbf{X}$ is

$$H(X) = - \sum_{x \in \Omega_X} p(x) \log p(x).$$

The entropy measures the uncertainty in the distribution of X . Given two variables $X_i, X_j \in \mathbf{X}$, the *conditional entropy* of X_i given X_j is

$$H(X_i|X_j) = - \sum_{x_j \in \Omega_{X_j}} p(x_j) \sum_{x_i \in \Omega_{X_i}} p(x_i|x_j) \log p(x_i|x_j),$$

and it quantifies the uncertainty that remains in the distribution of X_i after observing X_j .

The amount of information shared by two variables $X_i, X_j \in \mathbf{X}$ can be measured by their *mutual information* [25, 8]:

$$I(X_i, X_j) = H(X_i) - H(X_i|X_j).$$

Note that the mutual information is symmetric, $I(X_i, X_j) = I(X_j, X_i)$. It can be interpreted as the amount of uncertainty in X_i that is removed after observing X_j . Similarly, given $X_i, X_j, X_k \in \mathbf{X}$ the *conditional mutual information* between X_i and X_j given X_k can be defined as

¹ University of Almería, Spain, email: antonio.salmeron@ual.es

² HUGIN EXPERT A/S and Aalborg University, Denmark, email: alm@hugin.com

³ HUGIN EXPERT A/S, Aalborg, Denmark, email: fj@hugin.com

⁴ Norwegian University of Science and Technology, Trondheim, Norway, email: helgel@idi.ntnu.no

⁵ Aalborg University, Denmark, email: tdn@cs.aau.dk

⁶ University of Almería, Spain, email: dramosl@ual.es

⁷ Aalborg University, Denmark, email: ana@cs.aau.dk

⁸ Norwegian University of Science and Technology, Trondheim, Norway, email: andresrm@idi.ntnu.no

$$I(X_i, X_j | X_k) = H(X_i | X_k) - H(X_i | X_j, X_k).$$

Alternatively, the conditional mutual information can be written as

$$I(X_i, X_j | X_k) = \sum_{x_i, x_j, x_k} p(x_i, x_j | x_k) \log \frac{p(x_i, x_j | x_k)}{p(x_i | x_k) p(x_j | x_k)}.$$

The mutual information is defined analogously in the multivariate setting for random vectors \mathbf{X}_i , \mathbf{X}_j and \mathbf{X}_k as

$$I(\mathbf{X}_i, \mathbf{X}_j | \mathbf{X}_k) = \sum_{\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k} p(\mathbf{x}_i, \mathbf{x}_j | \mathbf{x}_k) \log \frac{p(\mathbf{x}_i, \mathbf{x}_j | \mathbf{x}_k)}{p(\mathbf{x}_i | \mathbf{x}_k) p(\mathbf{x}_j | \mathbf{x}_k)}.$$

A thorough analysis of information-theoretic filter methods for variable selection is reported in [6]. The analysis is based on the *conditional likelihood* of the class variable given \mathcal{S} (features included in the model) and τ (parameters of the distributions involved in the model) for a data set $\mathcal{D} = \{(\mathbf{x}^i, c^i), i = 1, \dots, n\}$, defined as

$$\mathcal{L}(\mathcal{S}, \tau | \mathcal{D}) = \prod_{i=1}^n q(c^i | \mathbf{x}^i, \tau),$$

where q is the learnt model. It is shown in [6] that the conditional likelihood is maximized by minimizing $I(\mathbf{X} \setminus \mathcal{S}, C | \mathcal{S})$. In other words, the conditional likelihood is maximized when the mutual information between the class and the features not included in the model, given the variables actually included, is minimized.

A general framework for filter-based feature selection can be established by making two technical assumptions about the independence structure of the domain [6]: For $X_i, X_j \in \mathcal{S}$, $X_k \in \mathbf{X} \setminus \mathcal{S}$ it holds that X_i and X_j are conditionally independent both when conditioning on X_k and on $\{X_k, C\}$. The framework is based on selecting features greedily, using a filter measure defined as

$$J_{\text{cmi}}(X_i) = I(X_i, C | \mathcal{S}), \quad (1)$$

where X_i is the candidate variable to include in the model.

Utilizing the assumptions mentioned above, $J_{\text{cmi}}(X_i)$ can also be calculated as

$$J_{\text{cmi}}(X_i) = I(X_i, C) - \sum_{X_j \in \mathcal{S}} (I(X_i, X_j) - I(X_i, X_j | C)). \quad (2)$$

The advantage of the definition in Equation (2) compared to Equation (1) is that the former does not require the computation of the conditional mutual information over multi-dimensional random variables. Several existing filter-based methods can be shown to be related to Equation (2). A remarkable example is the *joint mutual information* filter measure [28], defined as

$$\begin{aligned} J_{\text{jmi}}(X_i) &= \sum_{X_j \in \mathcal{S}} I(\{X_i, X_j\}, C) \\ &= I(X_i, C) - \frac{1}{|\mathcal{S}|} \sum_{X_j \in \mathcal{S}} (I(X_i, X_j) - I(X_i, X_j | C)), \quad (3) \end{aligned}$$

whose relation to Equation (2) is clear. Among the measures tested in [6], J_{jmi} is the one showing the best accuracy/stability tradeoff, and is therefore the one we utilize in the following. Note, however, that our approach works equally well for other measures, like J_{cmi} (when calculated using Equation (2)).

2.1 Scaling up filter based feature selection

Filter methods usually rank the features independently from each other. This is the approach taken in [22], where vertical parallelization is used. By *vertical parallelization* we mean distributing the variables among the computing units, corresponding to splitting the dataset by columns, while *horizontal parallelization* refers to splitting the dataset by rows and including all the variables in each split. In [22] different subsets of candidate variables are distributed through the available computing units, where the feature selection filters are actually applied. The selection is carried out through a voting process, where first the features that are discarded in each computing unit receive one vote, after which the features that have received the most votes are excluded from the final model.

The approach we adopt in this paper is different, as we propose to filter the variables globally and ensure scalability by distributing the calculations between the available computing units. More precisely, we follow the filter procedure described in Algorithm 1. It is based on a greedy strategy intended to select the feature that maximizes the joint mutual information measure in Equation (3). The filter measure is based on the mutual and conditional mutual information of each pair of features, computed in Steps 3 to 9. The variables are then included one by one (Steps 12 to 19) until a given maximum number of features have been selected.

1 Function Filter(\mathbf{X}, C, M)

Input: The set of features, $\mathbf{X} = \{X_1, \dots, X_N\}$. The class variable, C . The maximum number of features to be selected, M .

Output: \mathcal{S} , a set of selected features.

2 begin

3 for $i \leftarrow 1$ to N do

 Compute $I(X_i, C)$;

5 for $j \leftarrow i + 1$ to N do

 Compute $I(X_i, X_j | C)$;

 Compute $I(X_i, X_j)$;

8 end

9 end

$X^* \leftarrow \arg \max_{1 \leq i \leq N} I(X_i, C)$;

$\mathcal{S} \leftarrow \{X^*\}$;

12 for $i \leftarrow 1$ to $M - 1$ do

$\mathcal{R} \leftarrow \mathbf{X} \setminus \mathcal{S}$;

14 for $X \in \mathcal{R}$ do

 Compute $J_{\text{jmi}}(X)$ as in Equation (3) using the statistics computed in Steps 4, 6, and 7;

16 end

$X^* \leftarrow \arg \max_{X \in \mathcal{R}} J_{\text{jmi}}(X)$;

$\mathcal{S} \leftarrow \mathcal{S} \cup \{X^*\}$;

19 end

20 return \mathcal{S} ;

21 end

Algorithm 1: Filter-based feature selection based on conditional likelihood maximization.

Our proposal for scaling up Algorithm 1 consists of parallelizing Steps 4 to 9, which involve the calculation of the (conditional) mutual information between each pair of variables. These steps are also the most computationally demanding when the probabilities involved in the computations are estimated from data. An immediate approach for scaling up the algorithm could be to simply generate

one computing thread for each pair of variables and then process the threads in parallel. However, with n variables this approach would require accessing the underlying database $\binom{n}{2}$ times, inducing a significant overhead in terms of disk/network access. Alternatively, one might group the variables in blocks so that each block only accesses the data a single time in order to calculate the sufficient statistics required for estimating the (conditional) mutual information between all pairs of variables within the block. A key issue here is finding an appropriate block size and at the same time ensure that the blocks, in combination, guarantee that all pairs of variables are considered exactly once. This is the idea we propose.

To get an intuitive understanding of this process we can as an analogy consider the organization of the Speedway World Championship (SWC). After the initial pre-qualifying rounds for the SWC, the remaining 16 highest ranked riders should be compared to each other to obtain a final ranking of the riders. One approach to achieve this would be to pair-up the riders so that each rider will participate in 15 races, yielding a total of 120 rounds with two riders competing in each round. This setup would put a strain on the riders and not use the full capacity of the speedway track, which is designed to accommodate four riders simultaneously. Instead, the SWC employs a heat-system ensuring that each of the 16 riders will meet each of the other riders at some time during the competition. Specifically, the heat-system consists of 20 heats with four riders in a heat. Each rider participates in only five heats, and within a single heat all riders compete jointly, thereby meeting each other. After completing the 20 heats, all pairs of riders will have met exactly once. This can also be seen by labeling the riders $\{0, \dots, 15\}$ and constructing these heats: $H_1 = \{3, 6, 12, 15\}$, $H_2 = \{4, 5, 10, 13\}$, $H_3 = \{0, 4, 6, 7\}$, $H_4 = \{0, 10, 11, 15\}$, $H_5 = \{7, 10, 12, 14\}$, $H_6 = \{0, 8, 9, 14\}$, $H_7 = \{0, 1, 3, 13\}$, $H_8 = \{1, 6, 8, 10\}$, $H_9 = \{7, 9, 13, 15\}$, $H_{10} = \{1, 5, 14, 15\}$, $H_{11} = \{8, 11, 12, 13\}$, $H_{12} = \{5, 6, 9, 11\}$, $H_{13} = \{1, 4, 9, 12\}$, $H_{14} = \{3, 5, 7, 8\}$, $H_{15} = \{3, 4, 11, 14\}$, $H_{16} = \{2, 6, 13, 14\}$, $H_{17} = \{1, 2, 7, 11\}$, $H_{18} = \{0, 2, 5, 12\}$, $H_{19} = \{2, 4, 8, 15\}$, and $H_{20} = \{2, 3, 9, 10\}$.

Returning to the (conditional) mutual information calculations, the 16 riders correspond to variables and each heat represents a block consisting of four variables to be pairwise compared. Thus, rather than handling pairs of variables independently and having to make data access $\binom{16}{2} = 120$ times, we can instead make 20 blocks/heats of four variables each and thereby only having to access the data 20 times. Note that with the particular setup above, we are guaranteed not to make redundant calculations as the scores $I(X_i, X_j|C)$ and $I(X_i, X_j)$ are computed exactly once for all $1 \leq i, j \leq n$.

This approach of distributing features/riders into blocks/heats is an instance of a so-called *balanced incomplete block (BIB) design*; in fact the heat-system configuration employed by the Speedway World Championship correspond to a $(16, 4, 1)$ -BIB design (see Definition 2). BIB designs have already been used in related contexts like edge labeling of the spanning tree associated to TAN classifiers [20] and organizing the conditional independence tests in the PC algorithm for learning the structure of a Bayesian network [21]. In the following sections, we will give a more detailed specification of the approach and demonstrate how it can be used to ensure the scalability of Algorithm 1.

As a final note on computational efficiency, we would like to highlight that the filter approach in Algorithm 1 is especially appropriate for Bayesian classifiers and, in particular, for the Naive Bayes classifier, where the probability distribution of the class variable given the

features is modeled as

$$p(c|x_1, \dots, x_N) \propto p(c) \prod_{i=1}^N p(x_i|c).$$

For this model we see that all the required distributions have already been estimated when computing the conditional mutual information in Step 6. It means that some of the calculations carried out during the feature selection phase can be re-used when learning the model parameters. The same result also applies to TAN classifiers [12]. Furthermore, from the perspective of scalability, it is also worth pointing out that the information-theoretic score functions employed above can be efficiently updated after the arrival of new data if the distributions involved belong to the exponential family. It is enough to store a set of sufficient statistics corresponding to the parameters of the model, which are just counts for discrete variables.

2.2 Balanced incomplete block designs

The use of block designs dates back to the statistical theory of design of experiments [11], motivated in its origin by agricultural experiments. In this scenario the goal was to compare the yield of different plant varieties, considering that the yield could be significantly affected by the environment, i.e., the conditions under which the plants are grown. The idea was to compensate for the effect of the environment by setting up blocks of land small enough to assume uniform environmental conditions inside a block, and distribute the plant varieties among them. With space limitations inside each block, one may not be able to fit sufficient replications of all plant varieties inside a single block, and therefore rather required that each pair of plant varieties would be allocated at least once to the same block to facilitate a fair comparison between them. The relation to both the SWC and our calculation of (conditional) mutual information is evident.

BIB designs [26] can be applied to efficiently divide the computation of the (conditional) mutual information between all the pairs of features among a set of, for instance, threads on a shared memory system or processes on a distributed memory system. This section provides the necessary background information on BIB designs to follow the presentation of the method proposed. For ease of presentation, we focus on a shared memory system using threads to achieve parallelization.

Definition 1 (Design [26]) *A design is a pair (X, \mathcal{A}) s. t. the following properties are satisfied:*

1. X is a set of elements called points, and
2. \mathcal{A} is a collection of non-empty subsets of X called blocks.

In this paper, we only exploit cases where each block is a set, and not a multi-set (i.e., we do not allow multiple instances of the same element in the set). Nevertheless, some definitions will consider multi-sets. A BIB design is defined as follows:

Definition 2 (BIB design [26]) *Let v , k and λ be positive integers s. t. $v > k \geq 2$. A (v, k, λ) -BIB design is a design (X, \mathcal{A}) s. t. the following properties are satisfied:*

1. $|X| = v$,
2. each block contains exactly k points, and
3. every pair of distinct points is contained in exactly λ blocks.

The number of blocks in a design is denoted by b . Property 3 in the definition is the *balance* property that we will exploit. In Steps 6

and 7 of Algorithm 1, we want to compute the (conditional) mutual information between each pair X_i, X_j exactly once and therefore require $\lambda = 1$. A BIB design is *symmetric* when the number of blocks equals the number of points. This will not be the case in general.

Example 1 Consider the $(7, 3, 1)$ -BIB design. The blocks are (one out of a number of possibilities):

$\{1, 2, 3\}, \{1, 4, 5\}, \{0, 1, 6\}, \{2, 4, 6\}, \{0, 2, 5\}, \{0, 3, 4\}, \{3, 5, 6\}$.

This BIB design is symmetric as the number of blocks equals the number of points.

There is no single efficient method to construct all BIB designs. First, it is important to know that they do not exist for all combinations of v, k , and λ . Second, the problem of finding a BIB design is NP-complete [7]. To efficiently utilize them we have therefore pre-calculated a number of BIB designs, and utilized those at run-time. Instead of storing the full designs, it is sufficient to store *difference sets* that can be used to generate some symmetric BIB designs:

Definition 3 (Difference Set [26]) Assume $(G, +)$ is a finite group of order v in which the identity element is 0. Let k and λ be positive integers such that $2 \leq k < v$. A (v, k, λ) -difference set in $(G, +)$ is a subset $D \subseteq G$ that satisfies the following properties:

1. $|D| = k$,
2. the multi-set $[x - y : x, y \in D, x \neq y]$ contains every element in $G \setminus \{0\}$ exactly λ times.

In our case, we are restricted to using $(\mathbb{Z}_v, +)$, the integers modulo v . If $D \subseteq \mathbb{Z}_v$ is a difference set in group $(G, +)$, then $D + g = \{x + g | x \in D\}$ is a translate of D for any $g \in G$. The multi-set of all v translates of D is denoted $\text{Dev}(D)$ and called the development of D [26, page 42].

Theorem 1 ([26], Theorem 3.8 p. 43) Let D be a (v, k, λ) -difference set in an Abelian group $(G, +)$. Then $(G, \text{Dev}(D))$ is a symmetric (v, k, λ) -BIB design.

Example 2 The set $D = \{0, 1, 3\}$ is a $(7, 3, 1)$ -difference set in $(\mathbb{Z}_7, +)$. The blocks constructed by iteratively adding one to each element of D (modulo 7) are:

$\{0, 1, 3\}, \{1, 2, 4\}, \{2, 3, 5\}, \{3, 4, 6\}, \{4, 5, 0\}, \{5, 6, 1\}, \{6, 0, 2\}$.

Notice that the i th element of each block is unique across all blocks. This property will be used to assign blocks to threads in Section 2.3. This was not the case for the blocks presented in Example 1.

The concept of a difference set can be generalized to the concept of a *difference family*. A difference family is a set of base blocks. A difference family can be used to generate a BIB design similarly to how difference sets are used. Table 1 shows a set of difference families for BIB designs on the form $(q, 6, 1)$, which we will use later. Base blocks for generating BIB-designs are tabulated, e.g., [27], but can also be found computationally. The base blocks in Table 1 have been generated using SageMath⁹. The value $k = 6$ is chosen for practical reasons: First, difference families for generating the blocks need to be known to exist; second, we need to be able to store the count tables representing the joint distribution of the class and all the variables in a block in memory. The main idea for parallelization considered in this paper is to use the $(q, 6, 1)$ design to distribute the computations of the scores over a set of computing units such that each score is computed exactly once from a smaller intermediate table over six variables.

⁹ www.sagemath.org

2.3 Computing the (conditional) mutual information

The computation of the mutual and conditional mutual information scores $I(X_i, X_j)$ and $I(X_i, X_j | C)$ for all pairs of features X_i, X_j should be divided into tasks of (approximately) equal size such that the score for each pair X_i, X_j is computed exactly once. This is achieved using BIB designs of the form $(q, 6, 1)$ where q is at least the number of features. That is, q is selected as the smallest value larger than the number of variables such that a $(q, 6, 1)$ BIB design is known to exist. The blocks of the BIB design are generated using a difference family (e.g., Table 1). Each block is used to compute the marginal counts of the features represented in the block (and the class variable). If all features have the same state space size, then the count tables will be of equal size.

The computation of the mutual and conditional mutual information scores is parallelized assigning blocks to the computing units available (we assume threads) as each thread can compute the scores corresponding to a block in parallel with other threads. Blocks are assigned to threads using the unique rank of each thread. A thread with rank r iterates over the block array and considers only blocks where the array index modulus t equals r where t is the number of threads (the uniqueness means that there is no need for synchronization). When a thread has selected a block, it computes the scores for all the pairs of features using a $(3, 2, 1)$ -BIB design where the 6-block is marginalized to three blocks with four features each (in this case each point corresponds to two features). The tables of four variables are marginalized down to all pairs for computing the score where the first pair is ignored producing a total of $\binom{6}{2} = 15$ scores to be computed.

Figure 1 illustrates this principle, assuming an example with $q = 31$ features labelled as X_0, \dots, X_{30} . The first block (second row in the figure) is $\{X_1, X_2, X_7, X_{19}, X_{23}, X_{30}\}$, corresponding to the difference family for design $(31, 6, 1)$, as given in Table 1. The second block would be obtained by adding 1 to the index of the variable in each coordinate, modulo 31, i.e., $\{X_2, X_3, X_8, X_{20}, X_{24}, X_0\}$. According to the same procedure, the third block would be $\{X_3, X_4, X_9, X_{21}, X_{25}, X_1\}$ and so on.

Taking the first block, we form three pairs of features, $P_1 = \{X_1, X_2\}$, $P_2 = \{X_7, X_{19}\}$ and $P_3 = \{X_{23}, X_{30}\}$ and compute the blocks of a $(3, 2, 1)$ design, where each block has two pairs. These blocks are actually all the possible pairings of P_1, P_2 and P_3 , namely $\{P_1, P_2\}$, $\{P_2, P_3\}$ and $\{P_3, P_1\}$, placed on the third row of Figure 1. It can be seen that every three pairings we come up with $5 \times 3 = 15$ pairs of features for which the mutual and mutual information score is computed. In fact, each block corresponding to a pairing $\{P_i, P_j\}$ yields 6 pairs of features, but the first one is discarded in order to avoid repetitions. In Figure 1 it is indicated by marking both variables in red on the lower row.

Notice that $k = 6$ represents 15 pairs and the number of times we count is reduced by a factor of 15, but each count is a factor three more expensive (as we are counting six variables instead of two variables). In addition, there is the task of marginalizing the count tables to pairs. If the number of states for some features is high, then it may be more efficient to compute the score directly from the dataset instead of creating an intermediate table. More precisely, the table can become larger than the original dataset meaning that we would be computing counts from a table that is larger than the original dataset. In this case we may not obtain a time performance improvement.

Table 1. Examples of difference families for a set of $(q, 6, 1)$ -BIB designs.

BIB design	Difference family	#(base blocks)	$b = v \cdot \#(\text{base blocks})$
(31,6,1)	$\{(1, 2, 7, 19, 23, 30)\}$	1	31
(91,6,1)	$\{(0, 1, 3, 7, 25, 38), (0, 5, 20, 32, 46, 75), (0, 8, 17, 47, 57, 80)\}$	3	273
(151,6,1)	$\{(1, 32, 118, 7, 73, 71), \dots\}$	5	755
(211,6,1)	$\{(0, 1, 107, 55, 188, 71), \dots\}$	7	1477
(271,6,1)	$\{(1, 242, 28, 9, 10, 232), \dots\}$	9	2439

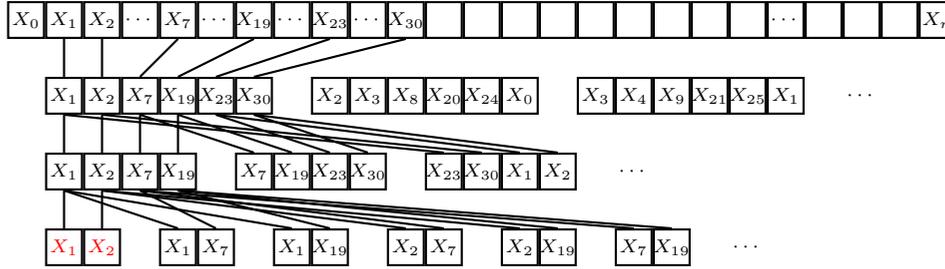


Figure 1. Example illustrating the use of $(q, 6, 1)$ and $(3, 2, 1)$ designs.

3 RESULTS

This section reports on the results of an empirical evaluation of the proposed parallel filter-based feature selection method based on BIB designs. The main focus of the empirical evaluation is to investigate the time performance improvement offered by the use of BIB designs to compute the score used by Algorithm 1. We consider the implementation of the proposed method on a shared memory computer with multiple cores such that threads can be used to achieve parallelization. This means that the entire dataset is loaded into the main shared memory of the computer where the process of the program is responsible for creating a set of POSIX threads to achieve parallelization.

Table 2. Networks from which datasets used in the experiments are generated.

Dataset	$ \mathcal{X} $	$ E $	Total CPT size
Munin1 [3]	189	282	19,466
Diabetes [2]	413	602	461,069
Munin2 [3]	1,003	1,244	83,920
SACSO [13]	2,371	3,521	44,274

The evaluation is performed using a total of fifteen datasets where twelve datasets are generated by sampling from known Bayesian network models [23, 9, 14, 16] and three are subsets of a real-world dataset over 1,823 variables from a Spanish bank. Random samples of data were generated from the four Bayesian networks of different sizes listed in Table 2 where $|\mathcal{X}|$ denotes the number of variables and $|E|$ denotes the number of edges in the Bayesian network. For each network a single variable is selected as target variable. Three datasets were generated at random for each network with 100,000, 250,000, and 500,000 cases. In addition, three samples from the bank dataset have been used. All datasets used are unless otherwise stated complete, i.e., there are no missing values in the data.

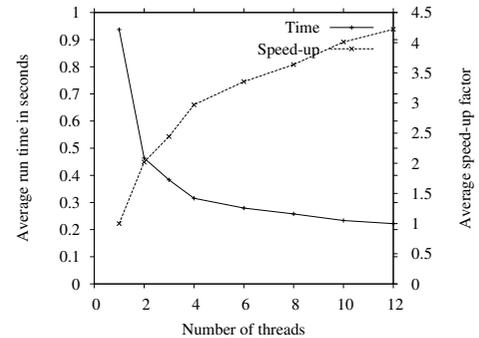


Figure 2. Munin1 with 100,000 cases.

The empirical evaluation is performed on a desktop computer running Red Hat Enterprise Linux 7 with a six-core Intel (TM) i7-5820K 3.3GHz processor and 64 GB RAM. The computer has six physical cores and twelve logical cores. For this reason, the number of threads used by the program is in the set $\{1, 2, 3, 4, 6, 8, 10, 12\}$ where the case of one thread is considered the baseline and corresponds to a sequential program.

The average computation time is calculated over ten runs with the same dataset. The computation time is measured as the elapsed (wall-clock) time of the algorithm. This means that other users of the computer may potentially impact the results. The speed-up factor is computed relative to the case of one thread.

Figures 2, 3, and 4 show the time performance and resulting speed-up factor achieved on the Munin1 dataset with 100,000, 250,000 and 500,000 cases, respectively.

The three performance figures for the Munin1 datasets clearly show an improvement in time performance as the number of threads

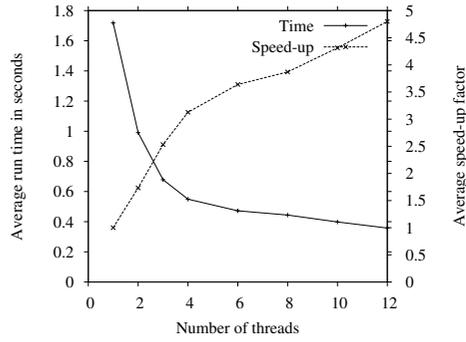


Figure 3. Munin1 with 250,000 cases.

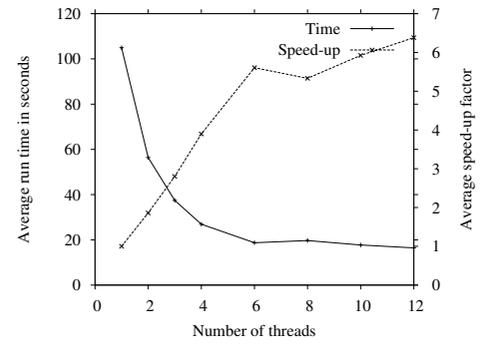


Figure 5. Munin2 with 500,000 cases.

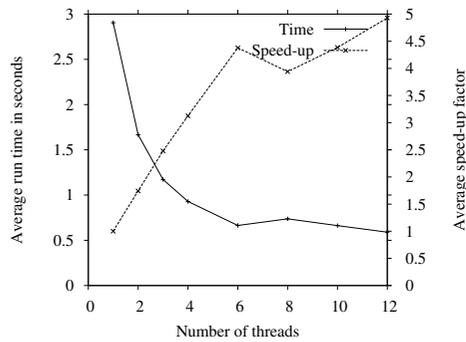


Figure 4. Munin1 with 500,000 cases.

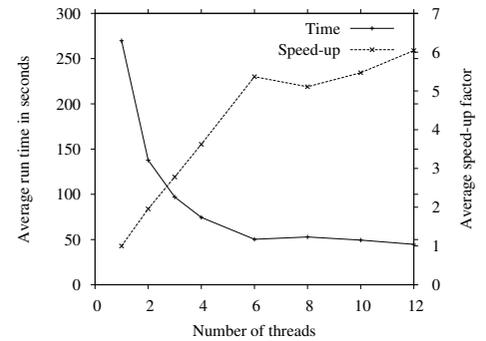


Figure 6. Bank with 500,000 cases.

used increases. The highest speed-up factor of approximately five is obtained using twelve threads and 500,000 cases. The most significant speed-up is obtained when going from one to two threads where the speed-up factor is close to two. After six threads there is in Figure 4 a small drop in performance. This is most likely related to the number of physical cores in the computer. Recall that the computer used in the test has six physical cores and twelve logical cores. We believe this is the explanation for the drop in performance going from six to eight threads.

Due to space restrictions, we do not show the time performance graphs for all tests. Instead we show the graphs for the largest datasets.

Figures 5, 6, and 7 show the time performance and resulting speed-up factor achieved on the Munin2 dataset with 500,000 cases, Bank with 500,000 cases, and SACS0 with 500,000 cases, respectively.

The three performance graphs for Munin2, SACS0 and Bank shown in Figures 5, 7, and 6, respectively show similar time performance improvements as Munin1. A highest speed-up factor of approximately six is obtained for all three networks.

Figure 8 shows the time performance and resulting speed-up factor achieved on the Diabetes dataset with 100,000 cases.

In this figure, we notice that the performance starts to deteriorate after six threads. Diabetes is a time-sliced model with a number of variables in each time slice having up to 21 states. This means that the use of $(q, 6, 1)$ designs could produce intermediate tables that are as large as or even are larger than the original dataset and the work could be unevenly distributed as a result of different table sizes.

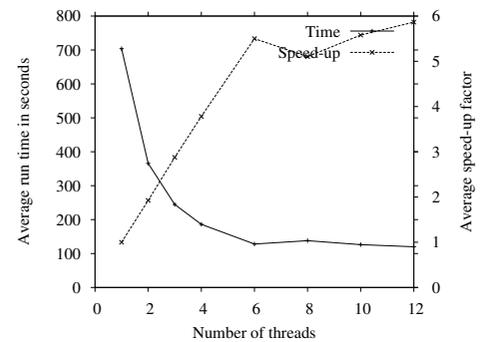


Figure 7. SACS0 with 500,000 cases.

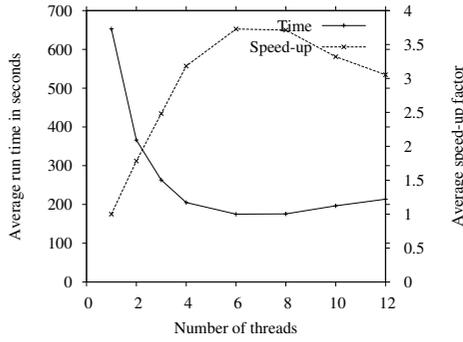


Figure 8. Diabetes with 100,000 cases.

Figure 8 shows the time performance and resulting speed-up factor achieved on the Diabetes dataset with one incomplete and 100,000 complete cases.

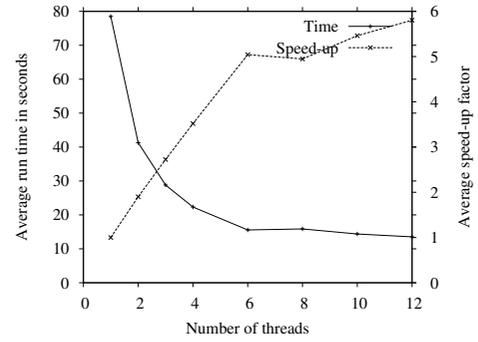


Figure 10. Bank with 100,000 complete cases.

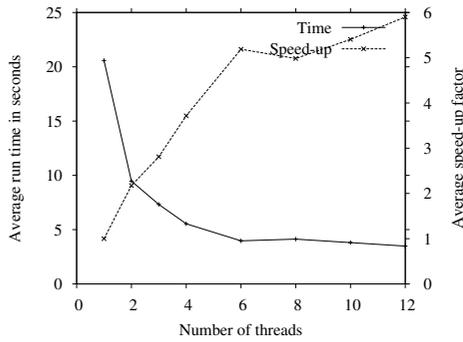


Figure 9. Diabetes with 100,000 complete and one incomplete cases.

Adding an incomplete case to the dataset means that $(q, 6, 1)$ designs are not used and the scores are computed directly from the original dataset. In this case, time performance is significantly better than in the case of complete data and the speed-up factor improves as the number of threads increases. Figure 10 shows the performance graphs for the Bank dataset with 100,000 complete cases while Figure 11 shows the performance for 100,000 complete cases and one incomplete case. It is clear from the two figures that the use of $(q, 6, 1)$ designs improves time performance significantly.

4 DISCUSSION

The results of the empirical evaluation reported in Section 3 demonstrates the scalability of the proposed approach to parallel filter-based feature selection using BIB designs. On the test computer the time performance clearly improves as the number of threads increases up to the number of physical cores. Thereafter, there is in some cases a minor drop in performance.

The proposed method uses BIB designs in two steps in order to ensure that the score is computed for each pair exactly once and such that the workload can be evenly distributed over the available computing units without any synchronization. In the first step a $(q, 6, 1)$ design is applied to create intermediate tables that are marginalised

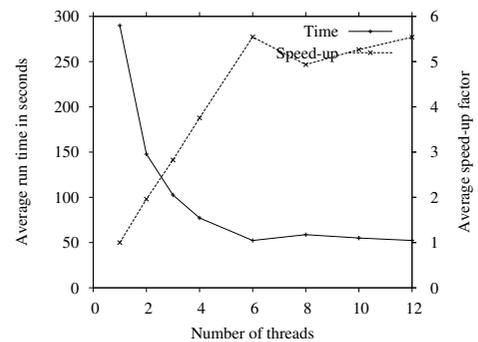


Figure 11. Bank with 100,000 complete and one incomplete cases.

to two or three variables to compute the score using a (3, 2, 1) design. That is, the $(q, 6, 1)$ design is used to create intermediate tables from which the scores for a specific set of pairs are calculated. In some cases, when variables have a high number of states as in the Diabetes network, the size of the intermediate tables may become too large to manage efficiently and performance can deteriorate. We can avoid this step by determining all pairs directly from the entire set of features. This would mean that the scores for each pair are computed over the entire dataset. A upper limit equal to the size of the original dataset could be put on the size of the intermediate tables.

If data has missing values, then we cannot exploit the $(q, 6, 1)$ designs as an intermediate table for the variables with missing values and would compute the counts for each pair directly from the original dataset ignoring cases with missing values. The performance improvement offered by the intermediate tables creating using $(q, 6, 1)$ designs is in most cases substantial.

Future work includes horizontal parallelization of the algorithm where each computing unit holds a subset of the data over all variables and is responsible for computing partial counts over the data it holds. This will support parallelization on a distributed memory system.

5 CONCLUSION

In this paper, we have proposed a method for scaling up filter-based feature selection in classification problems using parallelization. The principal idea is to use the conditional mutual information as a filter measure distributing the computation of the required statistics over a set of computing units. The distribution of the computations is managed using BIB designs requiring no synchronization and ensuring that each score is computed exactly once. We have demonstrated the scalability of the proposed method using both synthetic and real-world datasets.

ACKNOWLEDGEMENTS

This work was performed as part of the AMIDST project. AMIDST has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 619209. We thank BCC for providing access to an obfuscated sample dataset.

REFERENCES

- [1] S. Ahmed, H. Narasimhan, and S. Agarwal, 'Bayes optimal feature selection for supervised learning with general performance measures', in *Proceedings of the UAI'2015 Conference*, (2015).
- [2] S. Andreassen, R. Hovorka, J. Benn, K.G. Olesen, and E.R. Carson, 'A model-based approach to insulin adjustment', in *Proc. of the Third Conference on Artificial Intelligence in Medicine*, pp. 239–248, (1991).
- [3] S. Andreassen, F. V. Jensen, S. K. Andersen, B. Falck, U. Kjærulff, M. Woldbye, A. R. Sørensen, A. Rosenfalck, and F. Jensen, 'MUNIN — an expert EMG assistant', in *Computer-Aided Electromyography and Expert Systems*, chapter 21, Elsevier Science, (1989).
- [4] P. Bermejo, L. de la Ossa, J.A. Gámez, and J.M. Puerta, 'Fast wrapper feature subset selection in high-dimensional datasets by means of filter re-ranking', *Knowledge-Based Systems*, **25**, 35–44, (2012).
- [5] P. Bermejo, J.A. Gámez, and J.M. Puerta, 'A GRASP algorithm for fast hybrid (filter-wrapper) feature subset selection in high-dimensional datasets', *Pattern Recognition Letters*, **32**, 701–711, (2011).
- [6] G. Brown, A. Pocock, M. Zhao, and M. Luján, 'Conditional likelihood maximisation: A unifying framework for information theoretic feature selection', *Journal of Machine Learning Research*, **13**, 27–66, (January 2012).
- [7] D.G. Corneil and R.A. Mathon, 'Algorithmic techniques for the generation and analysis of strongly regular graphs and other combinatorial configurations', *Annals of Discrete Mathematics*, **2**, 1–32, (1978).
- [8] T.M. Cover and J.A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*, Wiley-Interscience, 2nd edn., 2006.
- [9] R.G. Cowell, A.P. Dawid, S.L. Lauritzen, and D.J. Spiegelhalter, *Probabilistic Networks and Expert Systems*, Springer, 1999.
- [10] W. Duch, 'Filter methods', in *Feature extraction. Foundations and applications*, eds., I. Guyon, S. Gunn, M. Nikraves, and L.A. Zadeh, volume 207 of *Studies in Fuzziness and Soft Computing*, 89–117, Springer, (2006).
- [11] R.A. Fisher, 'An examination of the different possible solutions of a problem in incomplete blocks', *Annals of Eugenics*, **10**, 52–75, (1940).
- [12] Nir Friedman, Dan Geiger, and Moises Goldszmidt, 'Bayesian network classifiers', *Machine Learning*, **29**(2–3), 131–163, (1997).
- [13] F. V. Jensen, C. Skaanning, and U. Kjærulff, 'The SACSO System for Troubleshooting of Printing Systems', in *In Proceedings of the Seventh Scandinavian Conference on Artificial Intelligence*, (2001).
- [14] F.V. Jensen and T.D. Nielsen, *Bayesian Networks and Decision Graphs*, Springer, 2nd edn., 2007.
- [15] G.H. John, R. Kohavi, and P. Pflieger, 'Irrelevant features and the subset selection problem', in *Machine Learning: Proceedings of the Eleventh International Conference*, pp. 121–129. Morgan Kaufmann, (1994).
- [16] U.B. Kjærulff and A.L. Madsen, *Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis*, Springer, 2nd edn., 2013.
- [17] Vipin Kumar and Sonajharia Minz, 'Feature selection: A literature review', *Smart Computing Review*, **4**(3), 211–229, (2014).
- [18] T.N. Lan, O. Chapelle, J. Weston, and A. Elisseeff, 'Embedded methods', in *Feature extraction. Foundations and applications*, eds., I. Guyon, S. Gunn, M. Nikraves, and L.A. Zadeh, volume 207 of *Studies in Fuzziness and Soft Computing*, 137–165, Springer, (2006).
- [19] J. Li, K. Cheng, S. Wang, F. Morstatter, R.P. Trevino, J. Tang, and H. Liu, 'Feature selection: A data perspective', *arXiv preprint arXiv:1601.07996*, (2016).
- [20] A.L. Madsen, F. Jensen, A. Salmerón, M. Karlsen, H. Langseth, and T. D. Nielsen, 'A new method for vertical parallelisation of TAN learning based on balanced incomplete block designs', in *PGM'2014. Lecture Notes in Artificial Intelligence*, volume 8754, pp. 302–317, (2014).
- [21] A.L. Madsen, F. Jensen, A. Salmerón, H. Langseth, and T.D. Nielsen, 'Parallelization of the PC algorithm', *CAEPIA'2015. Lecture Notes in Artificial Intelligence*, **9422**, 14–24, (2015).
- [22] L. Morán-Fernández, V. Bolón-Canedo, and A. Alonso-Betanzos, 'A time efficient approach for distributed feature selection partitioning by features', *CAEPIA 2015. Lecture Notes in Artificial Intelligence*, **9422**, 245–254, (2015).
- [23] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Series in Representation and Reasoning, Morgan Kaufmann, 1988.
- [24] H. Peng, F. Long, and C. Ding, 'Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**, 1226–1238, (2005).
- [25] C.E. Shannon, 'A mathematical theory of communication', *Bell Systems Technical Journal*, **27**(3), 379–423, (1948).
- [26] D. Stinson, *Combinatorial Designs — Constructions and Analysis*, Springer, 2003.
- [27] K. Takeuchi, 'A table of difference sets generating balanced incomplete block designs', *Review of the International Statistical Institute*, **30**(3), 361–366, (1962).
- [28] H. Yang and J. Moody, 'Data visualization and feature selection: New algorithms for non-Gaussian data', in *Advances in Neural Information Processing Systems*, volume 12, (1999).

Distributed Controllers for Norm Enforcement

Bas Testerink and Mehdi Dastani¹ and Nils Bulling²

Abstract. This paper focuses on computational mechanisms that control the behavior of autonomous systems at runtime without necessarily restricting their autonomy. We build on existing approaches from runtime verification, control automata, and norm-based systems, and define norm-based controllers that enforce norms by modifying system behavior at runtime to make it norm compliant. For many applications, an autonomous system should comply with a set of norms. We extend our approach to a distributed setting, where a set of norm-based controllers jointly modify the runtime behavior of an autonomous system. The norms that a set of norm-based controllers jointly enforce are investigated and characterized in terms of the norms that are enforced by individual norm-based controllers. We show that a set of norm-based controllers is able to modify the runtime behavior of an autonomous system to make it compliant with all norms that the individual norm-based controllers aim at enforcing.

1 Introduction

The emergence of autonomous systems requires mechanisms to monitor and control their behavior in order to ensure system level properties. Norms are a popular candidate for the specification of system level properties and can be seen as standards of behavior that distinguish good and bad behavior (see e.g., [25, 14, 6]). Various languages have been proposed to represent different classes of norms such as state-based norms, action-based norms, temporal norms, and for each class of norms, monitoring and enforcement models have been proposed to detect and control norm violations, respectively (e.g., [8, 1, 11, 5]).

Existing proposals for norm monitoring and enforcement are either based on logical models where norm violations are explained in terms of the satisfiability of a violation formula (i.e. a formula that characterizes violated states or executions) and norm enforcement is explained in terms of model updates [1, 3], or are concerned with practical frameworks for building norm-based systems (e.g., [16, 13, 7]). Most logical approaches to norm enforcement focus on infinite executions and are not concerned with runtime norm enforcement. An exception is the runtime model for norm enforcement proposed in [2]. However, in this work norms are only enforced by halting the system execution before a norm violation occurs. This paper differentiates from earlier work, both by ourselves and others, by focusing on a theoretical analysis of (distributed) runtime norm enforcement rather than an offline analysis on the effects of norm compliance of a system.

In other branches of computer science, such as runtime verification and control automata, the idea of monitoring and control at runtime has been extensively studied, albeit from a different perspective.

The main research problem in these areas is to ensure some system level properties for an untrusted system, also called the target system or the plant, when the system is expected to produce unwanted behavior. In these research areas runtime controllers are introduced to revise the behavior of the untrusted system to ensure the given system level properties. Such a controller reviews the actions that the system produces and may decide per action whether to allow the action, suppress the action or execute extra actions. These controllers have formally been studied, e.g., in [24, 21, 17, 9].

A characteristic feature of norms is that they may be enforced by means of regimentation and sanctioning. A norm can be either regimented in the sense that norm violations are prevented, or sanctioned in the sense that norm violations incur sanctions. Norm enforcement by means of sanctioning respects the autonomy of the controlled system as it allows violations to take place, but may intervene by adding corrective/repair actions. This paper builds on runtime controllers and applies them to norm enforcement. Regimenting and sanctioning controllers are distinguished, depending on whether norm regimentation or norm sanctioning is applied. We will use the term norm-based controller to indicate a controller that is enforcing a norm through regimentation or sanctioning. Our first contribution is to model norm regimentation and sanctioning in a consistent manner with respect to the aforementioned controller models of [21] or [17]. These models allow us to formally analyze regimenting and sanctioning controllers, and investigate the runtime enforcement of norms. A norm-based controller is defined as a computational entity that reviews the actions performed by the target system. If there is a violation about to happen, then the system execution is halted in case of regimentation, or the violating action is allowed and followed by a corrective/repair action in case of sanctioning.

For many applications, a target system should be compliant with a set of norms. This has led to various proposals for distributed architectures for norm enforcement (cf. [22, 18, 23, 28]). The general setup in these architectures is that multiple norm-based control mechanisms are applied concurrently on a target system. The benefits of concurrently applied control mechanisms include the removal of a single point of failure and a possible bottleneck at some central control mechanism. Concurrently applied control mechanisms independently process local observations and communicate them in order to collaborate on the task of enforcing norms. We follow this general architectural setup and use the term distributed norm-based controller to refer to a set of norm-based controllers. Our second contribution is the introduction of a framework that allows us to formally analyze a set of norm-based controllers that enforce their norms concurrently on one and the same target system. We propose the construction of collaborative automata to show that a set of norm-based controllers is able to modify the behavior of a target system to comply with the entire set of norms.

Section 2 discusses the relevant background theory. Section 3 pro-

¹ Utrecht University, The Netherlands, email: {B.J.G.Testerink,M.M.Dastani}@uu.nl

² TU Delft, The Netherlands, email: N.Bulling@tudelft.nl

poses our model for norms and norm enforcement. In Section 4 controller automata are introduced to specify norm-based controllers. In Section 5 collaborative automata are defined and analyzed. Finally Section 6 discusses related approaches and provides concluding remarks.

2 Property Enforcement

Throughout this paper we model the possible actions of a target system by a fixed global, non-empty set of actions \mathcal{A} . We denote with \mathcal{A}^* the finite sequences of elements of \mathcal{A} , referred to as *words*, including the empty word ε . For two words $\alpha, \alpha' \in \mathcal{A}^*$ we denote with $\alpha' \preceq \alpha$ and $\alpha' \prec \alpha$ that α' is a prefix and strict prefix of α , respectively. We assume a given equivalence relation $\sim \subseteq \mathcal{A}^* \times \mathcal{A}^*$, called *similarity relation*, that indicates when two words are semantically equivalent. We use $\alpha_{..i}$ and $\alpha_{j..}$ as the prefix of α up to and including index i and the suffix of α from and including index j , respectively. For two words $\alpha, \alpha' \in \mathcal{A}^*$ we use $\alpha\alpha'$ for the concatenation of α and α' . The next example illustrates the concepts.

Example 1 (Scenario) Consider a file system where an agent can manipulate a file. The possible actions are r (read), w (write), s (save) and b (backup), i.e. $\mathcal{A} = \{r, w, s, b\}$. The agent can attempt to execute these actions in any order and as often as it desires. It might be prescribed by the system designer that the file must be saved in-between any two write actions. That would mean that $rwrsw$ is a good word but $rwrws$ is a bad word. Moreover, the words s and ss might be considered semantically equivalent, thus $s \sim ss$.

We have opted for a simple scenario for clarity and conciseness of the examples. However, we emphasize that the contributions of this paper are applicable to more complex scenarios where concurrent processes perform sequences of actions that might be considered good or bad. Other example scenarios might be for instance electronic markets, social media and conference management systems.

2.1 Properties

We follow [21] and define a property as a set of words that are assumed to represent allowed system behaviors. We impose two constraints on a property. It must contain the empty word and must be closed under the similarity relation \sim .

Definition 1 (Property, P) A property is given by $P \subseteq \mathcal{A}^*$ such that $\varepsilon \in P$ and if $\alpha \in P$ and $\alpha \sim \alpha'$ then also $\alpha' \in P$ for all $\alpha, \alpha' \in \mathcal{A}^*$.

Example 2 (Ex. 1 cont., Property) Let property P contain all words where in-between every two write actions w there is a save action s . Therefore the word $rwrws$ is not in P , whilst $rwrsws$ and rwr are in P .

2.2 Controllers and Enforcement

A controller reviews an input word from \mathcal{A}^* of the target system from left to right and produces some output word from \mathcal{B}^* defined over some action set \mathcal{B} . It can be the case that $\mathcal{A}^* = \mathcal{B}^*$, but also that the controller introduces new actions that cannot be produced by the target system. A controller can deterministically revise the word into a new word. Thus, a controller can be seen as a mechanism to revise action executions at run-time. As such, a decision to previously output a word cannot be reverted in the future. This is captured by the formal definition below.

Definition 2 (Controller, m) A controller (function) over $(\mathcal{A}, \mathcal{B})$ is given by a function $m : \mathcal{A}^* \rightarrow \mathcal{B}^*$ such that if $\alpha' \preceq \alpha$ then $m(\alpha') \preceq m(\alpha)$ for all $\alpha, \alpha' \in \mathcal{A}^*$.

In the following, we assume, if not said otherwise, that a property P over $\mathcal{A}^* \cup \mathcal{B}^*$, a controller m over $(\mathcal{A}, \mathcal{B})$, and a word $\alpha \in \mathcal{A}^*$ are given. We simply say a controller over \mathcal{A} whenever $\mathcal{A} = \mathcal{B}$.

Example 3 (Ex. 2 cont., Controller) Let m be a controller over \mathcal{A} . Consider $\alpha = rwr$ and assume $m(\alpha) = \alpha' = rwsr$. This implies that given $\alpha'' = rwrws$ it cannot be the case that $m(\alpha'') = \alpha''$, because $\alpha \preceq \alpha''$ but $m(\alpha) \not\preceq m(\alpha'')$ (i.e. $rwsr \not\preceq rwrws$). Intuitively, when m is reviewing the word α'' and reaches the second read action, then it inserts a save action, because $m(rwr) = rwsr$. Then, when it reviews the s action in α'' it cannot retract the inserted save action. $rwsr$ has to be a prefix of the word that m returns given α'' .

The intended purpose of a controller is that it revises all words from \mathcal{A}^* such that the revisions satisfy some desired property P (i.e. $m(\alpha) \in P$ for all $\alpha \in \mathcal{A}^*$). This is called *soundness* [21]. A trivial way to achieve *soundness* is to revise any word to the empty word. However, we want a controller to not only ensure correct behavior, but also to be *transparent* in the sense that the target system's behavior is not meaningfully altered if there is no violation of the property [17]. In other words, if a word already satisfies the property, then it should remain unchanged or at least be revised to a similar word $wrt. \sim$. Otherwise the word is mapped to its longest prefix satisfying the property.

Definition 3 (Longest Correct Prefix) The longest correct prefix of $\alpha \in \mathcal{A}^* \cup \mathcal{B}^*$ wrt. P is $\alpha' \in P$ such that $\alpha' \preceq \alpha$ and for all $\alpha'' \in \mathcal{A}^* \cup \mathcal{B}^*$ with $\alpha' \prec \alpha'' \preceq \alpha$ it holds that $\alpha'' \notin P$.

Next we recall the definition of precise enforcement from [21]. If a controller precisely enforces a property, then any action in any correct input word must (immediately) be allowed.

Definition 4 (Precise Enforcement, [21]) m precisely enforces P if and only if for all $\alpha \in \mathcal{A}^*$ the following holds:

1. $m(\alpha) \in P$ and
2. if $\alpha \in P$ then for all $\alpha' \in \mathcal{A}^*$ with $\alpha' \preceq \alpha$ it holds that $m(\alpha') = \alpha'$.

Example 4 (Ex. 1 cont., Precise Enforcement) Let P be the property that contains all words in which each occurrence of w is preceded by b (before each write operation a backup must be performed). Let m be the controller over \mathcal{A} that revises word $\alpha \in \mathcal{A}^*$ as follows: if there exists a minimal index i such that $\alpha[i] \neq b$ and $\alpha[i+1] = w$ then $m(\alpha) = \alpha_{..i}$; otherwise $m(\alpha) = \alpha$. That is, if the agent tries to write without performing a backup, then m will suppress the write action as well as all subsequent actions. m precisely enforces P .

It is well known that exactly the class of safety properties from the safety-progress classification of [10] can be precisely enforced [21], where property P over \mathcal{A}^* is a *safety property* if for all $\alpha \in \mathcal{A}^*$ with $\alpha \notin P$ it holds that $\alpha \cdot \alpha' \notin P$ for all $\alpha' \in \mathcal{A}^*$ [20]. We observe that if a controller m over $(\mathcal{A}, \mathcal{B})$ precisely enforces P over $\mathcal{A}^* \cup \mathcal{B}^*$ then it will revise any word $\alpha \in \mathcal{A}^*$ to its longest correct prefix wrt. P .

It might be the case, however, that a controller withholds or inserts some actions upon revising a word without changing its semantic meaning. For those types of controllers the notion of effective

enforcement has been introduced [21]. A controller that effectively enforces some property will rewrite any correct word to a similar word according to \sim .

Definition 5 (Effective Enforcement, [21]) A controller m effectively enforces P wrt. \sim iff for all $\alpha \in \mathcal{A}^*$ it holds that: (1) $m(\alpha) \in P$ and (2) $\alpha \in P$ implies $\alpha \sim m(\alpha)$.

Example 5 (Ex. 2 cont., Effective Enforcement) Let m be a controller over \mathcal{A} such that for a given word any occurrence of w is revised to ws (i.e. a save is forced after each write action). We could argue that in our scenario saving the file multiple times between write operations equals saving the file once between write operations. This can be modelled by \sim . Then, m effectively enforces P wrt. \sim . Note that if we would take equality = as similarity relation then m would neither precisely enforce P nor would it effectively enforce P wrt. =.

Note that effective enforcement wrt. some property by a controller m does not require that all prefixes of correct words are rewritten by m to themselves as it was the case for perfect enforcement. Also note that if m precisely enforces P , then it also effectively enforces P up to any similarity relation. If m effectively enforces a safety property P wrt. =, then it also precisely enforces P . Properties for which violating words may not have a longest correct prefix cannot be effectively enforced by any controller [17].

3 Norms

Norms are a means to specify desirable behavior. Given a word representing a behavior, a norm might be violated multiple times. For instance norms that are represented as conditional obligations with deadlines (cf. [8, 4]) are violated each time that the deadline occurs and the obligation has not been satisfied since the last time the condition held. A controller that enforces a norm should either prevent violations, or sanction violations, which is known as regimentation or sanctioning, respectively.

For runtime control it is required that the violation of a norm should be detectable after a finite amount of actions. We therefore represent all violations of a norm as a set of words such that a violation occurs necessarily at the last action of those words, but possibly also earlier. A norm itself is represented as a tuple that contains the violations of the norm and the sanction that should be applied after a violation occurs, in case the norm's violations should be sanctioned.

We assume a global and fixed set of actions \mathcal{S} that can be used as sanctions and use $\mathcal{A}_{\mathcal{S}}$ as shorthand for $\mathcal{A} \cup \mathcal{S}$. We also assume that the target system will not by itself try to execute sanctions. Hence if \mathcal{A} are all actions that the target system may try to execute, then the set of possible sanctions \mathcal{S} is disjoint from \mathcal{A} . Another consequence is that we define a norm's violations as a subset of \mathcal{A}^* . Finally we assume that for two different sanctions $s, s' \in \mathcal{S}$ there is no two words $\alpha, \alpha' \in \mathcal{A}_{\mathcal{S}}$ such that $\alpha s \alpha' \sim \alpha s' \alpha'$.

Definition 6 (Norm, η) A norm is represented as a tuple $\eta = (V, s)$ where $V \subseteq \mathcal{A}^*$ is the set of violations and $s \in \mathcal{S}$ is a sanction. Furthermore if $\alpha \in V$ and $\alpha \sim \alpha'$ then also $\alpha' \in V$ for all $\alpha, \alpha' \in \mathcal{A}^*$. A word $\alpha \in \mathcal{A}^*$ is a violation of η iff $\alpha \in V$. Moreover, α violates η if there is a subword $\alpha' \preceq \alpha$ that is a violation of η .

It is important to note that our representation of norms as a set of violating behaviors with a sanction is general and covers many other possible representations. For instance, norms for which violations are regular, in the sense that they can be specified by a formula (e.g. by

using linear temporal logic), as well as norm for which violations are irregular are covered. For the irregular case, the violations of a norm might be specified by an enumeration of bad behaviors, e.g. a corpus of empirically collected bad behaviors or practices.

We stress that given our representation of norm violations, each norm violation is a word that violates the norm, but not every violating word is a norm violation. Next we need to define what it means for a word to be compliant with a norm. This is the case if the word does not violate the norm, or if each norm violation is immediately followed by the norm's sanction. Because sanctions can occur in compliant words we have that the set of compliant words for a norm is a subset of $\mathcal{A}_{\mathcal{S}}^*$. To define norm compliance we use $\alpha^{-\mathcal{S}}$ to refer to the word α' which equals α but in which all sanction actions from \mathcal{S} are removed.

Definition 7 (Compliant words, P_{η}) Let $\eta = (V, s)$. The set of η -compliant words $P_{\eta} \subseteq \mathcal{A}_{\mathcal{S}}^*$ is the set of words $\alpha \in \mathcal{A}_{\mathcal{S}}^*$ such that $\alpha^{-\mathcal{S}} \notin V$ or $\alpha[i+1] = s$ for all $i \in \{1, \dots, |\alpha|-1\}$ where $\alpha_{\dots i}^{-\mathcal{S}} \in V$.

Note that aside from words with correctly sanctioned violations, P_{η} also includes all words $\alpha \in \mathcal{A}^*$ that do not violate a norm η . We observe a connection between norms and properties. For a norm η the set of compliant words P_{η} is a property, because: (1) ε contains no violation of a norm η and hence must be in P_{η} and (2) if a compliant word is similar to another compliant word, then both are compliant words, and hence both are in P_{η} .

Example 6 (Norm) We assume that the set of sanctions is $\mathcal{S} = \{u\}$ where u stands for "undo the last write action that occurred". In other scenarios we may use more traditional sanctions such as fines or warnings. We will specify a norm that says that the file has to be saved between writes, and the sanction of which is the undo action. The set $V \subseteq \mathcal{A}^*$ contains all words $\alpha \in \mathcal{A}^*$ such that the final action is a write action and no save action has occurred since the last write action. Consider the norm $\eta = (V, u)$. The word $\alpha = wswrws$ violates η , because the prefix $wswrw$ of α is a violation of η . The η -compliant words P_{η} are those that do not violate η , such as $wswrsws$, or those where the sanction is applied after each violation, such as $wswrwus$. Note that the word $wswrwus$ still violates η , even though the sanction was applied.

3.1 Regimenting Controller

A regimenting controller for a norm prevents norm violations. Such a controller halts the system execution if it is about to violate a norm.

Definition 8 (Regimenting Controller) Let η be a norm and m be a controller over $\mathcal{A}_{\mathcal{S}}$. m is a regimenting controller for η iff for all $\alpha \in \mathcal{A}^*$ it holds that $m(\alpha) = \alpha'$ where α' is the longest prefix of α that does not violate η .

Example 7 (Ex. 6 Cont., Regimenting Controller) Let m_r be a regimenting controller for the norm η from Example 6. We can have for example that $m_r(wswrws) = wswr$. The norm's violation is prevented by blocking further execution when the norm is about to be violated by the third write action.

We shall now establish a connection between regimenting controllers and precise enforcement. We first observe that for a norm η the set of η -compliant words P_{η} is not in general a safety property. Hence precisely enforcing P_{η} is not in general possible. Consider the norm $\eta = (V, u)$ from Example 6 and the word ww . As ww

contains an unsanctioned violation it is not in P_η . However, wuu is in P_η , as all violations are properly sanctioned. Hence, an incorrect word might be extended to a correct one showing that P_η is not a safety property. The subset of P_η that contains no norm violations is however always a safety property.

Proposition 1 *Let η be a norm and $P \subseteq P_\eta$ be the set of all words not violating η . Then, P is a safety property.*

Proof: *If a word $\alpha \in \mathcal{A}_S^*$ violates η , then there is a prefix $\alpha' \preceq \alpha$ such that $\alpha' \in V$. It is impossible to add an extension to α such that α' is not a prefix anymore, hence any extension would be violating the norm as well and hence not be in P .*

A controller is a regimenting controller for a norm iff it precisely enforces the property that contains all words without norm violations. This means that we can apply security monitors [24]/truncation automata [21] for the regimentation of norms and make use of formal results of these techniques.

Proposition 2 *Let m be a controller over \mathcal{A}_S , $\eta = (V, s)$ be a norm and $P \subseteq P_\eta$ be the set of all words not violating η . Then, m is a regimenting controller for η iff m precisely enforces P .*

Proof: *Let $\alpha \in \mathcal{A}_S^*$ be an arbitrary word. By definition m can only revise a word α to its longest prefix $\alpha' \preceq \alpha$ that contains no violation of η . As α' has no violations of η we have that $\alpha' \in P$, hence $\forall \alpha \in \mathcal{A}_S^* : m(\alpha) \in P$, which is required for precise enforcement. If $\alpha \in P$ then all prefixes of α are in P . If a word contains no norm violations then it is its own longest correct prefix given P , and hence is mapped to itself by m . This holds for all prefixes of α , therefore: for all $\alpha' \in \mathcal{A}_S^*$ with $\alpha' \preceq \alpha$ it holds that $m(\alpha') = \alpha'$.*

For the other direction, if m precisely enforces P then each word in P is mapped to itself, and hence is its own longest correct prefix. If a word is not in P then it is rewritten to its longest correct prefix, which in this case is the longest prefix such that η is not violated. This matches the definition of a regimenting controller.

3.2 Sanctioning Controller

A sanctioning controller for a norm revises a word by inserting a sanction after each violation of the norm. Hence it will make any word a norm compliant one, but does not prevent the norm violation like a regimenting controller. As before, we use α^{-S} to refer to the result of removing all sanctions from α .

Definition 9 (Sanctioning Controller) *Let $\eta = (V, s)$ be a norm and m be a controller over \mathcal{A}_S . m is a sanctioning controller for η iff for all $\alpha \in \mathcal{A}_S^*$ it holds that $m(\alpha) \in P_\eta$ and $m(\alpha)^{-S} = \alpha^{-S}$.*

Example 8 (Ex. 6 Cont., Sanctioning Controller) *Let η be the norm from Example 6 and m_s be a sanctioning controller for η . A revision of m_s is $m_s(wswrws) = wswrwus$. The norm's violation is sanctioned by undoing the last write action when the violation occurred.*

We shall now establish the connection between sanctioning controllers and effective enforcement. Note first that a controller that effectively enforces P_η for some norm η is not necessarily a sanctioning controller. If m effectively enforces P_η then it is allowed that for a word $\alpha \notin P_\eta$ it holds that $m(\alpha) = \alpha'$ where α' is the longest correct prefix of α in P_η . However, the definition of sanctioning controllers requires that the controller injects sanctions which m does not do.

A sanctioning controller can possibly duplicate sanctions if they already occur in an input word. If a norm η has a sanction s which may be duplicated in any word in which s occurs without changing the word in a meaningful way wrt. \sim , then a sanctioning controller for a norm η effectively enforces P_η . This means that edit automata [21] can be used to implement sanctioning controllers and we can make use of results in that area to analyze sanctioning controllers.

Proposition 3 *Let m be a controller over \mathcal{A}_S , η be a norm, and \sim be the identity relation with the constraint that for all $\alpha, \alpha' \in \mathcal{A}_S^*$ and all $s \in S$ it holds that $\alpha s \alpha' \sim \alpha s \alpha'$. If m is a sanctioning controller for η then m effectively enforces P_η .*

Proof: *Let $\alpha \in \mathcal{A}_S^*$ be an arbitrary word and $m(\alpha) = \alpha'$. The definition of a sanctioning controller ensures that each violation in α is followed by s . Hence, for all $\alpha \in \mathcal{A}_S^*$ it holds that $m(\alpha) \in P_\eta$; the first constraint of effective enforcement. Second, if $\alpha \in P_\eta$ then any occurring violation of η in α is sanctioned. m inserts sanctions after each violation, even if a sanction already follows, hence duplications may occur in $m(\alpha)$ if sanctions already occur in α . But the duplication of a sanction was assumed to not change the word in a meaningful way wrt. \sim . Hence for all $\alpha \in P_\eta$ we have that $m(\alpha) \sim \alpha$; the second constraint of effective enforcement.*

4 Controller Automata

We introduce controller automata which are a formal tool to model runtime controllers in more detail. A controller automaton is essentially the same as an edit automaton introduced in [21]. The difference is of a syntactic rather than conceptual nature: they are equally expressive. Controller automata are labeled transition systems over some input alphabet \mathcal{A} and an output alphabet \mathcal{B} . Such an automaton makes a transition from one state to another state by reviewing an action and performing a revision. A revision is given by a/X where $a \in \mathcal{A} \cup \mathcal{B} \cup \{\varepsilon\}$ is an output action (or the empty word) to which an input action is revised and $X \in \{A, I, S, L\}$ is the name of the revision operation, where A stands for allow, I for insert, S for suppress and L for loop. Suppose some controller automaton is given the input word $\alpha = a_1 a_2 \dots a_k$. It reviews the input from left to right starting at a_1 . Then upon reviewing a_i , $i \in \{1, \dots, k\}$: a_i/A is read as “ a_i is allowed, continue with $\alpha_{i+1}..$ ”, ε/S is read as “ a_i is suppressed, continue with $\alpha_{i+1}..$ ”, a'/I is read as “ a' is inserted in the output word, keep reviewing $\alpha_i..$ ”, and ε/L is read as “do nothing, keep reviewing $\alpha_i..$ ”. The minor difference to edit automata is the loop revision operator (as first class citizen in the revision set). This is required for the distributed setting in the next section. We also omit the “halt” revision present in edit automata. A halt operation can be modeled by a special sink state that suppresses any action with a reflexive transition. We recall the definition of edit automata [21] with minor adjustments required for our setting. We require that for each possible input word the controller automaton halts at some point, which guarantees that the output word is always finite. This requirement is met if there is no transition in the controller automaton such that it can infinitely apply the insert revision. In practice this requirement is rarely limiting as control mechanisms are meant to modify executions, and not take over the flow of execution.

Definition 10 (Controller Automaton, c) *A controller automaton is a tuple $c = (\mathcal{A}, \mathcal{B}, Q, q_0, \delta)$ consisting of an input alphabet \mathcal{A} and an output alphabet \mathcal{B} , a countable set of (control) states Q , an initial state $q_0 \in Q$ and a transition function $\delta : Q \times \mathcal{A} \rightarrow \text{Rev} \times Q$ where $\text{Rev} = \{a/X \mid a \in \mathcal{A} \cup \mathcal{B}, X \in \{A, I\}\} \cup \{\varepsilon/S, \varepsilon/L\}$*

is the set of revisions. Moreover, we require that there is no infinite sequence of control states $q_1 q_2, \dots$ such that $\delta(q_1, a) = (\alpha/I, q_2)$, $\delta(q_2, a) = (\alpha'/I, q_3)$, etc.

In the following we assume, if not said otherwise, that a controller $c = (\mathcal{A}, \mathcal{B}, Q, q_0, \delta)$ is given. The operational semantics describes how the automaton behaves on an input word.

Definition 11 (Operational semantics) A configuration of c is a tuple $(\alpha, q) \in \mathcal{A}^* \times Q$ where α represents the input word that remains to be reviewed and q is the current control state. The operational semantics is defined by the following transition rule:

$$\frac{\delta(q, a) = (r, q')}{(a\alpha, q) \xrightarrow{r}_c (\alpha', q')} \quad (\text{Controller Transition})$$

where $\alpha' = \alpha$ if $r = a/A$ or $r = \varepsilon/S$, otherwise $\alpha' = a\alpha$. We write $\text{ctrl}_c(\alpha) = \alpha_1 \alpha_2 \dots \alpha_n$ iff controller c can make the transitions $(\alpha, q_0) \xrightarrow{\alpha_1/X_1}_c (\alpha', q_1) \xrightarrow{\alpha_2/X_2}_c \dots \xrightarrow{\alpha_n/X_n}_c (\varepsilon, q_n)$. Often, we also identify ctrl_c with c .

A transition $(a\alpha, q) \xrightarrow{r}_c (\alpha', q')$ represents that in state q , when action a is being reviewed, a transition to state q' takes place whilst revision r is executed, and the automaton continues reviewing the input α' . Note that for each $\alpha \in \mathcal{A}^*$ there is exactly one possible output word provided by $\text{ctrl}_c(\alpha)$.

Proposition 4 For every controller automaton c , we have that $\text{ctrl}_c : \mathcal{A}^* \rightarrow \mathcal{B}^*$ is a controller.

Given this result we will also say that “ c enforces...” when meaning that “ ctrl_c enforces...”, etc.

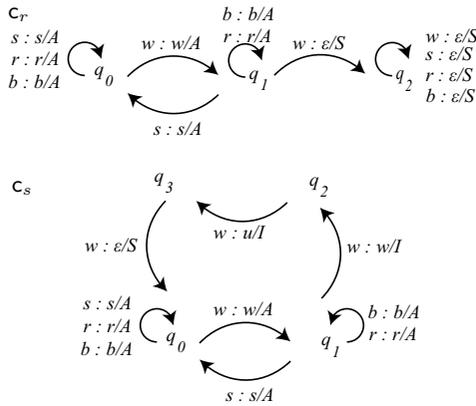


Figure 1. Top: controller automaton implementing m_r from Example 7, bottom: controller automaton implementing m_s from Example 8. A

transition from $q \xrightarrow{a:\alpha/X} q'$ indicates that upon reviewing action a in state q the controller transitions to q' whilst executing revision α/X .

Example 9 (Ex. 8 Cont., Controller automata) In Figure 1 two example controller automata c_r (top) c_s (bottom) are shown. We have that ctrl_{c_r} and ctrl_{c_s} are the controllers m_r and m_s from Examples 7 and 8, respectively. According to definition 10 states q_2 and q_3 in the automaton c_s should have transitions for each action. We omitted some because a transition that inserts an action upon reviewing w will not consume w , hence the next transition must be triggered by w again. We also omitted transitions for u as we assume that the target system will not produce this action.

The connection between regimenting and sanctioning controllers for norms and controller automata is that any regimenting or sanctioning controller can be implemented by a controller automaton.

Proposition 5 Let η be a norm and controller m be a regimenting or sanctioning controller for η . Then, there is a controller automaton c such that $\text{ctrl}_c = m$.

Proof (sketch): Recall from Proposition 2 that a regimenting controller for a norm η precisely enforces the set of words $P \subseteq P_\eta$ that do not violate η . A controller automaton is as expressive as an edit automaton. For edit automata it is shown that they can specify precisely enforcing controllers [21]. Therefore a regimenting controller for P can be implemented by an edit automaton and hence by a controller automaton. A controller automaton $c = (\mathcal{A}_S, \mathcal{A}_S, Q, q_0, \delta)$ such that $\text{ctrl}_c = m$ for a sanctioning controller m for $\eta = (V, s)$ can be constructed as follows: (1) for each word $\alpha \in \mathcal{A}_S^*$ assign a new state q in Q and for ε that state is q_0 , (2) for each action $a \in \mathcal{A}_S$ and word $\alpha \in \mathcal{A}_S^*$ let q and q' be the states belonging to α and $a\alpha$ respectively, define $\delta(q, a) = (a/A, q')$ if $a\alpha \notin V$, otherwise make two new states q_1, q_2 , add them to Q , and define: $\delta(q, a) = (a/I, q_1)$, $\delta(q_1, a) = (s/I, q_2)$, $\delta(q_2, a) = (\varepsilon/S, q')$. Note that by this construction Q is countably infinite.

5 Distributed Controllers

For many applications it is required that multiple norms are enforced. This can be achieved by deploying a distributed controller that consists of a set of concurrently applied individual controllers that collectively enforce norms. In this section we discuss collaborative automata that represent distributed controllers. It is important to note that a collaborative automaton is not an entity that has to be implemented alongside the individual controllers, but it is merely a representation of how the individual controllers are synchronized. Also, we assume here the extreme case where there is full synchronization among controllers and the target system. This is not always a requirement however when one implements the controllers that make up a collaborative automaton. It must be ensured, however, that the output of the collaborative automaton contains only norm compliant words, or restricts input words to words without any norm violations, for any norm. If multiple controllers, each of which is a sanctioning or regimenting controller for one of the norms, are applied concurrently on the same target system, then the compatibility of controllers must be ensured. For instance, when reviewing some action, some controllers may propose to suppress that action whilst other controllers propose to allow that action. A conflict resolution mechanism should in those cases decide which of the proposed revisions is applied. Note that this conflict resolution is on the level of action revisions, and not on a normative level. It is possible to have two controllers as part of a collaborative controller which enforce norms that cannot both be complied with. In case of regimentation it would then be ensured that at some point the controller halts the target system. In case of sanctioning it would mean that there will be a point where the sanction is applied. For a discussion on normative conflict, see for instance the work by Vasconcelos et al. [29].

Consider again the controller automata from Example 9 and Figure 1. If they simultaneously review ww then the revisions that the top controller automaton c_r executes are first w/A and then ε/S , whereas the bottom controller automaton c_s executes w/A , w/I , u/I and then ε/S . Both controller automata agree on the initial allow revision (w/A) but then execute different revisions. For this purpose, we introduce a *selection function* that decides which controller

automaton's revisions are performed. The main challenge when constructing a collaborative automaton is to define an appropriate selection function. Also note that the number of revisions for controller automata may not be equal as in the example above. This happens when some of the controllers perform insert revisions and thus do not move to the next input action, whilst other automata may perform allow or suppress revisions so that they do move on to next action. We make use of loop revisions to maintain synchronization. If some automata can perform an insert revision whilst others do not, then those others have to loop until all automata are ready with insertions and can allow or suppress the action that is being reviewed. In our example, the top controller automaton c_r should be forced to loop when the second action w is under review until c_s is at state q_3 and, just like c_r , is ready to process w by an allow or suppress revision.

5.1 Collaborative Automaton

A collaborative automaton models the collaboration between a set of concurrent controller automata. A collaborative automaton has a state space and transition function that given a state and action returns a label and next state. A transition is labeled by a vector of revisions, one for each controller automaton. A selection function can pick a revision from this vector for each transition. This selection can be interpreted as the result of coordination between controllers to decide upon a revision. The state space and transition function of a collaborative automaton are constructed from the individual controller automata. The state space is essentially the Cartesian product of the state spaces of the controller automata where we need to duplicate the local states of each controller automaton to allow them to loop if necessary. The duplicate of a state q is denoted by \hat{q} . In the following, for a state $x \in \{q, \hat{q}\}$ we write \bar{x} to refer to q , i.e., the overline removes the hat annotation. For a set of controller automata, the collaborative state is hence a snapshot of the states in which each controller automaton is at a certain moment in time. If a controller automaton's state in a collaborative state is \hat{q} then this can be interpreted as that the controller automaton is 'on hold'. The combination of all initial states of the controller automata is the initial state of the collaborative automaton.

We say a controller automaton proposes a certain revision given a collaborative state and action when, given the automaton's local state in the collaborative state and the action, the controller automaton can make a transition with that revision as a label. We say that the revisions in a label from a collaborative transition are assigned to the controller automata. However, in an application these revisions are the result of synchronization between controller applications. A transition label from one state of the collaborative automaton to the next for a given action is constructed as follows:

1. If there is a controller automaton that proposes an insert, then the collaborative automaton assigns to each controller automaton that proposes an insert their proposal. The other controller automata, which either propose an allow or suppress revision, are assigned loop revisions by the collaborative automaton.
2. If no controller automaton proposes an insert, then all controller automata propose either an allow or suppress revision. The collaborative automaton assigns to each controller automaton the revisions that they themselves propose.

The next state after a transition is determined by the label. Assume some given action. If a controller automaton is in state q in some collaborative state, and its assigned revisions given the collabor-

orative state and action is the loop revision, then its next state becomes \hat{q} in the next collaborative state. If, however, another revision was assigned to the controller automaton, then it must be a label of a transition that the controller automaton can make upon reading the action. The resulting state of that transition becomes the state of the controller automaton in the next collaborative state. We give the formal definition of a collaborative automaton. Recall that Rev is the set of possible revisions.

Definition 12 (Collaborative Automaton, C) Let $M = \{c_1, \dots, c_k\}$ be a set of controller automata such that $c_i = (\mathcal{A}, \mathcal{B}, Q^i, q_0^i, \delta^i)$ and $\hat{Q}^i = Q^i \cup \{\hat{q} \mid q \in Q^i\}$. A collaborative automaton C over M is a labeled transition system $C = (\mathcal{A}, \mathcal{B}, Q, q_0, \Delta, \sigma)$, where $Q = \hat{Q}^1 \times \dots \times \hat{Q}^k$ is the set of collaborative states, $q_0 = (q_0^1, \dots, q_0^k)$ is the initial collaborative state, and $\Delta : Q \times \mathcal{A} \rightarrow \text{Rev}^k \times Q$ is the transition function defined as follows. $\Delta((x_1, \dots, x_k), a) = ((r_1, \dots, r_k), (y_1, \dots, y_k))$ if, and only if, it holds that:

1. If there is an $i \in \{1, \dots, k\}$ such that $\delta^i(\bar{x}_i, a) = (a'/I, q')$ then for all $j \in \{1, \dots, k\}$ it holds that:

$$\begin{cases} r_j = a'/I, y_j = q' & \text{if } \delta^j(\bar{x}_j, a) = (a'/I, q') \\ r_j = \varepsilon/L, y_j = \hat{x}_j & \text{otherwise;} \end{cases}$$

2. otherwise, $\delta^j(\bar{x}_j, a) = (r_j, y_j)$, for all $j \in \{1, \dots, k\}$.

Finally, σ is a function $Q \times \mathcal{A} \rightarrow \text{Rev}$ such that $\sigma(q, a) \in \{r_1, \dots, r_k\}$ where $\Delta(q, a) = ((r_1, \dots, r_k), q')$. The function is called the selection function of C .

Each transition step of the collaborative automaton is labeled with a vector of revisions. We note that a collaborative automaton is completely specified by M apart from the selection function. In the following we assume that $C = (\mathcal{A}, \mathcal{B}, Q, q_0, \Delta, \sigma)$ is given as in the definition above.

Definition 13 (Operational Semantics) A configuration of C is a tuple $(\alpha, q) \in \mathcal{A}^* \times Q$ where $\alpha \in \mathcal{A}^*$ is the input word that remains to be reviewed and q is the current state of C . The operational semantics of a collaborative automaton is defined by the following transition rule:

$$\frac{\Delta(q, a) = ((r_1, \dots, r_k), q')}{(a\alpha, q) \xrightarrow{\sigma(q, a)}_C (\alpha', q')} \quad (\text{Col. Transition})$$

where $\alpha' = a\alpha$ if $\sigma(q, a) \in \{a'/I \mid a' \in \mathcal{A} \cup \mathcal{B}\} \cup \{\varepsilon/L\}$, and $\alpha' = \alpha$ otherwise. As before, we write $\text{ctrl}_C(\alpha) = \alpha_1\alpha_2 \dots \alpha_n$ iff C can make the transitions $(\alpha, q_0) \xrightarrow{\alpha_1/X_1}_C (\alpha', q_1) \xrightarrow{\alpha_2/X_2}_C \dots \xrightarrow{\alpha_n/X_n}_C (\varepsilon, q_n)$. Again, we often identify ctrl_C with C .

Proposition 6 For every collaborative automaton C , we have that $\text{ctrl}_C : \mathcal{A}^* \rightarrow \mathcal{B}^*$ is a controller.

Example 10 (Ex. 9 Cont., Collaborative Automaton) In Figure 2 two controller automata c_1 and c_2 with $c_i = (\mathcal{A}_S, \mathcal{A}_S, Q^i, q_0^i, \delta^i)$, are shown. Controller c_1 is controller c_s from Example 9 which is a sanctioning controller for $\eta_1 = (V_1, u)$. Controller automaton c_2 is an implementation of a sanctioning controller for the norm $\eta_2 = (V_2, u)$ where V_2 contains all words where a write action is not immediately preceded by a backup action. In Figure 2 there is also an example collaborative automaton $C = (\mathcal{A}_S, \mathcal{A}_S, Q, q_0, \Delta, \sigma)$ over

Now, suppose $\alpha \in \mathcal{A}^*$. We have that $c_i(\alpha) = \alpha^i \in P_i$ for all i . Let $\alpha' = a'_1 \dots a'_j$ be the longest common prefix of all the $c_i(\alpha)$. By precise enforcement, $\alpha' \in P_\cap$. On input α no controller automaton can suppress for the first j transitions. If $\alpha = \alpha'$ then also $\text{ctrl}_C(\alpha) = \alpha' \in P_\cap$, because no suppression occurs. If $\alpha' \prec \alpha$ then some controller automaton must suppress the next input action a_{j+1} . Moreover, this automaton will keep on suppressing actions from that moment on, and therefore so will C , showing that $\text{ctrl}_C(\alpha) = \alpha' \in P_\cap$. Secondly, suppose that $\alpha \in P_\cap$. Then for each i , also $\alpha \in P_i$ and thus for all $\alpha' \preceq \alpha$ it holds that $c_i(\alpha') = \alpha'$; in particular, c_i allows all actions of α' . Thus, by definition of the selection function, $\text{ctrl}_C(\alpha') = \alpha'$.

From Proposition 7 in combination with Proposition 2 it follows that a set of regimenting controllers $\{c_1, \dots, c_k\}$ for the norms $\eta_1 = (V_1, s_1), \dots, \eta_k = (V_k, s_k)$ can be combined to a collaborative automaton that prevents any violation of a norm η_i , $i \in \{1, \dots, k\}$. Note that this is also the same as stating that the collaborative automaton specifies a regimenting controller for a norm (V_\cup, s) , where $V_\cup = \bigcup_{i \in \{1, \dots, k\}} V_i$ and $s \in \mathcal{S}$ is an arbitrary sanction (because sanctions play no role in regimentation). For simplicity we again assume that the controllers do not use insert revisions.

Theorem 1 Let $M = \{c_1, \dots, c_k\}$ be a set of controller automata over \mathcal{A}_S which implement regimenting controllers for norms $\eta_1 = (V_1, s_1), \dots, \eta_k = (V_k, s_k)$ and do not use insert revisions, and $P_\cap = \bigcap_{i \in \{1, \dots, k\}} P_i$, where $P_i \subseteq P_{\eta_i}$ are all words that do not violate η_i . Then, there exists a collaborative automaton C over M such that $\text{ctrl}_C(\alpha) \in P_\cap$ for each $\alpha \in \mathcal{A}_S^*$.

Proof (sketch): Recall from Proposition 7 that each controller c_i is precisely enforcing the property P_i . A regimenting controller c_i also rewrites a word $\alpha \in \mathcal{A}_S^*$ to its longest correct prefix given P_i .

Therefore $M = \{c_1, \dots, c_k\}$ is a set of controller automata that do not perform insert revisions where c_i precisely enforces property P_i such that if $\alpha \notin P_i$ then c_i rewrites α to its longest correct prefix wrt. P_i . Following Proposition 7 there exists a collaborative automaton C over M such that ctrl_C precisely enforces P_\cap . Hence for that controller C we have $\text{ctrl}_C(\alpha) \in P_\cap$ for each $\alpha \in \mathcal{A}_S^*$.

For a set of sanctioning controllers we may run into a conflict if two controllers propose to insert a different sanction at the same time. The selection function can only select one of those sanctions. This issue will not occur if for any two sanctioning controllers for norms $\eta_1 = (V_1, s_1)$ and $\eta_2 = (V_2, s_2)$ there is no situation where they both propose to insert a sanction (i.e. $V_1 \cap V_2 = \emptyset$) or if the sanctions are the same. The latter case may occur if for instance the same sanction is used for all norms, such as a warning. Also, one can model different sanction procedures with the same symbol, and only use different symbols if the procedures are incompatible. If no two different sanction actions can be inserted concurrently, then a collaborative automaton can be constructed such that each input word is revised to a η_i compliant word for each $i \in \{1, \dots, k\}$.

Theorem 2 Assume that \sim is the identity relation with the constraint that $\forall s \in \mathcal{S}, \alpha \in \mathcal{A}_S^*, \alpha' \in \mathcal{A}_S^*: \alpha s \alpha' \sim \alpha s s \alpha'$. Let $M = \{c_1, \dots, c_k\}$ be a set of controller automata over \mathcal{A}_S which implement sanctioning controllers for norms $\eta_1 = (V_1, s_1), \dots, \eta_k = (V_k, s_k)$ and $P_\cap = \bigcap_{i \in \{1, \dots, k\}} P_{\eta_i}$. Then, there exists a collaborative automaton C over M such that $\text{ctrl}_C(\alpha) \in P_\cap$ for each $\alpha \in \mathcal{A}_S^*$ iff for each $i, j \in \{1, \dots, k\}$ if $V_i \cap V_j \neq \emptyset$ then $s_i = s_j$.

Proof (sketch): For simplicity we assume that a controller automaton uses a/A if possible and not the equivalent revisions a/I

followed by ε/S . In that case, note that the controller automata will only propose allow revisions, unless a violation is detected upon reviewing an action a , in which case a word of revisions equivalent to $a/I, s/I, \varepsilon/S$ is executed, where $s \in \mathcal{S}$ is a sanction. Under this assumption let $C = (\mathcal{A}_S, \mathcal{A}_S, Q, q_0, \Delta, \sigma)$ be the collaborative automaton. We define the following selection function σ :

$$\sigma(q, a) = \begin{cases} \varepsilon/S & \text{if } \exists i \in \{1, \dots, k\} : r_i = \varepsilon/S \\ a'/I & \text{if } \exists i \in \{1, \dots, k\} : r_i = a'/I \\ a/A & \text{otherwise} \end{cases}$$

where $\Delta(a, q) = ((r_1, \dots, r_k), q')$.

Consider an arbitrary word $\alpha \in P_\cap$ (such as ε), $q \in Q$ such that q is reached after reviewing α , and action $a \in \mathcal{A}_S$. If $\alpha a \in P_\cap$ then $\alpha a \in P_{\eta_i}$ for each $i \in \{1, \dots, k\}$, and hence each controller will allow the action. Therefore $\sigma(\alpha, a) = a/A$ and $\text{ctrl}_C(\alpha a) = \alpha a \in P_\cap$. If $\alpha a \notin P_\cap$ then, given our assumptions, for each $i \in \{1, \dots, k\}$ such that $\alpha a \notin P_{\eta_i}$ we know that c_i will sanction the violation with the same sanction $s \in \mathcal{A}_S$. Therefore $\alpha a s \in P_\cap$. These controllers will all first insert a , the others will have to loop. Hence $\sigma(q, a) = a/I$. Then, all these controllers will insert s and the others will loop, so $\sigma(q', a) = s/I$, finally, all the controllers that proposed inserting the action will now suppress a and the others will propose to allow a . Therefore $\sigma(q'', a) = \varepsilon/S$ and $\text{ctrl}_C(\alpha a) = \alpha a s \in P_\cap$. Hence for any word $\alpha \in \mathcal{A}_S^*$ if α violates some η_i then in $\text{ctrl}_C(\alpha)$ the violation will be followed by s_i . Therefore, $\text{ctrl}_C(\alpha) \in P_\cap$ for any word $\alpha \in \mathcal{A}_S^*$.

6 Concluding Remarks

Norm enforcement requires an enforcing controller to influence a target system's behavior. Enforcement through regimentation can halt target systems, and sanctioning can inject sanction actions. Our first contribution in this paper is an analysis of how regimentation and sanctioning relate to work on runtime control and verification. This allows us to analyze the runtime enforcement of norms. Our second contribution in this paper is a framework to specify the concurrent application of a set of controllers. We investigated the conditions under which the norm enforcement of a set of norms is possible.

Work on team automata [15, 26] to enforce properties distributively [27] relates to our collaborative automata. In particular in [30] edit automata are combined into a team edit automata. In this framework it is not possible for individual controllers to loop, as they do in our framework. How the team edit automaton's transition function can be constructed is not specified in [30], neither is a formal analysis given. In [19] a collaborative monitoring system, called a service automata framework, is described. The differences with our framework are that their controllers have an explicit local view of the target system, controllers are required to unanimously react to events through allow/suppress/insert, and interaction consists of sharing observations and delegating revision decisions. It is not analyzed how separately developed service automata can be combined into a service automata framework.

We are currently working on the translation of theoretical work on runtime norm enforcement to practical programming frameworks for implementations (cf. [12]). One of the challenges that remains is to develop a methodology to exploit the structure of a specific collaborative automaton in order to create an efficient communication protocol. For instance in the collaborative automaton from Example 9 c_2 only needs to execute its proposed revisions if both controllers are in state q_0 and the next action to review is w . Hence no full synchronization of the controllers is necessary, which is desirable.

REFERENCES

- [1] Thomas Ågotnes, Wiebe van der Hoek, and Michael Wooldridge, 'Normative system games', in *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '07)*, pp. 1–8, New York, NY, USA, (2007). ACM.
- [2] Natasha Alechina, Nils Bulling, Mehdi Dastani, and Brian Logan, 'Practical run-time norm enforcement with bounded lookahead', in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4-8, 2015*, pp. 443–451, (2015).
- [3] Natasha Alechina, Mehdi Dastani, and Brian Logan, 'Reasoning about normative update', in *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, (2013).
- [4] Natasha Alechina, Mehdi Dastani, and Brian Logan, 'Reasoning about normative update', in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pp. 20–26. AAAI Press, (2013).
- [5] Natasha Alechina, Mehdi Dastani, and Brian Logan, 'Norm approximation for imperfect monitors', in *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14, Paris, France, May 5-9, 2014*, pp. 117–124, (2014).
- [6] Guido Boella and Leendert van der Torre, 'Regulative and constitutive norms in normative multiagent systems', in *Proceedings of the Ninth International Conference on Principles of Knowledge Representation and Reasoning (KR'04)*, pp. 255–266, (2004).
- [7] Olivier Boissier, Rafael H. Bordini, Jomi Fred Hübner, Alessandro Ricci, and Andrea Santi, 'Multi-agent oriented programming with jacamò', *Science of Computer Programming*, **78**(6), 747–761, (2013).
- [8] Jan Broersen, Frank Dignum, Virginia Dignum, and John-Jules Ch Meyer, 'Designing a deontic logic of deadlines', in *Deontic Logic in Computer Science*, 43–56, Springer, (2004).
- [9] Christos G. Cassandras and Stephane Lafortune, *Introduction to Discrete Event Systems*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [10] Edward Chang, Zohar Manna, and Amir Pnueli, *The safety-progress classification*, Springer, 1993.
- [11] Mehdi Dastani, John-Jules Ch. Meyer, and Davide Grossi, 'A logic for normative multi-agent programs', *Journal of Logic and Computation*, **23**(2), 335–354, (2013).
- [12] Mehdi Dastani and Bas Testerink, 'From multi-agent programming to object oriented design patterns', in *International Workshop on Engineering Multi-Agent Systems*, pp. 204–226. Springer, (2014).
- [13] Mehdi Dastani, Nick AM Tinnemeier, and John-Jules Ch Meyer, 'A programming language for normative multi-agent systems', *Multi-Agent Systems: Semantics and Dynamics of Organizational Models*, 397–417, (2009).
- [14] Frank Dignum, 'Autonomous agents with norms', *Artificial Intelligence and Law*, **7**(1), 69–79, (1999).
- [15] Clarence Ellis, 'Team automata for groupware systems', in *Proceedings of the international ACM SIGGROUP conference on Supporting group work: the integration challenge*, pp. 415–424. ACM, (1997).
- [16] Marc Esteva, Julian Padget, and Carles Sierra, 'Formalizing a language for institutions and norms', in *Intelligent agents VIII*, 348–366, Springer, (2002).
- [17] Yliès Falcone, Laurent Mounier, Jean-Claude Fernandez, and Jean-Luc Richier, 'Runtime enforcement monitors: composition, synthesis, and enforcement abilities', *Formal Methods in System Design*, **38**(3), 223–262, (2011).
- [18] D. Gaertner, A. Garcia-camino, P. Noriega, and W. Vasconcelos, 'Distributed norm management in regulated multi-agent systems', in *AAMAS'07: Proceedings of the 6th International Conference on Autonomous Agents and Multiagent Systems*, pp. 624–631, (2007).
- [19] Richard Gay, Heiko Mantel, and Barbara Sprick, 'Service automata', in *Formal Aspects of Security and Trust*, 148–163, Springer, (2012).
- [20] Leslie Lamport, 'Proving the correctness of multiprocess programs', *Software Engineering, IEEE Transactions on*, (2), 125–143, (1977).
- [21] Jay Ligatti, Lujo Bauer, and David Walker, 'Edit automata: Enforcement mechanisms for run-time security policies', *International Journal of Information Security*, **4**(1-2), 2–16, (2005).
- [22] Naftaly H Minsky and Victoria Ungureanu, 'Law-governed interaction: a coordination and control mechanism for heterogeneous distributed systems', *ACM Transactions on Software Engineering and Methodology (TOSEM)*, **9**(3), 273–305, (2000).
- [23] F.Y. Okuyama, R.H. Bordini, and A.C. da Rocha Costa, 'A distributed normative infrastructure for situated multi-agent organisations', in *AAMAS'08: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems - Volume 3*, (2008).
- [24] Fred B Schneider, 'Enforceable security policies', *ACM Transactions on Information and System Security (TISSEC)*, **3**(1), 30–50, (2000).
- [25] Yoav Shoham and Moshe Tennenholtz, 'On social laws for artificial agent societies: off-line design', *Artificial Intelligence*, **73**(1-2), 231–252, (1995).
- [26] Maurice H Ter Beek, Clarence A Ellis, Jetty Kleijn, and Grzegorz Rozenberg, 'Synchronizations in team automata for groupware systems', *Computer Supported Cooperative Work (CSCW)*, **12**(1), 21–69, (2003).
- [27] Maurice H ter Beek, Gabriele Lenzini, and Marinella Petrocchi, 'Team automata for security:—a survey—', *Electronic Notes in Theoretical Computer Science*, **128**(5), 105–119, (2005).
- [28] B. Testerink and M. Dastani, 'A norm language for distributed organizations', in *BNAIC'12: Proceedings of the 24th Belgium-Netherlands Artificial Intelligence Conference*, pp. 234–241, (2012).
- [29] Wamberto W Vasconcelos, Martin J Kollingbaum, and Timothy J Norman, 'Normative conflict resolution in multi-agent systems', *Autonomous Agents and Multi-Agent Systems*, **19**(2), 124–152, (2009).
- [30] Zhenrong Yang, Aiman Hanna, and Mourad Debbabi, 'Team edit automata for testing security property', in *Information Assurance and Security, 2007. IAS 2007. Third International Symposium on*, pp. 235–240. IEEE, (2007).

Automatic Verification of Golog Programs via Predicate Abstraction

Peiming Mo and Naiqi Li and Yongmei Liu¹

Abstract. Golog is a logic programming language for high-level agent control. In a recent paper, we proposed a sound but incomplete method for automatic verification of partial correctness of Golog programs where we give a number of heuristic methods to strengthen given formulas in order to discover loop invariants. However, our method does not work on arithmetic domains. On the other hand, the method of predicate abstraction is widely used in the software engineering community for model checking and partial correctness verification of programs. Intuitively, the predicate abstraction task is to find a formula consisting of a given set of predicates to approximate a given first-order formula. In this paper, we propose a method for automatic verification of partial correctness of Golog programs which use predicate abstraction as a uniform method to strengthen given formulas. We implement a system based on the proposed method, conduct experiments on arithmetical domains and examples from the paper by Li and Liu. Also, we apply our method to the verification of winning strategies for combinatorial games.

1 Introduction

Among the various AI approaches, high-level programming tries to enable agents to understand and follow human's high-level instructions, and thus process intelligent behaviours. In the programming language Golog [12], the idea of high-level control is embodied in the statements of nondeterministic choices of actions and arguments, based on which the concrete actions to be performed are automatically generated by the interpreter. For robot programs which are expected to terminate, their partial correctness naturally becomes an important concern. Roughly speaking, a program is partially correct if desirable properties hold when it terminates on the condition that some properties hold before the execution of the program.

In a recent paper [13], we proposed a sound but incomplete method for automatic verification of partial correctness of Golog programs where we give a number of heuristic methods to strengthen given formulas in order to discover loop invariants. In our paper, the verification of Golog programs is achieved by the extended regression operator, which has a property similar to that of regression in the situation calculus: a formula holds after a program is executed, if its extended regression holds before the execution. With extended regression, we reduce partial correctness verification to a first-order theorem-proving task. When extended regression is applied to a loop statement, our method will repeatedly try to strengthen a candidate formula until it becomes a loop invariant. However, the heuristics that we developed to strengthen given formulas appear to be quite nonuniform, and our method does not work on arithmetic domains.

On the other hand, the method of predicate abstraction is widely used in the software engineering community for model checking and partial correctness verification of programs. Intuitively, the predicate abstraction task is to find a formula consisting of a given set of predicates to approximate a given first-order formula. The main practical problem in model checking is the state explosion problem, and predicate abstraction has been successfully used by Ball et al. [2], Clarke et al. [3, 4], etc, in reducing the size of the state space via capturing only relevant features. The most difficult step for partial correctness verification is to discover a proper loop invariant for each loop statement. While the search space of invariants is generally infinite, the technique of predicate abstraction tries to approximate the intended invariant based on a finite set of predicates, and thus the search space becomes finite. Flanagan and Qadeer [9] propose a heuristic method to generate appropriate predicates for each program loop, and then use these predicates to infer the loop invariants. According to their experimental results, predicate abstraction can lead to a very effective verification system, which can infer the necessary invariants for all except 31 of the 396 routines in a 44,380 LOC program within one hour. Srivastava and Gulwani [17] combine templates and predicate verification to infer very expressive loop invariants.

There are also a few works about the verification of Golog programs. Liu [14] presents a Hoare-style proof system for partial correctness of Golog programs. In [5], Claßen and Lakemeyer propose a logic based on the situation calculus variant called \mathcal{ES} to verify temporal properties of non-terminating Golog programs, and this line of research is continued by some recent publications [6, 19]. However, notably, all these works are mainly theoretic studies.

In this paper, we propose a method for automatic verification of partial correctness of Golog programs which uses predicate abstraction as a uniform method to strengthen given formulas. Specifically, the predicate abstraction method is used when a loop is being regressed. Obtaining the extended regression of a loop is equivalent to discovering a proper loop invariant, and we follow the idea proposed in [13]: start from an initial formula, and then repeatedly try to strengthen it until it becomes a loop invariant. While in that paper, the proper loop invariants are generated by handcrafted heuristics, in this paper we try to discover them by the method of predicate abstraction.

We also make use of small models and the technique of small model progression as in the previous paper. Intuitively, small models can be viewed as possible program states during the execution. Given a set of initial small models, which can be viewed as some initial test inputs, small model progression will run the given Golog program on the small models and thus associate a set of possible states with each loop statement. The associated small models can accelerate the process of predicate abstraction: if a candidate formula generated by the predicate abstraction is not satisfied by some of the small model-

¹ Dept. of Computer Science, Sun Yat-sen University, China, e-mail: mopeim3@mail2.sysu.edu.cn & linaiqi@mail2.sysu.edu.cn & ymliu@mail.sysu.edu.cn.

s, it is not a proper loop invariant and can be discarded immediately. It is important to emphasize that although our method makes use of small models, it is still capable of verifying properties of programs over arbitrarily large finite domains and even infinite domains. The reason is that small models only serve for the purpose of filtering out formulas which cannot be loop invariants, and we need to resort to first-order theorem-proving to verify a formula is a loop invariant. This also makes our work fundamentally different from the method of model checking, where the systems being verified must be finite.

We implement a verification system based on the proposed method and conduct a set of experiments on it. Experimental results show that our method can prove all the Hoare-triples reported in [13]. The potential of our approach is further demonstrated by experiments on some other domains that involves functions and arithmetical loop invariants. We also apply our system to verify properties of strategies in two famous games *Pick-up Stone* and *Chomp*, in which the verification of winning strategies is reduced to the verification of partial correctness of Golog programs.

The paper is organized as follows. In Section 2, we will introduce some background knowledge of our work. In Section 3, we will present the related concepts of predicate abstraction, and how to use them to infer loop invariants and thus verify the given Golog programs. In Section 4, we will discuss some technical details. Section 5 will present our experimental results in both arithmetical and non-arithmetical domains. In Section 6 we will show how to apply our method to verify game properties. Finally we will conclude this paper with a summary of contributions and a discussion of future work.

2 Preliminaries

In this section, we will first introduce the preliminaries knowledge. Then we will define Golog program and its partial correctness. In the third subsection we will discuss extended regression, which is introduced in our previous work and will play an important role in this paper. Finally we will introduce small model and its progression.

2.1 The Situation Calculus

The situation calculus [16] is a second-order language specifically designed for representing dynamic worlds. It includes a binary predicate $s \sqsubseteq s'$ meaning that situation s is a subhistory of situation s' ; a binary predicate $Poss(a, s)$ meaning that action a is possible in situation s ; a countable set of action functions, e.g., $move(x, y)$; a countable set of relational fluents, i.e., predicates taking a situation term as their last argument, e.g., $ontable(x, s)$; and a countable set of functional fluents, i.e., functions taking a situation term as their last argument, e.g., $height(x, s) = y$.

Often, we need to restrict our attention to formulas that do not refer to any situations other than a particular one τ , and we call such formulas uniform in τ . We use $\phi(\tau)$ to denote that ϕ is uniform in τ . We call a uniform formula ϕ with all situation arguments eliminated a *situation-suppressed* formula, and use $\phi[s]$ to denote the uniform formula with all situation arguments restored with term s . A situation s is executable if it is possible to perform the actions in s one by one: $Exec(s) \doteq \forall a, s'. do(a, s') \sqsubseteq s \supset Poss(a, s')$.

In the situation calculus, a particular domain of application is specified by a basic action theory (BAT) of the form: $\mathcal{D} = \Sigma \cup \mathcal{D}_{ap} \cup \mathcal{D}_{ss} \cup \mathcal{D}_{una} \cup \mathcal{D}_{S_0}$, where

1. Σ is the set of the foundational axioms for situations.

2. \mathcal{D}_{ap} contains a single precondition axiom of the form $Poss(a, s) \equiv \Pi(a, s)$, where $\Pi(a, s)$ is uniform in s .
3. \mathcal{D}_{ss} is a set of successor state axioms (SSAs).
For each relational fluent F :
 $F(\vec{x}, do(a, s)) \equiv \Phi_F(\vec{x}, a, s)$,
For each functional fluent f :
 $f(\vec{x}, do(a, s)) = y \equiv \Phi_f(\vec{x}, y, a, s)$,
where $\Phi_F(\vec{x}, a, s)$ and $\Phi_f(\vec{x}, y, a, s)$ are uniform in s .
4. \mathcal{D}_{una} is the set of unique names axioms for actions.
5. \mathcal{D}_{S_0} , the initial KB, is a set of sentences uniform in S_0 .

In the situation calculus, state constraints are formulas that hold true in every executable situation. We follow the definition of state constraints in [16].

Definition 1 (state constraint) Given a BAT \mathcal{D} and a formula $\phi(s)$, $\phi(s)$ is a state constraint for \mathcal{D} if $\mathcal{D} \models \forall s. Exec(s) \supset \phi(s)$.

We use \mathcal{D}_{SC} to denote a set of verified state constraints, and abuse \mathcal{D}_{SC} as its conjunction.

Regression is an important computational mechanism for reasoning about actions and their effects, and here we present the one step regression operator and a simple form of the regression theorem stated in [13], and we add the functional regression in the definition.

Definition 2 We use $\mathcal{R}_{\mathcal{D}}[\phi]$ to denote the formula obtained from ϕ by the following steps:

1. Replace each functional fluent atom $f(\vec{t}, do(\alpha, \sigma))$ with $(\exists y). \Phi_f(\vec{t}, y, \alpha, \sigma) \wedge \phi[f(\vec{t}, do(\alpha, \sigma))/y]$, where $\phi[x/y]$ means that all x 's in ϕ are replaced by y .
2. Replace each relational fluent atom $F(\vec{t}, do(\alpha, \sigma))$ with $\Phi_F(\vec{t}, \alpha, \sigma)$.
3. Replace each precondition atom $Poss(\alpha, \sigma)$ with $\Pi(\alpha, \sigma)$, and further simplify the result by using \mathcal{D}_{una} .

Theorem 1 If $\mathcal{R}_{\mathcal{D}}[\phi]$ is the formula regressed from ϕ , then $\mathcal{D} \models \phi \equiv \mathcal{R}_{\mathcal{D}}[\phi]$ holds.

2.2 Golog Programs and Partial Correctness

The formal semantics of Golog is specified by an abbreviation $Do(\delta, s, s')$, which is inductively defined as follows:

1. Primitive actions: For any action term α ,
 $Do(\alpha, s, s') \doteq Poss(\alpha, s) \wedge s' = do(\alpha, s)$.
2. Test actions: For any situation-suppressed formula ϕ ,
 $Do(\phi?, s, s') \doteq \phi[s] \wedge s = s'$.
3. Sequence:
 $Do(\delta_1; \delta_2, s, s') \doteq \exists s''. Do(\delta_1, s, s'') \wedge Do(\delta_2, s'', s')$.
4. Nondeterministic choice of two actions:
 $Do(\delta_1 | \delta_2, s, s') \doteq Do(\delta_1, s, s') \vee Do(\delta_2, s, s')$.
5. Nondeterministic choice of action arguments:
 $Do((\pi x)\delta(x), s, s') \doteq (\exists x) Do(\delta(x), s, s')$.
6. Nondeterministic iteration:
 $Do(\delta^*, s, s') \doteq (\forall P). \{ (\forall s_1) P(s_1, s_1) \wedge (\forall s_1, s_2, s_3) [P(s_1, s_2) \wedge Do(\delta, s_2, s_3) \supset P(s_1, s_3)] \} \supset P(s, s')$.

Conditionals and loops are defined as abbreviations:

if ϕ **then** δ_1 **else** δ_2 **fi** $\equiv [\phi?; \delta_1][\neg\phi?; \delta_2]$,
while ϕ **do** δ **od** $\equiv [\phi?; \delta]^*; \neg\phi?$.

Then, the partial correctness of a Hoare triple is defined as below:

Definition 3 A Hoare-triple is of the form $\{P\}\delta\{Q\}$, where P and Q are situation-suppressed formulas, and δ is a Golog program. A Hoare-triple $\{P\}\delta\{Q\}$ is said to be partially correct wrt \mathcal{D} if $\mathcal{D} \models \forall s, s'. P[s] \wedge Do(\delta, s, s') \supset Q[s']$.

2.3 Extended Regression

Li and Liu [13] extended the regression of primitive actions to that of programs, which is called extended regression:

Definition 4 Given \mathcal{D} and \mathcal{D}_{SC} , the extended regression of formula $\phi(s)$ wrt program δ , denoted as $\hat{\mathcal{R}}_{\mathcal{D}}[\phi(s), \delta]$, is defined as follows:

- $\hat{\mathcal{R}}_{\mathcal{D}}[\phi(s), \alpha] = \mathcal{R}_{\mathcal{D}}(Poss(\alpha, s) \supset \phi(do(\alpha, s)))$.
- $\hat{\mathcal{R}}_{\mathcal{D}}[\phi(s), \psi?] = \psi[s] \supset \phi(s)$.
- $\hat{\mathcal{R}}_{\mathcal{D}}[\phi(s), \delta_1; \delta_2] = \hat{\mathcal{R}}_{\mathcal{D}}[\hat{\mathcal{R}}_{\mathcal{D}}[\phi(s), \delta_2], \delta_1]$.
- $\hat{\mathcal{R}}_{\mathcal{D}}[\phi(s), \delta_1|\delta_2] = \hat{\mathcal{R}}_{\mathcal{D}}[\phi(s), \delta_1] \wedge \hat{\mathcal{R}}_{\mathcal{D}}[\phi(s), \delta_2]$.
- $\hat{\mathcal{R}}_{\mathcal{D}}[\phi(s), (\pi x)\delta(x)] = (\forall x)\hat{\mathcal{R}}_{\mathcal{D}}[\phi(s), \delta(x)]$.
- $\hat{\mathcal{R}}_{\mathcal{D}}[\phi(s), \mathbf{while} \varphi \mathbf{do} \delta \mathbf{od}]$ is a formula (denoted as $\eta(s)$) satisfying the following two conditions:
 1. $\models_{\text{FOL}} \forall s. \eta(s) \wedge \varphi[s] \wedge \mathcal{D}_{SC} \supset \hat{\mathcal{R}}_{\mathcal{D}}[\eta(s), \delta]$.
 2. $\models_{\text{FOL}} \forall s. \eta(s) \wedge \mathcal{D}_{SC} \supset \phi(s) \vee \varphi[s]$.

Intuitively, in the definition of loop statement the first condition ensures the regression is a loop invariant, and the second condition guarantees this invariant is strong enough to entail the formula being regressed when the loop ends.

In the situation calculus, a formula holds after a sequence of actions are performed iff its regression can be entailed by the initial knowledge base. The extended regression has a similar property [13]:

Theorem 2 A Hoare-triple $\{P\}\delta\{Q\}$ is partially correct wrt \mathcal{D} , if \mathcal{D}_{SC} is a set of verified state constraints, and $\models_{\text{FOL}} \forall s. P[s] \wedge \mathcal{D}_{SC} \supset \hat{\mathcal{R}}_{\mathcal{D}}[Q[s], \delta]$.

2.4 Small Model Progression

As aforementioned, small models can be viewed as possible program states during the execution. In this paper, a small model is represented as a finite set of ground atoms that excludes arithmetical relations. We also make the closed world assumptions to our small models, which means that any non-arithmetical relation that does not appear is regarded as false.

Li and Liu [13] present the notion of small model progression, and use $prog^S[M, \alpha]$ to denote the new model generated by updating small model M according to primitive action α , $prog[M, \delta]$ to denote a set of small models resulting from the progression of M wrt program δ . In this paper, the initial small models are provided by hand for the reason of convenience.

Definition 5 We assume the ground terms are all constants from a finite set D . Given a small model M and a program δ , the progression of M wrt δ , denoted as $prog[M, \delta]$, results in a set of small models:

- $prog[M, \alpha] = 1. \emptyset$ if $M[s] \not\models Poss(\alpha, s)$.
- 2. $\{prog^S[M, \alpha]\}$ if $M[s] \models Poss(\alpha, s)$.

- $prog[M, \psi?] = 1. \emptyset$ if $M[s] \not\models \psi[s]$.
- 2. $\{M\}$ if $M[s] \models \psi[s]$.
- $prog[M, \delta_1; \delta_2] = prog[prog[M, \delta_1], \delta_2]$.
- $prog[M, \delta_1|\delta_2] = prog[M, \delta_1] \cup prog[M, \delta_2]$.
- $prog[M, (\pi x)\delta(x)] = \bigcup \{prog[M, \delta(c)] \mid c \in D\}$.
- $prog[M, \delta^*] = \bigcup_{n \geq 0} prog[M, \delta^n]$, where δ^n is an abbreviation of jointing n copies of δ sequentially.

When \mathcal{M} is a set of small models, we define

$$prog[\mathcal{M}, \delta] = \bigcup \{M' \mid M \in \mathcal{M}, prog[M, \delta] = M'\}.$$

When progressing wrt a loop the computation may never stop. In practice we preset a constant K , and let $prog[M, \delta^*] = \bigcup_{n=0}^K prog[M, \delta^n]$.

The first usage of $prog[M, \delta]$ in our methods is that they can inform us the Hoare triple $\{P\}\delta\{Q\}$ is not partial correct before starting the static analyses, which derives from the following theorem [13]:

Theorem 3 If a Hoare-triple $\{P\}\delta\{Q\}$ is partially correct, and M is a small model that $M[s] \models P[s] \wedge \mathcal{D}_{SC}$, then for all $M' \in prog[M, \delta]$ we have $M'[s] \models Q[s]$, where $M[s]$ is a small model with situation s restored.

Besides, the small models can largely improve the efficiency of predicate abstraction because we use the small models to filter the wrong candidates before calling an SMT solver in the process of predicate abstraction. We can see these details in the next section.

3 Infer Invariants via Predicate Abstraction

In this section, we will first introduce the standard predicate abstraction, and how we adapt it to the bounded predicate abstraction to serve our purpose. And then we will present a concrete algorithm to compute the bounded predicate abstraction. Finally, we will see how it is used to discover loop invariants and thus verify the Hoare-triple.

The definition of standard predicate abstraction we use is the one presented in [9]:

Definition 6 (Standard Predicate Abstraction) Given a set of predicates $\mathcal{P} = \{p_1, \dots, p_n\}$, for any formula Q , its abstraction $\alpha(Q)$ is defined as the strongest boolean combination on \mathcal{P} such that $Q \supset \alpha(Q)$ is valid.

Now we present the predicate abstraction defined in our paper, and discuss how it is different from the standard one.

We say a formula ϕ is a $\forall^*\exists^*$ formula, if ϕ is of the form $\forall x_1 \dots \forall x_n \exists y_1 \dots \exists y_m \psi$, where $n, m \geq 0$ and ψ is a quantifier-free formula.

Definition 7 (Predicate Abstraction) Given a set of predicate \mathcal{P} , a formula ϕ , and a set of small models \mathcal{M} , we say a formula $\forall^*\exists^*C$ is the predicate abstraction of ϕ and \mathcal{M} under \mathcal{P} , if C is a clause over \mathcal{P} s.t. $\mathcal{M} \models \forall^*\exists^*C$ and $\mathcal{D}_{SC} \models \forall^*\exists^*C \supset \phi$, and there is no shorter clause C' over \mathcal{P} s.t. $\forall^*\exists^*C'$ satisfies these conditions.

The predicate abstraction we developed differs from the standard one in the following ways: 1. in the standard predicate abstraction all free variables in $\alpha(Q)$ are implicitly regarded as universally quantified, but in our definition both universal and existential quantifiers are possible; 2. the predicate abstraction in our approach results in a quantified clause, while the standard predicate abstraction results in

a formula; 3. the conditions in our definition are different from that in the standard definition.

The concept of predicate abstraction is already enough to solve many non-arithmetical verification problems. But during our research we observe that many loop invariants in arithmetical domains are of a particular form, which inspires us a more general definition.

Definition 8 (Bounded Predicate Abstraction) Let $Fun0$ and $Const$ denote the sets of 0-ary functional fluents and constants respectively, and \mathcal{P} , ϕ and \mathcal{M} as before. A bounded predicate abstraction of ϕ and \mathcal{M} under \mathcal{P} is a formula of the form $\varphi = \forall^* \exists^* bound(\vec{x}) \supset C(\vec{x})$ s.t. $\mathcal{M} \models \varphi$ and $\mathcal{D}_{SC} \models \varphi \supset \phi$, where $C(\vec{x})$ is a clause over \mathcal{P} , and $bound(\vec{x})$ is of the form $\bigwedge_i lb_i \leq x_i \leq hb_i$ where $lb_i, hb_i \in \{\vec{x}\} \cup Fun0 \cup Const$. There is no shorter $C'(\vec{x})$ s.t. it has a bound that makes $\forall^* \exists^* bound'(\vec{x}) \supset C'(\vec{x})$ satisfy these conditions.

In the algorithms below, we assume that \mathcal{D} and \mathcal{D}_{SC} are given without providing them in the arguments explicitly.

Algorithm 1 is used to compute the bounded predicate abstraction. In our implementation, we only consider \forall^* and $\forall^* \exists$ formulas because of efficiency. The idea is quite simple: we enumerate clauses from short ones to long ones, and then enumerate the different combinations of the quantifiers, until a formula that satisfies the conditions is found. In line 5, the algorithm calls an SMT solver to check whether the first-order entailment does hold (an SMT solver can decide whether a first-order formula is satisfiable with respect to some background theory such as linear arithmetic). If the SMT solver does not terminate in a given time, we treat the result as false. Similar cases are treated in the same way during algorithms 2 and 3.

Algorithm 1: $boundPA(\phi, \mathcal{M}, \mathcal{P})$

Input: ϕ - formula to be strengthened; \mathcal{M} - the set of small models; \mathcal{P} the set of predicate candidates
Output: The bounded predicate abstraction φ'

- 1 **repeat** Enumerate a clause $C(\vec{x})$ over \mathcal{P} (enumerate from short ones to long ones)
- 2 **repeat** Enumerate a bound $bound(\vec{x})$
- 3 Let $\varphi = bound(\vec{x}) \supset C(\vec{x})$
- 4 **repeat** Enumerate φ' as $\forall^* \varphi$ or $\forall^* \exists \varphi$
- 5 **if** $\mathcal{M} \models \varphi'$ and $\mathcal{D}_{SC} \models \varphi' \supset \phi$ **then**
- 6 **return** φ'
- 7 **return** false

Next we will see how to apply the bounded predicate abstraction to strengthen a formula into loop invariant. The procedure $infer$ listed as Algorithm 2 is similar to the corresponding algorithm in [13]. It is used to infer a loop invariant when a loop statement is being regressed.

In Algorithm 2, we use the association function $asso(\delta_l)$ to map each loop statement δ_l to a set of small models \mathcal{M} . Intuitively \mathcal{M} is the set of small models that are progressed to the beginning of the loop. For every loop, the loop invariant should be satisfied by all the associated small models. In Line 2, we will first discover a set of linear and inequality invariants Δ_{inv} with a high efficient method, the detail of which will be discussed in the next section. The idea here is to discover the simple linear and inequality invariants first, and then use them to discover the difficult ones later. In Line 3, we start with the candidate loop invariant as $\phi(s) \vee \varphi[s]$, and we try to repeatedly

Algorithm 2: $infer(\phi(s), \delta_l, asso)$

Input: $\phi(s)$ - formula being regressed; δ_l - loop statement being regressed; $asso$ - maps each **while** construct to a set of small models

Output: A loop invariant for δ_l .

- 1 Let $\delta_l = \mathbf{while} \varphi \mathbf{do} \delta \mathbf{od}$
- 2 $\Delta_{inv} \leftarrow getLinearInvs(\delta_l, asso(\delta_l))$
- 3 $\eta(s) \leftarrow \phi(s) \vee \varphi[s]; counter \leftarrow 0$
- 4 **while** $counter < K$ **do**
- 5 $counter \leftarrow counter + 1$
- 6 $reg(s) \leftarrow \hat{\mathcal{R}}_{\mathcal{D}}(\eta(s), \delta)$
- 7 **if** $\models_{FOL} \eta(s) \wedge \varphi[s] \wedge \mathcal{D}_{SC} \wedge \Delta_{inv} \supset reg(s)$ **then**
- 8 **return** $\eta(s) \wedge \Delta_{invs}$
- 9 Let $reg(s) \equiv A_1(s) \wedge \dots \wedge A_n(s)$, choose a $A_i(s)$ s.t.
 $\not\models_{FOL} \eta(s) \wedge \varphi[s] \wedge \mathcal{D}_{SC} \supset A_i(s)$
- 10 $\mathcal{P} \leftarrow genPredicate(A_i(s), \delta_l)$
- 11 $\varphi(s) \leftarrow boundPA(A_i(s), asso(\delta_l), \mathcal{P})$
- 12 **if** $\varphi(s) = \emptyset$ **then**
- 13 **return** *unknown*
- 14 $\eta(s) \leftarrow \eta(s) \wedge \varphi(s)$
- 15 **return** *unknown*

strengthen it until it becomes an invariant in the following loop. In the following loop that starts from Line 4, the variable $counter$ and a pre-set constant K are used to make sure that the procedure always terminates (with the premise that the underlying theorem prover always terminates). In each iteration, if the regressed result can be entailed at Line 7, we return $\eta(s) \wedge \Delta_{invs}$ as the loop invariant. If the entailment cannot be proved, in Line 9 we select one of the untailed conjunct $A_i(s)$, and then use the information of $A_i(s)$ and the loop δ_l to generate a predicate set \mathcal{P} . Line 11 is our abstraction algorithm. If the abstraction result is empty, it means the predicate set \mathcal{P} we generated is not precise or the Hoare-triple is wrong, then we return *unknown* in Line 13 under that situation, else we apply the predicate abstraction on it to obtain a strengthening at Line 14. The bounded predicate abstraction requires a set of small models and a set of predicate candidates. The set of small models is retrieved from the $asso$ function, and the predicate candidates are automatically generated in Line 10. We will discuss how to generate predicates in further details in the next section.

Finally a Hoare-triple is verified by the following main algorithm, which can be regarded as a simplified version of the main algorithm in [13]. Intuitively, the main algorithm will first try to use small models and progression to find a counterexample. If no counterexample is found, it will apply the extended regression and reduce the verification problem to first-order entailment problem. If all such attempts fail, the algorithm simply returns *non-deter*. In this algorithm, the procedure $prog$ is almost the same as the definition of small model progression, except that it also updates the association function during the process. The procedure returns *no* if a counterexample is found at Line 3, and returns *yes* if the regressed result can be entailed by the precondition. For other cases, the system returns *non-deter*.

Theorem 4 $veri(P, \delta, Q, \mathcal{M})$ returns *yes* only if $\{P\}\delta\{Q\}$ is partial correct; returns *no* only if $\{P\}\delta\{Q\}$ is not partially correct.

Proof: Firstly, if $veri(P, \delta, Q, \mathcal{M})$ returns *yes*, it means that $\models_{FOL} \forall s. P[s] \wedge \mathcal{D}_{SC} \supset reg(s)$. Then according to Theorems 2, $\{P\}\delta\{Q\}$

Algorithm 3: $veri(P, \delta, Q, \mathcal{M})$

Input: Hoare-triple $\{P\}\delta\{Q\}$; \mathcal{M} - initial small models that satisfy P and \mathcal{D}_{SC} .

Output: Returns *yes* if Hoare-triple is proved to be correct; returns *no* if Hoare-triple is proved to be wrong; otherwise returns *non-deter*

```

1 Let asso maps each while construct in  $\delta$  to  $\emptyset$ 
2  $\langle \mathcal{M}', \textit{asso} \rangle \leftarrow \textit{prog}(\mathcal{M}, \delta, \textit{asso})$ 
3 if  $\exists M' \in \mathcal{M}'$  s.t.  $M'[s] \not\models Q[s]$  then return no
4  $\textit{reg}(s) \leftarrow \hat{\mathcal{R}}_{\mathcal{D}}(Q[s], \delta, \textit{asso})$ 
5 if  $\models_{\text{FOL}} \forall s. P[s] \wedge \mathcal{D}_{SC} \supset \textit{reg}(s)$  then return yes
6 return non-deter

```

is partial correct. Secondly, according to Theorems 3, if $\{P\}\delta\{Q\}$ is partial correct, then $\forall M' \in \mathcal{M}'$ we should have $M'[s] \models Q[s]$ (we guarantee that all initial small models satisfy P and \mathcal{D}_{SC}). So if $\exists M' \in \mathcal{M}'$ s.t. $M'[s] \not\models Q[s]$, then $\{P\}\delta\{Q\}$ is not partial correct.

4 Technical Details

In this section, we will discuss some technical issues of our algorithm in further details. Firstly we will show how to generate predicate candidates. Secondly we will discuss the bound generation and disturbance problem. Finally we will present a method to efficiently discover linear and inequalities loop invariants.

4.1 Generate Predicate Candidates

The success of Algorithm 1 largely depends on the given predicate candidates. The following shows how we generate this set.

Let p be a predicate symbol with k arguments. We say a ground atom $p(t_1, \dots, t_k)$ is a predicate candidate, where t_1, \dots, t_k are terms. Our system will automatically generate a set of predicate candidates and use them in the predicate abstraction.

We use $genPredicate(\phi, \delta_i)$ to denote the set of predicate candidates generated by the formula ϕ and the loop statement δ_i . Intuitively, formula ϕ is the formula being abstracted and δ_i is the loop being regressed. Suppose the Hoare-triple being verified is $\{P\}\delta\{Q\}$, we regard formulas P and Q as globally accessible and will use them in the generation process. The set of predicate candidates is computed in the following steps:

1. Initialize \mathcal{P} as an empty set.
2. $\mathcal{P} \leftarrow \mathcal{P} \cup \{p \mid p \text{ is a predicate appears in } P, Q \text{ or } \phi\}$.
3. $\mathcal{P} \leftarrow \mathcal{P} \cup \{p \mid \psi? \text{ is a test action in } \delta_i \text{ and } p \text{ is a predicate in } \psi\}$.
4. Suppose that the predicate candidates generated in the previous steps are $\mathcal{P} = \{p_1(\vec{t}_1), \dots, p_n(\vec{t}_n)\}$. We collect all the terms that appear as arguments of some predicates, which is $\mathcal{T} \leftarrow \{t \mid \exists p_i(\vec{t}_i) \in \mathcal{P}. t \in \vec{t}_i\}$.
5. The final set of predicate candidates is $\mathcal{P}' \leftarrow \{p_i(\vec{t}_i') \mid \vec{t}_i' \in \mathcal{T}\}$.

Example 1 Suppose that the predicates generated in the first three steps are $\{in(x, f(y)), x < f(y)\}$. The set of terms is $\{x, f(y)\}$, so we will generate $\{in(x, x), in(x, f(y)), in(f(y), x), f(y) < x, x < f(y), in(f(y), f(y))\}$ as the set of new predicates. Theoretically $x < x$ and $f(y) < f(y)$ should also be generated, but with the knowledge of mathematics it is easy to verified that they are equivalent to *false*, so we will remove them during the process.

4.2 Generate Bounds

When generating bounds in the process of bounded predicate abstraction, there are two problems worth noting. The first problem is how to construct the set $Const$ as the constants being enumerated. The second problem is that in most verifications, the bounds will be ‘disturbed’ so we also need to consider the disturbance of bounds.

Recall that in bounded predicate abstraction, a bound is of the form $\bigwedge_i lb_i \leq x_i \leq hb_i$ where $lb_i, hb_i \in \{\vec{x}\} \cup Fun0 \cup Const$. In this formula, $\{\vec{x}\}$ and the set of 0-ary functional fluents $Fun0$ can be obtained from the clause $C(\vec{x})$ and the domain description respectively. However, it is impossible to construct $Const$ as all the constants in the domain, since the set of constants may be huge or even infinite. In order to restrict the number of bounds being enumerated, we heuristically assign different variables with different sets of $Const$, and make use of the information in the small models. The approach can be better demonstrated by the example below.

Example 2 Suppose the clause $C(\vec{x}) = vis(x, y)$, and the associated small models of the loop that being regressed is $\{vis(1, 1), vis(1, 2), vis(1, 3), vis(2, 1), vis(2, 2), vis(2, 3)\}$. Variable x is the first argument of predicate vis , and from the small model we know the range of the first argument of predicate vis is $[1, 2]$. So for variable x , we will construct its $Const$ as $\{1, 2\}$. Similarly, for variable y its $Const$ is $\{1, 3\}$. As a result, the bounded predicate abstraction method will finally generate formulas like $\forall x, y. 1 \leq x \leq 2 \wedge 1 \leq y \leq 3 \supset vis(x, y)$.

Another problem is the disturbance of the bounds. In many verifications, the actually desired bound may be just a little different from that generated by the previous method. For example, we may generate $0 \leq x \leq xpos$, while the desired bound is $1 \leq x \leq xpos + 1$. In order to enable our system discover such bounds, we will also enumerate bounds with disturbance $\pm k$, where k is a small constant, but we replace k with 1 under the consideration of efficiency and experience. To put it more formally, if $lb_i \leq x \leq hb_i$ is enumerated by the previous method, we will also enumerate other 4 bounds $lb_i \pm 1 \leq x \leq hb_i \pm 1$ as its disturbance.

4.3 Discover Linear Relations and Inequalities

To further improve the performance, we treat the loop invariants that are linear relations and inequalities specially. In the software engineering community, there are many available works on efficiently discovering linear and inequality invariants, such as [8] and [18].

Since the linear and inequality invariants can be discovered with special method much faster than by using predicate abstraction, our algorithm try to discover them in advance. This is demonstrated as function *getLinearInvs* at Line 2 of Algorithm 2. Our implementation of *getLinearInvs* follows the guess-and-check paradigm.

Firstly, the system will guess a set of candidate invariants based on dynamic checking and linear regression as follows. Suppose the set of small models is $\mathcal{M} = \{M_1, \dots, M_k\}$. From each small model M_i , we obtain a set of function-value pairs of all 0-ary functional fluents, such as $\{f_0 = v_{0i}, f_1 = v_{1i}, \dots, f_n = v_{ni}\}$. The inequality and linear invariant candidates are generated by the following methods:

- Inequality candidates: We consider inequality invariant candidates of the forms $f_u < f_v$ and $f_u < C$, where f_u, f_v are 0-ary functional fluents and C is a constant in the Hoare-triple. An inequality formula will be included as an invariant candidate if it is satisfied in every model.

- **Linear candidates:** In our implementation, we consider all linear relations of the form $A \cdot f_u + B \cdot f_v + C \cdot f_w + D = 0$. This is equivalent to solving the following system of linear equations:

$$\begin{aligned} A \cdot v_{u1} + B \cdot v_{v1} + C \cdot v_{w1} + D &= 0 \\ &\dots\dots \\ A \cdot v_{uk} + B \cdot v_{vk} + C \cdot v_{wk} + D &= 0 \end{aligned}$$

Suppose $\langle a, b, c, d \rangle$ is the solution of the system, we will include the linear relation $a \cdot f_u + b \cdot f_v + c \cdot f_w + d = 0$ as an invariant candidate.

After the guessing step, we obtain a set of candidate invariants Δ , and then each formula in Δ will be checked statically. The checking step is also an application of the extended regression. Suppose the loop is **while** φ **do** δ , for every candidate invariant $\eta \in \Delta$ we will compute its regression $reg(s) = \hat{\mathcal{R}}_{\mathcal{D}}[\eta[s], \delta]$ and then check that whether the entailment $\models_{\text{FOL}} \eta[s] \wedge \varphi[s] \wedge \mathcal{D}_{SC} \supset reg(s)$ can be proved. If we can prove the entailment, we know $\eta[s]$ is indeed a loop invariant. Otherwise, we will simply give up the candidate.

To ensure the efficiency, we will not try to strengthen the candidate invariant as Algorithm 2 does when the system fails to prove the entailment. However, experiments show that even such simple strategy is enough to discover and validate lots of useful linear and inequality invariants.

5 Experimental Results

We have implemented our algorithms to a system by using SWI-Prolog, Java and Z3. Java is used to discover linear relations because of its efficiency and powerful API, and Z3 [7] is the SMT solver which is used to prove the first-order entailments. They use the I/O interface to work together.

Our system has run on six arithmetical domains (with five succeed and one failed) and all the non-arithmetic domains used in [13]. Among all the arithmetical domains, *1D*, *PrizeA1*, *Arith* and *Sort* are adapted from those used in [10]. *Find* is a modified version from [9], and *Addition* is designed by ourselves. All experiments were conducted on a machine with 3.30 GHz CPU and 4.00GB RAM under Linux. In all our experiments the initial small models are manually provided, and only 1 or 2 simple initial small models are sufficient for each domain.

5.1 Arithmetical Domains

The arithmetic experiments are derived from the following domains:
1D: The program is to visit all the elements in an array from right to left.

PrizeA1: This program is to visit all the $N \cdot N$ cells from row 1 to row N in the outer loop, and from left to right during every row in the inner loop.

Arith: This program is to increase a variable *numy* (initialized as 0) to $2N$. It contains a single loop, and *numy* will increase by two during each iteration.

Find: This program sets the i -th element of a boolean array *inb* to be true if the i -th element of an integer array *ina* is nonzero, and to false otherwise. And we treat *ina*(X) as a 1-ary functional fluent and *inb*(X) as a 1-ary relational fluent.

Addition: This program is to increase a variable *sum* (initialized by 0) to $A \cdot B$ through a nested loop. The outer loop repeats A times and the inner loop B times, the variable *sum* is increased by one during every inner loop's iteration.

Sort: The program sorts an array by using a single loop.

In the experiments of the arithmetical domains, we provide the set of state constraints as \mathcal{D}_{SC} in advance. For all the experiments, we only provide constraints of type information. For example in the *1D* domain, the argument x of *vis*(x) is expected to be an integer, so we will include the constraint $\forall x. vis(x) \supset int(x)$. In the following, we will use $\forall x \in int. \phi(x)$ to denote $\forall x. int(x) \supset \phi(x)$.

We will take *1D* and *Sort* as examples to demonstrate our system, while the results are summarized in Table 1. We only mention the situation arguments explicitly in the precondition axioms and successor state axioms, and omit them in the Hoare-triple and the loop invariant formulas.

Figure 1 presents the Hoare-triple and verification results of the *1D* domain. It only contains one loop. The constant *len* denotes the length of the array. The action *move_left* has two effects: firstly it decreases *xpos*(s) by one, where *xpos*(s) is a 0-ary functional fluent used as a pointer, and then it makes the predicate *vis*(*xpos*(s), s) to be true. The resulted loop invariant is the conjunction of *Invs* and *Linear*, where *Invs* is discovered by the predicate abstraction algorithm, while *Linear* is a set and the conjunction of its elements denotes the useful linear relations between terms. This domain is verified by our system in 1.0s.

Hoare-triple:

```
{xpos = len}
while 0 < xpos do move_left od
{∀x ∈ int. 0 < x < len + 1 ⊃ vis(x)}
```

Action Precondition Axiom:

$Poss(move_left, s) \equiv xpos(s) > 0$

Successor State Axiom:

$xpos(do(a, s)) = y \equiv xpos(s) = y \vee$

$xpos(s) = y + 1 \wedge a = move_left$

$vis(x, do(a, s)) \equiv vis(x, s) \vee$

$\neg vis(x, s) \wedge a = move_left \wedge xpos(s) = x + 1$

Invs: $\forall x \in int. xpos < x < len + 1 \supset vis(x)$

Linear: $\{xpos < len + 1\}$

Figure 1. Example of the *1D* Domain

In Figure 2, the program sorts an array with length *len* in a single loop. During each iteration, *xpos*(s) moves right if $in(xpos(s), s) < in(xpos(s) + 1, s)$, otherwise, it swaps the value of $in(xpos(s), s)$ and $in(xpos(s) + 1, s)$ and then resets *xpos*(s) to be 1. After 224.0s, an invariant (the conjunction of *Invs* and *Linear*) is found and the triple is verified.

Now we report our experimental results on some other arithmetic domains. We can see in Table 1 that every domain has two rows in its Invariant column. They are *Invs* and *Linear*, and their meanings are as discussed in *1D* domain. Recall that to generate the invariants, our system firstly tries to discover linear and inequality relations with more efficient method. We can see the results of *Arith* and *Addition* in Table 1 that this process is efficient. If there is no any useful linear relations, the system then will go on to call the predicate abstraction method which can discover more general invariants, and it will still be a fast process if the predicate set is small and the length of the resulted clause is short, such as the case in *1D*. But it will cost lots of time if the predicate set is large and the resulted clause is complex, such as the cases in *Find* and *Sort*.

The failure of *PrizeA1* is mainly because its verification requires a conjunction of two clauses ($\varphi_1 \wedge \varphi_2$) to strengthen the given formula by doing the abstraction only once, but Algorithm 1 can merely output φ_i which can not be found here because neither

Hoare-triple:

```

{xpos = 1}
while xpos < len do
   $\pi(x_1, x_2 \in \text{int}. \langle x_{\text{pos}} = x_1 \wedge x_1 + 1 = x_2 \rangle?)$ 
   $\langle \text{in}(x_1) < \text{in}(x_2) \rangle?; \text{move\_right}$ 
   $\langle \neg(\text{in}(x_1) < \text{in}(x_2)) \rangle?; \text{swap}(x_1, x_2); \text{re\_set}$ 
od
 $\{\forall a1, a2 \in \text{int}. a1 < a2 \supset \text{in}(a1) < \text{in}(a2)\}$ 

```

Action Precondition Axiom:

$$\text{Poss}(\text{move_right}, s) \equiv \text{xpos}(s) < \text{len}$$

$$\text{Poss}(\text{re_set}, s) \equiv 0 < \text{xpos}(s) < \text{len}$$

$$\text{Poss}(\text{swap}(x, y), s) \equiv \text{true}$$
Successor State Axiom:

$$\text{xpos}(\text{do}(a, s)) = y \equiv \text{xpos}(s) = y \vee y = 1 \wedge a = \text{re_set} \vee$$

$$\text{xpos}(s) = y - 1 \wedge a = \text{move_right}$$

$$\text{in}(x, \text{do}(a, s)) = y \equiv$$

$$\text{in}(x, s) = y \vee \exists z. \text{in}(z, s) = y \wedge a = \text{swap}(x, z)$$

$$\text{Invs}: \forall a1, a2 \in \text{int}. a1 < a2 < \text{xpos} + 1 \supset \text{in}(a1) \leq \text{in}(a2)$$

$$\text{Linear}: \{0 < a1, 0 < \text{xpos}\}$$
Figure 2. Example of the *Sort* Domain

Domain	Invariant	Time
1D	$\forall x \in \text{int}. \text{xpos} < x \supset \text{vis}(x)$ $\{\text{xpos} < \text{len} + 1\}$	1.0
PrizeA1	-	-
Arith	$\text{numy} = 10 \vee i < N$ $\{0 \leq i, i < N + 1,$ $-10 \cdot i + 5 \cdot \text{numy} = 0\}$	1.0
Find	$\forall i \in \text{int}. i < \text{xpos} \supset$ $\neg \text{inb}(i) \vee \text{ina}(i) = 0$ $\{0 \leq \text{xpos}, \text{xpos} < \text{length} + 1\}$	444.0
Addition	inner $\text{sum} = A \cdot B \vee$ $i + 1 < A \vee j < B$ $\{0 \leq i, i < A, 0 \leq j,$ $j < B + 1, 0 \leq \text{sum},$ $i \leq \text{sum}, j \leq \text{sum},$ $20 \cdot i + 4 \cdot j - 4 \cdot \text{sum} = 0\}$	2.0
	outer $\text{sum} = A \cdot B \vee i < A$ $\{-25 \cdot i + 5 \cdot \text{sum} = 0\}$	
Sort	$\forall a1, a2 \in \text{int}. a1 < a2 < \text{xpos} + 1 \supset$ $\text{in}(a1) \leq \text{in}(a2)$ $\{0 < a1, 0 < \text{xpos}\}$	230.0

Table 1. Performance of the Arithmetic Domains

$\varphi_1 \supset \phi$ nor $\varphi_2 \supset \phi$ holds. If we adjust Algorithm 1 to search the conjunctions of clauses, its search space will explode.

5.2 Comparisons on the Non-Arithmetic Domains

In the experiments of non-arithmetic domains, we also manually provide the state constraints in advance. The constraints we use here are exactly those reported in our previous work [13], where all the constraints are automatically discovered and verified.

We can see in Table 2 that our method can successfully cover all the domains used in [13]. #A is the number of all Hoare-triples we tested, and T.avg and T.max are the average and maximal time costs of all verifications. Time costs are measured in seconds. All verifications in domains *CornerA*, *Transport* and *Trash* are trivial, i.e., during the extended regression, every $\hat{\mathcal{R}}_{\mathcal{D}}[\phi(s), \text{while } \varphi \text{ do } \delta \text{ od}]$ returns $\phi(s) \vee \varphi[s]$ as its loop invariant, so T.maxs and T.avg are reported to be 0. Because our method can generate more general predicates during the verification, it can discover more general invariants theoretically. But as we can see in Table 2 that the

generality is at the cost of efficiency due to the large search space.

When comparing our new experimental results with the previous ones, both machines we use are equipped with 4.00GB RAM and operate under Linux. But our new machine is equipped with an Intel i5, 3.30 GHz CPU, while the old one uses i7, 2.60 GHz CPU.

Domain	Old Method			New Method		
	#A	T.max	T.avg	#A	T.max	T.avg
CornerA	1	1.0	1.0	1	0	0
Delivery	4	3.0	1.5	4	514.0	150.2
Green	3	46.0	24.7	2	1388.0	1062.0
Gripper	3	6.0	3.0	3	2138.0	815.7
Logistics	2	7.0	6.0	2	23.0	11.5
Recycle	2	18.0	15.5	2	964.0	483.5
Transport	2	1.0	0.5	2	0	0
Trash	2	0	0	2	0	0

Table 2. Performance of the Non-Arithmetic Domains**6 Applications in Strategy Verification**

In this section we use our method to verify the winning property and executability of strategies in combinatorial games.

Giving a strategy, our first job is to encode it into a Golog program δ that can be recognized by our system.

Definition 9 A strategy S of a player A is a finite set (with size m) that denotes the moves of A under some given conditions, i.e., every element of S is a pair $\langle \phi_i, \sigma_i \rangle$ which means that A will choose σ_i as her next move under the condition of ϕ_i , where ϕ_i is a formula that is true in the given game state, and σ_i is an action according to ϕ_i .

Then a winning strategy (WS) of A is a strategy such that no matter how the opponent plays, A is guaranteed to win. It can be encoded into a Golog program δ_{WS} as below:

$$\bullet \delta_{WS} \doteq [\phi_1?; \sigma_1] \dots [\phi_m?; \sigma_m]$$

For example, in the *Pick-up Stone* game, there are two players A and B taking turns to pick up i stones on the table ($i \in \{1, 2, 3\}$). We use the variable n to denote the number of stones left on the table ($n > 0$ initially). The one who picks the last stone loses the game. We say that A has a winning strategy if n satisfies the property $(n\%4 \neq 1)$ every time in her turn. Then the WS of A should be $\{(n\%4 = 0, \text{pick}(3)), \langle n\%4 = 2, \text{pick}(1) \rangle, \langle n\%4 = 3, \text{pick}(2) \rangle\}$.

6.1 Partial Correctness of WS

To verify the partial correctness of WS , we should construct a Hoare-triple $\{P\} \delta_P \{Q\}$, where $\{P\}$ is the pre-condition that always denotes the initial state of the game, $\{Q\}$ is the post-condition that we call win-condition, and δ_P is as below:

$$\bullet \delta_{P_1} \doteq \text{while } \varphi \text{ do } (\pi \vec{X}) a(\vec{X}); \delta_{WS} \text{ od, or}$$

$$\bullet \delta_{P_2} \doteq \text{while } \varphi \text{ do } \delta_{WS}; (\pi \vec{X}) a(\vec{X}) \text{ od}$$

where φ is the termination condition of the game, $a(\vec{X})$ is the action in the game, and the choice of δ_{P_1} or δ_{P_2} depends on which player moves first. Specifically, if the player of the strategy being verified moves first we will use δ_{P_2} , and otherwise we will use δ_{P_1} .

Take the previous *Pick-up Stone* game as an example, the $\{P\} \delta_P \{Q\}$ of A can be encoded as:

$$\{n\%4 \neq 1 \wedge \text{turn}(A)\} \delta_{P_2} \{\text{turn}(A)\}$$

Domain	Partial Correctness		Executability	
	Inv	T	Inv	T
<i>Pick-up Stone</i>	$(n > 0 \vee \text{turn}(A)) \wedge$ $(n \% 4 = 0 \vee \text{turn}(B)) \wedge$ $(n \% 4 \neq 0 \vee n = 0)$	13.0	$\forall x \in \text{int.}$ $(n \geq x \wedge (x = 1 \vee x = 2 \vee x = 3)) \supset$ $(n - x) \% 4 = 0 \vee (n - x) \% 4 = 2 \vee (n - x) \% 4 = 3$	3.0
<i>Chomp</i> $2 \times N$	$(\text{length}(\text{row}[1]) > 0 \vee \text{turn}(A)) \wedge$ $(\text{length}(\text{row}[1]) = 0 \vee \text{turn}(B))$	6.0	#T	1.0
<i>Chomp</i> $N \times N$	$(\text{length}(\text{row}[1]) = 0 \vee \text{turn}(A)) \wedge$ $(\text{length}(\text{row}[1]) = 0 \vee \text{turn}(B))$	8.0	#T	1.0

Table 3. Experimental Results of WS s

where $\delta_{P_2} \doteq \mathbf{while} \ n > 0 \ \mathbf{do} \ \delta_{WS}; (\pi x) \text{pick}(x); \mathbf{od}$, and $\delta_{WS} \doteq [n \% 4 = 0?; \text{pick}(3)][n \% 4 = 2?; \text{pick}(1)][n \% 4 = 3?; \text{pick}(2)]$. Here the win-condition is $\{\text{turn}(A)\}$ because after B picking the last stone it should be A 's turn.

Nevertheless, how does it make sure that every time in her turn, A (the winner) can take a move according to her WS , i.e., every time in A 's turn, the game state s can entail the disjunction of situations in her WS ($s \supset \bigvee \phi_i$)? And is it enough to say that WS is really a winning strategy of A by only proving the partial correctness of $\{P\} \delta_P \{Q\}$? The answer is no. Assume that we change the WS of A to a wrong WS' by replacing its first element $\langle n \% 4 = 0, \text{pick}(3) \rangle$ by $\langle n \% 4 = 0, \text{pick}(2) \rangle$, there will be a counterexample if B performs $\text{pick}(1)$ under the pre-condition of $\{n = 4 \wedge \text{turn}(A)\}$, which leaves only one stone to A 's last turn. Since A has no strategy to deal with such situation in WS' , the executability of δ_P will be invalid.

Therefore, in order to ensure the correctness of a WS , it is the executability that should also be taken into consideration.

6.2 Executability of WS

Since we have specified WS , we can use our system to verify the executability of WS too. But now, the Hoare-triple may be a little different. Because it is the executability that we want to verify, the post-condition should be the disjunction of situations in WS , and the while loop in δ_P should be replaced by a non-deterministic loop. With the same pre-condition, now the Hoare-triple is $\{P\} \delta_E \{\bigvee \phi_i\}$, where δ_E is:

- $\delta_{E_1} \doteq [(\pi \vec{X})a(\vec{X}); \delta_{WS}]^*; (\pi \vec{X})a(\vec{X})$
- $\delta_{E_2} \doteq [\delta_{WS}; (\pi \vec{X})a(\vec{X})]^*$

Similarly, which of the two should be used depends on which player moves first. If player being analysed moves first we will use δ_{E_2} , and otherwise we will use δ_{E_1} .

When referring to the WS of A in the *Pick-up Stone* game, we can encode its Hoare-triple $\{P\} \delta_E \{Q\}$ as:

$$\{n \% 4 \neq 1 \wedge \text{turn}(A)\} \delta_{E_2} \{n \% 4 = 0 \vee n \% 4 = 2 \vee n \% 4 = 3\}$$

where $\delta_{E_2} \doteq [\delta_{WS}; (\pi x) \text{pick}(x)]^*$ and δ_{WS} as before.

6.3 Summary of Experiments

Besides *Pick-up Stone*, we have tried another game *Chomp* [15], which has two versions as below:

Chomp $2 \times N$: There is a $2 \times N$ grid, and each point contains a cookie. The positions of them can be represented as a two dimensional array $\text{row}[2][N]$ that starts from 1, and the one in $\text{row}[1][1]$ is poisonous. There are two players A and B taking turns to pick one of the remaining cookies, and once she picks $\text{row}[a][b]$, she must eat all the cookies in $\text{row}[i][j]$, where $a \leq i \leq 2$ and $b \leq j \leq N$. The one who has to eat the poisonous cookie loses the game. We say

that the first player A has a winning strategy by keeping the number of the first row of cookies one greater than the number of the second row every time after her move.

Chomp $N \times N$: This game is similar to *Chomp $2 \times N$* , but this time there are N rows of cookies initially. Player A has a winning strategy too, that is to eat $\text{row}[2][2]$ in her first step and then keep the length of the first row equivalent to the length of the first column every time after her move.

Table 3 shows that the WS s of these three games are correct. And interestingly, when proving their partial correctness, the discovered invariants are much simpler than we thought (for example in *Pick-up Stone*, the intuitive invariant is $n = 0 \supset \text{turn}(A) \wedge n > 0 \supset \text{turn}(B) \wedge n \% 4 = 1$), but the results are valid according to the proof system introduced in [1]. #T means that a verification is trivial, i.e., during the extended regression, every $\mathcal{R}_{\mathcal{D}}[\phi(s), \mathbf{while} \ \varphi \ \mathbf{do} \ \delta \ \mathbf{od}]$ returns $\phi(s) \vee \varphi[s]$ as its loop invariant.

7 Conclusion

In this paper, we propose a uniform method to verify the partial correctness of Golog programs that may involve functions and arithmetic. We summarize our main contributions as follows: Firstly, we combine the extended regression with the predicate abstraction method, which results in a uniform verification method that is capable of handling programs with functions and arithmetic. Secondly, to make our approach effective and feasible, we develop some techniques like generating predicates, generating bounds and efficiently discovering linear and inequality invariants. Thirdly, we have implemented a verification system, conducted a set of experiments and compared it with our previous work, showing the capability and potential of our approach. Lastly, we have also applied the method to the verification of winning strategies in two famous games, *Pick-up Stone* and *Chomp*.

As for the future work, we would like to further improve the performance of our system and make the approach more practical. One possible direction is to introduce human-machine interaction to the process of predicate abstraction. For example, a user may analyse the output of the system and add new predicates into the candidate set, or provide some invariant templates to accelerate the search. Another direction we have in mind is to combine our method with algorithmic learning, such as applying the CDNF algorithm as in the work of [11], where the intended loop invariant is learned by interacting with a mechanical teacher. We believe that by posing informative queries during the interaction, the algorithmic learning method can guide and thus speed up the search process.

Acknowledgments

We thank the anonymous reviewers for helpful comments. This work received support from the Natural Science Foundation of China under Grant No. 61572535.

REFERENCES

- [1] Krzysztof R. Apt, Frank S. de Boer, and Ernst-Rüdiger Olderog, *Verification of Sequential and Concurrent Programs*, Texts in Computer Science, Springer, 2009.
- [2] Thomas Ball, Rupak Majumdar, Todd D. Millstein, and Sriram K. Rajamani, ‘Automatic predicate abstraction of C programs’, in *Proceedings of the 2001 ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), Snowbird, Utah, USA, June 20-22, 2001*, pp. 203–213, (2001).
- [3] Edmund M. Clarke, Orna Grumberg, Somesh Jha, Yuan Lu, and Helmut Veith, ‘Counterexample-guided abstraction refinement’, in *Computer Aided Verification, 12th International Conference, CAV 2000, Chicago, IL, USA, July 15-19, 2000, Proceedings*, (2000).
- [4] Edmund M. Clarke, Orna Grumberg, Somesh Jha, Yuan Lu, and Helmut Veith, ‘Counterexample-guided abstraction refinement for symbolic model checking’, *J. ACM*, **50**(5), 752–794, (2003).
- [5] Jens Claßen and Gerhard Lakemeyer, ‘A logic for non-terminating golog programs’, in *Principles of Knowledge Representation and Reasoning: Proceedings of the Eleventh International Conference, KR 2008, Sydney, Australia, September 16-19, 2008*, pp. 589–599, (2008).
- [6] Jens Claßen, Martin Liebenberg, Gerhard Lakemeyer, and Benjamin Zariß, ‘Exploring the boundaries of decidable verification of non-terminating golog programs’, in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 1012–1019, (2014).
- [7] Leonardo Mendonça de Moura and Nikolaj Bjørner, ‘Z3: an efficient SMT solver’, in *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings*, pp. 337–340, (2008).
- [8] Michael D. Ernst, Jeff H. Perkins, Philip J. Guo, Stephen McCamant, Carlos Pacheco, Matthew S. Tschantz, and Chen Xiao, ‘The daikon system for dynamic detection of likely invariants’, *Sci. Comput. Program.*, **69**(1-3), 35–45, (2007).
- [9] Cormac Flanagan and Shaz Qadeer, ‘Predicate abstraction for software verification’, in *Conference Record of POPL-02*, pp. 191–202, (2002).
- [10] Yuxiao Hu, *Generation and Verification of Plans with Loops*, Ph.D. dissertation, Department of Computer Science, University of Toronto, 2012.
- [11] Wonchan Lee, Bow-Yaw Wang, and Kwangkeun Yi, ‘Termination analysis with algorithmic learning’, in *Computer Aided Verification - 24th International Conference*, pp. 88–104, (2012).
- [12] Hector J. Levesque, Raymond Reiter, Yves Lespérance, Fangzhen Lin, and Richard B. Scherl, ‘GOLOG: A logic programming language for dynamic domains’, *Journal of Logic Programming*, **31**, 59–83, (1997).
- [13] Naiqi Li and Yongmei Liu, ‘Automatic verification of partial correctness of golog programs’, in *Proc. of IJCAI-15*, pp. 3113–3119, (2015).
- [14] Yongmei Liu, ‘A hoare-style proof system for robot programs’, in *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence, July 28 - August 1, 2002, Edmonton, Alberta, Canada.*, pp. 74–79, (2002).
- [15] Richard J Nowakowski, *Games of no chance*, volume 29, Cambridge University Press, 1998.
- [16] Raymond Reiter, *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*, MIT Press, 2001.
- [17] Saurabh Srivastava and Sumit Gulwani, ‘Program verification using templates over predicate abstraction’, in *PLDI*, pp. 223–234, (2009).
- [18] Westley Weimer ThanhVu Nguyen, Deepak Kapur and Stephanie Forrest, ‘Using dynamic analysis to discover polynomial and array invariants’, in *ICSE*, pp. 683–693, (2012).
- [19] Benjamin Zariß and Jens Claßen, ‘Verifying CTL* properties of GOLOG programs over local-effect actions’, in *ECAI 2014 - 21st European Conference on Artificial Intelligence*, pp. 939–944, (2014).

An Assessment Study of Features and Meta-level Features in Twitter Sentiment Analysis

Jonnathan Carvalho^{1,2} and Alexandre Plastino²

Abstract. Sentiment analysis is the task of determining the opinion expressed on subjective data, which may include microblog messages, such as tweets. This type of message has been considered the target of sentiment analysis in many recent studies, since they represent a rich source of opinionated texts. Thus, in order to determine the opinion expressed in tweets, different studies have employed distinct strategies, which mainly include supervised machine learning methods. For this purpose, different kinds of features have been evaluated. Despite that, none of the state-of-the-art studies has evaluated distinct categories of features, regarding their similar characteristics. In this context, this work presents a literature review of the most common feature representation in Twitter sentiment analysis. We propose to group features sharing similar aspects into specific categories. We also evaluate the relevance of these categories of features, including meta-level features, using a significant number of Twitter datasets. Furthermore, we apply important and well-known feature selection strategies in order to identify relevant subsets of features for each dataset. We show in the experimental evaluation that the results achieved in this study, using feature selection strategies, outperform the results reported in previous works for the most of the assessed datasets.

1 Introduction

Sentiment analysis has been extensively used to determine the overall opinion expressed about different targets in many types of user-generated documents, generally on the Web, such as user reviews, blog comments, etc. Many companies have taken advantage of the area of sentiment analysis by automatically extracting the opinions expressed by consumers about their products and services, eliminating the need of extensive and expensive researches.

With the advent of social media and microblog platforms, such as Twitter³, not only the opinion about products and services can be tracked, but also the sentiment expressed about entities in real time events, such as politics debates, world disasters, etc. Twitter is a microblog platform, in which users can send short messages about any topic, limited to 140 characters. In order to determine the sentiment expressed in this type of message, the so-called tweets, different approaches have been proposed in the literature of Twitter sentiment analysis. These approaches mainly include supervised machine learning strategies and they usually focus on the polarity classification of tweets, that is, whether the sentiment expressed on them carries a positive or negative connotation.

Supervised machine learning strategies aim at determining the sentiment expressed in tweets by exploring their contents in order to learn characteristics, commonly referred to as features, that can distinguish the positive tweets from the negative ones. The most usual feature representation in the task of sentiment classification of tweets is the bag-of-words model, first employed by Go et al. [17], in which each token of a tweet is taken as a feature. Regarding the challenging nature of tweets, such as misspelling words and the 140-character limit of each message, different studies have proposed other types of features, trying to improve the performance of the sentiment classification. These features include, in their vast majority, n -grams [2, 6, 7, 12, 13, 17, 19, 22, 24, 25, 27, 29, 30, 38, 40, 44], part-of-speech tags [2, 5, 8, 17, 24, 27], punctuation [2, 5, 13, 19, 22, 27], specific characteristics of Twitter and microblog messages [2, 5, 19, 22, 24, 27, 44], and lexicon-based features [2, 8, 12, 19, 22, 23, 24, 27].

Although these features have been successfully employed in the sentiment classification of tweets, none of the state-of-the-art studies has organized into categories the large set of features used in the sentiment classification of tweets from distinct domains, including the most common features and meta-level features that have already been proposed in the literature. In this context, considering that many of these features share similar characteristics, we propose to group them into different categories, i.e., features that are similar in structural aspects make up the same category. Then, we investigate whether the classification of tweets from different domains can benefit from distinct categories of features and meta-level features. In addition, we explore the application of feature selection strategies in order to identify relevant subsets of features for each evaluated dataset.

The main contributions of this study are the following.

1. We present an extensive literature review of the most common feature representation of tweets for supervised sentiment classification, including meta-level features, which have been proposed and employed in a relevant set of well-referenced works.
2. We propose to group the features and meta-level features sharing similar aspects into specific categories, in order to investigate the importance of each of these categories in the sentiment classification of tweets from distinct domains.
3. We use a collection of sixteen Twitter datasets in the experimental evaluation of the categories of features and meta-level features. To the best of our knowledge, this is the first study that evaluates distinct categories of features and meta-level features for a significant number of Twitter datasets.
4. In order to identify relevant subsets of features for each dataset, we apply important feature selection strategies on the full set of features, including measures such as Information Gain, Chi-Squared, and Relief-F.

¹ Instituto Federal Fluminense, Itaperuna, Brazil, email: joncarv@iff.edu.br

² Universidade Federal Fluminense, Institute of Computing, Niterói, Brazil, emails: joncarv@ic.uff.br, plastino@ic.uff.br

³ <http://twitter.com>

5. We show that the results achieved in this study, using feature selection strategies, outperform previous results reported in the literature of Twitter sentiment analysis for the most of the assessed datasets.

The remainder of this work is organized as follows. Section 2 presents the related work on sentiment classification, focusing on the different types of features explored in the literature. In Section 3, we present the most common features and meta-level features identified in a set of well-referenced works. Besides, we propose the categorization of the features, based on their similar characteristics. The experimental evaluation of the different categories of features are reported in Section 4, as well as the results of the application of the feature selection strategies. Finally, in Section 5, we present the conclusions of this study and some future work.

2 Related Work

Sentiment analysis has been widely employed to determine the polarity of subjective data, that is, whether the sentiment expressed in opinionated text (movie reviews, blogs, microblogs, etc.) has a positive or negative connotation. For this purpose, different approaches have already been proposed, which mainly include supervised machine learning methods and lexicon-based strategies.

Regarding supervised machine learning methods, which are the main focus of this study, the precursor work by Pang et al. [31] applies different algorithms in the classification of movie reviews. In supervised methods, to classify a review as being positive or negative, a training dataset is used, which consists of pieces of texts, generally represented as a feature vector, whose polarities are already known. Such feature vector may contain relevant characteristics of each piece of text in the training dataset, the so-called features. In [31], in addition to the use of different machine learning algorithms (Naive Bayes, SVM, and Maximum Entropy), they also evaluate distinct sets of features, such as unigrams (bag-of-words), bigrams, and part-of-speech (POS) tags. Their experimental results show that better performance is achieved using only unigrams as features, and they conclude it is worthwhile to explore the data, in order to select good indicator features for sentiment classification.

More recent works explore the applicability of feature selection methods, attempting to improve the sentiment classification of movie reviews [3, 37]. Indeed, Sharma and Dey [37] show that feature selection may improve the performance of sentiment classification, although the improvement is dependent on the feature selection method employed and the number of features selected. Similarly, Agarwal and Mittal [3] study the effect of different feature selection methods and various sets of features from datasets of reviews in two distinct domains, such as movies and products. They show that features created from unigrams and bigrams achieves better results when compared to the use of unigrams or bigrams individually. Differently from [3] and [37], the strategy described in [1] does not evaluate distinct feature selection methods. In [1], Abbasi et al. propose a genetic algorithm that incorporates the Information Gain measure for feature selection on a corpus of movie reviews, using stylistic and syntactic features, such as word length distributions, and special character frequencies.

Beyond the classification of opinions expressed in movie and product reviews, the sentiment expressed in other types of documents has been evaluated, such as in tweets. The method presented by Go et al. [17] classifies the sentiment of tweets using a distant supervision approach, which relies on emoticons as noisy labels in a training dataset of 1,600,000 tweets. They also evaluate the performance of different sets of features, such as unigrams, bigrams, both unigrams

and bigrams, and part-of-speech tags. The experimental evaluation, in [17], shows that unigrams and bigrams together represent a good set of features for the sentiment classification of tweets.

Inspired by Go et al. [17], many other approaches explore the use of n -grams in Twitter sentiment classification [2, 6, 7, 12, 13, 17, 19, 22, 24, 25, 27, 29, 30, 38, 40, 44]. For example, Davidov et al. [13] use higher order n -grams, such as bigrams, trigrams, 4-grams, and 5-grams as features in the classification process. To avoid data sparsity, due to the use of different range of n -grams, they only extract n -grams features from words which have a training set frequency over 0.5%.

Although n -grams have been a kind of feature largely evaluated in the literature, some works incorporate different types of features in the sentiment classification task, such as punctuation, microblog and Twitter-specific features, and lexicon-based features [2, 5, 8, 12, 13, 19, 22, 23, 24, 27, 44]. They claim that using only n -grams may not be sufficient for this task, since tweets are very noisy and short messages. For example, Kouloumpis et al. [24] take into account the presence of positive and negative emoticons, as well as abbreviations and intensifiers (all-caps and character repetitions). They point out that the best performance on their experimental results comes from using n -grams together to the lexicon-based and the microblog features. In addition to the use of different sets of features, in [24], a feature selection method is applied in order to select only the 1,000 most representative n -grams for the classification of tweets.

Feature selection methods have also been tried in Twitter sentiment classification [8, 33], as mentioned before, concerning the vast amount of redundancy features and to reduce the feature space. For example, Bravo-Marquez et al. [8] first compute the information gain of the proposed set of lexicon-based meta-level features. Then, they test combinations of feature subsets, selecting arbitrary sets of features with a best-first strategy, based on the information gain of the primarily evaluated features, on three datasets of tweets. Prusa et al. [33] evaluate the impact of feature selection techniques on a corpus of 3,000 tweets, using only unigrams as features. In their experimental evaluation, they applied ten different filter-based feature selection techniques and ten different sizes of the feature subsets, varying from 5 to 200, and they show that feature selection methods can be effective in the sentiment classification of tweets.

Despite the previous use of feature selection methods in Twitter sentiment classification, to the best of our knowledge, this work is the first study that evaluates the importance of distinct categories of features and meta-level features for a significant number of Twitter datasets. These features and meta-level features, which include n -gram-based features, microblog and Twitter-specific features, part-of-speech tags, punctuation features, and lexicon-based features, were identified in a set of well-referenced works in the literature of supervised sentiment classification of tweets, after an extensive literature review. We also study the effect of the application of different feature selection strategies on the full set of features, in order to identify relevant subsets of features for each evaluated dataset.

3 Features and Meta-level Features

Different types of features have been engineered and used in Twitter sentiment analysis, from the most common representation, such as n -gram-based features, to meta-level features. Meta-level features are usually extracted from other features, and can capture insightful new information about the data [9]. In this study, we consider merely as features the information that can be extracted primarily from tweets, such as the presence or absence of some particular characteristic in a tweet. In the other hand, we consider as meta-level features those

referred to counts and summations, which are, in general, secondary information extracted from tweets. For readability reasons, meta-level features are referred to hereafter as meta-features.

In this section, we describe the features and meta-features we have examined in a set of well-referenced works in supervised sentiment classification of tweets. These works were identified after an extensive literature review, from which we have detected the most common types of features and meta-features used to determine the sentiment expressed in tweets. In order to describe and evaluate these features and meta-features, as shown in Table 1, we have grouped them into five categories, namely N-grams, Twitter and Microblog, Part-of-Speech, Punctuation, and Polarity, so that features that share structural aspects fall into the same category.

Table 1. Categories of features employed in the literature of supervised sentiment classification of tweets.

Reference	<i>f</i> 1	<i>f</i> 2	<i>f</i> 3	<i>f</i> 4	<i>f</i> 5
Agarwal et al. [2]	✓	✓	✓	✓	✓
Barbosa and Feng [5]		✓	✓	✓	
Birmingham and Smeaton [6]	✓				
Bifet et al. [7]	✓				
Bravo-Marquez et al. [8]			✓		✓
da Silva et al. [12]	✓				✓
Davidov et al. [13]	✓			✓	
Go et al. [17]	✓		✓		
Hagen et al. [19]	✓	✓		✓	✓
Jiang et al. [22]	✓	✓		✓	✓
Khuc et al. [23]	✓			✓	✓
Kouloumpis et al. [24]	✓	✓	✓		✓
Lin et al. [25]	✓				✓
Mohammad et al. [27]	✓	✓	✓	✓	
Narr et al. [29]	✓				
Pak and Paroubek [30]	✓				
Speriosu et al. [38]	✓				
Wang et al. [40]	✓				
Zhang et al. [44]	✓	✓			

*f*1 N-grams *f*2 Twitter and Microblog *f*3 POS *f*4 Punctuation *f*5 Polarity

3.1 N-grams Features

N-grams features are contiguous sequences of n tokens from a text. The n -gram-based features were first employed in sentiment classification of tweets by Go et al. [17]. Since then, this category of features has been the one most used by supervised sentiment learning strategies [2, 6, 7, 12, 13, 17, 19, 22, 24, 25, 27, 29, 30, 38, 40, 44].

In the bag-of-words model, that is, the unigram category ($n = 1$), each word or token is used as a feature. It is the basic representation of a tweet for the classification process, and it is adopted by many strategies [2, 6, 7, 12, 13, 17, 19, 22, 24, 27, 29, 30, 38, 40, 44]. In an attempt to capture more sentiment expressions, some studies have varied the value of n . For example, Davidov et al. [13] vary the value of n from 1 to 5, which means that each consecutive word sequence containing one to five words is taken as a feature. Table 2 presents an overview of the n -grams features used in the literature of Twitter sentiment classification.

3.2 Twitter and Microblog Features

The Twitter and Microblog category refers to those features related to the syntax and vocabulary used in tweets and microblog messages, as used in [2, 5, 19, 22, 24, 27, 44]. More specifically, some characteristics of how microblog posts are written may be good indicators of sentiment, such as emoticons and internet slang present in the vocabulary of this type of text. Furthermore, Twitter-specific tokens, such as

Table 2. Overview of the n -grams features used in Twitter sentiment classification.

Reference	$n = 1$	$n = 2$	$n = 3$	$n = 4$	$n = 5$
Agarwal et al. [2]	✓				
Birmingham and Smeaton [6]	✓	✓	✓		
Bifet et al. [7]	✓				
da Silva et al. [12]	✓				
Davidov et al. [13]	✓		✓	✓	✓
Go et al. [17]	✓	✓			
Hagen et al. [19]	✓	✓	✓	✓	
Jiang et al. [22]	✓	✓			
Kouloumpis et al. [24]	✓	✓			
Lin et al. [25]				✓	
Mohammad et al. [27]	✓	✓	✓	✓	
Narr et al. [29]	✓	✓			
Pak and Paroubek [30]	✓	✓	✓		
Speriosu et al. [38]	✓	✓			
Wang et al. [40]	✓				
Zhang et al. [44]	✓				

user mentions (followed by the special character @), retweets (indicated by RT), URLs, and hashtags (followed by the special character #) have also been explored in the literature.

Twitter hashtags, which are often used as keywords for tweets, are a very informative mechanism. Thus, they may be a good evidence of positive or negative sentiment, as employed in [2, 5, 19, 22, 27, 44]. Similarly, others Twitter-specific tokens are taken as features in the literature, such as the presence of user mentions and retweets [5].

Regarding the 140-character limit of tweets, a very common trick established among Twitter users is the use of word shortcuts and internet slang (for example, “love” becomes “luv”). Another interesting aspect of tweets is the use of repeated letters as intensifiers (for example, in “loooooove”). Thus, some works have defined these characteristics as meta-features as well [2, 19, 24, 27].

In this context, for this category, we identified the following set of nine features and meta-features, employed in the literature [2, 5, 19, 22, 24, 27, 44].

- *Whether the tweet has*: retweet, hashtag, user mentions, URL, emoticon, internet slang, repeated letters.
- *Number of*: internet slang, repeated letters.

3.3 Part-of-Speech Features

Although some studies have already acknowledged that part-of-speech (POS) features are not useful for sentiment classification [17, 31], this category of features is still used to determine the sentiment of tweets, in combination with other features [2, 5, 8, 17, 24, 27]. For example, assuming that some adjectives and verbs are good indicators of positive and negative sentiment, Barbosa and Feng [5] map each word in a tweet to its POS, using a POS database, which can identify nouns, verbs, adjectives, adverbs, interjections, and others. Similarly, Agarwal et al. [2] consider the number of adjectives, adverbs, verbs, and nouns as features. In order to capture the informal aspects of tweets, some works [8, 27] use a POS tagset, presented in [16], to identify some special characteristics of short and noisy texts, such as misspelling words.

In this context, for this category, we identified the following set of twenty-five features in the literature [2, 5, 8, 17, 24, 27].

- *Number of*: common noun, proper noun, personal pronoun, common noun + possessive, common noun + verb, proper noun + possessive, proper noun + verb, verb, adjective, adverb, interjection, determiner, pre or post-position, conjunction, verb particle, predeterminer, predeterminer + verb, hashtag, user mention, discourse marker (“RT” and “:” in retweet), URL or email address, emoticon, numeral, punctuation, abbreviation or symbol.

3.4 Punctuation Features

Punctuation may also play an important role in sentiment detection of microblog messages. For example, Bermingham and Smeaton [6] observed that the exclamation mark is the most discriminative unigram according to the Information Gain measure, in a corpus of 1,000 tweets labeled as being positive and negative. They also point out that the question mark and sequences of exclamation marks (for example, as “!!!”) are in the top 10 most relevant features.

In this context, punctuation features have also been explored in the literature [2, 5, 13, 19, 22, 27]. The most usual meta-features in this category are the number of exclamation and question marks, as appearing in [2, 5, 13, 19, 22]. The total count of quotes in tweets has also been used [13]. Some works have already proposed more sophisticated meta-features, such as the number of contiguous sequences of exclamation and question marks [19, 27], regarding their use in microblog messages to convey intonation. Therefore, to make out this category of features, we identified the following set of ten features and meta-features [2, 5, 13, 19, 22, 27].

- *Whether the tweet has*: question mark, exclamation mark.
- *Whether last token contains*: question mark, exclamation mark.
- *Number of*: question mark, exclamation mark, sequence of question marks, sequence of exclamation marks, sequence of both question and exclamation marks, quotes.

3.5 Polarity Features

A different manner of exploring the content of tweets, in order to determine the sentiment expressed in them, is from using existing sentiment lexical resources or dictionaries in the literature. These lexicons consist of lists of words with positive and negative terms, such as Bing Liu’s opinion lexicon [26], NRC-emotion [28], and OpinionFinder lexicon [43], as well as lexical resources containing words and phrases that are scored on a range of real values, such as SentiWordNet (SWN) [4], NRC-hashtag [27], and Sentiment140 lexicon (Sent140) [27]. Meta-features of this category have been widely explored in sentiment classification of tweets [2, 8, 12, 19, 22, 23, 24, 27], especially the total count of positive and negative words.

The polarity of emoticons may also be another relevant characteristic for Twitter sentiment analysis. Since emoticons are used by microblog users to summarize the sentiment they intend to communicate, some works have also extracted meta-features from emoticons, such as the number of positive and negative emoticons in a tweet, as employed in [2, 12, 19, 27].

Regarding negation, it has already been acknowledged it can affect the polarity of an expression [42]. Indeed, the expression “not good” is the opposite of “good”. In this context, an interesting meta-feature proposed in the literature to handle negation is the number of negated contexts [27]. Mohammad et al. [27] have defined a negated context as a segment of a tweet that starts with a negation word, such as “shouldn’t”, and ends on the first punctuation mark after the negation word. Moreover, in [27], negated contexts change the n -gram-based features, that is, they add the tag `_NEG` on each token into a negated context. More specifically, in a negated context, Mohammad et al. concatenate the tag `_NEG` to every token between the negation word and the first punctuation mark after it. For example, in the sentence “He isn’t a great book writer, but I read his books.”, the unigrams “great”, “book”, and “writer” become “great_NEG”, “book_NEG”, and “writer_NEG”, respectively.

Considering the polarity features identified in the literature [2, 8, 12, 19, 22, 23, 24, 27], the following features and meta-features compose

this category.

- *Whether the tweet has*: positive emoticon, negative emoticon.
- *Whether the last token is*: positive emoticon, negative emoticon.
- *Number of*: positive emoticon, negative emoticon, extremely positive emoticon, extremely negative emoticon, positive adjective, negative adjective, positive noun, negative noun, positive adverb, negative adverb, positive verb, negative verb, negated contexts.
- *Sum of the scores of the adjectives, adverbs, verbs, and nouns*.
- For each of the six aforementioned sentiment lexicons:
 - *Number of*: positive words, negative words.
 - *Total score of*: positive words, negative words.
 - *Score of*: last token.
 - *Maximal score of*: positive words, negative words.

3.6 Miscellaneous

Some other features reported in the literature [2, 5, 13, 19, 24, 27] do not fit in any of the aforementioned categories. Thus, we have created this category to put these features together. They include the presence of abbreviations, the number of capitalized text, and the number of words in a tweet, as follows:

- *Whether the tweet has*: abbreviation.
- *Number of*: words, abbreviations, capitalized words, capital letters, words with all letters capitalized (all caps).

4 Experimental Evaluation

This section describes the computational experiments conducted to evaluate the different categories of features and meta-features introduced in the previous section, as well as the results of the application of the feature selection strategies on the full set of features. We first present the datasets of tweets used and then we describe the settings adopted in the computational experiments. Finally, we present the results and the discussions.

4.1 Twitter Datasets

We used a set of sixteen datasets in the computational experiments reported in this section. These datasets have been extensively used in the literature of Twitter sentiment analysis. To the best of our knowledge, this is the first study using a significant number of Twitter datasets in the evaluation of different types of features and meta-features that have already been employed in the literature. These datasets are: Irony [18], Sarcasm [18], Aisopos⁴, SemEval-Fig⁵, Sentiment140 [17], Person [11], Movie [11], Sanders⁶, Narr [29], Obama-McCain Debate (OMD) [14], Health Care Reform (HCR) [38], STS-Gold [34], SentiStrength [39], Target-dependent [15], Vader [21], and SemEval13⁷. Some characteristics of these datasets are presented in Table 3, namely their total number of tweets, positive tweets and negative tweets.

4.2 Experimental Setting

In order to evaluate the different categories of features and meta-features described in Section 3, we applied the state-of-the-art machine learning algorithm Support Vector Machines (SVM), which

⁴ <http://grid.ece.ntua.gr>

⁵ <http://alt.qcri.org/semeval2015/task11>

⁶ <http://www.sananalytics.com/lab/twitter-sentiment>

⁷ <https://www.cs.york.ac.uk/semeval-2013/task2.html>

Table 3. Characteristics of the Twitter datasets.

Dataset	#tweets	#positive	#negative
Irony	65	22	43
Sarcasm	71	33	38
Aisopos	278	159	119
SemEval-Fig	321	47	274
Sentiment140	359	182	177
Person	439	312	127
Movie	561	460	101
Sanders	1,224	570	654
Narr	1,227	739	488
OMD	1,906	710	1,196
HCR	1,908	539	1,369
STS-Gold	2,034	632	1,402
SentiStrength	2,289	1,340	949
Target-dependent	3,467	1,734	1,733
Vader	4,196	2,897	1,299
SemEval13	4,378	3,183	1,195

has proven its robustness on large feature spaces [27]. In our experiments, we adopted the LIBSVM⁸ [10] implementation of SVM for Weka [20]. The regularization parameter of LIBSVM was set to its default value ($C = 1.0$) and we adopted the linear kernel.

As a preprocessing step, we used the same strategy as done in [27]. First, for each tweet in a given dataset, we replaced URLs by the token “http://someurl” and user mentions by the token “@someuser”. Then each tweet was tokenized and classified according to their part-of-speech tag, using the Twitter-specific part-of-speech tagset tool⁹ [16]. This tagset consists of twenty-five POS tags, specifically designed for tweets, that takes into account the different aspects that tweets have as compared to regular text, such as the lack of conventional orthography and the 140-character limit of each message [16]. Regarding stopwords removal, we discarded stopwords only as unigram features, since it has been acknowledged that stopwords can affect the polarity of some expressions in higher order n -grams [38].

The features used in the computational experiments are exactly those already proposed in the literature, as introduced in Section 3. In this context, for the category N-grams, we used as features: unigrams, bigrams, trigrams, 4-grams, and 5-grams. Considering that negation words (“shouldn’t”, for example) can affect the n -gram-based features, we handle negation by employing the same approach as used by Mohammad et al. [27], as described in Subsection 3.5. We used the SentiWordNet lexicon to extract the features of the Polarity category related to the number of positive and negative adjectives, nouns, adverbs, and verbs. Regarding the features related to internet slang and emoticons, we used the internet slang dictionary and the emoticon dictionary introduced and used in [2]. Similarly, we used the Internet Lingo Dictionary [41] for abbreviations, as done in [24].

In the experimental evaluation, the predictive performance of the sentiment classification is measured in terms of classification accuracy. For each evaluated dataset, the accuracy of the classification was computed as the ratio between the number of correctly classified tweets and the total number of tweets, after a 10-fold cross validation.

4.3 Results and Discussion

In this section, we present the computational results obtained in the set of experiments performed in this study. The conducted experiments aimed to answer two main questions:

1. How effective are the different categories of features and meta-features identified in the literature in the task of sentiment classification of tweets?

2. Can the sentiment classification of tweets benefit from the application of feature selection methods on the full set of features?

4.3.1 Analysis of the Categories of Features and Meta-features

In order to answer the first question, we evaluated the performance of each individual category by using its features and meta-features to train an SVM classifier for each dataset. Table 4 shows the results of this evaluation, as well as the number of features of each category (presented in the *Number of features* row, except for the category N-grams, which are presented in the *#features* column). The bold-faced values indicate the best accuracies. As can be observed, the best accuracies were achieved by the categories Polarity ($f5$ column) and N-grams ($f1$ column). The category Polarity achieved better results in ten out of the 16 datasets, while the category N-gram performed better in six out of the 16 datasets. None of the other categories, namely Twitter and Microblog ($f2$ column), POS ($f3$ column), and Punctuation ($f4$ column), achieved meaningful results.

We can also notice from Table 4 that the worst accuracies achieved with the features of the category N-grams are referred to the datasets Irony, Sarcasm, and Aisopos. For the datasets Irony and Sarcasm, it may be due to the few number of tweets they contain, that is, 65 and 71, respectively. It seems that the n -gram-based features are not representative enough when employed individually in the sentiment classification of the tweets from these datasets, since the classification is performed based on the vocabulary extracted from the training set, that is, the n -grams themselves. Regarding the dataset Aisopos, although the n -gram-based features achieved better results as compared to the categories Twitter and Microblog, POS, and Punctuation, there is a great difference between the performances of the categories N-gram and Polarity. The accuracy achieved with the polarity-based features surpassed in more than 20% the accuracy achieved with the n -grams features in this dataset. It may be due to the great number of emoticons that the tweets of this dataset contain. Since the polarities of emoticons are taken into account in the features of the category Polarity, this information may have improved the classification when using the features of this category.

Table 4. Accuracies (in %) achieved by evaluating each category of features and meta-features.

Dataset	$f1$		$f2$	$f3$	$f4$	$f5$
	Acc.	#features	Acc.	Acc.	Acc.	Acc.
Irony	66.2	3,457	66.2	66.2	64.6	84.6
Sarcasm	47.9	3,540	54.9	54.9	57.8	70.4
Aisopos	67.3	12,657	57.9	65.1	54.0	90.3
SemEval-Fig	88.2	17,592	86.9	84.7	85.4	84.1
Sentiment140	80.2	16,003	50.1	55.7	57.9	78.6
Person	78.1	22,624	71.1	70.8	70.8	79.3
Movie	83.1	24,952	81.8	82.0	82.0	83.8
Sanders	78.4	52,091	57.6	64.5	57.5	75.9
Narr	80.0	50,181	60.2	65.4	61.5	87.9
OMD	80.2	73,249	62.8	62.3	62.8	75.6
HCR	79.5	98,199	72.0	72.0	71.6	71.9
STS-Gold	82.5	83,882	68.7	68.8	69.3	89.4
SentiStrength	70.2	110,212	60.8	61.9	58.8	80.2
Target-dependent	81.5	163,101	50.3	57.9	54.6	79.9
Vader	81.6	155,429	69.0	69.0	69.3	87.9
SemEval13	78.4	248,807	72.7	72.7	72.7	84.5
Number of features	-		9	25	10	54

$f1$ N-grams $f2$ Twitter and Microblog $f3$ POS $f4$ Punctuation $f5$ Polarity

Although the n -gram-based features were not effective in the sentiment classification of the tweets in some datasets, other datasets are benefited from the use of n -grams features, such as SemEval-Fig, OMD, and HCR. The tweets of these three datasets are considered as

⁸ Available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

⁹ <http://www.ark.cs.cmu.edu/TweetNLP>

challenging tweets in sentiment classification because of their nature, that are: figurative language (irony, sarcasm, and metaphor), politics, and both politics and health, respectively. Since n -gram-based features are able to capture more context than the other features, such as expressions of irony and sarcasm, and specific vocabulary used in politics and health domains, these features seem to be more appropriated to be used in the sentiment classification of this type of tweets.

Another point we can observe is that the worst accuracy achieved for the dataset SemEval-Fig is the one obtained using the features of the category Polarity. Since most of the tweets of this dataset are related to irony and sarcasm, it is possible that polarity features were not helpful in the classification as in this type of tweets the polarity is usually reversed.

Besides the evaluation of the importance of each individual category, we also inverted this evaluation. More specifically, we investigate how each category contributes to the set of all features, by removing one of the categories at a time from the full set of features and meta-features. The results of this evaluation are presented in Table 5. The information in parentheses represents the lost or gain in accuracy when one category is removed, when compared to the accuracy achieved with the set of all features (*All* column). As we can see, in general, by removing the categories N-gram (*All-f1* column) and Polarity (*All-f5* column) the accuracies dropped considerably. In fact, these are the only categories in which their removal from the full set caused losses in accuracies for all datasets. It is in accordance with the previous results, in which the evaluation of these categories in isolation also performed better than the other categories.

Table 5. Accuracies achieved by evaluating different sets of features.

Dataset	All	All-f1	All-f2	All-f3	All-f4	All-f5
Irony	80.0	78.5 (-1.5)	80.0 (0.0)	81.5 (+1.5)	80.0 (0.0)	61.5 (-18.5)
Sarcasm	67.6	64.8 (-2.8)	71.8 (+4.2)	71.8 (+4.2)	69.0 (+1.4)	60.6 (-7.0)
Aisopos	92.8	87.4 (-5.4)	93.5 (+0.7)	92.4 (-0.4)	93.2 (+0.4)	71.9 (-20.9)
Sem-Fig	90.7	85.7 (-5.0)	90.3 (-0.4)	90.7 (0.0)	90.3 (-0.4)	88.5 (-2.2)
Sent140	81.9	79.1 (-2.8)	82.7 (+0.8)	81.3 (-0.6)	82.7 (+0.8)	73.5 (-8.4)
Person	84.1	78.8 (-5.3)	85.0 (+0.9)	82.7 (-1.4)	84.1 (0.0)	75.4 (-8.7)
Movie	85.6	83.2 (-2.4)	85.9 (+0.3)	85.0 (-0.6)	85.6 (0.0)	84.1 (-1.5)
Sanders	83.8	77.0 (-6.8)	84.2 (+0.4)	84.6 (+0.8)	83.5 (-0.3)	79.6 (-4.2)
Narr	88.0	86.8 (-1.2)	88.5 (+0.5)	87.8 (-0.2)	87.2 (-0.8)	77.6 (-10.4)
OMD	83.7	78.5 (-5.2)	83.9 (+0.2)	83.9 (+0.2)	83.4 (-0.3)	81.7 (-2.0)
HCR	80.5	74.2 (-6.3)	79.9 (-0.6)	80.1 (-0.4)	79.8 (-0.7)	79.6 (-0.9)
STS-Gold	90.7	88.8 (-1.9)	90.7 (0.0)	90.2 (-0.5)	90.2 (-0.5)	84.4 (-6.3)
SentiStr.	81.0	80.6 (-0.4)	81.1 (+0.1)	80.9 (-0.1)	80.6 (-0.4)	72.1 (-8.9)
Target-dep.	83.6	79.8 (-3.8)	83.7 (+0.1)	83.7 (+0.1)	83.7 (+0.1)	81.3 (-2.3)
Vader	88.6	87.6 (-1.0)	88.7 (+0.1)	89.0 (+0.4)	88.7 (+0.1)	82.3 (-6.3)
SemEval13	86.6	85.6 (-1.0)	86.7 (+0.1)	86.4 (-0.2)	85.8 (-0.8)	80.3 (-6.3)

f1 N-grams *f2* Twitter and Microblog *f3* POS *f4* Punctuation *f5* Polarity

The removal of the category POS (*All-f3* column) causes loss in accuracy in nine datasets, it was indifferent for only one, and it led to a better performance in six datasets. Similarly, by removing the category Punctuation (*All-f4* column), the accuracy dropped in eight datasets, it led to slightly higher accuracies in five datasets, and it was indifferent for three datasets.

The category Twitter and Microblog (*All-f2* column) seems to be the less important one. Removing its features and meta-features from the full set caused loss in accuracy only for the datasets SemEval-Fig and HCR. Considering that such datasets are more challenging than the others, as mentioned before, it is possible that the presence of the features of this category improves the overall classification accuracy. Differently, the absence of the category Twitter and Microblog in the classification was indifferent for two datasets (Irony and STS-Gold) and led to a better classification performance in twelve out of the 16 datasets. This is probably because the information in the features of this category is also captured by the features and meta-features

of other categories, that is, this category may add redundancy or inconsistency to the classification for the most of the datasets.

4.3.2 Application of the Feature Selection Methods

As mentioned before, it is possible that some redundant or inconsistent features are inserted into the classification when working with the full set of features. To further investigate this issue, and in order to answer the second research question, the next series of experiments aims at minimizing the redundancy and noise that distinct features may insert into the classification, by selecting the most relevant features for each dataset. To this purpose, we applied three commonly used feature selection measures, namely Information Gain (IG), Chi-Squared (CHI), and Relief-F. The adopted feature selection strategy ranks the features based on these measures and select the top n most relevant features for the classification, according to a predefined threshold (n). In this study, we have varied the threshold values from 75% of the full set of features to the top 5 features. More precisely, for each feature selection measure, we have used the following threshold values: 75%, 50%, 25%, 10%, 1000, 500, 100, 50, 25, 10, and 5. The accuracies achieved from using the measures Information Gain, Chi-Squared, and Relief-F are presented in Table 6, Table 7, and Table 8, respectively. For space reasons, we only reported the results for the threshold values from the top 5 to the top 1000 features. Moreover, the application of the omitted thresholds did not achieve meaningful results.

In some cases, when the results obtained by the application of two consecutive threshold values are very approximate and among the best, we varied the threshold between these two values. For example, regarding the dataset Vader, since the accuracies achieved by the IG measure using the threshold values 500 and 1000 were the best ones and very approximate (89.9% and 89.6%, respectively), we applied an extra variance in which the best accuracy was achieved using the top 700 features (90.0%). Similarly, regarding the Chi-Squared measure, the best results achieved for datasets Aisopos (93.9%) and Narr (88.6%) are from using the extra variance with the top 15 and 80 features, respectively. For space reasons, we only report the results of this extra variance when it led to a better result.

Table 6. Accuracies achieved by using Information Gain measure in the classification.

Dataset	#features							
	1000	700	500	100	50	25	10	5
Irony	64.6	-	76.9	80.0	78.5	83.1	72.3	73.9
Sarcasm	69.0	-	70.4	69.0	71.8	73.2	74.7	74.7
Aisopos	88.9	-	88.5	87.4	90.3	92.8	92.8	91.0
Sem-Fig	88.8	-	88.8	88.5	89.4	90.3	91.6	91.6
Sent140	84.4	85.0	84.7	84.7	82.5	80.2	81.1	79.1
Person	81.6	-	81.1	82.0	82.9	82.2	78.4	74.5
Movie	86.1	-	86.5	86.8	84.7	84.7	86.1	85.6
Sanders	83.1	-	82.9	80.0	79.9	79.2	75.3	73.9
Narr	87.3	-	88.0	88.4	87.9	86.9	86.6	84.9
OMD	83.1	-	84.3	82.8	81.4	80.9	78.8	69.4
HCR	77.6	-	78.3	77.7	75.4	74.0	71.8	71.8
STS-Gold	89.4	-	90.0	90.4	89.7	90.0	89.8	90.2
SentiStr.	79.1	-	79.5	81.4	81.0	80.2	77.6	76.8
Target-dep.	82.5	-	83.1	83.1	81.5	80.2	78.3	76.6
Vader	89.6	90.0	89.9	89.3	87.9	87.4	84.0	82.1
SemEval13	86.0	-	86.5	86.1	85.0	84.4	83.0	81.1

In general, regarding the three feature selection methods, the best results were achieved by using from the top 50 to the top 1000 features. However, we can observe that the datasets Irony, Sarcasm, Aisopos, and SemEval-Fig benefited from using a more compact set of features, ranging from 5 to 50 features, in general.

We can also note that the results achieved with feature selection for the dataset HCR did not surpass the accuracy from using the full set

of features (80.5%), shown in the previous experiment (Table 5). This may be an indication that using various and different kinds of features and meta-features is beneficial for this dataset, given its challenging and diverse domain (both politics and health).

Table 7. Accuracies achieved by using Chi-Squared measure in the classification.

Dataset	#features								
	1000	500	100	80	50	25	15	10	5
Irony	64.6	72.3	76.9	-	76.9	81.5	-	75.4	78.5
Sarcasm	67.6	71.8	73.2	-	71.8	74.7	-	73.2	74.7
Aisopos	89.2	88.5	88.5	-	89.6	93.2	93.9	92.8	91.0
Sem-Fig	88.8	88.8	88.2	-	91.0	91.9	-	91.6	91.6
Sent140	85.0	86.9	84.7	-	81.9	80.5	-	81.1	80.2
Person	80.6	81.1	83.6	-	83.6	82.7	-	77.9	75.2
Movie	85.4	84.9	85.0	-	85.0	84.7	-	85.7	85.6
Sanders	83.0	83.5	79.5	-	80.2	79.2	-	75.7	74.4
Narr	86.8	87.4	88.2	88.6	88.0	86.2	-	86.0	84.8
OMD	83.0	83.5	83.2	-	81.2	80.7	-	77.5	71.0
HCR	77.7	77.5	77.3	-	75.4	74.3	-	72.2	71.8
STS-Gold	88.7	90.1	90.4	-	89.8	90.0	-	89.7	90.2
SentiStr.	78.9	79.0	81.0	-	80.9	80.4	-	77.5	76.5
Target-dep.	82.5	83.0	82.8	-	81.5	80.2	-	78.2	76.8
Vader	89.8	89.7	89.4	-	88.3	87.2	-	83.8	82.1
SemEval13	85.7	86.8	86.1	-	84.9	84.5	-	82.3	81.1

Table 8. Accuracies achieved by using Relief-F measure in the classification.

Dataset	#features						
	1000	500	100	50	25	10	5
Irony	75.4	70.8	63.1	64.6	64.6	70.8	73.9
Sarcasm	71.8	67.6	67.6	71.8	70.4	67.6	71.8
Aisopos	87.4	88.5	92.1	92.8	90.3	91.0	91.4
SemEval-Fig	86.0	88.8	91.6	91.3	91.3	91.3	91.6
Sentiment140	79.4	78.6	84.7	82.2	81.6	79.1	79.4
Person	80.0	80.6	78.6	79.5	78.1	76.5	70.2
Movie	81.6	83.8	84.3	83.4	82.7	82.4	82.4
Sanders	78.8	78.9	80.7	78.4	75.2	71.2	61.0
Narr	87.1	87.8	87.0	87.8	88.2	80.0	75.4
OMD	81.0	81.4	80.3	77.3	74.7	69.8	68.5
HCR	76.9	77.2	74.7	74.2	74.1	72.9	71.8
STS-Gold	89.9	89.6	89.6	87.9	87.1	86.7	85.8
SentiStrength	80.0	81.4	79.4	76.5	74.6	74.6	65.9
Target-dep.	82.2	82.0	81.6	80.8	80.6	77.5	74.9
Vader	89.1	89.0	87.9	86.6	85.2	80.3	78.8
SemEval13	85.2	84.2	72.9	72.9	72.8	72.8	72.9

The best results achieved by each feature selection method are summarized in Table 9, as well as the results of the best category (Table 4) and the best set (Table 5), presented in the previous experiments. As we can see, the application of feature selection methods, in an attempting to minimize the redundancy and inconsistency that distinct features and meta-features may insert into the classification, led to better classification accuracies in ten of out the 16 datasets used in this evaluation. It means that while some datasets benefit with the presence of all features and meta-features of some categories, due to the nature of the tweets they contain, for other datasets this may cause the addition of noise and redundant features in the classification, which is not beneficial.

Besides improving the classification performance for the most of the datasets, it is important to highlight that the feature selection methods can also significantly reduce the feature space, that is, the number of features used in the classification. For example, for the dataset Narr, regarding the best result (88.6%), the application of the Chi-Squared measure have reduced the feature space of this dataset to only 80 features, as can be seen in Table 7, in contrast with more than 50,000 features from the full set of features.

Aiming at reporting the most relevant features selected for some datasets, Table 10 and Table 11 present the top features for datasets Aisopos and SemEval-Fig, respectively. For space reasons, we are

Table 9. Comparison among the best results achieved by each feature selection method, by the best category, and by the best set, respectively.

Dataset	IG	CHI	Relief-F	Best category	Best set
Irony	83.1	81.5	73.9	84.6	81.5
Sarcasm	74.7	74.7	71.8	70.4	71.8
Aisopos	92.8	93.9	92.8	90.3	93.5
SemEval-Fig	91.6	91.9	91.6	88.2	90.7
Sentiment140	85.0	86.9	84.7	80.2	82.7
Person	82.9	83.6	80.6	79.3	85.0
Movie	86.8	85.7	84.3	83.8	85.9
Sanders	83.1	83.5	80.7	78.4	84.6
Narr	88.4	88.6	88.2	87.9	88.5
OMD	84.3	83.5	81.4	80.2	83.9
HCR	78.3	77.7	77.2	79.5	80.7
STS-Gold	90.4	90.4	89.9	89.4	90.7
SentiStrength	81.4	81.0	81.4	80.2	81.1
Target-dep.	83.1	83.0	82.2	81.5	83.7
Vader	90.0	89.8	89.1	87.9	89.0
SemEval13	86.5	86.8	85.2	84.5	86.7

presenting the selected features for these two datasets only. For both datasets, we show the top features selected by the Chi-Squared measure, which achieved the best results for these datasets (Table 9). For presentation purposes, the features of each category are presented in the format $\langle category \rangle \cdot \langle featureName \rangle$, wherein *category* can be NGRAM, TWITTER, POS, PUNC, and POL, representing the categories N-grams, Twitter and Microblog, Part-of-Speech, Punctuation, and Polarity, respectively.

Analyzing the selected features for the dataset Aisopos (Table 10), we can see that the top 4 features are related to emoticons. As mentioned before, this dataset contains a great number of emoticons. Among the 119 negative tweets of this dataset, 97 tweets contain at least one negative emoticon. Differently, none of the 159 positive tweets contain any negative emoticons. Moreover, among those 97 negative tweets that contain negative emoticons, in 67 of them the emoticons appear as the last token. For this reason, the most discriminative negative features for this dataset are *POL_hasNegativeEmoticon*, *POL_numberOfNegativeEmoticons*, and *POL_isLastTokenNegativeEmoticon*.

Table 10. Top 15 features selected for dataset Aisopos.

Ranking	CHI	Feature
1	191.95	POL_hasNegativeEmoticon
2	191.95	POL_numberOfNegativeEmoticons
3	117.95	POL_isLastTokenNegativeEmoticon
4	115.58	POL_hasPositiveEmoticon
5	94.10	POL_totalScoreOfNegativeWordsInSent140
6	89.86	POL_maximalScoreOfNegativeWordsInSent140
7	84.70	POL_numberOfPositiveEmoticons
8	68.21	POL_totalScoreOfPositiveWordsInSent140
9	66.24	POL_isLastTokenPositiveEmoticon
10	62.01	POL_maximalScoreOfPositiveWordsInSent140
11	43.37	POL_sumOfScoresOfAdjAdvVerbNounFromSWN
12	40.97	POL_numberOfNegativeWordsInSent140
13	31.54	POL_totalScoreOfNegativeWordsInSWN
14	30.11	POL_numberOfPositiveWordsInSent140
15	29.40	POL_maximalScoreOfNegativeWordsInSWN

Regarding the dataset SemEval-Fig (Table 11), we can notice that 18 out of the 25 ranked features are from the N-grams category. It is consistent with the previous results reported for this dataset (from Table 4), in which the best accuracy was achieved when using the *n*-gram-based features in the classification. Among the *n*-grams, we can see the unigrams “#not” and “#sarcasm”, which are hashtags commonly used in tweets to express irony and sarcasm, respectively. We can also notice the trigram “pretty little liars”, which may be used as an expression of sarcasm. For this dataset, the most discriminative positive feature is the unigram “literally”, since this unigram has 21 occurrences among the 47 positive tweets of this dataset, and

occurs only eight times among the 274 negative tweets. The features *POS_numberOfHashtags* and *TWITTER_hasHashtag* are the most discriminative negative features for this dataset. This is probably because among the 274 tweets that contain hashtags, about 250 are related to negative tweets.

Table 11. Top 25 features selected for dataset SemEval-Fig.

Ranking	CHI	Feature
1	131.2	NGRAM_literally
2	72.9	POS_numberOfHashtags
3	72.9	TWITTER_hasHashtag
4	52.1	NGRAM_good
5	29.5	NGRAM_pretty
6	28.9	NGRAM_have a
7	28.8	POL_sumOfScoresOfAdjAdvVerbNounFromSWN
8	23.1	NGRAM_pretty little
9	22.9	POL_maximalScoreOfPositiveWordsInSWN
10	21.9	NGRAM_a good
11	21.4	NGRAM_#not
12	20.1	NGRAM_little
13	19.1	POL_numberOfPositiveWordsInSent140
14	19.0	NGRAM_one
15	18.0	NGRAM_#sarcasm
16	17.7	NGRAM_happy birthday
17	17.7	NGRAM_literally just
18	17.7	NGRAM_such a good
19	17.4	NGRAM_liars
20	17.4	NGRAM_pretty little liars
21	17.4	NGRAM_hilarious
22	17.4	NGRAM_little liars
23	15.9	POL_numberOfPositiveWordsInOpinionFinder
24	15.9	POL_scoreOfLastTokenInNRCHashtag
25	13.2	NGRAM_so good

4.3.3 Comparison with Results Reported in the Literature

In order to investigate the competitiveness of the computational results achieved in this work, we compared them with the results achieved in recent works in the literature. Specifically, we aim at comparing the classification performance using the most relevant features and meta-features we have identified using feature selection with the best results we found in recent works in the literature for this task. For space reasons, although we have found many works to compare with, we report only their best results. Moreover, we could not find any work to compare for some datasets because they have been used in the three-class classification problem, that is, the sentiment classification problem regarding the positive, negative, and neutral classes. Since we focus on the polarity classification (positive and negative classes, only), the results reported in such works are not comparable. The comparison is presented in Table 12.

Table 12. Comparison between the best results achieved in this work and the best results reported in the literature, in terms of accuracy and microF₁.

Dataset	Accuracy		MicroF ₁	
	Our results	Best in literature	Our results	Canuto et al. [9]
Aisopos	–	–	94.8	89.2
Sentiment140	86.9	86.3 [35]	87.1	86.9
Sanders	84.6	98.1 [8]	83.4	86.5
Narr	88.6	81.3 [29]	90.5	88.8
OMD	84.3	82.9 [36]	76.8	80.0
HCR	80.5	78.7 [34]	–	–
STS-Gold	90.7	85.7 [34]	–	–
SentiStrength	81.4	73.4 [34]	84.3	82.6
Vader	–	–	92.9	97.2
SemEval13	–	–	91.1	85.8
Win count	6	1	5	3

The results obtained in this work and the best results found in the literature are presented in terms of classification accuracy, except for the results achieved by Canuto et al. [9], which are presented in terms

of microF₁. For this reason, we split the table in two parts. The first part shows the comparison among the results in terms of accuracy. In the second part, we present the comparison between the results in terms of microF₁. We also show, in the last row (*Win count* row), the number of times that each compared work achieved the best results.

As we can see, the computational results achieved in this work, using different kinds of features and meta-features, are the best in six out of the seven datasets, regarding the results reported in terms of classification accuracy. However, for dataset Sanders, the best result was achieved by Bravo-Marquez et al. [8]. Regarding the comparison with the results reported in terms of microF₁, the results achieved in this work were among the best in five out of the eight compared datasets. Similarly to the results presented in terms of accuracy, we did not achieve the best result for dataset Sanders. These results confirm the importance of selecting the appropriate set of features in the context of Twitter sentiment analysis.

5 Conclusions and Future Work

In this work, we presented a literature review of the most common feature representation in the sentiment classification of tweets, including meta-features. We proposed to group these features and meta-features in specific categories, in order to evaluate the importance of each category in the polarity classification of tweets from distinct domains. These categories include N-grams, Twitter and Microblog, Part-of-Speech, Punctuation, and Polarity. We used sixteen datasets of tweets in the series of experiments reported in this study. To the best of our knowledge, this is the first work that evaluates distinct categories of features for a significant number of Twitter datasets.

Our experiments showed that the categories Polarity and N-grams are the most important ones, achieving the best results. Indeed, when considering the full set of features, the removal of the features from these two categories made the performance drop considerably for all datasets. We could also notice that some datasets, such as HCR, benefited from the presence of the full set of features in the classification. This may be an indication that tweets from challenging domains need to be represented by all types of features.

We also applied feature selection strategies in order to select the most relevant features for the classification. This set of experiments showed that tweets from distinct domains can benefit from using different subsets of features in the classification. Finally, we compared the results achieved in this work with the best results previously reported in the literature for some datasets. This comparison confirmed we have achieved meaningful results by evaluating different categories of features and also using feature selection strategies.

For future work, we intend to examine and incorporate new features from more recent studies, such as the meta-features recently proposed by Canuto et al. [9]. Another idea of future work is the application of lazy feature selection methods, based on the hypothesis that knowing the values of the features of a particular tweet at classification time may contribute to identify the best features for the correct classification of that specific tweet [32].

ACKNOWLEDGEMENTS

This work was supported by CAPES and CNPq research grants.

REFERENCES

- [1] A. Abbasi, H. Chen, and A. Salem, ‘Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums’, *ACM Transactions on Information Systems*, **26**(3), 12:1–12:34, (2008).

- [2] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau, 'Sentiment analysis of Twitter data', in *Proceedings of the Workshop on Languages in Social Media*, pp. 30–38. ACL, (2011).
- [3] B. Agarwal and N. Mittal, 'Optimal feature selection for sentiment analysis', in *Proceedings of the 14th International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 13–24. Springer, (2013).
- [4] S. Baccianella, A. Esuli, and F. Sebastiani, 'SentiWordNet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining', in *Proceedings of the 7th International Conference on Language Resources and Evaluation*, pp. 2200–2204, (2010).
- [5] L. Barbosa and J. Feng, 'Robust sentiment detection on Twitter from biased and noisy data', in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pp. 36–44. ACL, (2010).
- [6] A. Birmingham and A.F. Smeaton, 'Classifying sentiment in microblogs: is brevity an advantage?', in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pp. 1833–1836. ACM, (2010).
- [7] A. Bifet and E. Frank, 'Sentiment knowledge discovery in Twitter streaming data', in *Proceedings of the 13th International Conference on Discovery Science*, pp. 1–15. Springer-Verlag, (2010).
- [8] F. Bravo-Marquez, M. Mendoza, and B. Poblete, 'Meta-level sentiment models for big social data analysis', *Knowledge-Based Systems*, **69**, 86–99, (2014).
- [9] S. Canuto, M.A. Gonçalves, and F. Benevenuto, 'Exploiting new sentiment-based meta-level features for effective sentiment analysis', in *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*, pp. 53–62. ACM, (2016).
- [10] C. Chang and C. Lin, 'LIBSVM: A library for support vector machines', *ACM Transactions on Intelligent Systems and Technology*, **2**, 1–27, (2011).
- [11] L. Chen, W. Wang, M. Nagarajan, S. Wang, and A.P. Sheth, 'Extracting diverse sentiment expressions with target-dependent polarity from Twitter', in *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media*, pp. 50–57, (2012).
- [12] N.F.F. da Silva, E.R. Hruschka, and E.R. Hruschka Jr., 'Tweet sentiment analysis with classifier ensembles', *Decision Support Systems*, **66**, 170–179, (2014).
- [13] D. Davidov, O. Tsur, and A. Rappoport, 'Enhanced sentiment learning using Twitter hashtags and smileys', in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pp. 241–249. ACL, (2010).
- [14] N.A. Diakopoulos and D.A. Shamma, 'Characterizing debate performance via aggregated Twitter sentiment', in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1195–1198. ACM, (2010).
- [15] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, and K. Xu, 'Adaptive recursive neural network for target-dependent Twitter sentiment classification', in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: short papers*, pp. 49–54. ACL, (2014).
- [16] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N.A. Smith, 'Part-of-speech tagging for Twitter: annotation, features, and experiments', in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers*, pp. 42–47. ACL, (2011).
- [17] A. Go, R. Bhayani, and L. Huang, 'Twitter sentiment classification using distant supervision', Technical Report CS224N, Stanford, (2009).
- [18] P. Gonçalves, D. Dalip, J. Reis, J. Messias, F. Ribeiro, P. Melo, M. Gonçalves, and F. Benevenuto, 'Caracterizando e detectando sarcasmo e ironia no Twitter', in *Proceedings of the Brazilian Workshop on Social Network Analysis and Mining*, (2015).
- [19] M. Hagen, M. Potthast, M. Büchner, and B. Stein, 'Twitter sentiment detection via ensemble classification using averaged confidence scores', in *Proceedings of the 37th European Conference on IR Research*, pp. 741–754. Springer, (2015).
- [20] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten, 'The weka data mining software: an update', *SIGKDD Explorations Newsletter*, **11**(1), 10–18, (2009).
- [21] C.J. Hutto and E. Gilbert, 'Vader: A parsimonious rule-based model for sentiment analysis of social media text', in *Proceedings of the 8th International AAAI Conference on Weblogs and Social Media*, (2014).
- [22] L. Jiang, M. Yu, M. Zhou, X. Liu, and T. Zhao, 'Target-dependent Twitter sentiment classification', in *Proceedings of the 49th Annual Meeting of the ACL: Human Language Technologies*, pp. 151–160. ACL, (2011).
- [23] V.N. Khuc, C. Shivade, R. Ramnath, and J. Ramanathan, 'Towards building large-scale distributed systems for Twitter sentiment analysis', in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pp. 459–464. ACM, (2012).
- [24] E. Kouloumpis, T. Wilson, and J. Moore, 'Twitter sentiment analysis: The good the bad and the omg!', in *Proceedings of the 5th International AAAI Conference on Web and Social Media*, pp. 538–541, (2011).
- [25] J. Lin and A. Kolcz, 'Large-scale machine learning at Twitter', in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pp. 793–804. ACM, (2012).
- [26] B. Liu, 'Sentiment analysis and opinion mining', *Synthesis Lectures on Human Language Technologies*, **5**(1), 1–167, (2012).
- [27] S. Mohammad, S. Kiritchenko, and X. Zhu, 'Nrc-canada: building the state-of-the-art in sentiment analysis of tweets', in *Proceedings of the 7th International Workshop on Semantic Evaluation Exercises*, Atlanta, Georgia, USA, (2013).
- [28] S.M. Mohammad and P.D. Turney, 'Crowdsourcing a word-emotion association lexicon', *Computational Intelligence*, **29**(3), 436–465, (2013).
- [29] S. Narr, M. Hulpenhaus, and S. Albayrak, 'Language-independent Twitter sentiment analysis', in *Proceedings of the Workshop on Knowledge Discovery, Data Mining and Machine Learning*, (2012).
- [30] A. Pak and P. Paroubek, 'Twitter as a corpus for sentiment analysis and opinion mining', in *Proceedings of the 7th International Conference on Language Resources and Evaluation*, pp. 1320–1326, (2010).
- [31] B. Pang, L. Lee, and S. Vaithyanathan, 'Thumbs up? sentiment classification using machine learning techniques', in *Proceedings of the ACL-02 conference on Empirical Methods in Natural Language Processing*, pp. 79–86. ACL, (2002).
- [32] R.B. Pereira, A. Plastino, B. Zadrozny, L.H.C. Merschmann, and A.A. Freitas, 'Lazy attribute selection: Choosing attributes at classification time', *Intelligent Data Analysis*, **15**(5), 715–732, (2011).
- [33] J.D. Prusa, T.M. Khoshgoftaar, and D.J. Dittman, 'Impact of feature selection techniques for tweet sentiment classification', in *Proceedings of the 28th International FLAIRS Conference*, (2015).
- [34] H. Saif, M. Fernandez, Y. He, and H. Alani, 'Evaluation datasets for Twitter sentiment analysis: a survey and a new dataset, the STS-Gold', in *Proceedings of the 1st Workshop on Emotion and Sentiment in Social and Expressive Media*, (2013).
- [35] H. Saif, Y. He, and H. Alani, 'Alleviating data sparsity for Twitter sentiment analysis', in *Proceedings of the 2nd Workshop on Making Sense of Microposts*, pp. 2–9. EUR-WS, (2012).
- [36] H. Saif, Y. He, M. Fernandez, and H. Alani, 'Semantic patterns for sentiment analysis of Twitter', in *Proceedings of the 13th International Semantic Web Conference*, pp. 324–340. Springer, (2014).
- [37] A. Sharma and S. Dey, 'A comparative study of feature selection and machine learning techniques for sentiment analysis', in *Proceedings of the 2012 ACM Research in Applied Computation Symposium*, pp. 1–7. ACM, (2012).
- [38] M. Speriosu, N. Sudan, S. Upadhyay, and J. Baldrige, 'Twitter polarity classification with label propagation over lexical links and the follower graph', in *Proceedings of the 1st Workshop on Unsupervised Learning in NLP*, pp. 53–63. ACL, (2011).
- [39] M. Thelwall, K. Buckley, and G. Paltoglou, 'Sentiment strength detection for the social web', *Journal of the American Society for Information Science and Technology*, **63**(1), 163–173, (2012).
- [40] H. Wang, D. Can, A. Kazemzadeh, F. Bar, and S. Narayanan, 'A system for real-time Twitter sentiment analysis of 2012 US presidential election cycle', in *Proceedings of the ACL 2012 System Demonstrations*, pp. 115–120. ACL, (2012).
- [41] L. Wasden, 'Internet lingo dictionary: A parents' guide to codes used in chat rooms, instant messaging, text messaging, and blogs', Technical report, Idaho Office of the Attorney General, (2010).
- [42] M. Wiegand, A. Balahur, B. Roth, D. Klakow, and A. Montoyo, 'A survey on the role of negation in sentiment analysis', in *Proceedings of the Workshop on Negation and Speculation in Natural Language Processing*, pp. 60–68. ACL, (2010).
- [43] T. Wilson, J. Wiebe, and P. Hoffmann, 'Recognizing contextual polarity in phrase-level sentiment analysis', in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 347–354. ACL, (2005).
- [44] L. Zhang, R. Ghosh, M. Dekhil, M. Hsu, and B. Liu, 'Combining lexicon-based and learning-based methods for Twitter sentiment analysis', Technical Report HPL-2011-89, HP Laboratories, (2011).

Crowdfunding Public Projects with Provision Point: A Prediction Market Approach

Praphul Chandra¹ and Sujit Gujar² and Y. Narahari³

Abstract. Crowdfunding is emerging as a popular means to generate funding from citizens for public projects. This is popularly known as civic crowdfunding. In this paper, we focus on crowdfunding public projects with provision point: these are projects in which contributions must reach a predetermined threshold in order for the project to be provisioned. On web based civic crowdfunding platforms, the success of crowdfunding public projects has been somewhat mixed. In this paper, our objective is to design a mechanism that improves the success of crowdfunding public projects. In particular, we propose a class of mechanisms for crowdfunding platforms with sequentially arriving agents. This class of mechanisms induces an extensive form game for agents arriving on the platform and we show that the game has a non-empty set of sub-game perfect equilibria at which the project is fully funded. We call this new class of mechanisms *Provision Point Mechanism with Securities* (PPS). The novelty of PPS lies in the use of a prediction market to incentivize agents to contribute in proportion to their true value for the project and to contribute as soon as they arrive at the crowdfunding platform. Different variations of PPS are possible depending on the underlying prediction market. In this paper, we use a cost function (or equivalently, scoring rule) based prediction market; in fact, we specify the requirements that a cost function should satisfy to be used in PPS. We study and compare two specific instances of PPS: (1) Logarithmic Market Scoring Rule based and (2) Quadratic Scoring Rule based. We also discuss the considerations that should guide the choice of the cost function when deploying our mechanism on crowdfunding platforms.

1 INTRODUCTION

Civic crowdfunding platforms like Spacehive [1], Citizinvestor [11], Neighbourly [20] etc., aim to generate funding for public and community projects from citizens. The success of these platforms has been mixed. For example, in the United Kingdom, Spacehive has generated £4.4 million for public projects from citizen contributions across 68 cities with a 44% success rate (the fraction of posted projects that are fully funded)[1]. Thus, less than half the number of projects posted meet their funding targets. In this paper, our objective is to design a mechanism that can markedly improve the success rate of crowdfunding public projects. A typical process that is followed in crowdfunding of public projects is as follows:

1. *Requester posts public project:* A requester, seeking crowdfunding for a public project, posts a proposal. The proposal specifies

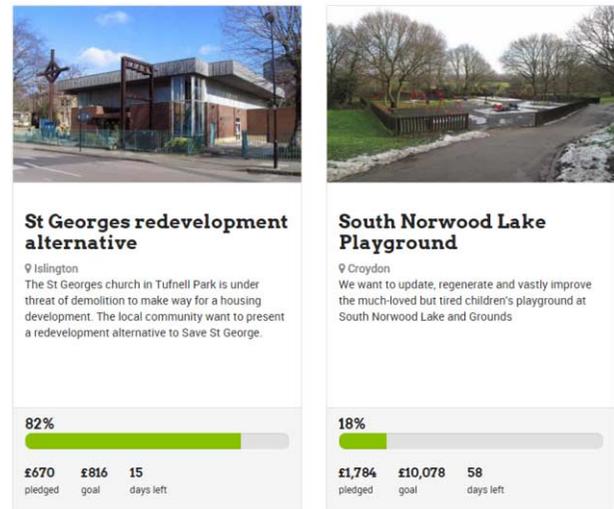


Figure 1. Public projects listed on a crowdfunding platform

a target amount of funds to be raised for the project to be provisioned: the target amount is thus also known as the provision point. The requester also specifies a deadline by which the funds need to be raised.

2. *Agents arrive:* Agents arrive over time to view the project and observe (i) the target amount, (ii) the amount pending to be funded, and (iii) the deadline. Figure 1 shows two different ongoing public projects from a crowdfunding platform [1] as they appear to an agent arriving at the platform.
3. *Agents contribute:* Each arriving agent may contribute a certain amount towards funding the project.
4. *Requester provisions or refunds:* If the funding target is achieved by the deadline, the requester provisions the project; otherwise, the contributions of all agents are refunded.

Two features in this process are notable: (i) Crowdfunding relies on voluntary contributions and hence neither coercion nor punishment is an option. (ii) Since contributions arrive over a period of time, an agent is able to observe the contributions of the agents who have contributed so far.

Relying on private contributions to fund a public project is not a new phenomenon and has been studied in the literature extensively [4, 5, 25, 26, 12, 6, 23, 27, 16]. The key challenge in relying on private contributions to fund a public project is the free-riding problem: since public projects are non-excludable and non-rival, agents have

¹ Indian Institute of Science, Bangalore, India, email: praphulcs@gmail.com

² International Institute of Information Technology, Hyderabad, India, email:sujit.gujar@iiit.ac.in (Work done while at IISc.)

³ Indian Institute of Science, Bangalore, India, email: hari@csa.iisc.ernet.in

an incentive to free-ride on the contributions of others. One approach to solve the free riding problem is using *assurance contracts* which allow agents to commit contributions conditional on sufficient contributions from others [5, 25]. The term *dominant assurance contracts* is used to refer to contracts which can ensure that the project gets funded at equilibrium [26]. Several mechanisms have been proposed which implement assurance contracts.

Provision Point Mechanism (PPM) [4] invites contributions for a target amount. If the target is met, the project is provisioned; otherwise, agents' contributions are refunded. Provision Point mechanism with Refund bonus (PPR), proposed by Zubrickas *et. al.* [27], invites contributions for a target amount. If the target is met, the project is provisioned; otherwise, agents' contributions are refunded and agents who volunteered to contribute are paid an additional *refund bonus* which depends on the quantum of the agent's contribution.

Neither of the above two mechanisms takes into account the sequential nature of agent contributions in crowdfunding platforms. The mechanisms also do not handle the fact that contributions, once made, are common knowledge. Applying these mechanisms in a sequential setting could hamper their success in crowdfunding of public projects (Section 3.2). This motivates the need for our proposed mechanism, in which, we retain the idea of a refund bonus⁴ from [27] but the way we compute the refund bonus makes the mechanism more attractive. Moreover our mechanism explicitly takes into account the sequential nature of arrivals of agent contributions. The novelty in our mechanism is in using a prediction market based approach for computing the refund bonuses. We award contingent securities to the agents who contribute to the public project: if the project is not funded, an agent is paid a unit amount for each unit of security held by the agent. The number of securities awarded to an agent is based on the quantum of the agent's contribution and the time at which the agent makes the contribution. As the securities are allotted only for the outcome that the project is not funded and not for the outcome that the project is funded, the prediction market under consideration is classified as *complex prediction market* [2].

1.1 Contributions and Outline

The following are the main contributions of this paper.

- We propose a class of mechanisms, named *Provision Point Mechanism with Securities* (PPS) for crowdfunding public projects. PPS induces an extensive form game and we show that the game has a non-empty set of sub-game perfect equilibria at which the project is fully funded (Theorem 3).
- PPS solves the free-riding problem when public projects are provisioned using private contributions since agents are incentivized to contribute *in proportion* to their true value for the project and to contribute *as soon as they arrive* at the crowdfunding platform.
- PPS uses a complex prediction market [2]. Different versions of PPS are possible depending on the underlying prediction market and the cost function. We study and compare two specific instances of PPS: (1) Logarithmic Market Scoring Rule (LMSR) [14, 15] based and (2) Quadratic Scoring Rule (QSR) based.

The rest of the paper is organized as follows. In Section 2, we summarize the notation we use and present some preliminaries. In Section 3, we position our work in relation to the existing literature. In Section 4, we review complex prediction markets and propose a

⁴ Even though PPS relies on a *sponsor* to offer a refund bonus while the funds are being collected, the bonus is not paid out at equilibrium.

new class of mechanisms, PPS, for crowdfunding a public project and show the existence of a non-empty set of sub-game perfect equilibria where the project gets fully funded. In Section 5, we compare the performance of PPS mechanism with two popular cost functions and discuss the impact of the cost function in PPS. We conclude in Section 6 with a summary.

2 PRELIMINARIES

We focus on crowdfunding projects which involve private provisioning of a public project without coercion and with agents arriving over time. Table 1 lists the key notation used in this work. Similar

Symbol	Definition
T	Time at which fund collection concludes
t	Epoch of time in the interval $[0, T]$
h^t	Amount that remains to be funded at t ;
h^0	Target amount (provision point)
$i \in \{0, 1, \dots, n\}$	Agent id; $i = 0$ refers to the requester
$\theta_i \in \mathbb{R}_+$	Agent i 's value for the project
$x_i \in \mathbb{R}_+$	Agent i 's contribution to the project
$a_i \in [0, T]$	Time at which agent i arrives at the platform
$t_i \in [a_i, T]$	Time at which agent i contributes to the project
$\psi_i = (x_i, t_i)$	Strategy of agent i
$\vartheta \in \mathbb{R}_+$	Net value for the project
$\chi \in \mathbb{R}_+$	Net contribution for the project

Table 1. Key notation

to previous work [4, 27], we assume that agents have quasi-linear utility (ASSUMPTION-1) and apart from knowing the history of contributions, agents do not have any information regarding whether the project will get funded or not (ASSUMPTION-2). We model the following sequence of events. At $t = 0$, the requester posts a proposal for funding a public project. This includes the target amount of funds h^0 (the provision point) and a deadline T till which agents may contribute to the project. h^t refers to the target amount that remains to be collected at time t : h^0 , T , and h^t are common knowledge. Agent $i \in \{1, 2, \dots, n\}$ arrives at time $a_i \in [0, T]$ and observes the funds that have been collected so far ($h^0 - h^{a_i}$). The value that an agent derives from the public project getting provisioned (θ_i) is his private information. Agent i may decide to contribute funds $x_i \in [0, h^{a_i}]$ to the project at any time $t_i \in [a_i, T]$. We assume that agents contribute only once to the project (ASSUMPTION-3). This assumption is reasonable in civic crowdfunding scenarios where agents typically visit the project website once and contribute if the project has value to them. From an analysis view point, the mechanism we design ensures that agents have no advantage in delaying or splitting up their contributions. We leave it for future work to study effect of spiteful contributions [8].

The strategy of agent i is $\psi_i = (x_i, t_i)$. $\mathbf{x} = (x_1, \dots, x_n)$ refers to the vector of agent contributions and $\psi = (\psi_1, \dots, \psi_n)$ denotes the strategy profile of agents. We use the subscript $-i$ to represent vectors without agent i ; so, for example, x_{-i} refers to the vector of contributions of all agents except i . The net value for the project among the agents is $\vartheta = \sum_{i=1}^n \theta_i$ and the net contribution is $\chi = \sum_{i=1}^n x_i$. The utility derived by agent i with value θ_i for the project, when agents use strategy profile ψ is $u_i(\psi; \theta_i)$.

2.1 Important Definitions

We seek to design mechanisms in a sequential setting such that a public project gets funded at equilibrium. Such mechanisms induce a game among the agents $\{1, 2, \dots, n\}$. With ψ_i s being agents' strategies and u_i s as their utilities, we define Pure Strategy Nash Equilibrium (PSNE) and Sub-Game Perfect Equilibrium (SGPE).

Definition: (Pure Strategy Nash Equilibrium) A strategy profile $\psi^* = (\psi_1^*, \dots, \psi_n^*)$ is said to be a Pure Strategy Nash Equilibrium (PSNE) if $\forall i, \forall \theta_i$

$$u_i(\psi_i^*, \psi_{-i}^*; \theta_i) \geq u_i(\tilde{\psi}_i, \psi_{-i}^*; \theta_i) \quad \forall \tilde{\psi}_i.$$

Let H^t be the history of the game till time t , that contains the agents' arrivals and their contributions, then we define:

Definition: (Sub-game Perfect Equilibrium) A strategy profile $\psi^* = (\psi_1^*, \dots, \psi_n^*)$ is said to be a sub-game perfect equilibrium if $\forall i, \forall \theta_i$

$$u_i(\psi_i^*, \psi_{-i|H^{a_i}}^*; \theta_i) \geq u_i(\tilde{\psi}_i, \psi_{-i|H^{a_i}}^*; \theta_i) \quad \forall \tilde{\psi}_i, \forall H^t$$

Here $\psi_{-i|H^{a_i}}^*$ indicates that the agents who arrive after a_i follow the strategy specified in ψ_{-i}^* .

3 RELATED WORK

Our work is related to the literature on provisioning of *public goods*. In this paper, we use the term *public project* instead, since it is more suitable for crowdfunding platforms. The literature deals with two kinds of public projects. For *discrete* public projects, a predetermined target amount must be collected for the project to be provisioned. For *continuous* public projects, the *extent* of project provisioned increases monotonically with net contributions, up to a threshold.

For *continuous*, public projects, one of the simplest mechanisms is the Voluntary Contribution Mechanism (VCM): agents voluntarily contribute and the extent of the public project provisioned corresponds to the aggregate funds collected. VCM induces a simultaneous move game which has multiple equilibria. Many of these equilibria lead to an under-provisioning of the public project, a result which has been verified empirically [17]. Morgan [19] studies the use of state lotteries for funding continuous public projects. Voluntary contributions are incentivized by offering an opportunity to win a fixed prize and an agent's contribution towards public project also determines the likelihood of his winning the prize. This game has a unique equilibrium which provisions a higher level of the public project than VCM.

In this paper, our focus is on *discrete* public projects (projects with a provision point) which are predominant on crowdfunding platforms. As discussed in Section 1, our work is motivated by the need to non-trivially extend the work of [4] and [27] to the realistic setting where agent contributions arrive *sequentially*. Marx and Matthews[18] consider a sequential setting where agents make repeated contributions to a project, taking turns in a round-robin fashion. They prove the existence of a Nash equilibrium where each agent contributes if and only if all the past agents have contributed their equilibrium contributions. Thus, it is not a sub-game perfect equilibrium. Our work differs from this in that, there is neither a pre-fixed order of contributions nor do agents contribute repeatedly and we look for sub-game perfect equilibria.

3.1 Provision Point Mechanism (PPM)

PPM [4] for discrete public projects collects voluntary contributions. The project is provisioned if the funding target is achieved. If the funding target is not achieved, the contributions are refunded. Let \mathcal{I}_X be an indicator random variable which takes the value 1 if X is true and 0 otherwise. Thus, for PPM, the project gets funded only if $\chi \geq h^0$, and agent i 's ($i > 0$) utility⁵ in PPM is:

$$u_i(\mathbf{x}; \theta_i) = \mathcal{I}_{\chi \geq h^0} \times (\theta_i - x_i) + \mathcal{I}_{\chi < h^0} \times 0 \quad (1)$$

In PPM, an agent's utility consists of a *funded utility* ($\theta_i - x_i$), which is the agent's utility if the project is provisioned and an *unfunded utility* (zero) which is the agent's utility if the project is not provisioned. PPM has been shown to have multiple equilibria, many of which are inefficient [4]: a result which has been verified empirically too [17].

3.2 Provision Point Mechanism with Refund bonus (PPR)

PPR [27] for discrete public projects collects voluntary contributions. The project is provisioned if the funding target is achieved. If the funding target is not achieved, the contributions are refunded and an additional refund bonus is paid to agents who volunteered to contribute, in proportion to their contribution. The refund bonus is $\frac{x_i}{\chi} B \forall i$ where $B > 0$ is the refund budget set aside by the requester at the beginning and is common knowledge among all agents. Thus, for PPR too, the project is provisioned only if $\chi \geq h^0$ and agent i 's utility in PPR is:

$$u_i(\mathbf{x}; \theta_i) = \mathcal{I}_{\chi \geq h^0} \times (\theta_i - x_i) + \mathcal{I}_{\chi < h^0} \times \left(\frac{x_i}{\chi} B \right) \quad (2)$$

In PPR, an agent's utility consists of a *funded utility* ($\theta_i - x_i$), which is the agent's utility if the project is provisioned and a strictly positive *unfunded utility* ($\frac{x_i}{\chi} B > 0$), which is the agent's utility if the project is not provisioned. The set of Pure Strategy Nash equilibria with PPR are characterized as follows:

Theorem 1 [27] Let $\vartheta > h^0$ and $B > 0$. In PPR, the set of PSNE are $\{(x_i^*) : x_i^* \leq \frac{h^0}{B+h^0} \theta_i \forall i; \chi = h^0\}$ if $B \leq \vartheta - h^0$. Otherwise the set of PSNE is empty.

Limitations of PPR

PPR considers a setup where agents decide their contributions *simultaneously* without knowledge of contributions made by the other agents. The game induced is thus a simultaneous move game. When applied in a sequential (discrete time) setting where agents can contribute over time and can observe previous contributions, agent strategies consist of the contribution amount and the interval in which they contribute.

Proposition: Let $\vartheta > h^0$ and $B > 0$. If $B \leq \vartheta - h^0$, the set $\{(x_i^*, T) : x_i^* \leq \frac{h^0}{B+h^0} \theta_i \forall i; \chi = h^0\}$ constitutes Pure Strategy Nash equilibria of PPR in the sequential setting.

⁵ As the strategy space in PPM consists only of contribution to be made, we drop ψ here.

Proof: In PPR, since the refund bonus does not depend on the time when the contribution is made, no agent has an incentive to invest earlier than the deadline, if all other agents do the same. Effectively, PPR in sequential setting collapses to a one shot simultaneous move game at $t = T$ where x_i^* s are given by Theorem (1). Thus, (x_i^*, T) constitute a set of PSNE of PPR in sequential setting. \square

The implication of the above proposition is that all the agents may delay their contribution as close to the deadline as possible in sequential settings and wait to free-ride till the end. This is undesirable since such temporal strategies can lead to the equilibrium not being achieved in practice. This shortcoming of PPR in sequential settings is because early contributions do not receive any advantage.

We seek to design crowdfunding mechanisms by explicitly capturing and taking advantage of the fact that on web based crowdfunding platforms, contributions are sequential rather than simultaneous. The key intuition in our approach is that by giving participants a pay-off structure which refunds them more generously if they contribute earlier (in the event the project is not funded), participants have an incentive to contribute early. This overcomes the serious limitation of PPR. We achieve our objective using a novel prediction market approach.

4 OUR APPROACH: PREDICTION MARKET FOR CROWDFUNDING MECHANISMS

We incorporate ideas from the literature on prediction markets with the key idea being that contributors actually buy contingent securities which each pay a unit amount if the project is not funded. As these securities are purchased, the price increases, thereby incentivizing the participants to contribute earlier rather than later. Our mechanism achieves an equilibrium at which the project is funded and thus the refund bonus is not paid out at equilibrium. Since our approach leverages prediction markets, we briefly explain important concepts from prediction market literature that are relevant to our crowdfunding mechanism design approach.

4.1 Cost function based Prediction Markets

A Prediction Market seeks to predict the outcome of an event in future. Let Ω be the set of mutually exclusive and exhaustive outcomes of the event. For example, in a prediction market designed to predict the outcome of a political election among two candidates, we would be interested in an outcome set $\Omega = \{\omega_A, \omega_B\}$ where ω_A is the outcome that candidate A wins the election and ω_B is the outcome that candidate B wins the election. Since $|\Omega| = 2$, we refer to this as a *binary* outcome event. A prediction market incentivizes agents to ex-

Symbol	Definition
$\Omega = \{\omega_j\}_{j \in \{1, \dots, \Omega \}}$	Set of possible outcomes of the event
π_{ω_j}	Payoff vector if outcome ω_j is realized
p_{ω_j}	Price of an infinitesimally small amount of security associated with outcome ω_j
$\mathbf{q} = \{q_{\omega_j}\}_{j \in \{1, \dots, \Omega \}}$	Vector of securities issued by the market maker
$C : \mathbb{R}^{ \Omega } \rightarrow \mathbb{R}$	Cost function used in the prediction market
\mathbf{r}	Bundle securities purchased by an agent
$\text{Cost}(\mathbf{r} \mathbf{q})$	Cost of purchasing a bundle of \mathbf{r} securities when \mathbf{q} securities are outstanding

Table 2. Important terms for prediction market

press their belief about the outcome of an event. One approach to realize a prediction market is by associating securities (Arrow-Debreu contracts [3]) with the outcomes of the event. A security associated with outcome ω_j pays a unit amount if ω_j is realized and zero otherwise. An agent with a belief different from the market belief can buy (or sell) securities to modify the market belief. An automated market maker is a software agent which automates pricing and order execution of such securities. An automated market maker can be realized using a cost function $C : \mathbb{R}^{|\Omega|} \rightarrow \mathbb{R}$ which is a potential function specifying the amount of money wagered in the market as a function of the number of securities that have been issued by the market for each outcome. In a market with a binary outcome event, $C : \mathbb{R}^2 \rightarrow \mathbb{R}$ is a function of the vector of outstanding securities, $\mathbf{q} \in \mathbb{R}^2$. Several authors [22, 21, 2, 10, 6] have studied conditions that a cost function must satisfy to be used in prediction markets.

4.1.1 Conditions on Cost Function

- **CONDITION-1 (PATH INDEPENDENCE)** This condition requires that the cost of acquiring a bundle of \mathbf{r} securities must be the same regardless of how an agent splits up the purchase. This condition implies that in a prediction market, prices can be represented by a cost function such that the cost of purchasing a bundle of \mathbf{r} securities is $\text{Cost}(\mathbf{r}|\mathbf{q}) = C(\mathbf{q} + \mathbf{r}) - C(\mathbf{q})$.
- **CONDITION-2 (CONTINUOUS AND DIFFERENTIABLE)** This condition requires that the gradient of the cost function ($\nabla C(\mathbf{q})$) is well defined everywhere so that it can be treated as a vector of instantaneous prices for securities associated with each outcome. Further $p_{\omega_j} = \partial C(\mathbf{q})/\partial(q_{\omega_j}) \geq 0 \quad \forall \omega_j \in \Omega$ represents the price per security of an infinitesimally small amount of security associated with outcome ω_j .
- **CONDITION-3 (INFORMATION INCORPORATION)** This condition requires that a purchase of a bundle of \mathbf{r} securities should never lower the price of \mathbf{r} , that is, for any \mathbf{q} and $\mathbf{r} \in \mathbb{R}^{|\Omega|}$ $C(\mathbf{q} + 2\mathbf{r}) - C(\mathbf{q} + \mathbf{r}) \geq C(\mathbf{q} + \mathbf{r}) - C(\mathbf{q})$. This condition is required to ensure that participating in the market is incentive compatible for a myopic agent. Further, this condition implies that the cost function used in prediction markets must be convex.
- **CONDITION-4 (NO ARBITRAGE)** This condition requires that it is never possible for an agent to purchase a bundle of securities \mathbf{r} and receive a positive payoff *regardless* of the outcome. For all \mathbf{q} and $\mathbf{r} \in \mathbb{R}^{|\Omega|}$, $\exists \omega_j \in \Omega$ such that $C(\mathbf{q} + \mathbf{r}) - C(\mathbf{q}) > \mathbf{r} \cdot \pi_{\omega_j}$.
- **CONDITION-5 (EXPRESSIVENESS)** This condition requires that any agent can set the market belief to reflect his belief about the expected outcome. Let Δ_n be the n dimensional probability simplex, then $\forall \mathbf{p} \in \Delta_{|\Omega|}$, $\exists \mathbf{q} \in \mathbb{R}^{|\Omega|}$ s.t. $\nabla C(\mathbf{q}) = \mathbb{E}_{\omega \sim \mathbf{p}}[\pi(\omega)]$.
- **CONDITION-6 (BOUNDED LOSS)** This condition requires that an automated market maker using a cost function can only lose a finite amount regardless of the transactions undertaken by the agents, that is, $\sup_{\mathbf{q}} [\max_{\omega_j} (q_{\omega_j}) - C(\mathbf{q})] < \infty$. If the market maker initializes the market with $\mathbf{q} = (0, 0)$, then the worst case loss for the market maker is $\sup_{\mathbf{q}} [\max_{\omega_j} (q_{\omega_j}) - (C(\mathbf{q}) - C(\mathbf{0}))]$.

With a binary outcome event, two popular cost functions that satisfy these conditions are [7]:

$$C_{LMSR}(\mathbf{q}) = b \ln(\exp(q_{\omega_0}/b) + \exp(q_{\omega_1}/b)) \quad (3)$$

$$C_{QSR}(\mathbf{q}) = \frac{q_{\omega_0} + q_{\omega_1}}{2} + \frac{q_{\omega_0}^2 + q_{\omega_1}^2}{4b} - \frac{(q_{\omega_0} + q_{\omega_1})^2}{8b} - \frac{b}{2} \quad (4)$$

where b is a parameter that controls how fast prices change.

4.2 Proposed Mechanism: Provision Point Mechanism with Securities (PPS)

We now introduce a new class of mechanisms which explicitly takes into account the sequential nature of contributions. Similar to PPM and PPR, in this new class of mechanism too, the project gets provisioned only if the net contributions reach the provision point ($\chi \geq h^0$). However, the refund bonus of a contributor is determined using securities from the complex cost based prediction market defined in Section 4.2.1. We refer to our mechanism as Provision Point Mechanism with Securities (PPS).

In PPS, we create a prediction market by associating securities with the binary outcome of the public project getting funded or not. We consider a binary outcome ($\Omega = \{\omega_0, \omega_1\}$) event where ω_0 refers to the (negative) outcome that the project is not funded by the deadline and ω_1 refers to the (positive) outcome that the project is funded by the deadline. The key intuition in PPS is to incentivize agents to contribute to public projects by treating every contribution towards the public project as simultaneously an investment in purchasing securities associated with the negative outcome (project not getting funded). We treat every contribution $x_i > 0$ at t_i towards the public project as simultaneously an investment in purchasing $r_i^{t_i} > x_i > 0$ securities associated with the negative outcome ω_0 : each of these securities pays out a unit amount if the project is not fully funded and zero otherwise. Thus, agent i 's utility who contributes x_i at t_i is:

$$u_i(\psi; \theta_i) = \mathcal{I}_{\chi \geq h^0} \times (\theta_i - x_i) + \mathcal{I}_{\chi < h^0} \times (r_i^{t_i} - x_i) \quad (5)$$

Equation (5) consists of two terms: the *funded utility* is $(\theta_i - x_i)$ and the *unfunded utility* is $(r_i^{t_i} - x_i)$. The funded utility is a monotonically decreasing function of x_i and is independent of the time the contribution is made (t_i) and the history of the game till time t_i (H^{t_i}).

The unfunded utility $(r_i^{t_i} - x_i)$ depends on $r_i^{t_i}$ which in turn depends on (i) the quantum of the contribution (x_i)⁶, (ii) the timing of the contribution (t_i) and (iii) the history of past contributions H^{t_i} via the total number of outstanding securities (q^{t_i}). CONDITION-3 ensures that $r_i^{t_i}$ is a monotonically decreasing function of q^{t_i} . This means that for a given contribution, the number of securities awarded to an agent cannot increase with time: intuitively, this is the reason why agents are incentivized to contribute early. It turns out that the cost function needed to determine $r_i^{t_i}$ in our crowdfunding mechanism needs to be a *complex* cost function [2].

4.2.1 Complex Prediction Markets

Abernethy et. al. [2] introduce the distinction between *complete* and *complex* cost function based prediction markets. In a complete cost function based market, the market maker offers a security corresponding to *each* potential outcome and *each* of these securities pays a unit amount if the associated outcome is realized. In a complex cost function based market, the market maker may offer $K < |\Omega|$ securities and/or a security may not necessarily pay a unit amount when an outcome $\omega_j \in \Omega$ is realized.

A key result from [2] imposes additional constraints on cost function to be used in complex prediction markets. Let $\pi_\Omega = \{\pi_{\omega_j} | \omega_j \in \Omega\}$ and let $\mathcal{H}(\pi_\Omega)$ be the convex hull of π_Ω . Then,

Theorem 2 [2] *If $\mathcal{H}(\pi_\Omega)$ is closed, then under CONDITIONS2-5, C must be convex with $\{\nabla C(\mathbf{q}) : \mathbf{q} \in \mathbb{R}^K\} = \mathcal{H}(\pi_\Omega)$.*

⁶ Even though $r_i^{t_i}$ depends on x_i , we use the notation $r_i^{t_i}$ for simplicity.

The following corollary, which can be derived from Theorem 2 and its proof in [2], will be useful in determining the price of securities awarded to agents who contribute to the public project.

Corollary: If $\mathcal{H}(\pi_\Omega)$ is closed, then under CONDITIONS2-4, C must be convex with $\{\nabla C(\mathbf{q}) : \mathbf{q} \in \mathbb{R}^K\} \subseteq \mathcal{H}(\pi_\Omega)$.

4.2.2 A Complex Prediction Market for Crowdfunding

Previous work has used complex cost function based markets in scenarios where the outcome space ($|\Omega|$) is very large [2, 9, 13]. Our use of complex prediction markets in the context of crowdfunding (binary outcome event) is motivated not by a large outcome space but with the explicit objective of limiting the expressiveness of agents.

For a binary outcome event, a *complete* cost function based prediction market offers $K = |\Omega| = 2$ securities where the securities associated with the negative outcome (q_{ω_0}) pay a unit amount of the project is not funded ($\pi_{\omega_0} = (1, 0)$) and securities associated with the positive outcome (q_{ω_1}) pay a unit amount of the project is funded ($\pi_{\omega_1} = (0, 1)$). Thus, $\pi_\Omega = \{(1, 0), (0, 1)\}$.

For a binary outcome event, we propose a *complex* cost function based prediction market that offers $K = |\Omega| = 2$ securities where the securities associated with the negative outcome (q_{ω_0}) pay a unit amount if the project is not funded ($\pi_{\omega_0} = (1, 0)$) but securities associated with the positive outcome (q_{ω_1}) never payout ($\pi_{\omega_1} = (0, 0)$). Thus, $\pi_\Omega = \{(1, 0), (0, 0)\}$ and $\mathcal{H}(\pi_\Omega) = [0, 1]$. Furthermore, agents are not allowed to sell securities. Thus, the design of our complex prediction market for crowdfunding has the following implications.

1. **FIXED POSITIVE OUTCOME SECURITIES** Since the payoff associated with the positive outcome are zero ($\pi_{\omega_1} = (0, 0)$) and since CONDITION-2 requires that the price of the positive security is non-negative ($\partial C(\mathbf{q})/\partial q_{\omega_1} \geq 0$), no agent will purchase securities associated with the positive outcome. For the rest of the paper we will assume that the market is initialized with $\mathbf{q} = (0, 0)$ ⁷. Thus, for the duration of the market $q_{\omega_1} = 0$ and $\mathbf{q} = (q_{\omega_0}, 0)$.
2. **LIMITED EXPRESSIVENESS** Since agents are not allowed to sell securities and since they can purchase securities associated with the negative outcome only, it follows that agents may not be able to express their true beliefs about the event and thus our complex market violates CONDITION-5.
3. **LOW RANK PRICE SPACE** Using the previous implication and Corollary from Section 4.2.1, we have that $\partial C(\mathbf{q})/\partial q_{\omega_0} \in [0, 1]$ and if the market is initialized with $\mathbf{q} = (0, 0)$, then $\partial C(\mathbf{q})/\partial q_{\omega_0} \in [0.5, 1]$.

To emphasize that securities related with the positive outcome are fixed at initialization and not traded for the duration of the market ($[0, T]$), we will refer to the cost function used in this proposed prediction market as C_0 . Such a cost function can be obtained by taking any cost function which satisfies CONDITIONS 1-4,6 and setting $q_{\omega_1}^t = 0 \quad \forall t \in [0, T]$. Thus, $C_0 : \mathbb{R} \rightarrow \mathbb{R}$. For the rest of the paper, since we will be using prediction market involving *only* negative outcome securities, we will use the following simplified notation:

$$\begin{aligned} q &\equiv q_{\omega_0} \\ \pi &\equiv \pi_{\omega_0} \\ p &\equiv p_{\omega_0} = \frac{\partial C_0(q)}{\partial q_{\omega_0}} \end{aligned}$$

⁷ If the market is initialized with a different number of positive securities, z , then $\mathbf{q} = (q_{\omega_0}, z)$. Section 5.4 discusses this scenario.

Proposition: Let C be a cost function which satisfies CONDITION-2 and C_0 be the corresponding cost function obtained by setting $q_{\omega_1}^t = 0 \quad \forall t \in [0, T]$, then C_0 is invertible.

Proof: Since $q_{\omega_1}^t = 0 \quad \forall t \in [0, T]$, $C_0 : \mathbb{R} \rightarrow \mathbb{R}$ is a one-to-one function of a single variable. By CONDITION-2, C_0 is continuous and differentiable. Thus, by the inverse function theorem [24], the inverse of C_0 exists. \square

We use the notation C_0^{-1} to refer to the inverse of the function C_0 . In PPS, the cost of purchasing $r_i^{t_i}$ securities at t_i when q^{t_i} securities are outstanding is:

$$\text{Cost}(r_i^{t_i} | q^{t_i}) = C_0(q^{t_i} + r_i^{t_i}) - C_0(q^{t_i})$$

Thus, an agent who contributes x_i at t_i receives $r_i^{t_i}$ securities where:

$$\begin{aligned} x_i &= C_0(q^{t_i} + r_i^{t_i}) - C_0(q^{t_i}) \\ r_i^{t_i} &= C_0^{-1}(x_i + C_0(q^{t_i})) - q^{t_i} \end{aligned} \quad (6)$$

Since the cost function must be path independent (CONDITION-1), the number of securities issued by a single contribution of h^0 is the same as the number of securities allocated if the contribution is split into any number of smaller contributions ($\sum_i x_i = h^0$). Hence, in PPS the total number of securities that will be issued for crowdfunding a project with a target amount h^0 is:

$$\sum_i r_i^{t_i} = C_0^{-1}(h^0 + C_0(0)) \quad (7)$$

4.2.3 Equilibrium Analysis of PPS

We now specify an additional condition that a cost function needs to satisfy to be used in PPS.

- **CONDITION-7 (SUFFICIENT LIQUIDITY)** This condition requires that a cost function should ensure that $\forall \theta_i < h^0$, an agent's unfunded utility ($r_i^{t_i} - x_i$) is monotonically increasing in x_i , that is, $\forall q^{t_i}, \forall x_i < h^0, \quad \frac{\partial}{\partial x_i}(r_i^{t_i} - x_i) > 0 \Rightarrow \frac{\partial r_i^{t_i}}{\partial x_i} > 1$.

Theorem 3 Let $C : \mathbb{R}^2 \rightarrow \mathbb{R}$ be a cost function that satisfies CONDITIONS 1-4,6 and $C_0 : \mathbb{R} \rightarrow \mathbb{R}$ be the cost function obtained from C by fixing the number of positive outcome securities. If C_0 satisfies CONDITION-7 and is used in PPS for crowdfunding a project with provision point h^0 when $\vartheta > C_0^{-1}(h^0 + C_0(0))$, the strategies in the set $\left\{ (\psi_i^* = \{x_i^*, a_i\}) : x_i^* \leq (C_0(\theta_i + q^{a_i}) - C_0(q^{a_i})) \text{ if } h^{a_i} > 0, \text{ otherwise } x_i^* = 0; \chi = h^0 \right\}$ are sub-game perfect equilibria.

Proof: First we claim in Step 1 that, at equilibrium, $\chi = h^0$. In Step 2, we characterize the equilibria strategy of agent i (ψ_i^*). Step 3 proves the upper bound on b . We show that these equilibria strategies are sub-game perfect in Step 4.

Step 1: In equilibrium, $\chi > h^0$ cannot hold since the requester stops collecting the funds at $\chi = h^0$ if this happens before the deadline T . In equilibrium, $\chi < h^0$ cannot hold since an agent can increase his utility by contributing more and receiving a higher r_i due to CONDITION-7. Thus, in equilibrium $\chi = h^0$.

Step 2: Due to ASSUMPTION-2, agents do not have any bias in believing whether the project will be funded, other than the contributions. From Step 1, the contributions would be such that the project

is funded in equilibrium. Thus, at equilibrium, an agent will contribute such that his funded utility is no less than the highest possible unfunded utility. That is, if (x_i^*, t_i^*) is agent's equilibrium strategy, $r_i^* - x_i^* \leq \theta_i - x_i^* \Rightarrow r_i^* \leq \theta_i$. Expressing r_i^* in terms of x_i^* and t_i^* using Equation (6), we get the condition:

$$\begin{aligned} C_0^{-1}(x_i^* + C_0(q^{t_i^*})) - q^{t_i^*} &\leq \theta_i \\ \text{or equivalently, } x_i^* &\leq C_0(\theta_i + q^{t_i^*}) - C_0(q^{t_i^*}) \end{aligned}$$

Note that (i) the RHS of Equation (8) is a monotonically decreasing function of $q^{t_i^*}$ and (ii) q^t , the number of securities allotted by the market at time t , is a monotonically non-decreasing function of t . Thus, an agent with value θ_i minimizes the RHS at $t_i^* = a_i$, that is, he contributes as soon as he arrives⁸. Thus, $\psi_i^* = (x_i^*, a_i)$ and at equilibrium:

$$x_i^* \leq C_0(\theta_i + q^{a_i}) - C_0(q^{a_i}) \quad (8)$$

Step 3: Summing up $r_i^* - x_i^* \leq \theta_i - x_i^*$ for all agents leads to the condition $\sum_{i=1}^n r_i^* \leq \vartheta$. Since securities are allocated using a path independent cost function (CONDITION-1), using Equation (7) the condition for Nash Equilibrium becomes:

$$C_0^{-1}(h^0 + C_0(0)) < \vartheta \quad (9)$$

Step 4: These equilibria, specified as a function of the aggregate history (\bar{h}^{a_i}), are also sub-game perfect. Consider agent j who arrives last at a_j . If $h^{a_j} = 0$, then his best strategy is $x_j^* = 0$. If $h^{a_j} > 0$, irrespective of H^{a_j} and h^{a_j} , his funded and unfunded utility are the same at x_j^* , defined in the theorem and still it is best response for j to follow the equilibrium strategy. With backward induction, by similar reasoning, it is best response for every agent to follow the equilibrium strategy irrespective of history. Thus, these equilibria are also sub-game perfect equilibria. \square

The above theorem characterizes a set of sub-game perfect equilibria at which crowdfunding projects using PPS gets fully funded. Since no agent, without any additional information regarding the project getting funded or not (ASSUMPTION-2), should invest no more than the bound of Equation (8), we believe that this is the only set of sub-game perfect equilibria of induced game at which the project gets fully funded. We are yet to identify any other Nash equilibria for the game induced by PPS.

5 PPS WITH DIFFERENT COST FUNCTIONS

In this section, we undertake a comparison of PPS instantiated using two popular cost functions: logarithmic scoring rule based and quadratic scoring rule based. In both cases, the cost functions satisfy CONDITIONS 1-4,6. A well known criterion for choosing the cost function in prediction markets is the trade off between the worst case loss and market liquidity [7]. In PPS, CONDITION-7 explicitly lower bounds the liquidity in the market. Interestingly, this upper bounds the refund budget and thus the worst case loss. Thus for PPS, the key consideration in choosing the cost function comes from the trade off between satisfying CONDITION-7 and the maximum bonus that can be offered to incentivize agents to contribute to the public project earlier.

⁸ For an intuitive explanation, See Section 5.3

5.1 LMSR-PPS

In our complex cost function based prediction market, LMSR-PPS, which uses the cost function specified in Equation (3) is specified as:

$$\begin{aligned} C_0(q^t) &= b \ln(1 + \exp(q^t/b)) \\ p^t &= \frac{\exp(q^t/b)}{1 + \exp(q^t/b)} \\ \text{Cost}(r^t|q^t) &= C_0(q^t + r^t) - C_0(q^t) \\ &= b \ln \left(\frac{1 + \exp(\frac{q^t+r^t}{b})}{1 + \exp(\frac{q^t}{b})} \right) \end{aligned}$$

An agent who contributes x_i at t_i receives $r_i^{t_i}$ securities where:

$$\begin{aligned} x_i &= b \ln \left(\frac{1 + \exp(\frac{q^{t_i} + r_i^{t_i}}{b})}{1 + \exp(\frac{q^{t_i}}{b})} \right) \text{ and} \\ r_i^{t_i} &= b \ln \left(\exp \left(\frac{x_i}{b} + \ln(1 + \exp(\frac{q^{t_i}}{b})) \right) - 1 \right) - q^{t_i} \end{aligned}$$

Proposition: LMSR-PPS satisfies CONDITION-7.

Proof: CONDITION-7 requires that $\forall q^{t_i}, \forall x_i < h^0$, $\frac{\partial r_i^{t_i}}{\partial x_i} > 1$. With LMSR-PPS,

$$\frac{\partial r_i^{t_i}}{\partial x_i} = \frac{\exp \left(\frac{x_i}{b} + \ln(1 + \exp(\frac{q^{t_i}}{b})) \right)}{\exp \left(\frac{x_i}{b} + \ln(1 + \exp(\frac{q^{t_i}}{b})) \right) - 1}$$

Since the RHS is always greater than 1, CONDITION-7 is always satisfied for LMSR-PPS. We note that this is an immediate implication of the infinite liquidity of LMSR based prediction markets [14]. \square

Corollary: If $\vartheta > h^0$ and $b > 0$, in LMSR-PPS, the strategies in the set $\left\{ (\psi_i^* = \{x_i^*, a_i\}) : x_i^* \leq b \ln \left(\frac{1 + \exp(\frac{\theta_i + q^{a_i}}{b})}{1 + \exp(\frac{q^{a_i}}{b})} \right) \text{ if } h^{a_i} > 0, \text{ otherwise } x_i^* = 0; \chi = h^0 \right\}$ are sub-game perfect equilibria if $b < \frac{\vartheta - h^0}{\ln 2}$.

Proof: Since LMSR-PPS satisfy CONDITIONS 1-4,6-7, Theorem 3 is applicable and Equation (8), the equilibrium contribution of agent i with value θ_i who arrives at a_i is:

$$x_i^* \leq C_0(\theta_i + q^{a_i}) - C_0(q^{a_i}) = b \ln \left(\frac{1 + \exp(\frac{\theta_i + q^{a_i}}{b})}{1 + \exp(\frac{q^{a_i}}{b})} \right)$$

The condition for attaining this equilibrium corresponding to Equation (9) is:

$$\begin{aligned} C_0^{-1}(h^0 + C_0(0)) &= b \ln \left(\exp \left(\frac{h^0}{b} + \ln(2) - 1 \right) \right) \\ &< b \ln \left(\exp \left(\frac{h^0}{b} + \ln(2) \right) \right) \\ &< \vartheta \\ \Rightarrow b &< \frac{\vartheta - h^0}{\ln 2} \end{aligned}$$

\square

5.2 QSR-PPS

In our complex cost function based prediction market, the QSR-PPS, which uses the cost function specified in Equation (4) is specified as:

$$\begin{aligned} C_0(q^t) &= \frac{q^t}{2} + \frac{(q^t)^2}{8b} - \frac{b}{2} \\ p^t &= \frac{1}{2} + \frac{q^t}{4b} \\ \text{Cost}(r^t|q^t) &= C_0(q^t + r^t) - C_0(q^t) \\ &= r^t \left(\frac{1}{2} + \frac{q^t}{4b} + \frac{r^t}{8b} \right) \end{aligned}$$

An agent who contributes x_i at t_i receives $r_i^{t_i}$ securities where:

$$\begin{aligned} x_i &= r_i^{t_i} \left(\frac{1}{2} + \frac{q^{t_i}}{4b} + \frac{r_i^{t_i}}{8b} \right) \text{ and} \\ r_i^{t_i} &= \sqrt{(q^{t_i} + 2b)^2 + 8bx_i} - (q^{t_i} + 2b) \end{aligned}$$

Proposition: QSR-PPS satisfies CONDITION-7 if $b > \frac{2}{3}h^0$

Proof: CONDITION-7 requires that $\forall q^{t_i}, \forall x_i \leq h^0$, $\frac{\partial r_i^{t_i}}{\partial x_i} > 1$. With QSR-PPS,

$$\frac{\partial r_i^{t_i}}{\partial x_i} = \frac{4b}{\sqrt{(q^{t_i} + 2b)^2 + 8bx_i}}$$

The RHS obtains its minimum value with $q^{t_i} = 0$ and $x_i = h^0$: this corresponds to the condition when the first agent who arrives contributes the whole amount for the public project. Ensuring that this minimum is greater than 1 leads to the condition $b > \frac{2}{3}h^0$. \square

Corollary: If $\vartheta > h^0$ and $b > 0$, in QSR-PPS, the strategies in the set $\left\{ (\psi_i^* = \{x_i^*, a_i\}) : x_i^* \leq \theta_i \left(\frac{1}{2} + \frac{q^{a_i}}{4b} + \frac{\theta_i}{8b} \right) \text{ if } h^{a_i} > 0, \text{ otherwise } x_i^* = 0; \chi = h^0 \right\}$ are sub-game perfect equilibria if $b < \frac{\vartheta^2}{8h^0}$.

Proof: With $b > \frac{2}{3}h^0$, since QSR-PPS satisfies CONDITIONS 1-4,6-7, Theorem 3 is applicable and Equation (8), the equilibrium contribution of agent i with value θ_i who arrives at a_i is:

$$x_i^* \leq C_0(\theta_i + q^{a_i}) - C_0(q^{a_i}) = \theta_i \left(\frac{1}{2} + \frac{q^{a_i}}{4b} + \frac{\theta_i}{8b} \right)$$

The condition for attaining this equilibrium corresponding to Equation (9) is:

$$\begin{aligned} C_0^{-1}(h^0 + C_0(0)) &= \sqrt{(2b)^2 + 8bh^0} - 2b \\ &< \sqrt{8bh^0} \\ &< \vartheta \\ \Rightarrow b &< \frac{\vartheta^2}{8h^0} \end{aligned}$$

\square

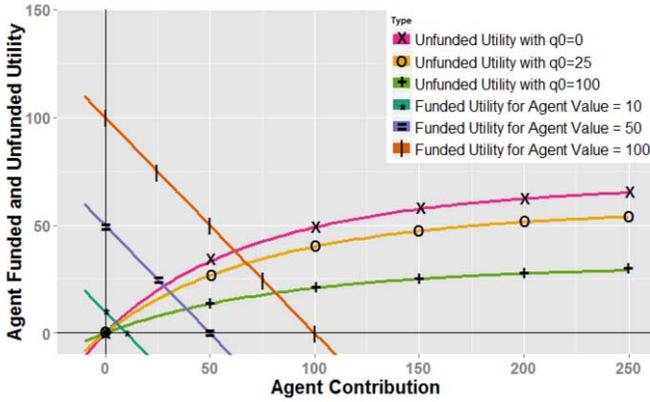


Figure 2. LMSR based PPS: utility vs. contribution

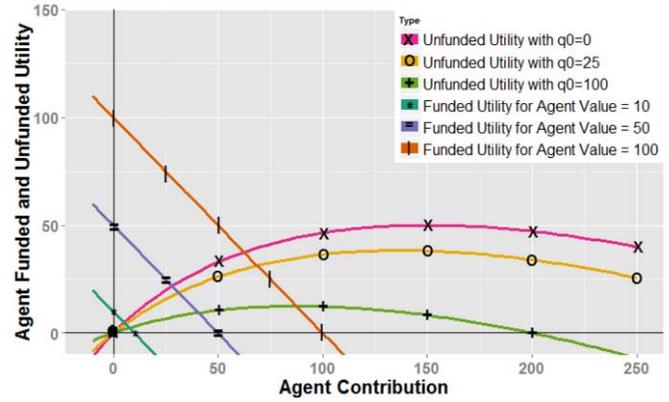


Figure 3. QSR based PPS: utility vs. contribution

5.3 Choosing a Cost Function for PPS

Sections 5.1 and 5.2 show that both LMSR and QSR based PPS can achieve successful crowdfunding of a public project under the right conditions. In both cases, the condition that $\vartheta > C_0^{-1}(h^0 + C_0(0))$ (Theorem 3) leads to an upper bound on b . For LMSR-PPS, $b < \frac{\vartheta - h^0}{\ln 2}$ and for QSR-PPS, $b < \frac{\vartheta^2}{8h^0}$. This upper bound ensures that the the refund bonus is not so high that it exceeds the net value of the project.

A lower bound on b (and hence the refund bonus) comes from CONDITION-7. In LMSR-PPS, the unfunded utility is always a monotonically increasing function of the contribution and CONDITION-7 is always satisfied. Thus, the lower bound on b is trivial ($b > 0$) and all agents have an incentive (no matter how small) to contribute. In practice, if the refund bonus in LMSR is too small to incentivize agents, then CONDITION-7 can be modified to require that $\forall q^{t_i}, \forall x_i < h^0, \frac{\partial r_{i}^{t_i}}{\partial x_i} > (1 + \epsilon)$ and we can show that $b > h^0 / \ln(\frac{1}{2\epsilon})$.

In QSR-PPS, the unfunded utility first increases then decreases and thus satisfying CONDITION-7 creates a lower bound on the refund budget ($b > \frac{2}{3}h^0$). Intuitively, if b is too low than agents who arrive after the contributions have crossed a certain threshold (but not yet reached the provision point) will not have an incentive to contribute.

Figure 2 and 3 compare the performance of LMSR-PPS and QSR-PPS with $b = 100$. In both the figures, the three straight lines correspond to funded utility of agents with different types. Since the unfunded utility depends both on the contribution and the timing of the contribution (via the number of outstanding securities at the time of the contribution), the unfunded utility as a function of the contribution for three different histories (different number of outstanding negative outcome securities) are shown. Note that the unfunded utility is independent of agent type.

If more securities have been sold at the time agent i contributes, he must contribute more to obtain the same unfunded utility. For a given history (number of outstanding securities), the point where the curve (unfunded utility) intersects the line (funded utility) is the contribution amount where the agent derives the same utility independent of whether or not get the project is funded. The set of equilibria x_i^* lie to the left of this intersection.

5.4 Price and Probability in PPS

In any cost function based prediction market, the gradient of the cost function is interpreted as both the instantaneous price of a security and the market probability of the associated event. In PPS however, this needs to be reinterpreted. For every contribution x_i towards the public project, PPS allocates securities associated with the project *not* getting funded. As the contributions near the target ($\chi \rightarrow h^0$), the price of the project *not* getting funded nears one ($p \rightarrow 1$): this is desirable since PPS is designed to incentivize early contributions. However, interpreting p as the probability of the project *not* getting funded is counter intuitive. Instead we propose to interpret $1 - p$ as the market probability that the project will not be funded. The range of instantaneous price (p) and probability ($1 - p$) also depend on the number of securities with which the market is initialized. If $\mathbf{q} = (0, 0)$, then $p \in [0.5, 1]$. If the market is initialized with $\mathbf{q} = (q_{\omega_1}, 0)$ then the price space expands or shrinks depending on the value of q_{ω_1} . Thus, initialization of \mathbf{q} can be used to control the price space in PPS.

6 Conclusion

In this work, we have proposed a class of provision point mechanisms, PPS for civic crowdfunding. PPS induces an extensive form game among the agents who arrive on the crowdfunding platform and achieves equilibria at which the project is funded. These equilibria have the desirable property that agents do not free ride but instead contribute in proportion to their true value for the project and do so as soon as they arrive. PPS achieves this by incentivizing agents with a refund greater than their contribution if the project is not funded. In PPS, securities issued in a cost function based prediction market determine the refund bonus. Even though PPS relies on a *sponsor* to offer a refund bonus while the funds are being collected, the bonus is not paid out at equilibrium. As these securities are purchased, the price increases, thereby incentivizing participants to contribute earlier. We specified the conditions that a cost function must satisfy to be used in PPS and compared PPS under two popular cost functions. Using these as the benchmark, we provided considerations to choose an optimal cost function. We believe that our work can significantly improve the success rate of provisioning public projects using private funds in scenarios like civic crowdfunding.

REFERENCES

- [1] Spacehive, 2015. Available online at <http://www.spacehive.com>.
- [2] Jacob Abernethy, Yiling Chen, and Jennifer Wortman Vaughan, 'Efficient market making via convex optimization, and a connection to online learning', *ACM Transactions on Economics and Computation*, **1**(2), 12, (2013).
- [3] Kenneth J Arrow, 'The role of securities in the optimal allocation of risk-bearing', *The Review of Economic Studies*, **31**(2), 91–96, (1964).
- [4] Mark Bagnoli and Barton L Lipman, 'Provision of public goods: Fully implementing the core through private contributions', *The Review of Economic Studies*, **56**(4), 583–601, (1989).
- [5] Earl R Brubaker, 'Free ride, free revelation, or golden rule?', *The Journal of Law & Economics*, **18**(1), 147–161, (1975).
- [6] Yan Chen, 'Incentive-compatible mechanisms for pure public goods: a survey of experimental research', *The Handbook of Experimental Economics Results*, Amsterdam: Elsevier Press, forthcoming, (1999).
- [7] Yiling Chen, 'A utility framework for bounded-loss market makers', in *In Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*, (2007).
- [8] Yiling Chen, Stanko Dimitrov, Rahul Sami, Daniel M Reeves, David M Pennock, Robin D Hanson, Lance Fortnow, and Rica Gonen, 'Gaming prediction markets: Equilibrium strategies with a market maker', *Algorithmica*, **58**(4), 930–969, (2010).
- [9] Yiling Chen, Lance Fortnow, Nicolas Lambert, David M Pennock, and Jennifer Wortman, 'Complexity of combinatorial market makers', in *Proceedings of the 9th ACM conference on Electronic commerce*, pp. 190–199. ACM, (2008).
- [10] Yiling Chen and Jennifer Wortman Vaughan, 'A new understanding of prediction markets via no-regret learning', in *Proceedings of the 11th ACM conference on Electronic commerce*, pp. 189–198. ACM, (2010).
- [11] citizeninvestor.com. citizeninvestor, 2015. Available online at <http://www.citizeninvestor.com>.
- [12] Theodore Groves and John Ledyard, 'Optimal allocation of public goods: A solution to the "free rider" problem', *Econometrica: Journal of the Econometric Society*, 783–809, (1977).
- [13] Mingyu Guo and David M Pennock, 'Combinatorial prediction markets for event hierarchies', in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pp. 201–208. International Foundation for Autonomous Agents and Multiagent Systems, (2009).
- [14] Robin Hanson, 'Combinatorial information market design', *Information Systems Frontiers*, **5**(1), 107–119, (2003).
- [15] Robin Hanson, 'Logarithmic markets coring rules for modular combinatorial information aggregation', *The Journal of Prediction Markets*, **1**(1), 3–15, (2012).
- [16] Robin Hanson et al., 'Shall we vote on values, but bet on beliefs?', *Journal of Political Philosophy*, (2003).
- [17] Paul J Healy, 'Learning dynamics for mechanism design: An experimental comparison of public goods mechanisms', *Journal of Economic Theory*, **129**(1), 114–149, (2006).
- [18] Leslie M Marx and Steven A Matthews, 'Dynamic voluntary contribution to a public project', *The Review of Economic Studies*, **67**(2), 327–358, (2000).
- [19] John Morgan, 'Financing public goods by means of lotteries', *The Review of Economic Studies*, **67**(4), 761–784, (2000).
- [20] neighbourly.com. neighbourly, 2015. Available online at <http://www.neighbourly.com>.
- [21] A Othman and T Sandholm, 'Homogeneous risk measures and liquidity-sensitive automated market makers', in *Proceedings of the 7th Workshop on Internet and Network Economics*, pp. 314–325, (2011).
- [22] Abraham Othman, David M Pennock, Daniel M Reeves, and Tuomas Sandholm, 'A practical liquidity-sensitive automated market maker', *ACM Transactions on Economics and Computation*, **1**(3), 14, (2013).
- [23] Thomas R Palfrey and Howard Rosenthal, 'Participation and the provision of discrete public goods: a strategic analysis', *Journal of public Economics*, **24**(2), 171–193, (1984).
- [24] Walter Rudin, *Real and complex analysis*, Tata McGraw-Hill Education, 1987.
- [25] David Schmidtz. The limits of government (boulder, 1991).
- [26] Alexander Tabarrok, 'The private provision of public goods via dominant assurance contracts', *Public Choice*, **96**(3–4), 345–362, (1998).
- [27] Robertas Zubrickas, 'The provision point mechanism with refund bonuses', *Journal of Public Economics*, **120**, 231–234, (2014).

Welfare of Sequential Allocation Mechanisms for Indivisible Goods

Haris Aziz¹ and Thomas Kalinowski² and Toby Walsh³ and Lirong Xia⁴

Abstract. Sequential allocation is a simple and attractive mechanism for the allocation of indivisible goods used in a number of real world settings. In sequential allocation, agents pick items according to a policy, the order in which agents take turns. Sequential allocation will return an allocation which is Pareto efficient – no agent can do better without others doing worse. However, sequential allocation may not return the outcome that optimizes the social welfare. We consider therefore the relationship between the welfare and the efficiency of the allocations returned by sequential allocation mechanisms. We then study some simple computational questions about what welfare is possible or necessary depending on the choice of policy. Over half the problems we study turn out to be tractable, and we give polynomial time algorithms to compute them. We also consider a novel control problem in which the Chair chooses a policy to improve social welfare. Again, many of the control problems we study turn out to be tractable, and our results give polynomial time algorithms. In this case, tractability is a good thing so that the Chair can improve the social welfare of the allocation.

INTRODUCTION

The fair division of resources is a central problem in social choice. One challenging case in fair division is when the goods being allocated are indivisible. For instance, we might want be interested in allocating courses to students at an university, time windows on an expensive scientific instrument to different groups of scientists, or landing slots on a runway to different airlines.

A simple mechanism to allocate indivisible goods like this is *sequential allocation* [9]: agents simply take turns to pick items. This leaves open the particular order (the “policy”) used to pick items. For example, in a balanced alternating policy, agents pick items in rounds, every agent picks one item in each round, and the order of agents is reversed between rounds. On the other hand, in a balanced policy, the agents simply have the same number of turns, and there is no restriction that picking happens in rounds with each agent getting one item in each round. Throughout this paper to make life simple and ensure balance is indeed possible, we will assume that the number of items is an integer multiple of the number of agents.

The actual policy used may not be fixed in advance. For example, sequential allocation is used to allocate courses to students at the Harvard Business School [10], and the policy used is chosen uniformly from the space of all balanced alternating policies by randomly ordering the students in the initial round. Whilst this may be perceived

to be procedurally fair, it does not necessarily maximize the welfare of the agents.

This suggests a number of questions about the social welfare that can or must be achieved by sequential allocation mechanisms. Do we *necessarily* achieve a minimum acceptable welfare whatever policy is chosen? Is it *possible* that the welfare is above some given amount? What is the *maximum* or *minimum* welfare that can be achieved? These questions are closely related to an interesting control problem. Can a (benevolent) chair choose a policy not at random but to improve or maximize welfare? They are also related to the expected welfare when the policy is chosen at random, as at the Harvard Business School. The expected welfare is between the minimum welfare that is necessary and the maximum welfare that is possible. Indeed, if the minimum and maximum welfare are different, then the expected welfare is strictly between them.

We study these problems about the welfare possible or necessary from a computational perspective. We consider classes of policies considered in previous work (e.g. [8, 5, 1]), and used in real life setting like the previously mentioned course allocation mechanism from Harvard Business School. As our results show (summarized in Table 1 at the end of the paper), over half of these problems are polynomial time solvable. This is a good thing. We want to be able to compute the welfare possible or necessary. We want the Chair potentially to be able to improve the welfare by choosing a good policy. We want to be able to compute the expected welfare of the agents. Our results provide efficient algorithms to compute answers to these questions.

Sequential allocation is an ordinal mechanism. That is, it merely requires agents to declare an ordering over items. It does not require the agents to declare their actual utilities. However, to compute the welfare of an outcome, we need to know the utilities of the agents. This does not necessarily mean we need to elicit the utilities explicitly. A significant body of work in the fair division literature supposes agents have utilities that are simply derived from their ordinal preferences (e.g. an agent’s utility is simply the sum of the Borda scores for the items). For example, Brams *et al.* [7] study fairness criteria using, amongst others, Borda utilities derived from the ordinal preferences. As a second example, Bouveret and Lang [5] consider sequential allocation mechanisms with Borda, lexicographical or quasi-indifferent utilities. As a third example, Kalinowski *et al.* [13] compute the optimal policies for sequential allocation supposing agents have Borda utilities. As a fourth example, Baumeister *et al.* [3] study ordinal mechanisms based on Borda, lexicographical, quasi-indifferent or *k*-approval utilities. As a fifth example, Darmann and Schauer [11] consider mechanisms that maximize the Nash product social welfare supposing Borda, lexicographical or 0/1 utilities. As a sixth example, Fujita *et al.* [12] study mechanisms for house allocation supposing lexicographical utilities. Many of our results hold

¹ Data61 and UNSW, email: haris.aziz@data61.csiro.au

² University of Newcastle, email: thomaskalinow@gmail.com

³ UNSW and Data61, email: toby.walsh@data61.csiro.au

⁴ RPI, email: xialirong@gmail.com

in the special case of Borda or lexicographical utilities, and thus only require agents to declare ordinal preferences.

Even if we elicit general utilities as opposed to ordinal preferences, there are some advantages to an ordinal mechanism like sequential allocation. First, it is easier for agents to verify that the sequential allocation mechanism has been applied correctly compared to, say, a black-box cardinal mechanism that maximizes welfare. The chair can nevertheless still choose a policy that maximizes welfare. Second, the sequential allocation mechanism can easily ensure additional constraints like, for instance, that all agents receive the same number of items, or that an agent is not allocated two items that are incompatible with each other. Such constraints can significantly increase the complexity of applying a cardinal mechanism. Third, sequential allocation mechanisms, especially when restricted to a class like balanced alternation policies, may be perceived to be procedurally fair. Fourth, there is typically less opportunity for agents to act strategically with an ordinal mechanism like sequential allocation.

To demonstrate that ordinal mechanisms may offer less opportunity for strategic behaviour than cardinal mechanisms, consider 2 agents and 4 items, a to d . Suppose agent 1 sincerely declares Borda utilities: 4 for a , 3 for b , 2 for c and 1 for d . If we use the sequential allocation procedure with a balanced alternation policy then agent 1 always gets one of its top two choices or both, irrespective of how agent 2 acts, strategically or sincerely. Consider now the cardinal mechanism that maximizes the utilitarian welfare. Suppose agent 2 has the same Borda utilities as agent 1 but strategically declares an utility of 5 for a , 4 for b , 1 for c and 0 for d . Note that the total utility declared by agent 2 is the same as the sum of the Borda scores of agent 1. With these declared utilities and the cardinal mechanism that maximizes the utilitarian welfare, agent 2 now gets both of agent 1's top two choices. Thus, we see more strategic outcomes are possible with this cardinal mechanism.

WELFARE AND EFFICIENCY

When agents pick sincerely, sequential allocation is guaranteed to return a Pareto efficient outcome. No agent can do better without at least one being worse off. However, sequential allocation is not guaranteed to maximize the social welfare of the outcome. We consider therefore the precise relationship between social welfare and efficiency. We suppose that there are n agents being allocated $m = nk$ items for integer $k \geq 1$. Agents have additive utilities over the items. Agents convert these into a strict ordinal ranking over items, breaking any ties in utility in some fixed way.

The welfare of an agent is simply the sum of the utilities of the items allocated to that agent. The utilitarian welfare is the sum of the welfare of the agents, whilst the egalitarian welfare is that of the worst off agent (or agents). The sequential allocation mechanism is parameterized by the policy, the order in which agents pick items. For example, with the policy 123321, agent 1 picks first, then agent 2, then agent 3 before we repeat in reverse. An allocation is an assignment of items to agents. One allocation *Pareto improves* another iff each agent has at least the same utility in the first, and there is at least one agent where the utility is greater. An allocation is *Pareto efficient* iff there is no allocation which Pareto improves it. For every Pareto efficient allocation, there exists a policy such that sincere picking with this policy generates this allocation. We can construct this policy using the greedy algorithm in the proof of Proposition 1 in [8]. The reverse, however, is not true. Sincere picking may not return a Pareto efficient allocation.

Remark 1. *Sincere picking can generate allocations that are not*

Pareto efficient.

Proof: Consider the policy 1221. Suppose the agents' utilities are as follows

	a	b	c	d
1	5	4	2	0
2	8	2	1	0

Both agents have the same total utility over the items. Sincere picking gives items a and d to agent 1 and items b and c to agent 2. This gives an utility of 5 to agent 1 and of 3 to agent 2. If they swap allocations, then the utility of agent 1 increases to 6, and of agent 2 to 8. Hence, sincere picking leads to an allocation that is not Pareto efficient, and does not have the optimal egalitarian or utilitarian social welfare. \diamond

We contrast this observation with Proposition 1 in [8]. This looks just at the rank of items in an agent's preference ordering, ignoring their precise utilities. Given two sets of items S and S' with $|S| = |S'|$, an allocation of items S to an agent *dominates* the allocation of items S' iff for every item in $S - S'$ there is a different item in $S' - S$ that is strictly less preferred. They then define an ordering, *ordinal efficiency* in terms of such domination. This is a strictly weaker ordering than Pareto efficiency which is defined in terms of utilities rather than ordinal rankings.

Proposition 1 in [8] demonstrates that ordinal efficiency corresponds *exactly* to allocations generated by sequential allocation supposing sincere picking. On the other hand, only a subset of the allocations returned by sequential allocation are Pareto efficient, and only a subset again maximize the egalitarian social welfare. However, one of these allocations is certainly Pareto efficient.

Remark 2. *There exists an allocation with the maximum possible egalitarian social welfare that is also Pareto efficient.*

It follows quickly that there always exists a policy for sequential allocation that gives an allocation with the maximum possible egalitarian social welfare supposing sincere picking. Note that this does not rule out other allocations which maximize egalitarian social welfare which are not ordinal efficient, and which cannot be generated by sequential allocation with sincere picking.

Example 1. *Suppose we have three agents (1 to 3), three items (a to c), and Borda utilities. Let agent 1 have a preference order bac , agent 2 have abc , and agent 3 have acb . Then allocating a to 1, b to 2 and c to 3 maximizes the egalitarian social welfare. However, there is no policy for sequential allocation that will return such an allocation supposing agents pick sincerely as no agent gets a first choice item.*

Maximizing the utilitarian social welfare also does not conflict with Pareto efficiency. In this case, we point out the well-known fact that *any* allocation that maximizes utilitarian social welfare is Pareto efficient.

Remark 3. *Any allocation with the maximum possible utilitarian social welfare is also Pareto efficient.*

Again it follows quickly that there exists a policy for sequential allocation that gives an allocation with the maximum possible utilitarian social welfare supposing sincere picking.

POSSIBLE AND NECESSARY WELFARE

Since sequential allocation may not return allocations that are optimal from either an egalitarian or utilitarian perspective, we turn to the (computational) questions of what social welfare is possible or

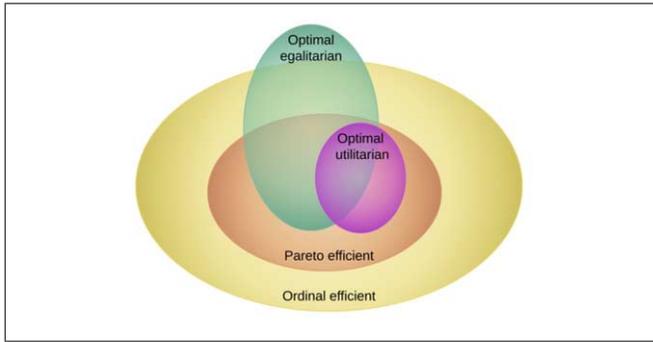


Figure 1: Inclusion relationship between different allocations. Sequential allocation with sincere picking can generate any ordinally efficient allocation.

necessary. Note that throughout this paper, we suppose agents pick sincerely. Whilst strategic behaviour may be beneficial, risk averse agents will tend to pick sincerely, especially when the policy and/or utilities are private information. Nevertheless, it is interesting future work to consider agents acting strategically [14]. We consider four decision problems related to the egalitarian or utilitarian welfare possible or necessary. We also consider different class of sequential mechanisms that depend on the class of the picking sequences allowed.

POSSIBLE EGALITARIAN WELFARE

Input: a set of n items, m agents each with utilities over the items, a class of policies, and an integer t .

Question: Is there a policy in the class that results in an allocation with an egalitarian social welfare of t or greater supposing agents pick items sincerely?

POSSIBLE UTILITARIAN WELFARE

Input: a set of n items, m agents each with utilities over the items, a class of policies, and an integer t .

Question: Is there a policy in the class that results in an allocation with an utilitarian social welfare of t or greater supposing agents pick items sincerely?

NECESSARY EGALITARIAN WELFARE

Input: a set of n items, m agents each with utilities over the items, a class of policies, and an integer t .

Question: Does every policy in the class result in an allocation with an egalitarian social welfare of t or greater supposing agents pick items sincerely?

NECESSARY UTILITARIAN WELFARE

Input: a set of n items, m agents each with utilities over the items, a class of policies, and an integer t .

Question: Does every policy in the class result in an allocation with an utilitarian social welfare of t or greater supposing agents pick items sincerely?

The possible and necessary welfare questions answer a *policy control* problem: can the chair choose a policy to achieve a given social welfare? Similar control problems have been considered previously [1] but with the goal of allocating particular items to agents, rather than, as here, of achieving a particular welfare. Note that we suppose we know the (private) utilities of the agents. As mentioned before, we may relax this assumption by supposing that the utilities are simple functions of the ordinal rank (e.g. Borda, lexicographical or

quasi-indifferent scores). As this is a special case of general utilities, any result that control takes polynomial time in the general case will map onto a polynomial time result in this more restricted setting. When we prove that a particular possible or necessary welfare problem takes polynomial time to solve, we will typically do so by answering a closely related maximization or minimization problem. Such problems are interesting in their own right. We consider four such function problems that compute the maximal or minimal social welfare.

MINIMUM EGALITARIAN WELFARE

Input: a set of n items, m agents each with utilities over the items, and a class of policies.

Output: The minimum egalitarian social welfare possible over all policies supposing agents pick items sincerely.

MINIMUM UTILITARIAN WELFARE

Input: a set of n items, m agents each with utilities over the items, and a class of policies.

Output: The minimum utilitarian social welfare possible over all policies supposing agents pick items sincerely.

MAXIMUM EGALITARIAN WELFARE

Input: a set of n items, m agents each with utilities over the items, and a class of policies.

Output: The maximum egalitarian social welfare possible over all policies supposing agents pick items sincerely.

MAXIMUM UTILITARIAN WELFARE

Input: a set of n items, m agents each with utilities over the items, and a class of policies.

Output: The maximum utilitarian social welfare possible over all policies supposing agents pick items sincerely.

Throughout the paper, we assume that agents have strict ordinal preferences. In some of the proofs, whenever there are ties in the *utilities*, we can obtain strict ordinal preferences by perturbing the utilities by an arbitrarily small margin. The arguments for the reductions are not affected.

ALL POSSIBLE POLICIES

We first consider the case when any policy, balanced or unbalanced is possible. In this case it is easy to maximize the utilitarian social welfare. The chair just need to choose a policy that gives items to the agents which value them most. Recall that we are assuming throughout that agents are behaving sincerely.

Theorem 1. *The MAXIMUM and POSSIBLE UTILITARIAN WELFARE problems take polynomial time to solve.*

Proof: We order the items by the maximum utility assigned by any agent. Ties can be broken in any way. We then construct the policy that allocates items in this order choosing the agent who gives an item the greater utility. No allocation can do better than this. \diamond

The MINIMUM EGALITARIAN WELFARE problem also takes polynomial time to solve. It is always zero as there are policies in which one agents gets no turns to pick items. On the other hand, the POSSIBLE EGALITARIAN WELFARE problem is intractable in general, even in the special case that all the agents have identical utilities for the items. We note that this problem has previously been shown to be NP-complete but not in the strong sense (Theorem 5.1 in [16]).

Theorem 2. *The POSSIBLE EGALITARIAN WELFARE problem for m items and n agents is strongly NP-complete when $m \geq 2n$ even when agents have the same ordinal preferences.*

Proof: Membership in NP is shown by giving the policy. To show NP-hardness, we consider $m = 2n$. For larger m , we add dummy items to which all agents assign the same zero utility. Recall that we have some fixed tie breaking mechanism to order items with the same utility. In this and other proofs that follow, we can replace items with zero utility by items with some uniform small, non-zero utility. This merely makes the algebra a little more complex so, to ease exposition, we present proofs here using items of zero utility. The proof uses a reduction from numerical 3-dimensional matching. Given an integer t and 3 multisets $X = \{x_1, \dots, x_n\}$, $Y = \{y_1, \dots, y_n\}$ and $Z = \{z_1, \dots, z_n\}$ of integers with $\sum_{i=1}^n (x_i + y_i + z_i) = nt$, this problem asks if there are permutations σ and π such that $x_i + y_{\sigma(i)} + z_{\pi(i)} = t$ for all $i \in [n]$. We construct an allocation problem over n agents and $m \geq 2n$ items as follows. Let $u = 1 + \sum_{i=1}^n z_i$. For every $j \in [n]$, there is a “big” item with utility $u + x_i + y_j$ for agent i ($i = 1, \dots, n$) and a “small” item which all agents give utility z_j . Finally, there are $m - 2n$ items with zero utility for all agents. We ask if we can achieve an egalitarian welfare of $u + t$. To achieve this, each agent must get precisely a utility of $u + t$. This is only possible if each agent gets one big item and one small item, and $x_i + y_{\sigma(i)} + z_{\pi(i)} = t$ where $\sigma(i)$ and $\pi(i)$ denote are the indices of the big and the small item obtained by agent i . Therefore, we can achieve the egalitarian welfare of $u + t$ iff there is a solution of the original numerical 3-dimensional matching problem. \diamond

Computing the maximum egalitarian welfare possible is also intractable in general.

Theorem 3. *The MAXIMUM EGALITARIAN WELFARE problem for m items and n agents is NP-hard to compute when $m \geq 2n$ even when the agents have the same ordinal preferences.*

Proof: We use the same reduction as in the last proof. \diamond

In the more restricted setting that utilities are Borda scores, computing the possible egalitarian welfare remains intractable. We thank an anonymous reviewer of an earlier version of this paper for suggesting this result.

Theorem 4. *With Borda utilities, the POSSIBLE EGALITARIAN WELFARE problem for m items and n agents is NP-complete when $m \geq \frac{12n+4}{5}$.*

Proof: We use the reduction in the proof of Theorem 3 in [4]. This reduction proves that deciding if there is an allocation with an egalitarian social welfare greater than or equal to some constant t is NP-complete even when utilities are Borda scores. Note that, unlike the previous two proofs, agents in this reduction do not share the same ordinal preferences. It is easy to show that there is a policy for sequential allocation that finds the precise allocation constructed in this reduction. Note also that $\frac{12n+4}{5} > 2n$ so this reduction uses slightly more items than the previous two proofs, in addition to requiring agents to have different ordinal preferences. \diamond

It follows that the MAXIMUM EGALITARIAN WELFARE problems is NP-hard to compute in this setting. Similarly, with lexicographical utilities it is intractable to compute the egalitarian welfare possible,

Theorem 5. *With lexicographical utilities, the POSSIBLE EGALITARIAN WELFARE problem for m items and n agents is NP-complete when $m \geq 3n$.*

Proof: This follows from Theorem 1 in [4]. Note that we again need more than the $2n$ items used in the first two reductions. We have also again relaxed the assumption that the agents have the same ordinal preferences. \diamond

It follows immediately that the MAXIMUM EGALITARIAN WELFARE problem is NP-hard to compute with lexicographical utilities. Finally, the NECESSARY EGALITARIAN WELFARE problem is trivial: if an agent does not get a turn, their welfare is zero.

BALANCED POLICIES

It might be considered unfair to use any policy, for example one in which one agent gets many more items than another. Whilst looking for allocations that maximize fairness and efficiency, Brams and King [8] observe that “*the symbolic value of giving players equal numbers of items, such as landing slots at an airport, may be important*”. We therefore consider the restricted class of balanced policies. In a *balanced* policy, each agent gets the same number of items. Recall that we suppose the number of items is an integer multiple of the number of agents. Hence, we can give each agent the same number of items. Of course, limiting sequential allocation to balanced policies impacts the social welfare that can be obtained.

To maximize utilitarian welfare, we cannot simply give items to the agents that value them most. This may violate balance. Despite this restriction, we can still find the policy that maximizes the utilitarian welfare in polynomial time.

Theorem 6. *The MAXIMUM and POSSIBLE UTILITARIAN WELFARE problems for balanced policies take polynomial time to solve.*

Proof: We suppose that there are kn items to divide between the n agents. We set up a min cost max flow problem. We connect the source node to nodes representing the agents, each with a capacity of k and no cost. We connect the nodes representing agents to nodes representing the items. Each edge has a capacity of 1, and a cost equal to minus the utility that the agent assigns to the item. Finally we connect the nodes representing the items to the target node, each with an edge of capacity 1 and zero cost. We find a Pareto efficient allocation from any such flow using the top trading cycle algorithm [17]. A policy can be constructed that achieves this Pareto efficient allocation by again exploiting Proposition 1 in [8]. \diamond

By comparison, the NECESSARY UTILITARIAN WELFARE problem is intractable for balanced policies.

Theorem 7. *The NECESSARY UTILITARIAN WELFARE problem for balanced policies is coNP-complete for $m \geq 2n$.*

Proof: We reduce from the NECESSARY ITEM problem. This asks if a given agent necessarily gets an given item irrespective of the policy used. The NECESSARY ITEM problem for balanced policies is coNP-complete even when limited to an agent’s most preferred item and $m = 2n$ [1]. Let one agent have utility of 1 for her most preferred item, zero utility for all others, and the other agents all have utility 1 for every item. Then the NECESSARY ITEM problem is equivalent to asking if an utilitarian welfare of m or more is necessary. \diamond

It follows that the MINIMUM UTILITARIAN WELFARE problem for balanced policies is NP-hard to compute. Restricting to balanced policies also does not change the intractability of computing the egalitarian welfare that is possible.

Theorem 8. *The POSSIBLE EGALITARIAN WELFARE problem for balanced policies is NP-complete for $m \geq 2n$ even when agents have the same ordinal preferences.*

Proof: This follows almost immediately from the reduction used in the proof of Theorem 2. Note that this reduction uses policies in which not all agents get the same number of items. However, such unbalanced policies can be ignored as they result in poor egalitarian welfare. Note also that when a numerical 3-dimensional matching exists, the corresponding successful policy constructed in the reduction is balanced. \diamond

It follows immediately that MAXIMUM EGALITARIAN WELFARE problem is NP-hard to compute for balanced policies. Note that an easy reduction from the EQUI-PARTITION problem demonstrates that the POSSIBLE EGALITARIAN WELFARE problem for balanced policies is NP-complete even for just two agents with identical utilities. On the one hand, this is a more restricted setting than Theorem 8 as we now have only 2 agents and they have identical utilities. On the other hand, this is a weaker result, as it is not strong NP-completeness, and dynamic programming will return a result in polynomial time supposing utilities are specified in unary.

With lexicographical utilities and balanced policies, it remains intractable to compute the egalitarian welfare possible.

Theorem 9. *With lexicographical utilities, the POSSIBLE EGALITARIAN WELFARE problem for balanced policies is NP-complete for $m \geq 3n$.*

Proof: We can adapt the reduction in the proof of Theorem 1 in [4]. Note that we have again relaxed the assumption from Theorem 8 that agents have the same ordinal preferences. \diamond

Finally, computing the egalitarian welfare necessary is intractable for balanced policies.

Theorem 10. *The NECESSARY EGALITARIAN WELFARE problem for balanced policies is coNP-complete for $m \geq 2n$.*

Proof: The same reduction as in the proof of Theorem 7. \diamond

RECURSIVELY BALANCED POLICIES

Balanced policies might still be considered unfair. For example, a policy like 11112222 favours the first agent even though it is balanced, guaranteed to return a Pareto efficient allocation, and is strategy proof. We therefore consider an even more restrictive class: recursively balanced policies. In such a policy, items are allocated in rounds, and each agent appears once in each round. For simplicity, we again suppose that the number of items is an integer multiple of the number of agents and add dummy items of no utility otherwise. When the number of items equals the number of agents, all balanced policies are recursively balanced. For this reason, we focus on problems where the number of items exceeds the number of agents.

Formally, a policy is *recursively balanced* iff it is the empty policy, or it is non-empty and every agent has exactly one turn in the first n picks and the remaining policy is also recursively balanced. Recursively balanced policies include the alternating policy (12121212...), the balanced alternating policy (12211221...), as well as the Thue-Morse sequence (1221211221...). With two agents, recursively balanced policies are concatenations of 12 and 21. Other simple properties of recursively balanced policies follow immediately from their definition. For example, no agent has more than two successive picks in a recursively balanced policy. Limiting sequential allocation to recursively balanced policies may further impact the social welfare that can be obtained.

There are several situations where focusing on recursively balanced policies does not hurt welfare. For example, with Borda utilities, the expected utilitarian social welfare for two agents is not impacted by limiting allocation to recursively balanced policies. The

simple alternating policy which is recursively balanced is optimal in expectation [13]. Similarly for Borda utilities and small n , the expected egalitarian social welfare for two agents is not impacted. The authors [13] computed the policies that maximize expected egalitarian social welfare for up to 12 items and for each n , at least one optimal policy is recursively balanced.

In general, restricting to recursively balanced policies results in it being intractable to decide if a given egalitarian or utilitarian welfare can or must be achieved.

Theorem 11. *The POSSIBLE EGALITARIAN and POSSIBLE UTILITARIAN WELFARE problems for recursively balanced policies are NP-complete for $m \geq 2n$, whilst the NECESSARY EGALITARIAN and NECESSARY UTILITARIAN WELFARE are coNP-complete for $m \geq 3n$.*

Proof: We reduce from the corresponding problem of deciding whether the top k most preferred items of an agent are possible or necessary [1]. TOP- k POSSIBLE SET for recursively balanced policies is NP-complete for $m \geq 3n$. We reduce this to POSSIBLE EGALITARIAN WELFARE as follows. Let one agent have utility of k^2 for their i th most preferred items ($i \leq k$), zero utility for all others, and the other agents all have utility k^3 or greater for any item. Then TOP- k POSSIBLE SET is equivalent to asking if an egalitarian welfare of k^3 or more is possible. We also reduce TOP- k POSSIBLE SET to POSSIBLE UTILITARIAN WELFARE as follows. Let one agent have utility of mk^2 for their i th most preferred items ($i \leq k$), zero utility for all others, and all other agents have utility of k or less for any item. Then TOP- k POSSIBLE SET is equivalent to asking if an utilitarian welfare of mk^3 or more is possible.

TOP- k NECESSARY SET is coNP-complete for recursively balanced policies when $m \geq 2n$. We reduce this to NECESSARY EGALITARIAN WELFARE as follows. Let one agent have total utility of k^2 for their k most preferred items zero utility for all others, and the other agents all have utility k^3 or greater for any item. Then TOP- k NECESSARY SET is equivalent to asking if an egalitarian welfare of k^2 is necessary. We also reduce TOP- k NECESSARY SET to NECESSARY UTILITARIAN WELFARE as follows. Let one agent have utility of mk^2 for their i th most preferred items ($i \leq k$), zero utility for all others, and all other agents have utility of k or less for any item. Then TOP- k NECESSARY SET is equivalent to asking if an utilitarian welfare of mk^3 or more is necessary. \diamond

Even when agents have identical utilities, these problems can remain intractable.

Theorem 12. *When allocating $2n$ items between two agents, the POSSIBLE EGALITARIAN WELFARE problem for recursively balanced policies is NP-complete even when agents have identical utilities given in binary.*

Proof: Membership in NP is clear. For the hardness we use reduction from PARTITION: for positive integers a_1, \dots, a_n with $a_1 + \dots + a_n = 2B$, the problem is to decide if there is a nonempty set $I \subseteq [n]$ with $\sum_{i \in I} a_i = B$. We reduce this to the POSSIBLE EGALITARIAN WELFARE problem for two agents and $2n$ items with utilities $c_1 = 2B, c_{2n} = 0$, and

$$c_{2j} = c_{2j+1} = c_{2j-1} - a_j \quad \text{for } j = 1, 2, \dots, n-1.$$

Let $C = \sum_{i=1}^{2n} c_i$ be the sum of the utilities, and u_i be the utility received by agent i in a given allocation. Note that an egalitarian welfare of $C/2$ is equivalent to $u_1 = u_2$. In round j , the items with utilities c_{2j-1} and c_{2j} are allocated. From $c_{2j-1} - c_{2j} = a_j$ it follows

that the difference $u_1 - u_2$ between the agents' utilities increases by a_j if agent 1 starts and decreases if agent 2 starts. Let $I \subseteq [n]$ be the set of rounds in which agent 1 starts. An egalitarian social welfare of $C/2$ is achieved if and only if

$$0 = u_1 - u_2 = \sum_{j \in I} a_j - \sum_{j \in [n] \setminus I} a_j,$$

That is, if and only if there is a perfect partition. \diamond

BALANCED ALTERNATING POLICIES

The final and most restricted class of policies we consider is that of balanced alternating. This is the subclass of recursively balanced policies in which each round is the reverse of the previous. When allocating students to courses at the Harvard Business School, such a policy is chosen uniformly at random from the space of all possible balanced alternating policies. This gives a form of procedural fairness.

Theorem 13. *The POSSIBLE EGALITARIAN and POSSIBLE UTILITARIAN WELFARE problems for balanced alternating policies are NP-complete for $m \geq 2n$, whilst the NECESSARY EGALITARIAN and NECESSARY UTILITARIAN WELFARE are coNP-complete again for $m \geq 2n$.*

Proof: As in proof of Theorem 11, by reduction from corresponding TOP- k POSSIBLE or NECESSARY SET problem for balanced alternating policies. Given an allocation problem, preference profiles for all the agents, a class of policies, and a designated agent, the TOP- k POSSIBLE problem asks if there is a policy such that the agent gets their top k most preferred items. Given an allocation problem, preference profiles for all the agents, a class of policies, and a designated agent, the TOP- k NECESSARY SET problem asks if the agent necessarily gets their top k most preferred items irrespective of the policy used. TOP- k POSSIBLE for balanced alternating policies is NP-complete for $m \geq 2n$, whilst TOP- k NECESSARY SET is coNP-complete [1]. \diamond

It follows that it is NP-hard to compute the probability that the Harvard Business School course allocation mechanism returns an allocation with egalitarian or utilitarian welfare greater than or equal to some given value t .

TWO AGENTS

We now consider some special and more tractable cases. With two agents, we can find a balanced policy that maximizes the egalitarian or utilitarian welfare in polynomial time.

Theorem 14. *The MAXIMUM EGALITARIAN and MAXIMUM UTILITARIAN WELFARE problems with balanced policies can be solved in $O(w^2n^3)$ and $O(wn^2)$ time respectively when allocating $2n$ items between two agents with utilities that are (possibly different) integers taken from $[0, w]$.*

Proof: We put the items into some (arbitrary) order and consider how each item is allocated in turn. We construct a $2n$ step dynamic program in which the i th step corresponds to the decision of where to allocate the i th item in this order. The states of this dynamic program are triples containing the number of items allocated to the first agent, the sum of the utilities of the items so far allocated to the first agent, and the sum of the utilities of the items so far allocated to the second agent. We can compute the number of items allocated to the

second agent from this. As both sums are bounded in size by $2wn$, this dynamic program has $O(w^2n^3)$ states. For the maximum utilitarian welfare, the states of the dynamic program can be simpler and just need to be pairs containing the number of items allocated to the first agent, and the sum of the utilities of the items so far allocated to both agents. \diamond

This result generalizes to a bounded number of agents. On the other hand, when utilities are specified in binary, an easy reduction from the PARTITION problem demonstrates that the POSSIBLE EGALITARIAN WELFARE problem is NP-complete even when the two agents have identical utilities. This is almost identical to Proposition 2 in [5] which shows that deciding if there is a policy that ensures a given expected egalitarian welfare is NP-complete when the utilities of the two agents are identical.

With recursively balanced policies, we consider the case where agents have the same ordinal ranking over items.

Theorem 15. *The MAXIMUM and POSSIBLE EGALITARIAN WELFARE problems for recursively balanced policies can be solved in $O(w^2n^2)$, whilst the MAXIMUM and POSSIBLE UTILITARIAN WELFARE problems can be solved in just $O(wn)$ time when allocating $2n$ items between two agents when agents have the same ordering over items but possibly different utilities, and utilities are integers drawn from $[0, w]$.*

Proof: We construct a n step dynamic program in which each step corresponds to allocating one item to each of the agents. The states of this program are pairs containing the sums of the utilities of items so far allocated to the two agents. As both sums are bounded by wn , there are $O(w^2n^2)$ states. By scanning the final step of the dynamic program that computes all possible partitions, we can compute the optimal egalitarian social welfare. To compute the optimal utilitarian social welfare, we can use a simpler dynamic program where the states are just the sum of the utilities allocated to the agents. \diamond

This result again generalizes to a bounded number of agents easily.

HOUSE ALLOCATION

Another more tractable case is house allocation, when we have only as many items as agents and each agent gets one item. Results for this setting imply results for recursively balanced and balanced alternating policies for $m = n$. For example, sequential allocation is used in many universities and residential colleges to assign rooms to students. In this case, we can solve the MAXIMUM and POSSIBLE EGALITARIAN WELFARE problems over all possible policies in polynomial time. We construct a graph between agents and items with edges for all items that have a utility greater than or equal to the desired egalitarian social welfare. The POSSIBLE EGALITARIAN WELFARE problem is solvable if we can find a perfect matching in this graph. To construct a satisfying policy, we find a Pareto efficient allocation from this matching using the top trading cycle algorithm [17]. A policy can be constructed that achieves this Pareto efficient allocation using Proposition 1 in [8].

We also show that NECESSARY EGALITARIAN WELFARE is polynomial-time solvable for house allocation.

Theorem 16. *NECESSARY EGALITARIAN WELFARE is polynomial-time solvable for house allocation.*

Proof. We first show that it can be checked in polynomial time whether $n - 1$ agents get allocated a target $n - 1$ set of items in the first $n - 1$ turns. The problem reduces to checking whether there

	all policies	balanced	recursively balanced	balanced alternating
POSSIBLE EGALITARIAN WELFARE	NPC P	NPC P (for $m = n$)	NPC P (for $m = n$)	NPC P (for $m = n$)
POSSIBLE UTILITARIAN WELFARE	P P	P P (for $m = n$)	NPC P (for $m = n$)	NPC P (for $m = n$)
NECESSARY EGALITARIAN WELFARE	P P	coNPC P (for $m = n$)	coNPC P (for $m = n$)	coNPC P (for $m = n$)
NECESSARY UTILITARIAN WELFARE	?	coNPC	coNPC	coNPC

Table 1. Summary of results when allocating m items between n agents: NPC=NP-complete, coNPC=coNP-complete, P=polynomial-time. Note that over half the entries in this table identify problems which can be solved in polynomial time.

exists a matching in which each of the $n - 1$ agents gets a more preferred item than the one not in the target set. If the allocation of the n agents is not ordinally efficient, it can be made ordinally efficient via trading cycles none of which will involve the n -th agent's item. This means that we can check whether there exists a sequence such that the n -th agent can get a particular item in the last turn. In order to get minimum egalitarian welfare, some agent i has to get a low preferred item o . If there exists a policy in which i gets o in some turn that is not the last turn, then i gets o or an even worse item if i 's turn is moved to the end of same policy. Thus we can check in polynomial time for each i and o whether i gets o in the last turn and identify the i 's and o 's for which this is possible and $u_i(o)$ is minimum. \square

The same statement also applies to recursively balanced and balanced alternative policies provided that $m = n$.

OTHER RELATED WORK

As mentioned earlier, Bouveret and Lang [5] consider the case in which the utilities of items are simply functions of the ordinal rankings. They prove that any recursively balanced policy tends to an allocation giving the optimal expected egalitarian or utilitarian welfare as the number of items grows, supposing sincere picking, utilities that are Borda scores and all ordinal rankings being equiprobable. In addition, they compute the optimal policies for maximizing the expected egalitarian or utilitarian welfare under the same assumptions for up to 12 items. The optimal policies for two agents and an even number of items are recursively balanced. Kalinowski *et al.* [13] prove that the alternating policy maximizes the expected utilitarian welfare under these same assumptions. Note that such results are about maximizing the *expected* welfare supposing limited knowledge about the utilities, whilst the results here about maximizing the *exact* welfare supposing the chair knows the *actual* utilities.

There has been some study of strategic behaviour of agents (as opposed to the chair) in the sequential allocation mechanism. It can, for example, be viewed as a repeated game. When all agents have complete information, we can compute the subgame perfect Nash equilibrium. This is unique and takes polynomial time to compute for two agents [15, 14], but for an arbitrary number of agents, there can be an exponential number of equilibria and computing even one is PSPACE-hard [14]. More recently, Bouveret and Lang [6] consider how an agent or coalition of agents can strategically mis-report their preferences in a sequential allocation mechanism supposing the other agents act sincerely. They show that the loss of social welfare caused by such manipulation is not great. For example, with Borda scoring, two agents, and the alternating policy, there was at most a 33% loss in the utilitarian welfare.

More recently, a family of rules for dividing indivisible goods among agents has been proposed that take as input the agents' ordinal rankings over the items, a scoring vector, and a social welfare aggregation function [4, 2]. They return the allocation that maximizes the social welfare according to this scoring rule and aggregation function. Whilst such rules have a number of desirable properties like monotonicity, it is computationally challenging to compute the actual allocation (unless we have a bounded number of agents in which case we can typically use dynamic programming). This contrasts with sequential allocation where computing the allocation take just linear time. Baumeister *et al.* [2] also compute the multiplicative/additive "price of elicitation-freeness", the worst-case ratio/difference in social welfare between such allocations and the allocation returned by sequential allocation. Whilst their results are limited to simple alternating policies, the prices are typically not great. For example, the optimal utilitarian welfare with Borda scores is at most twice that returned by sequential allocation using simple alternation.

CONCLUSIONS

We have considered the implications on social welfare of choosing different policies when using a sequential mechanism to allocate indivisible goods. In particular, we consider the (computational) questions of what welfare is possible or necessary. The former is related to the control problem in which a (benevolent) chair chooses a policy for the sequential allocation mechanism to improve the social welfare. These questions are also related to the expected welfare when we choose a policy uniformly at random. Our results are summarized in Table 1. We note again that over half the entries in this table are polynomial time algorithms. Many of these questions are computationally tractable, and our results provide efficient algorithms to answer them. There are many interesting open questions. For example, how difficult is it to find a recursively balanced policy that returns a Pareto efficient allocation supposing sincere agents? As a second example, how can the chair adaptively compute a policy as agents pick items to maximize social welfare? And what is the "price of adaptability", the ratio between the welfare achieved when the chair is adaptive and when the chair has to declare and fix the policy in advance?

REFERENCES

- [1] H. Aziz, T. Walsh, and L. Xia, 'Possible and necessary allocations via sequential mechanisms', in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, (IJCAI 2015)*, eds., Q. Yang and M. Wooldridge, (2015).

- [2] D. Baumeister, S. Bouveret, J. Lang, N. Nguyen, T. Nguyen, J. Rothe, and A. Saffidine, 'Axiomatic and computational aspects of scoring allocation rules for indivisible goods', in *5th International Workshop on Computational Social Choice (COMSOC)*, pp. 1–22, Pittsburgh, USA, (June 2014).
- [3] D. Baumeister, S. Bouveret, J. Lang, T. Nguyen, N. Nguyen, and J. Rothe, 'Scoring rules for the allocation of indivisible goods', in *Proceedings of 21st European Conference on Artificial Intelligence (ECAI 2014)*, eds., T. Schaub, G. Friedrich, and B. O'Sullivan, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pp. 75–80. IOS Press, (2014).
- [4] D. Baumeister, S. Bouveret, J. Lang, T. Nguyen, J. Rothe, and A. Saffidine, 'Positional scoring rules for the allocation of indivisible goods', in *11th European Workshop on Multi-Agent Systems (EUMAS)*, pp. 1–14, Toulouse, France, (December 2013).
- [5] S. Bouveret and J. Lang, 'A general elicitation-free protocol for allocating indivisible goods', in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, ed., Toby Walsh, pp. 73–78. IJCAI/AAAI, (2011).
- [6] S. Bouveret and J. Lang, 'Manipulating picking sequences', in *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI'14)*. IOS Press, (2014).
- [7] S. Brams, P. Edelman, and P. Fishburn, 'Fair division of indivisible items', *Theory and Decision*, **55**(2), 147–180, (2003).
- [8] S.J. Brams and D.L. King, 'Efficient fair division: Help the worst off or avoid envy?', *Rationality and Society*, **17**(4), 387–421, (2005).
- [9] S.J. Brams and A.D. Taylor, *Fair Division: From cake-cutting to dispute resolution*, Cambridge University Press, Cambridge, 1996.
- [10] E. Budish and E. Cantillon, 'The Multi-Unit Assignment Problem: Theory and Evidence from Course Allocation at Harvard', *American Economic Review*, **102**(5), 2237–71, (2012).
- [11] A. Darmann and J. Schauer, 'Maximizing nash product social welfare in allocating indivisible goods', *European Journal of Operational Research*, **247**(2), 548 – 559, (2015).
- [12] E. Fujita, J. Lesca, A. Sonoda, T. Todo, and M. Yokoo, 'A complexity approach for core-selecting exchange with multiple indivisible goods under lexicographic preferences', in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, eds., B. Bonet and S. Koenig, pp. 907–913. AAAI Press, (2015).
- [13] T. Kalinowski, N. Narodytska, and T. Walsh, 'A social welfare optimal sequential allocation procedure', in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI-2013)*. International Joint Conference on Artificial Intelligence, (2013).
- [14] T. Kalinowski, N. Narodytska, T. Walsh, and L. Xia, 'Strategic behavior when allocating indivisible goods sequentially', in *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence (AAAI 2013)*. AAAI Press, (2013).
- [15] D.A. Kohler and R. Chandrasekaran, 'A Class of Sequential Games', *Operations Research*, **19**, 270–277, (1971).
- [16] M. Roos and J. Rothe, 'Complexity of social welfare optimization in multiagent resource allocation', in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems, AAMAS10*, pp. 641–648, (2010).
- [17] L. Shapley and H. Scarf, 'On cores and indivisibility', *Journal of Mathematical Economics*, **1**(1), 23 – 37, (1974).

A Computational Approach to Consensus-Finding

Eric Grégoire¹ and Jean-Marie Lagniez²

Abstract. Consensus-finding plays a ubiquitous role in A.I. In this paper, a consensus among agents is defined as a non-contradictory fragment of all the information conveyed by the agents such that this fragment does not logically conflict with any of the agents. This concept is investigated in modal logic S5 in order to meet representation needs that are put in light by this concept of consensus itself. Interestingly, an optimization-based approach to compute maximal consensus is developed and shown experimentally efficient very often for both the standard Boolean and S5 frameworks.

1 INTRODUCTION

Consensus-finding plays a ubiquitous role in A.I. For example, interacting agents [4, 17] can need to target a consensual, shared, goal whereas negotiation can amount to finding and settling on a consensual agreement. More generally, a notion of consensus can prove helpful for reconciling several information³ sources, hereafter often simply called sources or agents. For example, cautious A.I. systems might rely on uncontroversial, consensual, fragments of the global information conveyed by several mutually conflicting belief sources [10, 7].

However, consensus paradigms are rarely defined in a precise way, especially in the context of several agents who are equipped with inferential reasoning capabilities. In this paper, a consensus among several sources is defined as a non-contradictory fragment of all the information conveyed by the sources such that this fragment does not logically conflict with any source. Interestingly, such a consensus as proposed in [5], might not contain only the information shared by every source; it can also contain some additional information that is in some sense possibly acceptable from the point of view of each source since it does not contradict it. Hence, each source might endorse the information in the consensus as this information either also belongs to the source or does not conflict with it.

First, we discuss this notion of consensus that has been defined in a Boolean setting in [5]. Then, it is extended to modal logic S5 (see for example [3, 2] for an introduction to modal logics) in order to meet representation needs that are put in light by this concept of consensus itself. The focus is on the practical computational extraction of maximal consensus. Noticeably, it is stressed that the computation of one maximal consensus diverges from the well-studied search for maximal satisfiable subsets of all the information conveyed by the sources. Interestingly, we provide an optimization schema that proves experimentally efficient even for very large sources for both the standard Boolean and S5 frameworks, extending also [5] in this latter aspect.

The paper is organized as follows. In the next section, we provide the main logical preliminaries and notations used throughout the paper. In section 3, we introduce several basic concepts of consensus before we push the envelope in section 4 by investigating consensus in modal logic S5. Section 5 presents a practical computational approach for the extraction of one consensus whereas section 6 extends it to S5. Section 7 reports our experimental study. The paper ends with a discussion and some promising perspectives for further research.

2 LOGICAL PRELIMINARIES

We consider the standard (Boolean logic) language \mathcal{L} of formulas, based on a denumerable set of Boolean variables \mathcal{P} , which are written a, b, \dots and can be assigned either *true* or *false*. The conjunctive, disjunctive, negation, material implication and equivalence connectives are written $\wedge, \vee, \neg, \rightarrow, \equiv$, respectively. Formulas and sets of formulas are denoted α, β, \dots and Φ, Γ, \dots , respectively. In the following, we assume n agents Φ_i where $i \in [1..n]$ and identify each agent with her knowledge (actually, with the part of her knowledge that is concerned by the search for one consensus). Without loss of generality, we often assume that each Φ_i is under clausal form (CNF): namely, is a conjunction of clauses, where a clause is a disjunction of literals and a literal is a possibly negated Boolean variable.

Φ_i is satisfiable iff there exists a truth value assignment of every variable such that all formulas in Φ_i are *true* according to usual compositional rules. Such an assignment is called a model of Φ_i . Any Boolean formula can be rewritten in linear time in CNF that is equivalent with respect to satisfiability. \vdash denotes the deduction relation: $\Phi_i \vdash \alpha$ iff α is *true* in all models of Φ_i . A tautology is *true* under any assignment; in other words, we have that $\vdash \alpha$ for every tautology α . Logically equivalent formulas are considered indistinguishable: for example, we do not distinguish between $a \vee b$ and $b \vee a$. $Th(\Phi_i)$ represents the set of deductive consequences (also called the deductive closure) of Φ_i , namely $Th(\Phi_i) = \{\alpha \in \mathcal{L} \text{ s.t. } \Phi_i \vdash \alpha\}$. Arrays of sets of formulas are called profiles and noted $\mathcal{S}, \mathcal{V}, \dots$. The cardinality of a set Φ is noted $\#\Phi$. We will use the concept of prime implicate, defined as follows. A prime implicate of a finite set Δ of formulas is any clause δ such that $\Delta \vdash \delta$, and, at the same time, $\vdash (\delta' \equiv \delta)$ for every clause δ' s.t. $(\Delta \vdash \delta' \text{ and } \delta' \vdash \delta)$. For readability reason, we will present elements of modal logic when needed.

3 CONSENSUS: BASIC DEFINITIONS AND PROPERTIES

Assume $\mathcal{S} = [\Phi_1, \dots, \Phi_n]$ represents n sources Φ_i where each Φ_i is such that $\Phi_i \subset \mathcal{L}$ and is satisfiable. In this paper, we consider consensus that are included within $\bigcup_{i=1}^n \Phi_i$, as defined in [5]:

¹ CRIL, Univ. Artois & CNRS, 62300 Lens, France gregoire@cril.fr

² CRIL, Univ. Artois & CNRS, 62300 Lens, France lagniez@cril.fr

³ In this paper, no difference is made between information, plans, desires, goals, knowledge and beliefs.

Definition 1. A set $\Gamma \subset \mathcal{L}$ is a consensus for \mathcal{S} iff $\Gamma \subseteq \bigcup_{i=1}^n \Phi_i$ and $\forall \Phi_i \in \mathcal{S} : \Gamma \cup \Phi_i$ is satisfiable.

Accordingly, a consensus is always satisfiable since no set is satisfiable together with an unsatisfiable one. Since each Φ_i is satisfiable, there exists always at least one consensus, which can be the empty set. When $\bigcup_{i=1}^n \Phi_i$ is satisfiable, it forms the only maximal consensus. In the general case, a consensus need not be unique. No logical consequence of a consensus contradicts any source: hence, a consensus can be identified with its deductive closure. Notice that it would be possible to require a consensus to be a subset of $Th(\bigcup_{i=1}^n \Phi_i)$ vs. a subset of $\bigcup_{i=1}^n \Phi_i$, however such an extended definition would not allow for the practical computational approach that we develop in this paper. Two natural definitions for maximal consensus are as follows [5], depending on whether maximality is considered with respect to set-theoretic inclusion or set cardinality.

Definition 2. A consensus Γ for \mathcal{S} is max_{\subseteq} iff $\forall \Theta$ s.t. $\Gamma \subset \Theta \subseteq \bigcup_{i=1}^n \Phi_i$, $\exists \Phi_i \in \mathcal{S}$ s.t. $\Theta \cup \Phi_i$ is unsatisfiable. A consensus Γ for \mathcal{S} is $max_{\#}$ iff $\forall \Theta$ s.t. $\Theta \subseteq \bigcup_{i=1}^n \Phi_i$ and $\#\Theta > \#\Gamma$, $\exists \Phi_i \in \mathcal{S}$ s.t. $\Theta \cup \Phi_i$ is unsatisfiable.

Clearly, any $max_{\#}$ consensus is a max_{\subseteq} consensus whereas the converse does not hold. From now on, we consider max_{\subseteq} and $max_{\#}$ consensus, only. We write *consensus* when there is no need to differentiate between max_{\subseteq} and $max_{\#}$. Notice that consensus and maximal (w.r.t. \subseteq or $\#$) satisfiable subsets of $\bigcup_{i=1}^n \Phi_i$ (in short MSSes) are closely related but different concepts. In the general case, a consensus is not one MSS that is satisfiable with each Φ_i : although every consensus is included in some MSSes, an MSS is not necessary a consensus. Actually, the sets of MSSes and of consensus can be disjoint.

Example 1. Assume that three political groups are negotiating for a possible coalition program while their individual political agendas are mutually conflicting. One maximal consensus is searched within all elements of the programs such that it does not contradict any of the agendas. Although a political group might find within the consensus some elements that do not belong to its own agenda, it might endorse the consensus since these elements do not contradict its own objectives. Assume for example that $\{it, tss, ids\}$ is a set of Boolean variables standing for Increase Taxation, Trim Social Security and Increase Defense Spendings, respectively. Let \mathcal{S} be the agendas of these three political groups: $\mathcal{S} = [\Phi_1, \Phi_2, \Phi_3]$ with $\Phi_1 = \{it, \neg tss, \neg it \rightarrow \neg ids\}$, $\Phi_2 = \{tss, it \rightarrow ids\}$ and $\Phi_3 = \{\neg ids\}$. It is easy to see that there are three $max_{\#}$ consensus for \mathcal{S} : $\Gamma_1 = \{\neg it \rightarrow \neg ids, it \rightarrow ids\}$, $\Gamma_2 = \{\neg ids, \neg it \rightarrow \neg ids\}$ and $\Gamma_3 = \{it, \neg it \rightarrow \neg ids\}$. For example, Γ_3 states “Increase taxation and if we do not increase taxation then we do not increase the defense spendings”. Remember that any consensus can be identified with its deductive closure: hence Γ_3 can be identified with $Th(\{ids\})$. Notice that none of the consensus is an MSS of $\bigcup_{i=1}^3 \Phi_i$.

The consensus concept is easily extended by requiring forms of integrity constraints to be obeyed. In the following, we assume that a set of integrity constraints, noted Ψ , is included in $\bigcup_{i=1}^n \Phi_i$ [5]. Ψ can need to be included in any consensus or, as in the next definition, be a deductive consequence of any consensus (definitions for maximal consensus are easily adapted, too). In a negotiation setting, integrity constraints can express elements that must belong to any consensus.

Definition 3. A set $\Gamma \subset \mathcal{L}$ is a consensus for \mathcal{S} under the constraints Ψ iff $\Gamma \subset \mathcal{L}$ is a consensus for \mathcal{S} and $\Gamma \vdash \Psi$.

Example 2. In the previous example, $\Gamma = \{it, \neg it \rightarrow \neg ids\}$ is a consensus for \mathcal{S} under the constraint $\Psi = \{it\}$. For example, there is no consensus for \mathcal{S} under the constraint $\Psi = \{\neg tss\}$ since tss is logically conflicting with Φ_2 .

Notice that the last definition could be easily extended in such a way that a consensus includes or entails a set of integrity constraints that is not included in $\bigcup_{i=1}^n \Phi_i$ or even $Th(\bigcup_{i=1}^n \Phi_i)$.

From a computational point of view, extracting one max_{\subseteq} consensus amounts to computing one MSS under an additional constraint of satisfiability with each Φ_i . Consequently, computing such consensus in this way is as hard as computing MSSes, which is known to be intractable in the worst case. Indeed, the computation of one MSS_{\subseteq} belongs to the $FP^{NP}[wit, log]$ class, i.e., the set of function problems that can be computed in polynomial time by executing a logarithmic number of calls to an NP oracle that returns a witness for the positive outcome [13]. Even worse, computing one $MSS_{\#}$ belongs to the $Opt-P$ class of problems [16], i.e., the class of functions computable by taking the maximum of the output values over all accepting paths of an NP machine.

It is also important to notice that the number of different consensus is exponential in the number of clauses in $[\Phi_1, \dots, \Phi_n]$ in the worst case. However, in many difficult negotiation problems and applications where a consensus needs to be found, extracting one maximal consensus is often the actual and sufficient problem to be solved. It can also be a useful starting point for further successful discussions. Accordingly, the focus in the rest of the paper is on the search for one maximal consensus; more precisely, we investigate the computation of one $max_{\#}$ consensus (remember that any $max_{\#}$ consensus is also a max_{\subseteq} consensus).

4 CONSENSUS IN A MODAL LOGIC OF POSSIBILITY AND NECESSITY

4.1 Motivations

We claim that the *possibility* and *necessity* modalities (noted \diamond and \square , respectively) of standard modal logics can be of specific interest in situations where consensus-finding must take place. In this section, we motivate the needs for the additional expressive power provided by these modal logics; we take advantage of these motivating examples to provide some early intuitive grasp about how consensus-finding will be actually implemented by reduction to standard Boolean logic. To this end, the reader only needs to remember at this stage that a formula $\diamond\alpha$ is intended to assert that α is “possible”, i.e., is true in some possible world.

Actually, the concept of consensus already allows an agent Φ_i to express that a formula α should be “possible” with respect to any consensus in the following sense. By definition, when $\alpha \in \Phi_i$ (and more generally when $\Phi_i \vdash \alpha$), no consensus for \mathcal{S} contradicts α or, equivalently, for any consensus Γ , we have that $\Gamma \not\vdash \neg\alpha$. In the example from the previous section, if one political agenda Φ_i contains $\neg taxes-increase$ then $taxes-increase$ does not belong to any consensus Γ and is not inferable from Γ . Hence, any consensus conveys the information that $\neg taxes-increase$ remains possible. This feature will be exploited to some extent to handle some occurrences of the modality operator \diamond in the search for consensus for \mathcal{S} . However, this cannot be extended in the general case. Indeed, inserting every formula of a set Θ inside Φ_i to express that each formula Θ must be possible does not always yield the intended result. The simplest example

of this is as follows. Assume Φ_i requires both a formula α and its contrary $\neg\alpha$ to be *true* in some (possibly different) possible worlds. Inserting both α and its contrary $\neg\alpha$ inside Φ_i will make Φ_i become unsatisfiable and collapse: no consensus will exist. On the contrary, both modal formulas $\diamond\alpha$ and $\diamond\neg\alpha$ are not mutually contradictory since they express that α and $\neg\alpha$ need to be *true* in some (different) possible worlds. Interestingly, by means of additional Boolean variables, it will be possible to reduce \mathcal{S} expressed using modal logic S5 into mere (clausal) Boolean logic, and reuse a computational approach to consensus-finding that has been developed in the standard Boolean framework [5].

In the political agenda example, Φ_i might require both *taxes-increase* and \neg *taxes-increase* to remain possible: any consensus should prevent any of those formulas from being derivable. In this way, the taxation issue will remain an open question for possible further negotiations since it is not hindered by any consensus, which can form a first useful acquired result among the agents. Note that if both formulas were inserted within Φ_i then Φ_i would become unsatisfiable and no consensus could exist, hence the use of modalities to prevent this from happening. More generally, we claim that in negotiation and consensus-finding situations, there can be a need to express and handle requirements from agents asserting that some formulas should be possible, or in a dual way, that some formulas should not be derivable. Indeed, an agent might not only express negative desires under the form of standard logic formulas $\neg\alpha$ but also weaker desires that simply require some α not to be derivable. Additionally, since the latter form of desire is logically weaker than the former one, weakening some desires into mere “possible” or “not derivable” forms can be a way to allow for consensus to exist when no consensus for the agents’ initial requirements exists.

Let us also give another motivating example for the use of the possibility modality \diamond . Assume that a consensus is to be found among two engineers who have conflicting diagnoses about a same device fault. The first one claims that the reason for failure is to be found in three (possibly cumulative) device faults. Let us tentatively represent this by $\Phi_1 = \{cause_1 \vee cause_2 \vee cause_3\}$. The second one is convinced that the reasons for failure are to be found in some of the first two causes, only. Assume that this is represented by $\Phi_2 = \{cause_1 \vee cause_2\}$. The only *max* consensus is $\Phi_1 \cup \Phi_2$ since $\Phi_1 \cup \Phi_2$ is satisfiable. From this consensus we can deduce $cause_1 \vee cause_2$. It can be argued that this result is counter-intuitive with respect to the intended role of a consensus since Φ_1 requires $cause_3$ to be also considered as a possible cause of failure whereas the same agent Φ_1 does not agree that the only two possible reasons for failure are $cause_1$ and $cause_2$. Actually, what the engineer Φ_1 needs to express is that $\alpha = cause_1 \vee cause_2 \vee cause_3$ is a prime implicate of her own knowledge: namely, no strict sub-clause of α is derivable. She must thus express that $\Phi_1 \not\vdash \alpha$ for any α that is a strict sub-clause of $cause_1 \vee cause_2 \vee cause_3$, or equivalently, that any corresponding formula $\neg\alpha$ must be *possible*. In this way, no consensus can allow one to conclude for example $cause_1 \vee cause_2$ without contradicting her own requirements. Notice that, like in the example about *taxes-increase* and \neg *taxes-increase*, the set of standard logic formulas $\neg\alpha$ such that α is a strict sub-clause of $cause_1 \vee cause_2 \vee cause_3$ is unsatisfiable. Hence, introducing this set within Φ_i would make Φ_i collapse and prevent any consensus from existing: this also justifies our use of modal logic.

4.2 Consensus in modal logic S5

Modal logic S5 is a canonical logic of possibility and necessity that can also be considered as a logic of knowledge. Its language \mathcal{L}_M extends the language of standard Boolean logic by means of two modality connectives: \diamond and \square . Main basic definitions and concepts about S5 are recalled in Appendix: for the understanding of this paper, it is sufficient to know that (1). \diamond and \square are used as additional unary connectives: for example $\diamond(\square a \vee b \vee \diamond\neg c)$ is a well-formed formula of \mathcal{L}_M . (2). modalities are dual in the sense that $\diamond\alpha = \neg\square\neg\alpha$. (3). Truth values of modal formulas can vary depending on the considered so-called possible world, these worlds being connected by an equivalence accessibility relation in S5; the satisfiability paradigm is adapted accordingly. Interestingly, the satisfiability problem in S5 is NP-complete [11], just like SAT.

We assume that $\forall i \in [1..n] : \Phi_i \subset \mathcal{L}_M$ is S5-satisfiable and we consider $[\Phi_1, \dots, \Phi_n]$ as the profile \mathcal{S} for which a consensus needs to be found. Now, all definitions from Section 3 directly apply in this modal framework, using the satisfiability paradigm of S5. In this last respect, let us just introduce the two basic cases involving formulas with modalities. We will indicate how to handle more complex modal formulas later in the paper by reduction to these formulas and non-modal ones. Assume that Φ_i contains the formula $\diamond\alpha$ where α does not contain any modalities: any consensus for \mathcal{S} cannot allow one to deduce $\neg\alpha$ since any consensus should not conflict with Φ_1 , which asserts that α is possible, i.e., is *true* in some possible world. When Φ_i contains $\square\alpha$ then any non-empty consensus must contain α since it cannot conflict with Φ_1 , which asserts that α is *true* in any possible world. Consequently, even when all Φ_i are satisfiable, the existence of a non-empty consensus is not guaranteed: when one agent asserts $\square\alpha$ and a second one can deduce $\neg\alpha$, no non-empty consensus can exist. In this respect, $\square\alpha$ can be interpreted as requiring α to be one integrity constraint for any non-empty consensus. Similarly, if one agent can deduce $\square\alpha$ whereas another one can deduce $\diamond\neg\alpha$ then no non-empty consensus exists.

Proposition 1. *Let $\mathcal{S} = [\Phi_1, \dots, \Phi_n]$ such that $\forall i \in [1..n] : \Phi_i \subset \mathcal{L}_M$ is satisfiable. There is no non-empty consensus for \mathcal{S} iff $\exists i, j \in [1..n]$ s.t. for some α in \mathcal{L}_M we have $(\Phi_i \models_{S5} \neg\alpha$ and $\Phi_j \models_{S5} \square\alpha)$ or $(\Phi_i \models_{S5} \diamond\neg\alpha$ and $\Phi_j \models_{S5} \square\alpha)$.*

Example 3. *Let us come back to Example 1 and assume now that agent Φ_3 strengthens her desires and does not want to leave open any possibility in the consensus of having an increase of defense spendings: Φ_3 is now $\{\square\neg ids\}$ (or equivalently $\{\neg\diamond ids\}$). There remains only one *max*_# consensus, namely $\Gamma_2 = \{\square\neg ids, \neg it \rightarrow \neg ids\}$. Note that $\square\neg ids$ entails $\neg ids$ in S5. Now, if any Φ_i is then augmented with $\diamond ids$ then no non-empty consensus exists anymore.*

5 COMPUTING ONE MAX_# CONSENSUS

Let us focus first on the computation of one *max*_# consensus in the standard Boolean framework. We assume that each Φ_i in \mathcal{S} is a satisfiable set of clauses of \mathcal{L} .

If one MSS_# of $\bigcup_{i=1}^n \Phi_i$ were to be extracted, we could directly use SAT-related techniques to address this issue. Specifically, the problem would amount to solving a variant of Max-SAT($\bigcup_{i=1}^n \Phi_i$) that would return one MSS_# of its argument (instead of yielding merely the cardinality of this MSS). Actually, in order to deliver one *max*_# consensus, the computation of one MSS_# must also take into account the additional constraint requiring the result to be satisfiable with each Φ_i taken individually. Notice that since the Φ_i can be

mutually conflicting, in the general case it is not possible to simply replace this multiple constraint by a unique one stating that the result should be satisfiable with $\bigcup_{i=1}^n \Phi_i$. A naive direct approach to compute one $max_{\#}$ consensus could however consist of the following steps. First, initialize Γ with $\bigcup_{i=1}^n \Phi_i$. Then, consider each Φ_j , successively. At each step, trim the current contents of Γ so that it becomes satisfiable with the current Φ_j . Clearly, at the end of the process, Γ would be satisfiable with each Φ_j and would be a consensus for \mathcal{S} . However, there would be no guarantee that Γ is a maximal consensus. Indeed, some clauses from Γ might have been dropped to ensure satisfiability with, say, Φ_j ; at some subsequent step, some other clauses could have been discarded from Γ to ensure satisfiability with another source whereas dropping only these other clauses would have been sufficient to ensure that Γ is satisfiable with Φ_j . Hence to ensure that Γ is a $max_{\#}$ consensus, we would need to consider every possible ordering of all Φ_i and for each of them, consider every Φ_j and record the corresponding various minimal subsets of clauses to be dropped in order to ensure satisfiability with Φ_j . Based on all this information, we might finally select the clauses to be expelled to give rise to one maximal consensus. Clearly, such an approach is doomed to face a combinatorial blow-up very often, especially since the number of possible orderings to consider is exponential.

We have followed another path in [5] and have adapted a method, called *Transformational Method*, introduced in [1] to circumvent a close combinatorial issue consisting in extracting one maximal subset Γ from a set of clauses such that, at the same time, Γ is satisfiable with several possibly mutually conflicting contexts. The method is based on the transformation of the initial problem into one single-step optimization problem. Intuitively, the satisfiability of Γ with one given Φ_i is interpreted as a sub-problem, using its own range of fresh Boolean variables. All sub-problems are then linked together with the use of additional variables, called linking variables. The use of a Partial Max-SAT solver allows then to extract one subset of clauses that is a solution to the initial problem. There has been very significant progress these last years about the design of experimentally efficient (Partial) Max-SAT solvers: see for example [15] and the related international competitions <http://www.maxsat.udl.cat/>. Interestingly, despite the increase of the problem size that is linear with respect to the number of contexts, this approach proves experimentally far more efficient and scalable than the above naive method [1]. In order to adapt this method to compute one $max_{\#}$ consensus, we use Partial Max-SAT, which belongs to the *Opt-P* class of problems [16].

Definition 4. Let Σ_1 and Σ_2 be two sets of clauses. *Partial Max-SAT*(Σ_1, Σ_2) computes one cardinality maximal subset of Σ_1 that is satisfiable with Σ_2 . Σ_1 and Σ_2 are called the sets of soft and hard constraints, respectively.

Algorithm 1 depicts the method. The problem of having Γ being a subset of $\bigcup_{i=1}^n \Phi_i$ that is satisfiable with each Φ_j is first treated as n independent subproblems; these subproblems will be then linked together to form one single optimization problem through one single call to Partial Max-SAT. Each clause δ_i^j from any Φ_i is augmented with an additional disjunct $\neg e_i^j$ using a new fresh variable (line 2): this yields a set Σ . These e_i^j variables will be used to *link* the various subproblems. Each subproblem is created by unioning Σ with one Φ_i and by renaming all variables except the e_i^j (l. 4-7). All together, the subproblems form the set of hard clauses; these ones are all simultaneously satisfiable (just assign all e_i^j to *true*). The set of soft clauses is made of all unit clauses e_i^j (l. 3). The instance of the Partial

Max-SAT problem with these sets of hard and soft clauses will search one truth-value assignment such that all hard clauses and one maximal number of clauses e_i^j are satisfied. Accordingly, all clauses δ_i^j corresponding to the satisfied e_i^j form one $max_{\#}$ consensus for \mathcal{S} . Notice that the use of additional variables and clauses is a paradigm that has long been exploited in Max-SAT computation and the extraction of minimal unsatisfiable sets of clauses by other authors, see for example [6] and [14].

Algorithm 1: Compute one $max_{\#}$ consensus for \mathcal{S}

input : $\mathcal{S} = [\Phi_1, \dots, \Phi_n]$: a profile of n satisfiable sets of Boolean clauses;
 Assume that the clauses of Φ_i are noted $\delta_i^1, \delta_i^2, \dots$;
output: One $max_{\#}$ consensus for \mathcal{S} ;

- 1 $\Gamma_{\text{Hard}} \leftarrow \emptyset$; $\Gamma_{\text{Soft}} \leftarrow \emptyset$;
- 2 $\Sigma \leftarrow \bigcup_{\Phi_i \in \mathcal{S}} \{-e_i^j \vee \delta_i^j \mid \delta_i^j \in \Phi_i \text{ and } e_i^j \text{ are new variables}\}$;
- 3 $\Gamma_{\text{Soft}} \leftarrow \{e_i^j\}_{i,j}$;
- 4 **foreach** $\Phi_i \in \mathcal{S}$ **do**
- 5 $\Phi_i \leftarrow \Sigma \cup \Phi_i$;
- 6 Rename all var. in Φ_i (except the e_i^j) with new ones;
- 7 $\Gamma_{\text{Hard}} \leftarrow \Gamma_{\text{Hard}} \cup \Phi_i$;
- 8 $\Sigma \leftarrow \text{Partial Max-SAT}(\Gamma_{\text{Soft}}, \Gamma_{\text{Hard}})$;
- 9 **return** ($\{\delta_i^j \in \mathcal{S} \mid e_i^j \in \Sigma\}$);

Proposition 2. Let m be the number of clauses in $\mathcal{S} = [\Phi_1, \dots, \Phi_n]$. The Transformational Approach computes one $max_{\#}$ consensus for \mathcal{S} . It requires one call to a Partial Max-SAT solver on a set of hard constraints made of mn clauses and a set of soft constraints made of m clauses.

6 COMPUTING ONE $MAX_{\#}$ CONSENSUS IN S5

Assume now that $\forall i \in [1..n]: \Phi_i \in \mathcal{L}_M$ and Φ_i is S5-satisfiable.

Consensus-finding grounds itself in multiple occurrences of satisfiability constraints since it amounts to finding a subset of formulas that is satisfiable with each source. Interestingly, as already mentioned, the satisfiability problem for S5 is NP-complete [11]: hence, there exists a polynomial transformation allowing this latter problem to be rewritten as SAT. Accordingly, there has been already much research about SAT-based deduction and satisfiability in S5 and other usual modal logics: see for example [8] and [9]. In the same vein, one way to compute consensus in S5 can thus amount to translating the profile \mathcal{S} in \mathcal{L}_M into a profile in \mathcal{L} , while preserving (un)satisfiability.

The challenge is to devise such a transformation so that the practical efficiency of the approach in Boolean logic to extract one $max_{\#}$ consensus remains experimentally efficient. Especially, the number of additional variables and clauses that need to be introduced by the translation process into standard Boolean logic must be as minimal as possible to address large instances since the consensus-finding method in the Boolean framework itself roughly multiplies the size of the instance by the number of sources. Interestingly, the subsequent optimization technique to extract consensus does not require the so-called *nominals* and other related concepts that are often introduced in the Boolean language by existing techniques, like for example [19], when modal S5 formulas are translated into Boolean ones. On the contrary, we use a plain direct translation, as proposed in [18], to transform the S5 modal formulas into equi-satisfiable CNF.

The first step consists in rewriting the formulas of \mathcal{S} into modal CNF that preserves (un)satisfiability. We used the axiom schemata of S5, as well as De Morgan laws, which are valid in S5, to simplify nested occurrences of connectives and modalities, and transform each Φ_i into its Normal Negation Form (NNF), which requires (1). \neg to only occur immediately before a Boolean variable and (2). the absence of occurrences the \rightarrow connectives. In a second step, the NNF was then transformed into a set of Boolean clauses using the Tseitin encoding technique [20], which introduces new variables to encode subformulas, and using other additional variables to encode the various possible worlds as follows. It is easy to see that it is sufficient to consider a number of different possible worlds that is bounded by the number of occurrences of modal operators in the NNF. Intuitively, a formula $\Diamond a$ is rewritten as $a_{pw1} \vee \dots \vee a_{pwk}$ where a_{pwi} are new fresh variables referring to a in world i , and k is the total number of possible worlds that need be considered.

Notice that this transformation procedure of an S5 formula into an equi-satisfiable CNF standard Boolean logic is linear in the number of occurrences of modal operators in the initial formula. The number of Boolean variables in the NNF Boolean form depends on the number of variables in the initial modal formula with a proportional factor that is the number of involved possible worlds.

7 EXPERIMENTAL STUDY

All the experimentations have been conducted on Intel Xeon E5-2643 (3.30GHz) processors with 8Gb RAM with Linux CentOS. We used the Weighted Partial Max-SAT solver MaxHS from <http://www.maxhs.org/> and have implemented all the tested algorithms in C++ on top of *Glucose* (<http://www.labri.fr/perso/lisimon/glucose/>). All software, data and results are available at <http://cril.univ-artois.fr/consensus>.

7.1 $max_{\#}$ consensus in the Boolean framework

The profiles \mathcal{S} in the standard Boolean case were based on the 291 (mostly real-world) unsatisfiable instances from the 2011 MUS competition <http://www.satcompetition.org/2011/> about the extraction of (set-inclusion) MUSes (Minimal Unsatisfiable Subsets). MUSes and MSSes are naturally related: each MUS is a minimal hitting set on the set of Co-MSSes (a Co-MSS is the set-theoretical complement of an MSS) whereas Co-MSS are hitting sets on the set of MUSes (see [12] for more on the use of this duality to compute MUSes). The selected instances are highly challenging: they are made of up to more than 15983000 clauses and 4426000 variables (457459 clauses using 139139 different variables on average). Each instance has been randomly divided into $n \in [3, 5, 7, 10]$ same-size (modulo n) sources Φ_i to yield all the \mathcal{S} . Time-out for each single $max_{\#}$ consensus extraction was set to 900 seconds.

Table 1 summarizes the average results for the extraction of one $max_{\#}$ consensus per value of n . It lists the number of successful extractions, the average time in seconds to extract one $max_{\#}$ consensus, the average numbers of clauses and variables in the transformed instance and, finally, the average number of clauses to drop to deliver the consensus. A drop of performance can be observed when n increases (from 235 successful extractions to 207): this is due to both the increase of size of the representation of the transformed instance and additional satisfiability tests when n increases. These results show the viability of the approach and its scalability. Let us stress again that these benchmarks were selected to test extreme computa-

tional limits of the approach. Hopefully, these benchmarks should be harder and bigger than most real-life consensus-finding applications.

	$n = 3$	$n = 5$	$n = 7$	$n = 10$
#solved	235	223	210	207
time (seconds)	96	109	119	150
#variables	303643	329599	380110	460194
#clauses	1325632	1855884	2386137	3181517
#clauses _{removed}	7	2	2	2

Table 1. Computing one $max_{\#}$ consensus in the standard Boolean framework.

7.2 $max_{\#}$ consensus in S5

For the S5 logic framework, we have considered all the unsatisfiable modal logic benchmarks from <http://www.ps.uni-saarland.de/theses/goetzmann/> that we were able to split into $n \in [2, 3, 5]$ S5-satisfiable Φ_i using the following procedure (there were 93 such successfully split benchmarks). For each of them, we have built the various modal logic profiles \mathcal{S} made of n sources as follows. \mathcal{S} is initialized with one source made of the initial benchmark, which is treated as a unique formula. We define the size of a modal formula as the number of edges within its usual NNF representation. While the number of sources in \mathcal{S} is less than n , the largest formula Σ in \mathcal{S} is replaced by two formulas obtained from Σ in the following way: let Σ' be a sub-formula of Σ such that the size of Σ' is as close as possible to half the size of Σ . Then, we replace Σ' in \mathcal{S} by a fresh variable $s_{\Sigma'}$ and we insert Σ' within \mathcal{S} as an additional source, together with the information that $s_{\Sigma'} \equiv \Sigma'$. In the whole process, we make sure that all Φ_i are S5-satisfiable. Then, each resulting modal profile was translated into a standard Boolean logic one according to the aforementioned transformation technique. Finally, one $max_{\#}$ consensus was then searched using our transformational Partial Max-SAT-based technique.

One $max_{\#}$ consensus was delivered for all the 93 benchmarks when $n = 2$. One $max_{\#}$ consensus was found for 88 and 74 benchmarks when $n = 3$ and $n = 5$, respectively. We explain the drop of performance when n increases by the additional clauses that are needed to express the $s_{\Sigma'} \equiv \Sigma'$ constraints in NNF format, and by the fact that the Partial Max-SAT step increases the size of the representation by a factor n . Figure 1 illustrates the number of successful extractions of $max_{\#}$ consensus according to the CPU time spent (in seconds) to compute such a consensus for these instances, and according to n .

Table 2 summarizes the average parameters values of the benchmarks and of their transformations after the different steps, as well as the average time spent in the process. The first column gives n . The four next ones list the main parameters of the initial modal formulas and of their corresponding modal profiles \mathcal{S} : namely, $\#vars$ is the average number of variables in the initial instance and $\#\{\Diamond, \Box\}$ is the average number of occurrences of modal operators; $\text{sum}(|\Phi_i|)$ is the average total size of the n Φ_i (size is the number of edges in the NNF representation) and, for convenience (since this can be computed from the previous columns) $\text{avg}(|\Psi_i|)$ gives the average size of each Φ_i . The next columns list the average values about the CNF transformation: namely, $\#vars$ gives the average number of Boolean variables used to encode the S5 formula into one CNF; $\text{sum}(\#\text{cls}_i)$ is the average total number of clauses in the standard logic profile whereas $\text{avg}(\#\text{cls}_i)$ is the average number of clauses in each

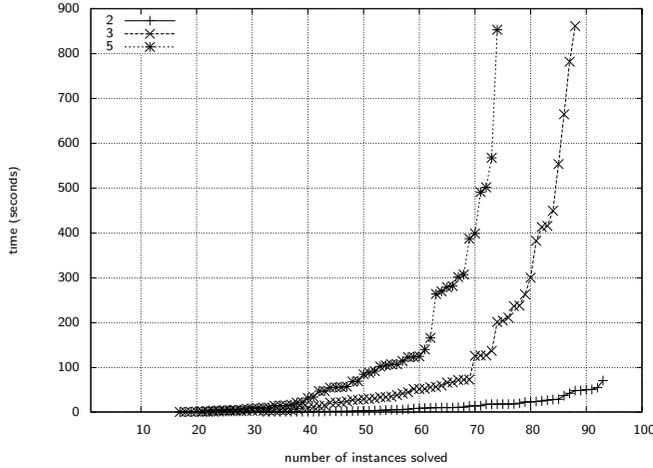


Figure 1. Number of modal profiles \mathcal{S} for which one $\max\#$ consensus was extracted, depending on the CPU time spent and n .

Φ_i . Then, Time-Tr. gives the average CPU time in seconds spent to achieve the CNF encoding. The next group of columns provide average values about the Partial Max-SAT step: #vars and #cls are the average numbers of Boolean variables and clauses used in the encoding of the optimization problem. Finally, Time-Opt. is the average time in seconds to extract a consensus from the standard Boolean representation whereas Time-Opt. + Time-Tr. gives the average time required to compute the consensus plus the time spent for the translation process, when the whole operation was successful. Hence the values in this column slightly differ from the sum of values from the corresponding columns Time-Opt. and Time-Tr. when $n = 3$ and $n = 5$.

It is important to stress that the increase of size of the modal representation with n is simply due to the additional clauses that are needed to express equivalences during the splitting process to form the modal profiles \mathcal{S} . Accordingly, the actual “size” that is relevant is not the “size” of the modal benchmarks but the number of total clauses in the modal \mathcal{S} , namely $\text{sum}(\Phi_i)$ in Table 2. For the same reason, the decrease of performance with the increase of n should not be interpreted as being only an increase of difficulty due to a larger number of sources, as the Partial Max-SAT step increases the size of the CNF representation by a factor n . Actually, this is also in part the expected consequence of having to consider larger modal profiles in these specific experimentations. The number of occurrences of modal operators in the benchmarks also plays a crucial role: it directly influences the size of the CNF representation. Actually, this parameter appears to be the most limiting one, together with n . Accordingly, we believe that these experimentations show the viability of the approach for real-life applications, whose size in terms of formulas, numbers of sources and of occurrences of modal operators remain in the range of the ones in the tested benchmarks.

8 CONCLUSION AND PERSPECTIVES

Consensus-finding is a ubiquitous issue in real-life and in many A.I.-related applications. In this paper, a concept of consensus has been investigated, with a focus on practical computational issues. This concept of consensus does not merely capture what is shared by sev-

eral agents: it provides a satisfiable fragment of all the information conveyed by the agents that is satisfiable with every agent. Hence, such a consensus might be endorsed by the agents as it does not contradict them. The paper focused on the most appealing consensus, namely maximal ones. They have been studied in both standard Boolean logic and in modal logic S5, as the necessity and possibility representation paradigms are highly relevant in consensus-related domains, like negotiation. Interestingly, we have proposed and experimented a single-step optimization technique that delivers one maximal consensus in an efficient way very often, for both logical frameworks.

At this point, this study opens paths for various promising further research. Let us here simply mention three of them. First, a natural extension of this study would consist in allowing various maximality preferences in consensus. For example, an agent might pre-order her desires according to her priorities and any consensus should attempt to obey these preferences. Interestingly, the use of *weighted* Partial Max-SAT vs. Partial Max-SAT could allow for a direct handling of these kinds of preferences, not only in the standard Boolean framework as this has been done in [5], but also in the S5 logic. Another natural challenging issue would be the extension of the concept of consensus for a multiple-agents modal S5. However, extending our transformational approach accordingly does not seem viable since satisfiability in such a logic is P-SPACE complete, making a polynomial translation into SAT out of reach. Finally, a promising challenge would be to build more elaborate transformation procedures for the modal logic framework, so that the structure of the initial formulas is not lost in the CNF representation and is fully exploited in the checks for satisfiability during the subsequent operations in the search for a maximal consensus.

ACKNOWLEDGEMENTS

We would like to thank the referees for their useful comments, which helped improve the presentation of this paper. All experimentations have been conducted on a cluster that has been funded in part by the *Conseil Régional du Nord/Pas-de-Calais* and an EC FEDER grant.

APPENDIX

The language \mathcal{L}_M of modal logic S5 extends the language \mathcal{L} of Boolean logic by allowing two modalities \diamond and \square to be used as additional unary connectives and that are dual in the sense that $\square \stackrel{def}{=} \neg\diamond\neg$. An axiomatic system for S5 is given by the following axiom schemata and rules:

All Boolean logic tautologies. $\mathcal{K} : \square(A \rightarrow B) \rightarrow (\square A \rightarrow \square B)$

$\mathcal{T} : \square A \rightarrow A$ $\mathcal{B} : A \rightarrow \square\diamond A$ $\mathcal{4} : \square A \rightarrow \square\square A$

(or $\mathcal{5} : \diamond A \rightarrow \square\diamond A$ instead of \mathcal{B} and $\mathcal{4}$)

$MP : \frac{A \quad (A \rightarrow B)}{B}$ $Nec : \frac{A}{\square A}$

The *possible worlds* semantics of S5 is based on Kripke frames, which are pairs (W, R) where W is a non-empty set of possible worlds and R is an accessibility relation between worlds. In S5, the relation R is reflexive, transitive and symmetric, i.e. is an equivalence relation. We note wRw' to express that w' is accessible from w . A valuation V assigns a subset of W to each atomic proposition p : namely, the worlds where p is *true*. Given a Kripke frame (W, R) , a valuation V and a world w , the satisfaction relation \models is defined inductively as follows.

$[(W, R), V, w] \models p$ iff $w \in V(p)$

n	Initial modal formulas and modal profiles				CNF transformation step				Partial Max-SAT step			Total (secs.)
	#vars	# $\{\diamond, \square\}$	sum($ \Phi_i $)	avg($ \Phi_i $)	#vars	sum($ \#cls_i $)	avg($ \#cls_i $)	Time-Tr.	#vars	#cls	Time-Opt.	Time-Tr. + Time-Opt.
2	80	85	464	232	22022	279983	139991	1.28	324029	1119933	10.30	11.58
3	81	101	569	189	68695	632146	210715	2.07	838233	3160731	90.64	92.49
5	83	106	640	128	114239	924416	184883	2.70	1495611	6470913	96.08	97.90

Table 2. Computing one $max_{\#}$ consensus in the S5 framework.

$[(W, R), V, w] \models \neg p$ iff $w \notin V(p)$
 $[(W, R), V, w] \models A \wedge B$ iff $[(W, R), V, w] \models A$ and $[(W, R), V, w] \models B$
 $[(W, R), V, w] \models A \vee B$ iff $[(W, R), V, w] \models A$ or $[(W, R), V, w] \models B$
 $[(W, R), V, w] \models \diamond A$ iff $\exists w' \in W$ s.t. wRw' : $[(W, R), V, w'] \models A$
 $[(W, R), V, w] \models \square A$ iff $\forall w' \in W$ s.t. wRw' : $[(W, R), V, w'] \models A$

The S5 satisfiability problem is: given a formula α of \mathcal{L}_M , determine whether there exists a Kripke frame (W, R) , a valuation V and a world $w \in W$ s.t. $[(W, R), V, w] \models \alpha$.

The deduction relation in S5 is noted \models_{S5} .

REFERENCES

- [1] Philippe Besnard, Éric Grégoire, and Jean-Marie Lagniez, ‘On computing maximal subsets of clauses that must be satisfiable with possibly mutually-contradictory assumptive contexts’, in *Proceedings of the Twenty-Ninth National Conference on Artificial Intelligence AAAI '15*, pp. 3710–3716, (2015).
- [2] Patrick Blackburn, M. de Rijke, and Y. Venema, *Modal Logic*, Cambridge University Press, 2001.
- [3] Brian F. Chellas, *Modal Logic*, Cambridge University Press, 1980.
- [4] Eithan Ephrati and Jeffrey S. Rosenschein, ‘Deriving consensus in multi-agent systems’, *Artificial Intelligence*, **87**(1-2), 21–74, (1996).
- [5] Éric Grégoire, Sébastien Konieczny, and Jean-Marie Lagniez, ‘On consensus extraction’, in *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI'16)*, (2016).
- [6] Zhaohui Fu and Sharad Malik, ‘On solving the partial MAX-SAT problem’, in *Proceedings of the 9th International Conference on Theory and Applications of Satisfiability Testing (SAT 2006)*, pp. 252–265, (2006).
- [7] Olivier Gauwin, Sébastien Konieczny, and Pierre Marquis, ‘Conciliation through iterated belief merging’, *Journal of Logic and Computation*, **17**(5), 909–937, (2007).
- [8] E. Giunchiglia, F. Giunchiglia, R. Sebastiani, and A. Tacchella, ‘SAT vs. translation based decision procedures for modal logics: a comparative evaluation’, *Journal of Applied Non-Classical Logics*, **10**(2), (2000).
- [9] E. Giunchiglia, A. Tacchella, and F. Giunchiglia, ‘SAT-based decision procedures for classical modal logics’, *Journal of Automated Reasoning*, **28**(2), 143–171, (2000).
- [10] Audun Jøsang, ‘The consensus operator for combining beliefs’, *Artificial Intelligence*, **141**(1/2), 157–170, (2002).
- [11] R.E. Ladner, ‘The computational complexity of provability in systems of modal propositional logic’, *SIAM Journal of Computing*, **6**(3), 467–480, (1977).
- [12] Mark H. Liffiton and Kareem A. Sakallah, ‘On finding all minimally unsatisfiable subformulas’, in *Proceedings of the 8th International Conference on Theory and Applications of Satisfiability Testing (SAT'05)*, volume 3569 of *Lecture Notes in Computer Science*, pp. 173–186. Springer, (2005).
- [13] Joao Marques-Silva and Mikolás Janota, ‘On the query complexity of selecting few minimal sets’, *Electronic Colloquium on Computational Complexity (ECCC)*, **21**, 31, (2014).
- [14] Alexander Nadel, ‘Boosting minimal unsatisfiable core extraction’, in *Proceedings of 10th International Conference on Formal Methods in Computer-Aided Design (FMCAD 2010)*, pp. 221–229, (2010).
- [15] Nina Narodytska and Fahiem Bacchus, ‘Maximum satisfiability using core-guided maxsat resolution’, in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI'14)*, pp. 2717–2723, (2014).
- [16] Christos H. Papadimitriou and Mihalis Yannakakis, ‘Optimization, approximation, and complexity classes’, *Journal of Computer and System Sciences*, **43**(3), 425 – 440, (1991).
- [17] Wei Ren, R.W. Beard, and E.M. Atkins, ‘A survey of consensus problems in multi-agent coordination’, in *Proceedings of the 2005 American Control Conference*, volume 3, pp. 1859–1864, (2005).
- [18] Yakoub Salhi, Saïd Jabbour, and Lakhdar Sais, *Information Search, Integration and Personalization: International Workshop, ISIP 2012, Sapporo, Japan, October 11-13, 2012. Revised Selected Papers*, chapter Graded Modal Logic GS5 and Itemset Support Satisfiability, 131–140, Springer, 2013.
- [19] Yakoub Salhi and Michael Sioutis, ‘A resolution method for modal logic S5’, in *Proceedings of the Global Conference on Artificial Intelligence (CCAI 2015)*, volume 36 of *EPiC Series in Computer Science*, pp. 152–262, (2015).
- [20] G.S. Tseitin, *Structures in Constructives Mathematics and Mathematical Logic, Part II*, chapter On the Complexity of Derivations in the Propositional Calculus, 115–125, Steklov Mathematical Institute, 1968.

Complexity and Tractability Islands for Combinatorial Auctions on Discrete Intervals with Gaps

Janosch Döcker¹ and Britta Dorn² and Ulle Endriss³ and Dominikus Krüger⁴

Abstract. Combinatorial auctions are mechanisms for allocating bundles of goods to agents who each have preferences over these goods. Finding an economically efficient allocation, the so-called winner determination problem, is computationally intractable in the general case, which is why it is important to identify special cases that are tractable but also sufficiently expressive for applications. We introduce a family of auction problems in which the goods on auction can be rearranged into a sequence, and each bid submitted concerns a bundle of goods corresponding to an interval on this sequence, possibly with multiple gaps of bounded length. We investigate the computational complexity of the winner determination problem for such auctions and explore the frontier between tractability and intractability in detail, identifying tractable, intractable, and fixed-parameter tractable cases.

1 INTRODUCTION

Combinatorial auctions [8] are mechanisms to allocate goods in which bidders are permitted to place bids on bundles of goods. They are widely used in practice, e.g., to auction off radio spectrum licences or the rights to serve different bus routes. The design of combinatorial auctions poses many challenges that are relevant to Artificial Intelligence (AI). This includes algorithm design based on AI techniques such as heuristic-guided search [e.g., 26], the design of expressive bidding languages building on insights from knowledge representation [e.g., 6], and the analysis of the strategic behaviour of agents participating in an auction [e.g., 28].

Due to the combinatorial structure of the bids, the *winner determination problem* (WDP), i.e., the problem of computing an allocation that maximises the revenue for the auctioneer is \mathcal{NP} -hard in the general case [25, 18] (e.g. Rothkopf et al. [25] observed that the problem is equivalent to a weighted set packing problem). It therefore is important to identify special cases, so-called “tractability islands”, that permit efficient solutions but that are also sufficiently expressive for applications of interest. This approach was pioneered by Rothkopf et al. [25] who identified several structural restrictions on the range of permitted bids that render the WDP polynomial, and it was further refined by, amongst others, Conitzer et al. [7] and Gottlob and Greco [14].

In this paper we introduce a family of auction problems located at the frontier between tractability and intractability. Consider the example on the lefthand side of Figure 1, where six bidders each submit a bid for several connected cells on a construction ground put up for

auction. If we rearrange the cells as shown on the righthand side of Figure 1, to obtain a sequence of cells (i.e., of goods on auction), we find that some of the bids (e.g., *A* and *B*) end up as intervals. By a result of Rothkopf et al. [25], the WDP is polynomial in case there is such a mapping under which all bids end up as intervals. In our example, however, some of these intervals have (small) gaps. A gap consists of consecutive positive integers missing within a bid, e.g., the bid *C* on the left-hand side of Figure 1 is mapped to an interval with a gap of size 2. We study the family of combinatorial auction problems that can be mapped into a linear structure such that all bids correspond to discrete intervals with a number of gaps of bounded length. Note that the example shown in Figure 1 requires one of the dimensions of the ground to be constant. This happens, for instance, in auctions for a swath of offshore waters (see Rothkopf et al. [25]).

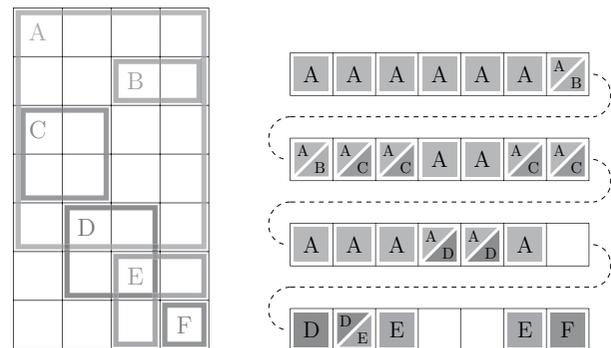


Figure 1. Bids *A–F* on cells of a construction ground (left), and the same auction problem mapped to a linear structure (right).

Depending on what restrictions we impose exactly on these gaps, we obtain either tractability,⁵ intractability, or fixed-parameter tractability results. For the most part, we assume that the goods are represented by distinct positive integers $1, \dots, n$, such that the bids have the structure we are presenting results for. An exception is Theorem 10, where we do not assume such an ordering to be given.

In Section 3 we show that the WDP can be solved in polynomial time if all bids correspond to intervals with multiple gaps of length at most ℓ each, for some fixed integer ℓ (as we will see, the case of $\ell = 2$ is of particular interest). This result thus significantly extends the original result of Rothkopf et al. [25]. It has immediate applications by identifying a large family of auction problems that can be solved efficiently in practice.

¹ University of Tübingen, Germany, email: janosch.doecker@uni-tuebingen.de

² University of Tübingen, Germany, email: britta.dorn@uni-tuebingen.de

³ University of Amsterdam, The Netherlands, email: ulle.endriss@uva.nl

⁴ Ulm University, Germany, email: dominikus.krueger@uni-ulm.de

⁵ By tractable we mean that the problem can be solved in polynomial time with respect to the size of the input.

In Section 4 we identify several cases for which the decision variant of the WDP is \mathcal{NP} -complete. These are negative results, but they nevertheless are important in that they clarify the transition between tractable and intractable cases. All of our results of this kind deal with auction problems where bids concern small combinations of goods that are very close to the case of intervals with gaps covered by our positive tractability result mentioned earlier:

- Every bid is of the form $\{i, i + 1\} \cup \{j, j + 1\}$ with $j - i = \lfloor \sqrt{n} \rfloor$ (where n is the number of goods). Thus, every bid concerns two intervals of length 2 each, and the results applies even when we fix the distance between these two intervals to always be $\lfloor \sqrt{n} \rfloor$.
- Every bid is for an interval of length 2 together with an arbitrary third good, i.e., it is of the form $\{i, i + 1, j\}$.
- Every bid is either for an interval of length 3, i.e., it is of the form $\{i, i + 1, i + 2\}$, or for a set of 2 arbitrary goods, i.e., it is of the form $\{i, j\}$. Thus, a single auction instance may include bids of both of these types.

We thus refine known results showing that the WDP is \mathcal{NP} -hard if bidders may bundle two intervals together [7] and if bids on three arbitrary goods are permitted [25]. As both 3-intervals and 2-sets *alone* have a polynomial WDP [25], our third intractability result also shows that tractable instances are not closed under taking unions.

Finally, in Section 5 we provide new insights on how parts of the input influence the computational hardness by showing that the WDP is fixed-parameter tractable with respect to the following parameters:

- the maximum length s of the section in each interval within which any subset of elements may be missing;
- the combined parameter (g, k) consisting of the maximum number g of goods in a bid and the minimum number k of bids that have to be deleted such that the remaining problem can be represented using intervals.

Thus, for auction instances where the above parameters are small constants, the WDP can again be solved efficiently. These results complement previous work on the parameterized complexity of combinatorial auctions [19]. Amongst the results of Loker and Larson [19] are that the WDP is $W[1]$ -complete with respect to the revenue and is in \mathcal{FPT} with respect to the number of distinct atomic bids. Moreover, they present complexity results for different restrictions of the so-called *bid graph*.

The remainder of this paper thus is organised as follows. Section 2, besides covering relevant background material on combinatorial auctions and parameterized complexity, formally introduces our notation and terminology. Our tractability results are presented in Section 3, our intractability results in Section 4, and our fixed-parameter tractability results in Section 5. Section 6 concludes.

2 PRELIMINARIES

In this section, we introduce relevant notation and terminology for combinatorial auctions as well as a number of specific structures we will use to describe auction instances of special interest. We also recall basic concepts from the theory of parameterized complexity.

2.1 Combinatorial auctions, WDP

By $\mathbb{N} := \{0, 1, 2, \dots\}$, we denote the set of non-negative integers. Our notation is similar to that of Rothkopf et al. [25]. A *combinatorial auction* can be described as a triple $\mathcal{C} = (A, P, b)$, where

$A := \{1, \dots, n\}$ is the set of *goods* (or *assets*, or *items*) to be sold by auction (subsets of which are called *combinations*), P is the set of *permitted combinations* $C \subseteq A$ on which bids may be placed, and $b: P \rightarrow \mathbb{N}$ is the mapping representing the bids (which can be thought of as a list of pairs, each consisting of a combination and a price). The number $b(C)$ is the largest bid submitted for C , and $b(C) = 0$ if no bid is submitted for C . We will not model the bidders, but the set of submitted bids only. This can be thought of as each bidder being single-minded and submitting one ‘atomic bid’ (one desired combination), or as fewer bidders submitting a union (an OR-expression) of such atomic bids each. An *outcome* W of an auction is a set of pairwise disjoint combinations of P . The *revenue* $rev(W)$ of an outcome W is defined as $rev(W) := \sum_{C \in W} b(C)$.

The WINNER DETERMINATION problem (WDP) asks, given a combinatorial auction $\mathcal{C} = (A, P, b)$, for an outcome maximising the revenue. Its decision version asks, given a combinatorial auction $\mathcal{C} = (A, P, b)$ and a positive integer k , whether there is an outcome W such that $rev(W) \geq k$.

2.2 Discrete intervals, longest paths, item graphs

For $i, j \in A$, let $[i, j] := \{x \in A \mid i \leq x \leq j\}$ denote a discrete interval, i.e., $A = [1, n]$. Thus, $[i, j] = \emptyset$ if $i > j$.

We will use the problem LONGEST PATH for directed acyclic graphs (DAGs) to derive some of our tractability results. Let $G = (V, E)$ be a directed graph with edge weights given by $g: E \rightarrow \mathbb{N}$, and let $\pi = (v_1, v_2, \dots, v_{|\pi|})$ be a directed path in G , where $|\pi|$ is the number of vertices on the path. We define the *length* of π as the sum of its edge weights $\sum_{l=1}^{|\pi|-1} g((v_l, v_{l+1}))$. The problem LONGEST PATH then asks, given a DAG $G = (V, E)$ with edge weight function g and two vertices $v_i, v_f \in V$, for a directed path $\pi = (v_i = v_1, v_2, \dots, v_f = v_{|\pi|})$ of maximum length. This problem can be solved in linear time $O(|V| + |E|)$ [27, p. 661].

Given a combinatorial auction $\mathcal{C} = (A, P, b)$, a graph with vertex set A is called an *item graph* if the bids induce connected subgraphs [7]. A *structured* item graph has bounded treewidth.

2.3 Parameterized complexity

The computational complexity of a problem is usually studied with respect to the size of the input I of the problem. Parameterized complexity theory [10, 12, 22] additionally takes into account the size of a so-called parameter which is a certain part of the input, e.g., the size of the solution set or the maximum number of goods in a combination in a combinatorial auction. A problem is called *fixed-parameter tractable* (is in the class \mathcal{FPT}) with respect to a parameter k if it can be solved in time $f(k)|I|^{O(1)}$, where f is a computable function and $|I|$ is the length of the encoding of I . This means that the running time of the corresponding algorithm is polynomial in the size of the input, but may be exponential or worse in terms of the parameter k ; hence, for small values of k , the problem might be solvable efficiently. We will extend this definition to two-dimensional parameter spaces [23], considering a *combined* parameter of the form (k_1, k_2) .

3 TRACTABLE CASES

Rothkopf et al. [25] show that WINNER DETERMINATION is solvable in quadratic time if all permitted combinations are discrete intervals. In this section, we extend their result to the case of discrete intervals with multiple gaps of length at most ℓ each, for some fixed integer ℓ .

The case of intervals with up to one missing element can be solved in time $\mathcal{O}(n^3)$ by adapting the dynamic programming algorithm from Rothkopf et al. [25]. We first consider intervals with gaps of combined length at most 2. More precisely, we show that the WDP is solvable in polynomial time for combinations

$$C_{i,j}^M := [i, j] \setminus M, 1 \leq i \leq j \leq n,$$

with $\emptyset \subseteq M \subset [i, j]$, $M \cap \{i, j\} = \emptyset$, and $0 \leq |M| \leq 2$.

Theorem 1. *Let $\mathcal{C} = (A, P, b)$ be a combinatorial auction with $P := \{C_{i,j}^M : 1 \leq i \leq j \leq n \wedge M \subseteq [i+1, j-1] \wedge |M| \leq 2\}$, where $C_{i,j}^M := [i, j] \setminus M$. Then an optimal outcome for \mathcal{C} can be found in time $\mathcal{O}(n|P|) = \mathcal{O}(n^5)$, where $|P|$ is the cardinality of P .*

Proof. We construct a DAG so that all possible outcomes of the auction are represented as an edge-weighted path, where the first and the last vertex are fixed. The bids themselves will be represented as edges with positive weights.

Let us first look at how an outcome W is represented as a path π_W . Let $W := \{C_{i_1, j_1}^{M_1}, C_{i_2, j_2}^{M_2}, \dots, C_{i_{|W|}, j_{|W|}}^{M_{|W|}}\}$ be any outcome and w.l.o.g. we assume $i_1 < i_2 < \dots < i_{|W|}$. The bids on combinations of W will be represented in the same sorted order on π_W . A vertex on π_W represents which items of the auction are assigned at this stage of the path. We will only consider paths starting in the vertex where no items are assigned and ending in the vertex where all items are assigned. Items are assigned when an edge is used to reach the next vertex of a path. We will use two kinds of edges: Edges with weight 0 assign items to the auctioneer, i.e., these items are not sold, and edges with positive weight assign a combination to the corresponding bidder—the vertex reached via the edge reflects this change. The sum of the edge weights of π_W will then correspond to the revenue of W .

The following observation on “overlapping” combinations allows us to limit the number of required vertices in the graph. We say two combinations $C_{i,j}^M, C_{i',j'}^{M'} \in P$ overlap if they are disjoint and $[i, j] \cap [i', j'] \neq \emptyset$.

Observation 2. *The union of two overlapping combinations yields again an interval with at most two missing elements.*

This is obvious if one combination is contained in the missing elements of the other. Otherwise we consider two overlapping combinations $C_{i,j}^M, C_{i',j'}^{M'} \in P$ with $i < i'$ (the case $i > i'$ is analogous). Since the two combinations are disjoint, we have $i' \in M$ and $j \in M'$. Thus, the missing elements of the union are a subset of $M \setminus \{i'\} \cup M' \setminus \{j\}$ with cardinality ≤ 2 .

The set of vertices of the graph is

$$V := \{v_i^M : 1 \leq i \leq n+1 \wedge M \subseteq [2, i-2] \wedge |M| \leq 2\}.$$

In the vertex v_i^M , all items of $[1, i-1] \setminus M$ are assigned. We write

$$\text{asgd}(v_i^M) := [1, i-1] \setminus M$$

for the set of assigned items in a vertex $v_i^M \in V$. The first vertex of every path we consider is v_1^\emptyset , where the set of assigned items is empty and the last vertex is v_{n+1}^\emptyset , where every item is assigned. The reason for condition $M \subseteq [2, i-2]$ in the definition of V is that every set of assigned items is an interval with at most two missing inner elements. Let

$$\mathfrak{C} := \{C \in P : b(C) > 0\}$$

be the set of combinations with non-zero bids. We represent the bids with weighted edges. For every combination $C_{i,j}^M \in \mathfrak{C}$ we add a directed edge from v_i^\emptyset to v_{j+1}^M with weight $b(C_{i,j}^M)$:

$$E_1 := \{(v_i^\emptyset, v_{j+1}^M) : C_{i,j}^M \in \mathfrak{C}\}.$$

Furthermore, for every combination $C_{i',j'}^{M'} \in \mathfrak{C}$, we add an edge from v_i^M to $v_{i'+1}^{M'}$ with weight $b(C_{i',j'}^{M'})$, if $i' \in M$ and

$$\text{asgd}(v_i^M) \dot{\cup} C_{i',j'}^{M'} = \text{asgd}(v_{i'+1}^{M'}) \quad (1)$$

holds, where $\dot{\cup}$ denotes the disjoint union of two sets. The weight of this edge is uniquely defined, since the union is disjoint. In order to extend the outcome with a combination $C_{i',j'}^{M'}$, it is necessary that the union in Equation (1) is disjoint, since every item can be sold only once. From Observation 2, we know that the union of two overlapping combinations is again an interval with at most 2 missing elements. So we add the following edges:

$$E_2 := \left\{ (v_i^M, v_{i'+1}^{M'}) : M \neq \emptyset \wedge \left(\exists C_{i',j'}^{M'} \in \mathfrak{C} : i' \in M \wedge \text{asgd}(v_i^M) \dot{\cup} C_{i',j'}^{M'} = \text{asgd}(v_{i'+1}^{M'}) \right) \right\}.$$

The edges in the graph with weight greater than 0 are

$$E_b = E_1 \dot{\cup} E_2.$$

There can be edges with weight 0, since not all items have to be sold. We set

$$E_0 := \left(\{(v_i^\emptyset, v_{i+1}^\emptyset) : 1 \leq i \leq n\} \cup \{(v_i^M, v_i^\emptyset) : 4 \leq i \leq n+1 \wedge M \neq \emptyset\} \right) \setminus E_b.$$

The reason for condition $i \geq 4$ is that for smaller i the set of assigned items cannot have missing inner elements.

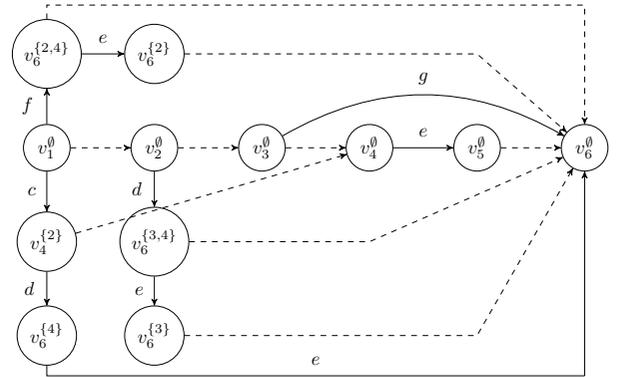


Figure 2. Winner determination for intervals with gap ≤ 2 : The path with maximal weight from v_1^0 to v_6^0 yields the winners of the auction with the bids $b(C_{1,3}^{\{2\}}) = c$, $b(C_{2,5}^{\{3,4\}}) = d$, $b(C_{4,4}^\emptyset) = e$, $b(C_{1,5}^{\{2,4\}}) = f$ and $b(C_{3,5}^\emptyset) = g$. The dashed edges have weight 0. The vertices not reachable from v_1^0 are omitted.

It is not hard to see that a path π with maximal weight from v_1^0 to v_{n+1}^\emptyset in the graph $G := (V, E_b \cup E_0)$ corresponds to an optimal outcome W of the auction (see Figure 2 for an example) that can be retrieved by calculating $\mathcal{O}(n)$ differences:

$$W := \{D_p : e_p \text{ is an edge of the path } \pi \text{ with weight } > 0\},$$

where $D_p := \text{asgd}(v_i^{M'}) \setminus \text{asgd}(v_i^M)$ for the edge $e_p = (v_i^M, v_i^{M'})$. A detailed analysis yields $|P| \in \mathcal{O}(n^4)$, $|V| \in \mathcal{O}(n^3)$ and $|E_b \cup E_0| \in \mathcal{O}(n^5)$. The graph can be constructed in time $\mathcal{O}(n^5)$ and it is directed and acyclic—note that for every edge $(v_i^M, v_i^{M'})$ we have $\text{asgd}(v_i^M) \subsetneq \text{asgd}(v_i^{M'})$. Therefore, by finding the longest path in G , we can solve the WDP in $\mathcal{O}(n^5)$. This concludes the proof of Theorem 1. \square

For the following tractability result, we make use of structured item graphs. We recall the corresponding definitions: For a given combinatorial auction $\mathcal{C} = (A, P, b)$, a graph with vertex set A is called an item graph if the bids induce connected subgraphs. A structured item graph has bounded treewidth. Conitzer et al. [7] show that the WDP is tractable if a structured item graph is given, i.e., they solve the WDP for this case in $\mathcal{O}(|T|^2(|B| + 1)^{tw+1})$ using dynamic programming, where T is a tree decomposition with width tw of an item graph for the instance of the WDP and $|B|$ is the number of bids. Note that, however, it is \mathcal{NP} -complete to check whether a combinatorial auction has a structured item graph of treewidth k , even for $k = 3$ [14]. Thus, identifying natural classes of auction instances for which we actually know the treewidth of the corresponding structured item graph is important and of immediate practical interest. Next, we identify a large class of combinatorial auctions that have structured item graphs. Since we explicitly give an item graph with bounded treewidth (see Figure 3 for an example), we obtain that the WDP can be solved efficiently for this class. The permitted combinations are discrete intervals with gaps consisting of at most ℓ elements each, with ℓ being a fixed positive integer. These gaps have to be separated by at least one element that is contained in the combination.

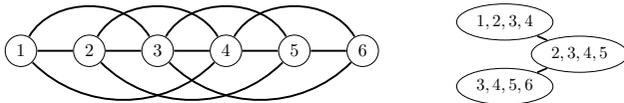


Figure 3. Item graph for bids on intervals with gaps of size at most 2 (left). The graph is chordal and the maximum clique size is 4, so the treewidth is 3 (see tree decomposition on the right). For criteria on upper and lower bounds for the treewidth of a graph see Bodlaender and Koster [4, 5].

Theorem 3. Let $\ell \in \mathbb{N}$, $\mathcal{C} = (A, P, b)$ be a combinatorial auction with $P := \{C \subseteq A : \forall k \in [\min(C), \max(C) - \ell] : C \cap [k, k + \ell] \neq \emptyset\}$. Given m bids, an optimal outcome can be found in time $\mathcal{O}(n^2(m + 1)^{\ell+2})$.

Proof. First, we construct an item graph for this class of permitted combinations. Consider the graph $G := (A, E)$ with

$$E := \{\{a, b\} \subseteq A : 1 \leq |a - b| \leq \ell + 1\}.$$

We show by contrapositive that G is an item graph. Let $C \subset A$ be an arbitrary set of items not inducing a connected component of G . Then there are two items $c, d \in C$ with $c < d, |c - d| \geq \ell + 2$ and $f \notin C$ for all $c < f < d$ (otherwise there is a path connecting the items of C). This implies

$$C \cap [c + 1, c + 1 + \ell] = \emptyset.$$

Hence, C is not a permitted combination.

In order to show that G is a structured item graph, we construct a tree decomposition of G with bounded width. For the underlying structure we choose the path $(X_1, X_2, \dots, X_{n-\ell-1})$, where the bags are defined as

$$X_i := \{j \in A : i \leq j \leq i + \ell + 1\}.$$

It is easy to see that the conditions of a tree decomposition are satisfied. Every bag contains exactly $\ell + 2$ elements, the width of the decomposition is therefore $\ell + 1$. For any bag X_i , the induced subgraph $G[X_i]$ is a clique, thus $\ell + 1$ is also a lower bound for the treewidth. Consequently, the treewidth of G is $\ell + 1$.

For combinatorial auctions with structured item graphs, we can solve the WDP with the algorithm of Conitzer et al. [7, p.214]. The running time of this algorithm for the structured item graph G with treewidth $\ell + 1$ and the tree decomposition we constructed is $\mathcal{O}(n^2(m + 1)^{\ell+2})$. \square

Note that Theorem 1 is a special case of Theorem 3. However, the proof of Theorem 1 will be useful for deriving Theorem 9. It also yields a better running time than Theorem 3.

4 INTRACTABLE CASES

In this section, we further explore the frontier of tractability for the WDP for intervals with gaps and consider very restricted instances for which the WDP nevertheless becomes \mathcal{NP} -complete. The following theorem can be obtained by a simple reduction from the WDP for 2×2 rectangles [25, Theorem 9], by mapping these rectangles to a linear structure as depicted in Figure 1.

Theorem 4. The WDP is \mathcal{NP} -complete for combinatorial auctions $\mathcal{C} = (A, P, b)$ with $P := \{\{i, i + 1\} \cup \{j, j + 1\} : j = i + \lfloor \sqrt{n} \rfloor\}$, even if all bids have value 1.

The WDP is equivalent to solving the maximum weighted independent set problem in a bid graph, where each vertex corresponds to a bid, weighted with the value of the bid, and there is an edge between two vertices if and only if the corresponding combinations have a non-empty intersection [cf., e.g., 19]. This connection implies, by a result of Fellows et al. [11, Theorem 1] on the parameterized complexity of k -INDEPENDENT SET for multiple-interval graphs, $W[1]$ -hardness—with respect to the revenue of the auction—of the WDP for combinations of the form $I_1 \cup I_2$, where I_1, I_2 are discrete intervals and where all bids have value 1. Further, the work by Bar-Yehuda et al. [1, Corollary 2.3] implies \mathcal{NP} -hardness (and \mathcal{APX} -hardness) of the WDP for combinations of the form $\{i, i + 1\} \cup \{j, j + 1\}$ and bids with value 1. However, in Theorem 4 we restrict the distance between the two intervals of each bid, i.e., we require $j = i + \sqrt{n}$ for each combination $\{i, i + 1\} \cup \{j, j + 1\}$.

The WDP is also \mathcal{NP} -complete if bids are allowed on intervals of length 3 and on combinations with two elements.

Theorem 5. The WDP is \mathcal{NP} -complete for combinatorial auctions $\mathcal{C} = (A, P, b)$ with $P := \{\{i, i + 2\} : 1 \leq i \leq n - 2\} \cup \{\{i, j\} : i, j \in A\}$, even if all bids have value 1.

Proof. To show \mathcal{NP} -hardness, we use a reduction from the \mathcal{NP} -complete problem 2P1N-SAT, a variant of the satisfiability problem for a collection of clauses where each variable occurs exactly two times as a positive literal and one time as a negative literal [29, p.238f]. Let \mathcal{I}_{SAT} be an instance of 2P1N-SAT. Let

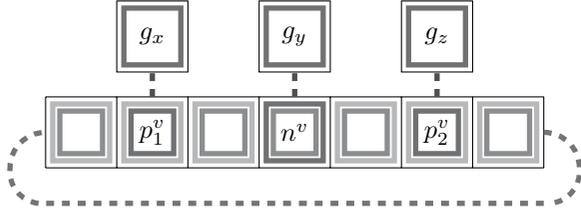


Figure 4. Construction of the bids for a variable $v \in \mathcal{V}$. The items p_1^v and p_2^v represent the occurrences of v , and n^v the occurrence of $\neg v$, respectively. Furthermore, the items g_x and g_z correspond to the clauses containing v , and g_y corresponds to the clause containing $\neg v$. If a clause contains more than one variable, the corresponding clause item occurs in the construction for multiple variables. The dashed lines indicate that the two connected items belong to the same bid. Additionally, three consecutive items of the same shade of grey represent one bid each. Altogether there are seven bids shown in the figure: Two medium grey, two light grey and three dark grey bids. The light grey and the medium grey bids simulate possible variable assignments: $v \mapsto 1$ ($v \mapsto 0$) corresponds to an outcome where the two medium (light) grey bids are accepted. In the first case the items p_1^v and p_2^v can be used to accept the bids containing g_x and g_z . In the second case the situation is analog for n^v and g_y .

$\mathcal{V} := \{v_1, v_2, \dots, v_n\}$ denote the set of n variables and $\mathcal{C} := \{c_1, c_2, \dots, c_m\}$ the collection of m clauses of \mathcal{I}_{SAT} . For every variable $v_i \in \mathcal{V}$, we create seven consecutive items

$$7(i-1)+1, 7(i-1)+2, \dots, 7i,$$

and for every clause $c_j \in \mathcal{C}$ one item $g_j := 7n + j$. Overall, we create $n := 7n + m$ items. For better readability we define

$$\begin{aligned} p_1^{v_i} &:= 7(i-1) + 2, \\ p_2^{v_i} &:= 7(i-1) + 6, \\ n^{v_i} &:= 7(i-1) + 4, \end{aligned}$$

where $p_1^{v_i}$ and $p_2^{v_i}$ represent the occurrences of v_i and n^{v_i} the occurrence of $\neg v_i$, respectively. Let x_i, z_i and y_i be the indices of the clauses containing the literals represented by $p_1^{v_i}, p_2^{v_i}$ and n^{v_i} , respectively. We create seven bids with value 1 for every variable $v_i \in \mathcal{V}$ (see Figure 4 for a visualisation):

$$\begin{aligned} B_{v_i \mapsto 1} &:= \{7(i-1) + 3, n^{v_i}, 7(i-1) + 5\}, \\ B'_{v_i \mapsto 1} &:= \{7(i-1) + 1, 7i\}, \\ B_{v_i \mapsto 0} &:= \{7(i-1) + 1, p_1^{v_i}, 7(i-1) + 3\}, \\ B'_{v_i \mapsto 0} &:= \{7(i-1) + 5, p_2^{v_i}, 7i\}, \\ B_{v_i}^1 &:= \{p_1^{v_i}, g_{x_i}\}, \\ B_{v_i}^2 &:= \{p_2^{v_i}, g_{z_i}\}, \\ B_{\neg v_i} &:= \{n^{v_i}, g_{y_i}\}. \end{aligned}$$

We now show that \mathcal{I}_{SAT} is satisfiable if and only if there is an outcome of the auction with revenue $k \geq 2n + m$.

“ \Rightarrow ”: Let $\phi: \mathcal{V} \rightarrow \{0, 1\}$ be an assignment that satisfies all clauses. For every variable $v_i \in \mathcal{V}$ we accept the bids on $B_{v_i \mapsto \phi(v_i)}$ and $B'_{v_i \mapsto \phi(v_i)}$. This is possible since

$$B_{v_i \mapsto 1} \cap B'_{v_i \mapsto 1} = B_{v_i \mapsto 0} \cap B'_{v_i \mapsto 0} = \emptyset$$

and because these bids are on subsets of $[7(i-1) + 1, 7i]$, i.e., items that only occur in bids created for the variable v_i . This yields a revenue of $2n$. For every clause $c_j \in \mathcal{C}$ there is at least one literal satisfied by ϕ , otherwise the clause would not be satisfied. Let l_j be any satisfied literal of the clause c_j . By construction, the item representing l_j is still available and only contained in exactly one bid involving the item g_j . Therefore, we can accept one additional bid for every clause, resulting in an outcome with revenue $2n + m$.

“ \Leftarrow ”: Let W be an outcome of the auction with revenue $\geq 2n + m$.

We first show that all clause items must have been sold. Suppose to the contrary that there is an item g_j which is not contained in an accepted bid. Then, we can accept at most $m - 1$ bids involving clause items. The remaining bids are $B_{v_i \mapsto 1}, B'_{v_i \mapsto 1}, B_{v_i \mapsto 0}$ and $B'_{v_i \mapsto 0}$ for every variable $v_i \in \mathcal{V}$. Since $(B_{v_i \mapsto 1}, B'_{v_i \mapsto 1})$ and $(B_{v_i \mapsto 0}, B'_{v_i \mapsto 0})$ are the only pairs of these bids having an empty intersection, we can accept at most $2n$ additional bids. Consequently, the revenue is at most $2n + m - 1$. This is a contradiction, thus all clause items have been sold.

We show that the accepted bids involving the clause items induce a partial mapping $\phi_p: \mathcal{V} \rightarrow \{0, 1\}$ such that at least one literal is satisfied in each clause. Every accepted bid involving a clause item contains exactly one other item representing a literal that determines ϕ_p for the corresponding variable, i.e., ϕ_p assigns to the variable the truth value that makes the literal satisfied. This cannot result in an inconsistent assignment, as we will show now. Suppose the result is an inconsistent assignment. Then, there is a variable $v_i \in \mathcal{V}$ such that $B_{\neg v_i}$ and at least one of the bids $B_{v_i}^1$ and $B_{v_i}^2$ are accepted. In this case only one of the bids $B_{v_i \mapsto 1}, B'_{v_i \mapsto 1}, B_{v_i \mapsto 0}$ and $B'_{v_i \mapsto 0}$ can be accepted, so the achievable revenue is at most $2n + m - 1$. Again, this is a contradiction, so the assignment is consistent. Every variable not assigned by ϕ_p may be chosen arbitrary, since ϕ_p already satisfies one literal in each clause. So $\phi: \mathcal{V} \rightarrow \{0, 1\}$ with

$$\phi(v_i) = \begin{cases} \phi_p(v_i) & \text{if } \phi_p(v_i) \in \{0, 1\}, \\ 1 & \text{otherwise,} \end{cases}$$

is an assignment satisfying all clauses of the \mathcal{I}_{SAT} instance.

This construction can clearly be done in polynomial time, so the WDP is \mathcal{NP} -complete as claimed. \square

We could have obtained a similar result by adjusting the proof of Conitzer et al. [7] showing that permitting bidders to bundle two intervals together yields a \mathcal{NP} -hard instance of the WDP: The same structure arises if we restrict VERTEX COVER to cubic graphs in their reduction, but then the proof still contains bids of value 1 and 2.

Since the WDP is solvable in polynomial time if either all permitted combinations are intervals or if all permitted combinations consist of at most two elements only [25], Theorem 5 implies the following corollary.

Corollary 6. *The class of tractable instances of WDP is not closed under union.*

In the next theorem, bids are allowed only on combinations consisting of two consecutive elements and an arbitrary third.

Theorem 7. *The WDP is \mathcal{NP} -complete for combinatorial auctions $\mathcal{C} = (A, P, b)$ with $P := \{\{i, j, k\} \subseteq A : j = i + 1\}$, even if all bids have value 1.*

In order to prove this theorem, we make use of the following lemma.

Lemma 8. *Let $G = (V, E)$ be a connected cubic graph. There is a bijection $\beta: E \rightarrow \{1, 2, \dots, |E|\}$ so that for each vertex $v \in V$*

there are two incident edges that are mapped to consecutive integers. Such a bijection can be found in linear time.

Proof. Let $G = (V, E)$ be a cubic graph with vertices v_1, v_2, \dots, v_n . By the handshaking lemma, every cubic graph has an even number of vertices. Hence, we have $n = 2m$ for some positive integer m . We extend G , so that every vertex has even degree: For every $i = 1, 2, \dots, m$ we add a vertex w_i and connect it to the vertices v_{2i-1} and v_{2i} (see Figure 5).

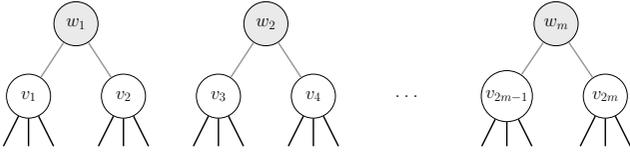


Figure 5. Extension of a cubic graph G with vertices v_1, v_2, \dots, v_{2m} to a (non-cubic) graph G' , so that every vertex has even degree. The edges of G are hinted at below the vertices (every vertex is incident to three edges) and depend on the specific graph. For every $i = 1, 2, \dots, m$ we add a vertex w_i and connect it to the vertices v_{2i-1} and v_{2i} (depicted in grey).

The resulting graph $G' = (V', E')$ is connected and every vertex has even degree, because every $v \in V$ has degree 4 in G' and every $w \in V' \setminus V$ has degree 2. Thus, G' has an Eulerian circuit and we can find one in linear time with Hierholzer's algorithm [17]. Consider an arbitrary Eulerian circuit in G' . Starting from w_1 (or any other vertex $w_i \in V' \setminus V$), we traverse the edges in the order in which they appear on the circuit. Let j be a counter initially set to 1. Each time we encounter an edge $e \in E$, we set $\beta(e) := j$ and increment j by 1. Every vertex $v \in V$ is once both entered and left via edges contained in E , since exactly one incident edge is not in E . Therefore, two of the incident edges are assigned to consecutive integers.⁶ An Eulerian circuit visits every edge of a graph exactly once, so this approach yields a bijection with the desired property. With suitable data structures, the traversal of the edges can be done in $\mathcal{O}(E') = \mathcal{O}(E)$ and the test whether $e \in E$ in $\mathcal{O}(1)$. Overall, the construction needs linear time. \square

Blumrosen and Nisan [3, p.271] use a reduction from INDEPENDENT SET to prove that the WDP is \mathcal{NP} -complete. Since INDEPENDENT SET remains \mathcal{NP} -hard when restricted to 2-connected cubic planar graphs [20, p. 10 f.], we can apply Lemma 8 and prove Theorem 7 using a reduction from the restricted version of INDEPENDENT SET in an analogous manner.

We can further strengthen some of these results, using refined \mathcal{NP} -completeness results for several variants of the tiling problem [2, 21]. We then obtain \mathcal{NP} -completeness of the WDP with permitted combinations $P_1 := \{\{i, i + 1, j\} \subseteq A : j = i + r \wedge r \in \{\lfloor \sqrt{n} \rfloor, \lfloor \sqrt{n} \rfloor + 1\}\}$ and $P_2 := \{\{i, i + 2\} : 1 \leq i \leq n - 2\} \cup \{\{i, i + \lfloor \sqrt{n} \rfloor\} : 1 \leq i \leq n - \lfloor \sqrt{n} \rfloor\}$, respectively, even if we only allow bids with value 1. If we replace $\lfloor \sqrt{n} \rfloor$ in the definition of P_i , $i = 1, 2$, with a constant value, the WDP can be solved in polynomial time even if we remove the restriction to bids of value 1 (see Theorem 3).

⁶ At this point, it is important that we chose a vertex $w \notin V$ as a starting point. Otherwise, this would not necessarily hold for the start vertex.

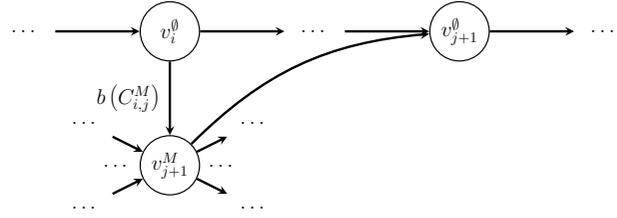


Figure 6. Representing partial assignments where the set of assigned items is not necessarily an interval. We create an edge from v_i^0 to v_{j+1}^M if and only if there is a bid on $C_{i,j}^M$. For $M = \emptyset$, this is an interval bid. If $M \neq \emptyset$, we create an edge from v_{j+1}^M to v_{j+1}^0 , possibly with weight 0 (if this edge does not represent an existing bid). The outgoing edges from v_{j+1}^M for the case $M \neq \emptyset$ are explained in Figure 7. We do not have to consider the incoming edges separately, because they can be regarded as outgoing edges from other vertices.

5 FIXED-PARAMETER TRACTABLE CASES

In this section, we investigate the influence of certain parts of the input on the computational hardness of the WDP. We consider two parameters and obtain fixed-parameter tractability with respect to each of them.

Theorem 9. For $s \in \mathbb{N}$, let $\mathcal{C} = (A, P, b)$ be a combinatorial auction with $P := \{C_{i,j}^M : 1 \leq i \leq j \leq n \wedge \exists k \in A : M \subseteq [k, k + s - 1] \subseteq [i + 1, j - 1]\}$, where $C_{i,j}^M := [i, j] \setminus M$. Given m bids, an optimal outcome for \mathcal{C} can be found in time $\mathcal{O}(n^3 s^2 4^s)$, which can also be expressed in terms of the number m of submitted bids as $\mathcal{O}(n^2 m 2^s)$. Hence, the WDP is fixed-parameter tractable with respect to the length s of the section in each interval within which any subset of items may be missing. More precisely, the parameter s is defined as follows. For each combination C corresponding to a bid, let s_C denote the minimum natural number such that all missing elements of C are within an interval $[k_C, k_C + s_C - 1]$ with $k_C \in A$. Then, the parameter s is the maximum s_C for the given instance.

Proof. We will only sketch the proof, since it is similar to the proof of Theorem 1. Given an instance of WDP as described, we create a graph in which the longest path between two designated vertices corresponds to an optimal outcome of the auction.

Again, in a vertex v_i^M all items of $[1, i - 1] \setminus M$ are assigned. We start by creating a path from v_1^0 to v_{n+1}^0 . An edge on this path has weight 0 if and only if there is no bid on the corresponding item. The two designated vertices are again v_1^0 and v_{n+1}^0 , i. e., the longest path from v_1^0 to v_{n+1}^0 in the final graph will correspond to an optimal outcome of the auction. Now, we will create further vertices and edges such that every outcome of the auction is represented as an weighted path from v_1^0 to v_{n+1}^0 in a specific order, i. e., if we sort an arbitrary outcome by the first element of each combination, then the contained combinations appear in that order on the corresponding path. For every bid on a combination, we create vertices and edges as shown in Figures 6 and 7.

If an edge has non-zero weight, it is weighted with the maximum value that someone has bid on a combination leading to this edge. Since there might be several different combinations leading to the same edge, we label the edge also with the corresponding bid.

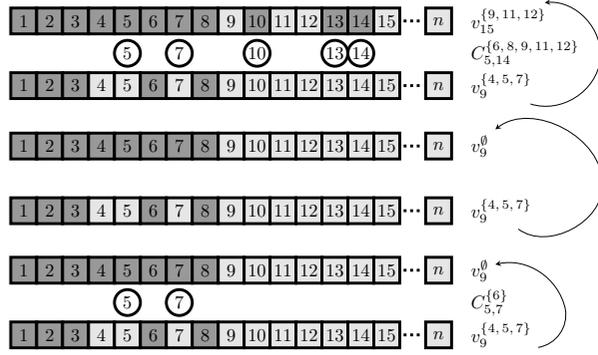


Figure 7. Possible transitions from a vertex v_i^M with $M \neq \emptyset$. We consider an example which can be easily generalised; here we have $i = 9$ and $M = \{4, 5, 7\}$. The assigned items are dark grey and the unassigned ones are light grey. The circles in one row visualise the corresponding combination with a non-zero bid on the right. On the bottom we have the case of a bid on a subset of M . Since we construct the outcomes of the auction in a specific order, we do not have to consider item 4 on this path any more. If there is a bid on $C \subset M$ with $4 \in C$, then we create a separate edge, e.g., a bid on $\{4, 5\}$ would lead to an edge to $v_9^{\{7\}}$. It can happen that several bids lead to the same successor; in this case we consider only the one with the highest value and set the weight of the edge accordingly. If there is no bid as depicted in the middle, we create an edge to v_9^0 with weight 0. This ensures that the path can be extended by combinations that do not overlap with $v_9^{\{4, 5, 7\}}$. Finally, there may be bids on combinations overlapping with $v_9^{\{4, 5, 7\}}$ that are not subsets of $M = \{4, 5, 7\}$ (shown at the top). Item 4 is again omitted.

The correctness of this construction can be shown in a similar fashion as for the proof of Theorem 1. We will briefly comment on the correctness of the case depicted on top of Figure 7. The set of vertices in the constructed graph is

$$V := \{v_i^M : 1 \leq i \leq n + 1 \wedge \exists k \in A : M \subseteq [k, k + s - 1] \subseteq [2, i - 2]\}.$$

Actually, fewer vertices may suffice, e.g., if V contains vertices that are not reachable from v_1^0 . The important property in the top case is that the successor, say $v_{j+1}^{M'}$, reached via a combination, say $C_{i,j}^M$, is also an element of V , i.e., the length of the section with missing elements does not increase. Since we represent the combinations of an outcome in the order described above, we may disregard all items lower than i . Consequently, M' is a proper subset of M —due to the overlap we have $|M'| \leq |M| - 1$. Hence, $v_{j+1}^{M'} \in V$.

A major difference to Theorem 1 is that V and E are not of polynomial size. Here, the number of vertices is exponential in s . We have $|V| \in \mathcal{O}(n^2 2^s)$. It is not hard to see that the asymptotic number of edges is dominated by the top case in Figure 7. Let us consider an arbitrary vertex v_i^M with $M \neq \emptyset$. The section with missing elements of a combination, say $C_{i',j'}^{M'}$, as depicted in the top case is partly fixed because it has to contain $i - 1$. For this reason, there are only $\mathcal{O}(s)$ possible starting points for M' . Further, there are $\mathcal{O}(s)$ possible values for i' , since $i' \in M$, and $\mathcal{O}(2^s)$ possibilities for M' . Finally, there are $\mathcal{O}(n)$ possible values for j' . Thus, we can conclude $|E| \in \mathcal{O}(|V|s^2 2^s n) = \mathcal{O}(n^3 s^2 4^s)$. It is easy to see that $|E|$ is also in $\mathcal{O}(n^2 m 2^s)$, since there are $\mathcal{O}(m)$ outgoing edges for every vertex. The graph obviously is directed and acyclic, therefore computing a longest path from v_1^0 to v_{n+1}^0 that yields an optimal outcome of the

auction requires time $\mathcal{O}(n^3 s^2 4^s)$ or, expressed in terms of the number m of submitted bids, time $\mathcal{O}(n^2 m 2^s)$. \square

We can represent the combinations C_i of a combinatorial auction with $b(C_i) > 0$, $1 \leq i \leq m$, as a binary matrix $(c_{ij})_{1 \leq i \leq m, 1 \leq j \leq n}$ with $c_{ij} = 1 \Leftrightarrow j \in C_i$ (cf. the integer program of Rothkopf et al. [25]). If this matrix has the consecutive ones property (CIP), i.e., if the columns can be permuted so that all ones appear consecutively in every row, the WDP can be solved in polynomial time [25]. Dom [9, p. viii and pp. 79–118] shows that the problem of deleting a minimum number of rows—corresponding to combinations in our case—to transform a given matrix into one with the CIP is fixed-parameter tractable with respect to the combined parameter (Δ, d) , where Δ is the maximum number of 1-entries per row and d is the number of rows that may be deleted. Building on this result we can prove the following theorem.

Theorem 10. *The WDP is in \mathcal{FPT} for combinatorial auctions $\mathcal{C} = (A, P, b)$ with unrestricted P w.r.t. the combined parameter (g, k) consisting of the maximum number g of goods in a bid and the minimum number k of bids that have to be deleted such that the remaining problem can be represented using intervals without gaps.*

The parameter k above can be thought of as a measure for the “distance” [16] of a given instance from an instance that is tractable; here the tractable instance consists of intervals and k describes the similarity to such an instance.

A result by Fomin et al. [13] implies that the WDP is in \mathcal{FPT} with respect to the number of bids that need to be deleted so that the corresponding bid graph becomes an interval graph. However, Fomin et al. make the assumption that “an interval deletion set is provided as a part of the input, as it is an open question whether INTERVAL VERTEX DELETION is \mathcal{FPT} ” [13, p. 352]; this is a restriction we do not use in Theorem 10.

6 CONCLUSION

We have introduced a new domain into the study of combinatorial auctions, consisting of all those allocation problems in which the goods can be arranged in a sequence in such a way that every bid concerns a discrete interval with multiple gaps of bounded length each. As already pointed out by Rothkopf et al. [25], even the simplest such scenario, namely the one without any gaps at all, is of some practical interest, e.g., for selling licenses for radio frequencies. Allowing for gaps increases flexibility and thereby makes this model relevant to a wider range of applications, as illustrated, for instance, by our introductory example on auctioning off the cells making up a construction ground.

For the case without any gaps at all, the problem of computing an optimal allocation was previously known to be solvable in polynomial time. We have systematically explored the extent to which this positive result can (and cannot) be generalised. If each interval has arbitrarily many gaps of at most some fixed length, there still is a polynomial algorithm. On the other hand, for several other seemingly mild extensions, we have established \mathcal{NP} -completeness results, thereby demonstrating just how subtle the difference between tractability and intractability can be. This complements previous work on the fine-grained complexity analysis of group decision making in AI [see, e.g., 15]. Finally, we identified two parameters that, when kept fixed, render the WDP tractable.

Our proofs employ both familiar and novel techniques. Specifically, the proof of Theorem 1 makes use of the tractability of the

LONGEST PATH problem for directed acyclic edge-weighted graphs, which is a helpful graphical representation of a dynamic programming approach.

An interesting question remains for most of the (fixed-parameter) tractable results, Theorem 10 being the exception: What is the computational complexity of recognizing instances with the considered structures if an ordering of the goods is not provided as part of the input? It may be the case that such a decision procedure is not constructive in the sense that it can be used to find an ordering with the desired properties if one exists. Then it would be interesting to know how hard it is to find such an ordering.

Acknowledgements

This work has been partly supported by COST Action IC1205 on Computational Social Choice. Further, we thank the anonymous referees for their valuable comments and suggestions.

REFERENCES

- [1] Reuven Bar-Yehuda, Magnús M. Halldórsson, Joseph Seffi Naor, Hadas Shachnai, and Irina Shapira, ‘Scheduling split intervals’, *SIAM Journal on Computing*, **36**(1), 1–15, (2006).
- [2] Danièle Beauquier, Maurice Nivat, Eric Rémila, and Mike Robson, ‘Tiling figures of the plane with two bars’, *Computational Geometry*, **5**(1), 1–25, (1995).
- [3] Liad Blumrosen and Noam Nisan, ‘Combinatorial auctions’, In Nisan et al. [24], chapter 11, 267–300.
- [4] Hans L. Bodlaender and Arie M.C.A. Koster, ‘Treewidth computations I. upper bounds’, *Information and Computation*, **208**(3), 259 – 275, (2010).
- [5] Hans L. Bodlaender and Arie M.C.A. Koster, ‘Treewidth computations II. lower bounds’, *Information and Computation*, **209**(7), 1103 – 1119, (2011).
- [6] Craig Boutilier and Holger H. Hoos, ‘Bidding languages for combinatorial auctions’, in *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-2001)*, (2001).
- [7] Vincent Conitzer, Jonathan Derryberry, and Tuomas Sandholm, ‘Combinatorial auctions with structured item graphs’, in *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI-2004)*, (2004).
- [8] *Combinatorial Auctions*, eds., Peter Cramton, Yoav Shoham, and Richard Steinberg, MIT press, 2006.
- [9] Michael Dom, *Recognition, Generation, and Application of Binary Matrices with the Consecutive-Ones Property*, Ph.D. dissertation, Friedrich-Schiller-Universität Jena, 2008.
- [10] Rodney G. Downey and Michael R. Fellows, *Fundamentals of Parameterized Complexity*, Springer, 2013.
- [11] Michael R. Fellows, Danny Hermelin, Frances Rosamond, and Stéphane Vialette, ‘On the parameterized complexity of multiple-interval graph problems’, *Theoretical Computer Science*, **410**(1), 53–61, (2009).
- [12] Jörg Flum and Martin Grohe, *Parameterized Complexity Theory*, Texts in Theoretical Computer Science, Springer, New York, 2006.
- [13] Fedor V. Fomin, Serge Gaspers, Petr A. Golovach, Karol Suchan, Stefan Szeider, Erik Jan van Leeuwen, Martin Vatshelle, and Yngve Villanger, ‘k-gap interval graphs’, in *LATIN 2012: Theoretical Informatics*, 350–361, Springer, (2012).
- [14] Georg Gottlob and Gianluigi Greco, ‘Decomposing combinatorial auctions and set packing problems’, *J. ACM*, **60**(4), 24:1–24:39, (2013).
- [15] Gianluigi Greco and Jérôme Lang, ‘Group decision making via weighted propositional logic: Complexity and islands of tractability’, in *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI-2015)*, (2015).
- [16] Jiong Guo, Falk Hüffner, and Rolf Niedermeier, ‘A structural view on parameterizing problems: Distance from triviality’, in *Proceedings of the 1st International Workshop on Parameterized and Exact Computation (IWPEC ’04)*, (2004).
- [17] Carl Hierholzer, ‘Ueber die Möglichkeit, einen Linienzug ohne Wiederholung und ohne Unterbrechung zu umfahren’, *Mathematische Annalen*, **6**(1), 30–32, (1873).
- [18] Daniel Lehmann, Rudolf Müller, and Tuomas Sandholm, ‘The winner determination problem’, In Cramton et al. [8], chapter 12.
- [19] David Loker and Kate Larson, ‘Parameterizing the winner determination problem for combinatorial auctions’, in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2010)*, (2010).
- [20] Bojan Mohar, ‘Face covers and the genus problem for apex graphs’, *Journal of Combinatorial Theory, Series B*, **82**(1), 102–117, (2001).
- [21] Christopher Moore and John Michael Robson, ‘Hard tiling problems with simple tiles’, *Discrete & Computational Geometry*, **26**(4), 573–590, (2001).
- [22] Rolf Niedermeier, *Invitation to Fixed-Parameter Algorithms*, Oxford Lecture Series in Mathematics and its Applications, Oxford University Press, 2006.
- [23] Rolf Niedermeier, ‘Reflections on multivariate algorithmics and problem parameterization’, in *Proceedings of the 27th International Symposium on Theoretical Aspects of Computer Science (STACS-2010)*, (2010).
- [24] *Algorithmic Game Theory*, eds., Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani, Cambridge University Press, 2007.
- [25] Michael H. Rothkopf, Aleksandar Pekeč, and Ronald M. Harstad, ‘Computationally manageable combinatorial auctions’, *Management Science*, **44**(8), 1131–1147, (1998).
- [26] Tuomas Sandholm, ‘Algorithm for optimal winner determination in combinatorial auctions’, *Artificial Intelligence*, **135**(1), 1–54, (2002).
- [27] Robert Sedgewick and Kevin Wayne, *Algorithms*, Addison-Wesley Professional, 4th edn., 2011.
- [28] Makoto Yokoo, ‘Characterization of strategy/false-name proof combinatorial auction protocols: Price-oriented, rationing-free protocol’, in *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-2003)*, (2003).
- [29] Ryo Yoshinaka, ‘Higher-order matching in the linear lambda calculus in the absence of constants is NP-complete’, in *Proceedings of the 16th International Conference on Term Rewriting and Applications (RTA-2005)*, Springer, (2005).

Efficient SAT Approach to Multi-Agent Path Finding Under the Sum of Costs Objective

Pavel Surynek¹ and Ariel Felner² and Roni Stern³ and Eli Boyarski⁴

Abstract. In the *multi-agent path finding* (MAPF) the task is to find non-conflicting paths for multiple agents. In this paper we present the first SAT-solver for the *sum-of-costs* variant of MAPF which was previously only solved by search-based methods. Using both a lower bound on the sum-of-costs and an upper bound on the makespan, we are able to have a reasonable number of variables in our SAT encoding. We then further improve the encoding by borrowing ideas from ICTS, a search-based solver. Experimental evaluation on several domains showed that there are many scenarios where the new SAT-based method outperforms the best variants of previous sum-of-costs search solvers - the ICTS and ICBS algorithms.

1 Introduction and Background

The *multi-agent path finding* (MAPF) problem consists of a graph, $G = (V, E)$ and a set $A = \{a_1, a_2, \dots, a_m\}$ of m agents. Time is discretized into time steps. The arrangement of agents at time-step t is denoted as α_t . Each agent a_i has a start position $\alpha_0(a_i) \in V$ and a goal position $\alpha_+(a_i) \in V$. At each time step an agent can either *move* to an adjacent empty location⁵ or *wait* in its current location. The task is to find a sequence of move/wait actions for each agent a_i , moving it from $\alpha_0(a_i)$ to $\alpha_+(a_i)$ such that agents do not *conflict*, i.e., do not occupy the same location at the same time. Formally, an MAPF instance is a tuple $\Sigma = (G = (V, E), A, \alpha_0, \alpha_+)$. A *solution* for Σ is a sequence of arrangements $\mathcal{S}(\Sigma) = [\alpha_0, \alpha_1, \dots, \alpha_\mu]$ such that $\alpha_\mu = \alpha_+$ where α_{t+1} results from valid movements from α_t for $t = 1, 2, \dots, \mu - 1$. An example of MAPF and its solution are shown in Figure 1.

MAPF has practical applications in video games, traffic control, robotics etc. (see [17] for a survey). The scope of this paper is limited to the setting of *fully cooperative* agents that are centrally controlled. MAPF is usually solved aiming to minimize one of the two commonly-used global cumulative cost functions:

(1) sum-of-costs (denoted ξ) is the summation, over all agents, of the number of time steps required to reach the goal location [8, 23, 18, 17]. Formally, $\xi = \sum_{i=1}^m \xi(a_i)$, where $\xi(a_i)$ is an *individual path cost* of agent a_i .

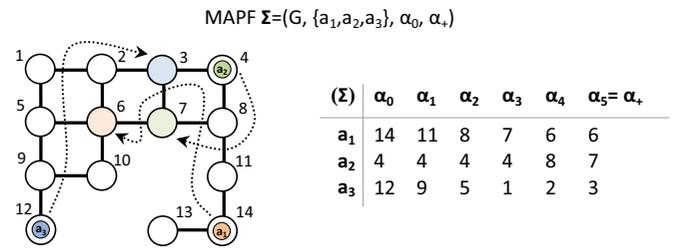


Figure 1. Example of MAPF for agents a_1 , a_2 , and a_3 over a 4-connected grid (left) and its optimal solution (right)

(2) makespan: (denoted μ) is the total time until the last agent reaches its destination (i.e., the maximum of the individual costs) [25, 27, 30].

It is important to note that in any solution $\mathcal{S}(\Sigma)$ it holds that $\mu \leq \xi \leq m \cdot \mu$. Thus the optimal *makespan* is usually smaller than the optimal *sum-of-costs*.

Finding optimal solutions for both variants is NP-Hard [34, 25]. Therefore, many suboptimal solvers were developed and are usually used when m is large [15, 6, 20, 14, 12, 32]

1.1 Optimal MAPF Solvers

The focus of this paper is on optimal solvers which are divided into two main classes:

(1) Reduction-based solvers. Many recent optimal solvers reduce MAPF to known problems such as COP [15], SAT [26], Inductive Logic Programming [33] and Answer Set Programming [9]. These papers mostly prove a polynomial-time reduction from MAPF to these problems. These reductions are usually designed for the *makespan* variant of MAPF; they are not applicable for the sum-of-costs variant.

(2) Search-based solvers. By contrast, many recent optimal MAPF solvers are search-based. Some are variants of the A* algorithm on a global *search space* – all different ways to place m agents into V vertices, one agent per vertex [23, 31]. Other employ novel search trees [18, 17, 5]. These search-based solvers are usually designed for the *sum-of-costs* MAPF variant.

A major weakness is that connection/comparison between different algorithms was usually done only within a given class of algorithms and cost variant but not across these two classes.

¹ Charles University Prague, Malostranské náměstí 25, 11800, Praha, Czech Republic, email: pavel.surynek@mff.cuni.cz

² Ben Gurion University, Beer-Sheva, Israel 84105, email: felner@bgu.ac.il

³ Ben Gurion University, Beer-Sheva, Israel 84105, email: sternron@bgu.ac.il

⁴ Bar-Ilan University, Ramat-Gan, Israel, email: eli.boyarski@gmail.com

⁵ Some variants of MAPF relax the empty location requirement by allowing a chain of neighboring agents to move, given that the head of the chain enters an empty location. Most MAPF algorithms are robust (or at least easily modified) across these variants.

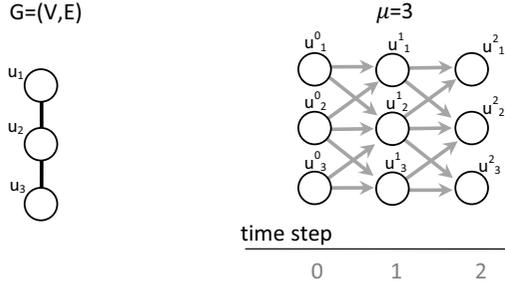


Figure 2. An example of time expansion graph.

1.2 Contributions

This paper aims to start and close the gap. Most of the search-based algorithms can be easily modified to the makespan variant by modifying the cost function and the way the state-space is represented. Some initial directions are given by [17]. By contrast, the reduction-based algorithms are not trivially modified to the sum-of-costs variant and sometimes a completely new reduction is needed.

In this paper we develop the first SAT-based solvers for the sum-of-costs variant which is based on adding *cardinality constraints* [3, 19] for bounding the sum-of-costs. We show how to use known lower bounds on the sum-of-costs to reduce the number of variables that encode these cardinality constraints so as to be practical for current SAT solvers. We then present an *enhanced SAT-solver* which adapts ideas from the ICTS algorithm [18] and uses *multi-value decision diagrams* (MDDs) [22] to further reduce the encoding. This demonstrates the potential of combining ideas from both classes of approaches (search-based and SAT solvers). Experimental results show that our enhanced SAT solver outperforms the best existing search-based solvers for the sum-of-costs variant on many scenarios.

2 SAT Encoding for Optimal Makespan

SAT solvers encompass boolean variables and answer binary questions. The challenge is to apply SAT for MAPF where there is a cumulative cost function. This challenge is stronger for the sum-of-costs variant where each agent has its own cost. We first describe existing SAT encodings for makespan. Then, we present our SAT encoding for sum-of-costs.

A *time expansion graph* (denoted TEG) is a basic concept used in SAT solvers for makespan [27]. We use it too in the sum-of-costs variant below. A TEG is a directed acyclic graph (DAG). First, the set of vertices of the underlying graph G are duplicated for all time-steps from 0 up to the given bound μ . Then, possible actions (move along edges or wait) are represented as directed edges between successive time steps. Figure 2 shows a graph and its TEG for time steps 0, 1 and 2 (vertical layouts). It is important to note that in this example (1) horizontal edges in TEG correspond to *wait* actions. (2) diagonal moves in TEG correspond to real moves. Formally a TEG is defined as follows:

Definition 1 *Time expansion graph of depth μ is a digraph (V, E) where $V = \{u_j^t | t = 0, 1, \dots, \mu \wedge u_j \in V\}$ and $E \subseteq \{(u_j^t, u_k^{t+1}) | t = 0, 1, \dots, \mu - 1 \wedge (\{u_j, u_k\} \in E \vee j = k)\}$.*

The encoding for MAPF introduces propositional variables and constraints for a single time-step t in order to represent any possible arrangement of agents at time t . Given a desired makespan μ , the formula represents the question of whether there is a solution in the TEG of μ time steps. The search for optimal makespan is done by iteratively incrementing μ ($=0, 1, 2, \dots$) until a satisfiable formula is obtained. This ensures optimality in case of a solvable MAPF instance. More information on SAT encoding for the makespan variant can be found, e.g. in [27, 28, 29]

3 Basic-SAT for Optimal Sum-of-costs

The general scheme described above for finding optimal makespan is to convert the optimization problem (finding minimal makespan) to a sequence of decision problems (is there a solution of a given makespan μ). We apply the same scheme for finding optimal sum-of-costs, converting it to a sequence of decision problems – is there a solution of a given sum-of-costs ξ .

However, encoding this decision problem is more challenging than the makespan case, because one needs to both bound the sum-of-costs, but also to predict how many time expansions are needed. We address this challenge by using two key techniques described next: (1) Cardinality constraint for bounding ξ and (2) Bounding the Makespan.

3.1 Cardinality Constraint for Bounding ξ

The SAT literature offers a technique for encoding a *cardinality constraint* [3, 19], which allows calculating and bounding a numeric cost within the formula. Formally, for a bound $\lambda \in \mathbb{N}$ and a set of propositional variables $X = \{x_1, x_2, \dots, x_k\}$ the *cardinality constraint* $\leq_\lambda \{x_1, x_2, \dots, x_k\}$ is satisfied iff the number of variables from the set X that are set to TRUE is $\leq \lambda$.

In our SAT encoding, we bound the sum-of-costs by mapping every agent’s action to a propositional variable, and then encoding a cardinality constraint on these variables. Thus, one can use the general structure of the makespan SAT encoding (which iterates over possible makespans), and add such a cardinality constraint on top. Next we address the challenge of how to connect these two factors together.

3.2 Bounding the Makespan for the Sum of Costs

Next, we compute how many time expansions (μ) are needed to guarantee that if a solution with sum-of-costs ξ exists then it will be found. In other words, in our encoding, the values we give to ξ and μ must fulfill the following requirement:

R1: *all possible solutions with sum-of-costs ξ must be possible for a makespan of at most μ .*

To find a μ value that meets R1, we require the following definitions. Let $\xi_0(a_i)$ be the cost of the shortest individual path for agent a_i , and let $\xi_0 = \sum_{a_i \in A} \xi_0(a_i)$. ξ_0 was called the *sum of individual costs* (SIC) [18]. ξ_0 is an admissible heuristic for optimal sum-of-costs search algorithms, since ξ_0 is a lower bound on the minimal sum-of-costs. ξ_0 is calculated by relaxing the problem by omitting the other agents. Similarly, we define $\mu_0 = \max_{a_i \in A} \xi_0(a_i)$. μ_0 is length of the *longest* of the shortest individual paths and is thus a lower bound on the minimal makespan. Finally, let Δ be the extra cost over SIC (as done in [18]). That is, let $\Delta = \xi - \xi_0$.

Algorithm 1: SAT consult

```

1 MAPF-SAT(MAPF  $\Sigma = (G = (V, E), A, \alpha_0, \alpha_+)$ )
2    $\mu_0 = \max_{a_i \in A} \xi_0(a_i); \Delta \leftarrow 0$ 
3   while Solution not found do
4      $\mu \leftarrow \mu_0 + \Delta;$ 
5     for each agent  $a_i$  do
6       build  $TEG_i(\mu);$ 
7     end
8     Solution=Consult-SAT-SOLVER( $\Sigma, \mu, \Delta$ );
9     if Solution not found then
10       $\Delta++;$ 
11    end
12  end
13  return (Solution);
14 end

```

Proposition 1 For makespan μ of any solution with sum-of-costs ξ , RI holds for $\mu \leq \mu_0 + \Delta$.

Proof outline: The worst-case scenario, in terms of makespan, is that all the Δ extra moves belong to a single agent. Given this scenario, in the worst case, Δ is assigned to the agent with the largest shortest-path. Thus, the resulting path of that agent would be $\mu_0 + \Delta$, as required. \square

Using Proposition 1, we can safely encode the decision problem of whether there is a solution with sum-of-costs ξ by using $\mu = \mu_0 + \Delta$ time expansions, knowing that if a solution of cost ξ exists then it will be found within $\mu = \mu_0 + \Delta$ time expansions. In other words, Proposition 1 shows relation of both parameters μ and ξ which will be both changed by changing Δ . Algorithm 1 summarizes our optimal sum-of-costs algorithm. In every iteration, μ is set to $\mu_0 + \Delta$ (Line 4) and the relevant TEGs (described below) for the various agents are built. Next a decision problem asking whether there is a solution with sum-of-costs ξ and makespan μ is queried (Line 8). The first iteration starts with $\Delta = 0$. If such a solution exists, it is returned. Otherwise ξ is incremented by one, Δ and consequently μ are modified accordingly and another iteration of SAT consulting is activated.

This algorithm clearly terminates for solvable MAPF instances as we start seeking a solution of $\xi = \xi_0$ ($\Delta = 0$) and increment Δ (which increments ξ and μ as well) to all possible values. The unsolvability of an MAPF instance can be checked separately by a polynomial-time complete sub-optimal algorithm such as PUSH-AND-ROTATE [7].

3.3 Efficient Use of the Cardinality Constraint

The complexity of encoding a cardinality constraint depends linearly in the number of constrained variables [19, 21]. Since each agent a_i must move at least $\xi_0(a_i)$, we can reduce the number of variables counted by the cardinality constraint by only counting the variables corresponding to extra movements over the first $\xi_0(a_i)$ movement a_i makes. We implement this by introducing a TEG for a given agent a_i (labeled TEG_i).

TEG_i differs from TEG (Definition 1) in that it distinguishes between two types of edges: E_i and F_i . E_i are (directed) edges whose destination is at time step $\leq \xi_0(a_i)$. These are called *standard edges*. F_i denoted as *extra edges* are directed edges whose destination is at time step $> \xi_0(a_i)$. Figure 3 shows an underlying graph for agent a_1 (left) and the corresponding TEG_1 . Note that the optimal solution of cost 2 is denoted by the diagonal path of the TEG. Edges that belong to F_i are those that their destination is time step 3 (dotted lines). The

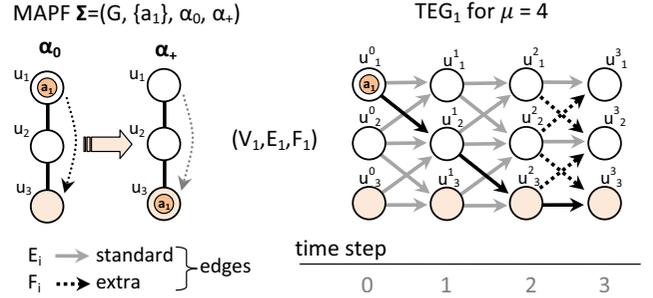


Figure 3. A TEG for an agent that needs to go from u_1 to u_3 .

key in this definition is that the cardinality constraint would only be applied to the extra edges, that is, we will only bound the number of extra edges (they sum up to Δ) making it more efficient.

There are various possibilities to define what happens to an agent when it reaches the goal (disappears, waits etc.). In all cases, edges in TEGs corresponding to wait actions at the goal are not marked as extra. Importantly, our SAT approach is robust across all these variants.

3.4 Detailed Description of the SAT Encoding

Agent a_i must go from its initial position to its goal within TEG_i . This simulates its location in time in the underlying graph G . That is, the task is to find a path from $\alpha_0^0(a_i)$ to $\alpha_+^\mu(a_i)$ in TEG_i . The search for such a path will be encoded within the Boolean formula. Additional constraints will be added to capture all movement constraints such as *collision avoidance* etc. And, of course, we will encode the cardinality constraint that the number of extra edges must be exactly Δ .

We want to ask whether a sum-of-costs solution of ξ exists. For this we build TEG_i for each agent $a_i \in A$ of depth $\mu_0 + \Delta$. We use V_i to denote the set of vertices in TEG_i that agent a_i might occupy during the time steps. Next we introduce the Boolean encoding (denoted BASIC-SAT) which has the following Boolean variables:

1: $\mathcal{X}_j^t(a_i)$ for every $t \in \{0, 1, \dots, \mu\}$ and $u_j^t \in V_i$ — Boolean variable of whether agent a_i is in vertex v_j at time step t .

2: $\mathcal{E}_{j,k}^t(a_i)$ for every $t \in \{0, 1, \dots, \mu - 1\}$ and $(u_j^t, u_k^{t+1}) \in (E_i \cup F_i)$ — Boolean variables that model transition of agent a_i from vertex v_j to vertex v_k through any edge (standard or extra) between time steps t and $t + 1$ respectively.

3: $\mathcal{C}^t(a_i)$ for every $t \in \{0, 1, \dots, \mu - 1\}$ such that there exist $u_j^t \in V_i$ and $u_k^{t+1} \in V_i$ with $(u_j^t, u_k^{t+1}) \in F_i$ — Boolean variables that model cost of movements along **extra edges** (from F_i) between time steps t and $t + 1$.

We now introduce constraints on these variables to restrict illegal values as defined by our variant of MAPF. Other variants may use a slightly different encoding but the principle is the same. Let $T_\mu = \{0, 1, \dots, \mu - 1\}$. Several groups of constraints are introduced for each agent $a_i \in A$ as follows:

C1: If an agent appears in a vertex at a given time step, then it must follow through exactly one adjacent edge into the next time step. This is encoded by the following two constraints, which are posted

for every $t \in T_\mu$ and $u_j^t \in V_i$

$$\mathcal{X}_j^t(a_i) \Rightarrow \bigvee_{(u_j^t, u_k^{t+1}) \in E_i \cup F_i} \mathcal{E}_{j,k}^t(a_i), \quad (1)$$

$$\bigwedge_{(u_j^t, u_k^{t+1}), (u_l^t, u_m^{t+1}) \in E_i \cup F_i \wedge k < l} \neg \mathcal{E}_{j,k}^t(a_i) \vee \neg \mathcal{E}_{l,m}^t(a_i) \quad (2)$$

C2: Whenever an agent occupies an edge it must also enter it before and leave it at the next time-step. This is ensured by the following constraint introduced for every $t \in T_\mu$ and $(u_j^t, u_k^{t+1}) \in E_i \cup F_i$:

$$\mathcal{E}_{j,k}^t(a_i) \Rightarrow \mathcal{X}_j^t(a_i) \wedge \mathcal{X}_k^{t+1}(a_i) \quad (3)$$

C3: The target vertex of any movement except wait action must be empty. This is ensured by the following constraint introduced for every $t \in T_\mu$ and $(u_j^t, u_k^{t+1}) \in E_i \cup F_i$ such that $j \neq k$.

$$\mathcal{E}_{j,k}^t(a_i) \Rightarrow \bigwedge_{a_l \in A \wedge a_l \neq a_i \wedge u_k^{t+1} \in V_i} \neg \mathcal{X}_k^{t+1}(a_l) \quad (4)$$

C4: No two agents can appear in the same vertex at the same time step (although the previous constraint ensures that an agent does not collide with an agent currently residing in a vertex it does not prevent simultaneous entering of the same vertex by multiple agents). That is the following constraint is added for every $t \in T_\mu$ and pair of agents $a_i, a_l \in A$ such that $i \neq l$:

$$\bigwedge_{u_j^t \in V_i \cap V_l} \neg \mathcal{X}_j^t(a_i) \vee \neg \mathcal{X}_j^t(a_l) \quad (5)$$

C5: Whenever an extra edge is traversed the cost needs to be accumulated. In fact, this is the only cost that we accumulate as discussed above. This is done by the following constraint for every $t \in T_\mu$ and extra edge $(u_j^t, u_k^{t+1}) \in F_i$.

$$\mathcal{E}_{j,k}^t(a_i) \Rightarrow \mathcal{C}^t(a_i) \quad (6)$$

C6: Cardinality constraint. Finally the bound on the total cost needs to be introduced. Reaching the sum-of-costs of ξ corresponds to traversing exactly Δ extra edges from F_i . The following cardinality constraint ensures this:

$$\leq_\Delta \left\{ \begin{array}{l} \mathcal{C}^t(a_i) | i = 1, 2, \dots, n \wedge t = 0, 1, \dots, \mu - 1 \\ \wedge \{(u_j^t, u_k^{t+1}) \in F_i\} \neq \emptyset \end{array} \right\} \quad (7)$$

Final formula. The resulting Boolean formula that is a conjunction of $C1 \dots C6$ will be denoted as $\mathcal{F}_{BASIC}(\Sigma, \mu, \Delta)$ and is the one that is consulted by Algorithm 1 (line 4).

The following proposition summarizes the correctness of our encoding.

Proposition 2 *MAPF $\Sigma = (G = (V, E), A, \alpha_0, \alpha_+)$ has a sum-of-costs solution of ξ if and only if $\mathcal{F}_{BASIC}(\Sigma, \mu, \Delta)$ is satisfiable. Moreover, a solution of MAPF Σ with the sum-of-costs of ξ can be extracted from the satisfying valuation of $\mathcal{F}_{BASIC}(\Sigma, \mu, \Delta)$ by reading its $\mathcal{X}_j^t(a_i)$ variables.*

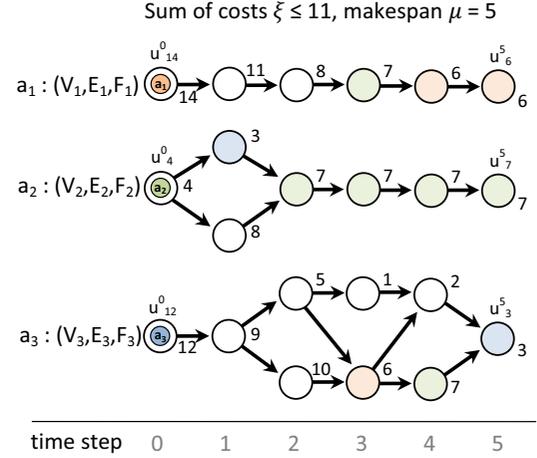


Figure 4. MDDs for agents a_1 , a_2 , and a_3 for the MAPF from Figure 1 for sum of individual cost $\xi \leq 11$.

Proof: The direct consequence of the above definitions is that a valid solution of a given MAPF Σ corresponds to non-conflicting paths in the TEGs of the individual agents. These non-conflicting paths further correspond to satisfying the variable assignment of $\mathcal{F}_{BASIC}(\Sigma, \mu, \Delta)$, i.e., that there are Δ extra edges in TEGs of depth $\mu = \mu_0 + \Delta$. \square

Proposition 3 *Let D be the maximal degree of any vertex in G and let m be the number of agents. If $m \cdot |E| \geq \Delta$ and $m \geq D$ then the number of clauses in $\mathcal{F}_{BASIC}(\Sigma, \mu, \Delta)$ is $O(\mu \cdot m^2 \cdot |E|)$, and the number of variables is $O(\mu \cdot |E| \cdot m)$.*

Proof: The components of $\mathcal{F}_{BASIC}(\Sigma, \mu, \Delta)$ is described in equations 1–7. Equation 1 introduces at most $O(m \cdot \mu \cdot |E|)$ clauses. Equation 2 introduces at most $O(m \cdot \mu |E| \cdot D)$ clauses. Equation 3 introduces at most $O(m \cdot \mu \cdot |E|)$ clauses. Equation 4 introduces at most $O(m^2 \cdot \mu \cdot |E|)$ Equation 5 introduces at most $O(m^2 \cdot \mu \cdot |V|)$ clauses. Equation 6 introduces at most $O(m \cdot \mu \cdot |E|)$ clauses. Equation 7 introduces at most $O(m \cdot \mu \cdot (\xi - \xi_0))$ clauses, since a cardinality constraint checking that n variables has a cardinality constraint of m requires $O(n \cdot m)$ clauses [21]. Summing all the above results in a total of $O(\mu \cdot m \cdot (|E| \cdot (D + m) + (\xi - \xi_0)))$. If we assume that $m > D$ and that $m \cdot |E| > (\xi - \xi_0)$ then the number of clauses is $O(\mu \cdot m^2 \cdot |E|)$. The number of variables is easily computed in a similar way. \square

4 Improving Basic SAT by Adding MDDs

A major parameter that affects the speed of solving of Boolean formulae is their size [13]. The size of formulae in the BASIC-SAT encoding is affected mostly by the size of the TEGs (this is embodied in the $|E|$ factor in the encoding size). To obtain a significant speedup we reduce the size of TEG_i for agent a_i in terms of number of vertices while the soundness of encoding is preserved. To do this we borrow the ideas of *multi-value Decision Diagram* (MDD) from the search-based MAPF algorithm ICTS [18]. This shows the advantage of combining techniques from both classes of approaches (search-based and SAT).

Let TEG_i^μ denote TEG_i for μ time expansions. We set $\mu = \mu_0 + \Delta$ in our solution. The data structure we use for reducing

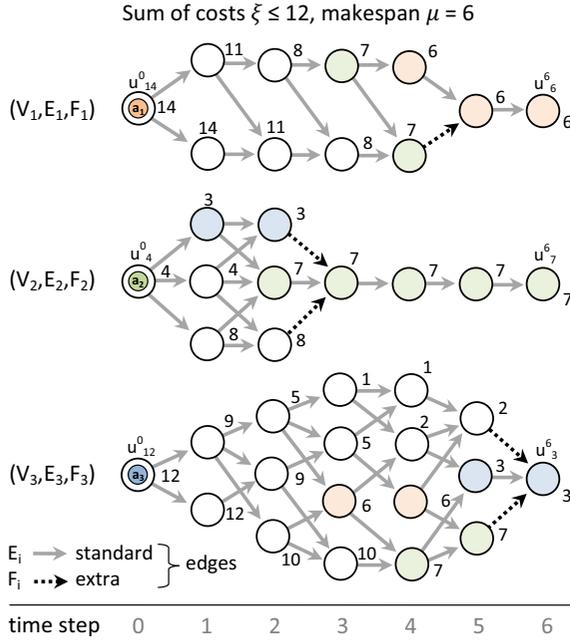


Figure 5. MDDs from Figure 4 for the incremented *sum of individual cost* from 11 to 12 ($\xi \leq 12$).

TEG_i^μ is a *multi-value Decision Diagram* (MDD). MDDs were already used in the search-based MAPF algorithm ICTS [18]. In our context, MDD_i^μ is a digraph that represents all possible valid paths from $\alpha_0(a_i)$ to $\alpha_+(a_i)$ of cost μ for agent a_i . MDD_i^μ has a single *source node* at level 0 and a single *sink node* at level μ . Every node at depth t of MDD_i^μ corresponds to a possible location of a_i at time t , that is on a path of cost μ from $\alpha_0(a_i)$ to $\alpha_+(a_i)$. It is easy to see that MDD_i^μ is subgraph of TEG_i . While TEG_i includes all vertices of G at each time step, MDD_i^μ includes only those vertices and edges that represent possible valid paths, and thus vertices not in MDD_i^μ can be ignored.

Moreover, the maximum cost that can be consumed by single agent a_i under given sum-of-costs bound ξ is $\xi_0(a_i) + \Delta$ where, as defined above, $\xi_0(a_i)$ is the shortest path connecting $\alpha_0(a_i)$ with $\alpha_+(a_i)$ in G (assuming no other agent exist). Thus, it is sufficient to replace TEG_i^μ with $MDD_i^{\xi_0(a_i)+\Delta}$, which is useful since $\xi_0(a_i) + \Delta \leq \mu_0 + \Delta = \mu$.

MDDs for the agents of Figure 1 are shown in Figures 4 and 5. Indeed, the size of the MDDs is much smaller than the corresponding TEGs which include all states for all time steps. Though the increase in size caused by ability to reach more vertices under given sum-of-costs bounds is observable between Figures 4 and 5.

Grid 8x8 m	BASIC-SAT		MDD-SAT	
	Variables	Clauses	Variables	Clauses
1	1 552.8	11 617.6	20.6	27.9
4	14 712.0	127 732.2	276.5	554.0
8	226 391.2	2 099 127.6	18 355.6	68 826.0
16	4 075 187.2	32 108 347.2	2 253 508.2	13 128 646.9

Table 1. The effect of using MDDs in the encoding in terms of the number of variables and clauses.

The encoding that uses MDD-based time expansion will be called

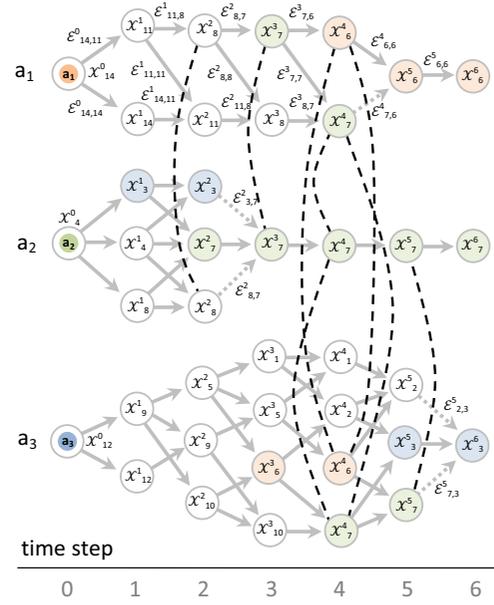


Figure 6. An illustration of MDD-SAT encoding using MDDs from Figure 5. Mutual exclusion constraints (C4) that prevent multiple agent occurrence in the same vertex are shown using dashed edges.

MDD-SAT and the corresponding formulae will be denoted as $\mathcal{F}_{MDD}(\Sigma, \mu, \Delta)$. $\mathcal{F}_{MDD}(\Sigma, \mu, \Delta)$ are similar to BASIC-SAT. The only difference is that in BASIC-SAT there is a variable for all vertices and edges of the TEGs while in MDD-SAT, only variables for the vertices and edges of the MDDs are needed. This difference can be significant. Table 1 presents the number of propositional variables and clauses accumulated over all the constructed formulae for a given MAPF instance for BASIC-SAT and for MDD-SAT over 8×8 grid with 10% obstacles. The average values out of 10 random instances per number of agents is shown. Up to two orders of magnitude reduction is shown.

An illustration of the $\mathcal{F}_{MDD}(\Sigma, \mu, \Delta)$ formula is shown in Figure 6. It is particularly observable that MDDs reduce the number of mutual exclusion constraints (dashed edges) by omitting unreachable vertices (and all the constraints incident with them).

5 Experimental Evaluation

We experimented on 4-connected grids with randomly placed obstacles [20, 23] and on *Dragon Age* maps [17, 24]. Both settings are a standard MAPF benchmarks. The initial position of the agents was randomly selected. To ensure solvability the goal positions were selected by performing a long *random walk* from the initial arrangement.

We compared our SAT solvers to several state-of-the-art search-based algorithms: the *increasing cost tree search* - ICTS [18], *Enhanced Partial Expansion A** - EPEA* [10] and *improved conflict-based search* - ICBS [5]. For all the search algorithms we used the best known setup of their parameters and enhancements suitable for solving the given instances over 4-connected grids.

The SAT approaches were implemented in C++. The implementation consists of a top level algorithm for finding the optimal sum-of-costs ξ and *CNF* formula generator [4] that prepares input formula for a SAT solver into a file. The SAT solver is an external module

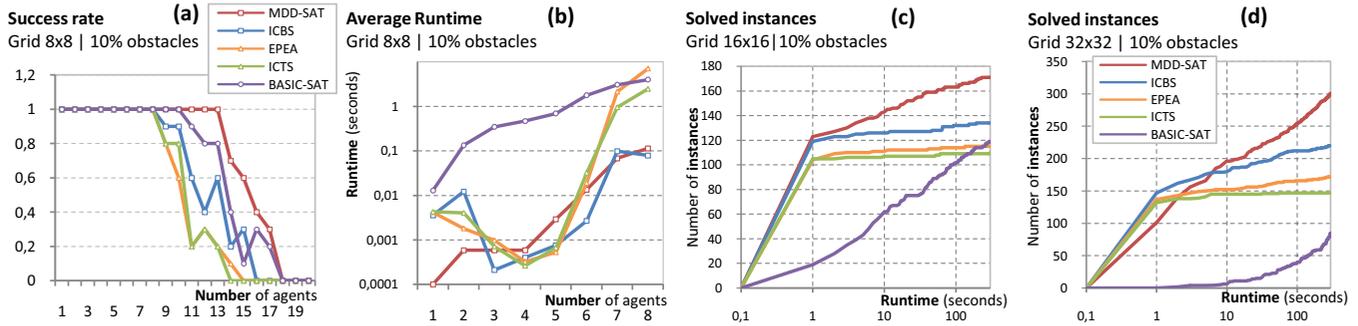


Figure 7. Results on 8×8 grid (left). Number of solved instances in the given runtime on 16×16 and 32×32 grids. (right)

of this architecture. We used `Glucose 3.0` [2, 1] which is a top performing SAT solver in the *SAT Competition* [11, 27].

The cardinality constraint was encoded using a simple standard circuit based encoding called *sequential counter* [21]. In our initial testing we considered various encodings of the cardinality constraint such as those discussed in [3, 19]. However, it turned out that changing the encoding has a minor effect.⁶

ICTS and ICBS were implemented in C#, based on their original implementation (here we used a slight modification in which the target vertex of a move must be empty). All experiments were performed on a Xeon 2Ghz, and on Phenom II 3.6Ghz, both with 12 Gb of memory.

5.1 Square Grid Experiments

We first experimented on 8×8 , 16×16 , and 32×32 grids with 10% obstacles while varying the number of agents from 1 up to the number where at least one solver was able to solve an instance (in case of the 8×8 grid this is 20 agents; and 32 and 58 in case of 16×16 and 32×32 grids respectively). For each number of agents 10 random instances were generated.

Figure 7 presents results where each algorithm was given a time limit of 300 seconds (as was done by [18, 5, 16]). The leftmost plot (Plot (a)) shows the *success rate* (=percentage out of given 10 random instances solved within the time limit) as a function of the number of agents for the 8×8 grid (higher curves are better). The next plot (Plot (b)) reports the average runtime for instances that were solved by all algorithms (lower curves are better). Here, we required 100% success rate for all the tested algorithms to be able to calculate average runtime; this is also the reason why the number of agents is smaller. The two right plots visualize the results on 16×16 grid (Plot (c)) and 32×32 grid (Plot (d)) but in a different way. Here, we present the number of instances (out of all instances for all number of agents) that each method solved (*y*-axis) as a function of the elapsed time (*x*-axis). Thus, for example Plot (c) says that MDD-SAT was able to solve 145 instances in time less than 10 seconds (higher curves are better).

The first clear trend is that MDD-SAT significantly outperforms BASIC-SAT in all aspects. This shows the importance of developing enhanced SAT encodings for the MAPF problem. The performance of the BASIC-SAT encoding compared to the search-based algorithm degrades as the size of the grids grow larger: in the 8×8 grids it is

⁶ Due to the knowledge of lower bounds on the sum-of-costs, the number of variables involved in the cardinality constraint is relatively small and hence the different encoding style has not enough room to show its benefit.

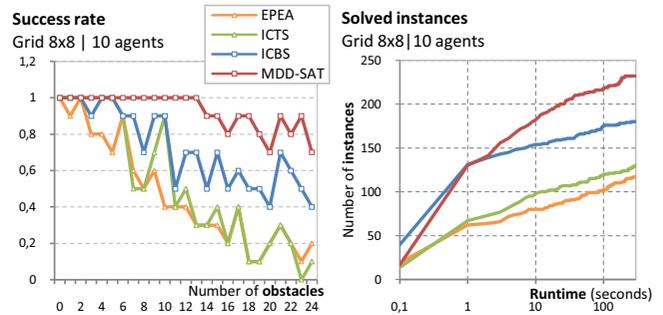


Figure 8. Success rate and runtime on the 8×8 grid with increasing number of obstacles (out of 64 cells).

second only to MDD-SAT, in the 16×16 grid it is comparable to most search-based algorithms, and in the 32×32 grid it is even substantially worse. For the rest of the experiments we did not activate BASIC-SAT.

In addition, a prominent trend observed in all the plots is that MDD-SAT has higher success rate and solves more instances than all other algorithms. In particular, in highly constrained instances (containing many agents) the MDD-SAT solver is the best option.

However, on the 32×32 grid (rightmost figure) for easy instances when the available runtime was less than 10 seconds, MDD-SAT was weaker than the search-based algorithms. This is mostly due to the architecture of the MDD-SAT solver which has an overhead of running the external SAT solver and passing input in the textual form to it. This effect is also seen in the 8×8 plot (Plot (b)) as these were rather easy instances (solved by all algorithms) and the extra overhead of activating the external SAT solver did not pay off.

Next, we varied the number of obstacles for the 8×8 grid with 10 agents to see the impact of shrinking free space and increasing the frequency of interactions among agents. Results are shown in Figure 8. Again, MDD-SAT clearly solves more instances over all settings. MDD-SAT was always faster except for some easy instances (that needed up to 1 second) where ICBS was slightly faster which is again due to the overhead in setup of the SAT solving by an external solver. Interestingly, increasing the number of obstacles reduces the number of open cells. This is an advantage for the SAT formula generator in MDD-SAT as the formula has less variables and constraints. By contrast, the combinatorial difficulty of the instances increases with adding obstacles for all the solvers as it means that the graphs gets denser and harder to solve.

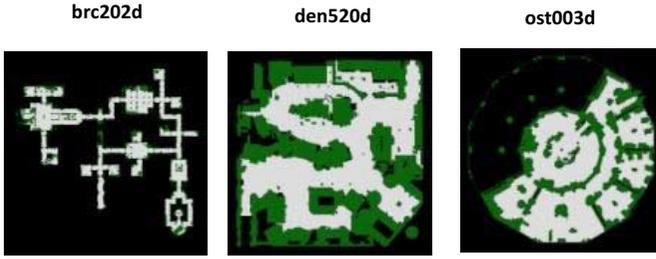


Figure 9. Three structurally diverse Dragon-Age maps used in the experimental evaluation. This selection includes: narrow corridors in `brc202d`, large open space in `den520d`, and open space with almost isolated rooms in `ost003d`.

5.2 Results on the Dragon Age Maps

Next, we experimented on three structurally different Dragon-Age maps - `ost003d`, `den520d`, and `brc202d`, that are commonly used as testbeds [18, 10, 5] - see Figure 9. On these maps we only evaluated the most efficient algorithms, namely, MDD-SAT, ICTS, and ICBS. Generally, in these maps there is a large number of open cells but the graph is sparse with agents but there are topological differences. `brc202d` has many narrow corridors. `ost003d` consists of few open areas interconnected by narrow doors. Finally, `den520d` has wider open areas.

To obtain instances of various difficulties we varied the distance between start and goal locations. Ten random instances were generated for each distance in the range: $\{8, 16, 24, \dots, 320\}$ in order to have instances of different difficulties (total of 400 instances). With larger distances, the problems are more difficult as the probability for interactions (avoidance) among agents increases as they need to travel through a larger part of the graph.

The results for the three Dragon-Age maps are shown in Figure 10 (`brc202d`), Figure 11 (`den520d`), and Figure 12 (`ost003d`). Two setups were used for each map - one with 16 agents, the other with 32 agents. The left plot of each figure shows the number of solved instances (y -axis) as a function of the elapsed time (x -axis). Again, higher curves correspond to better performance). The right plot is interpreted as follows. For each solver the 400 instances are ordered in increasing order of their solution time (this has strong correlation with the distance between the start and goal configurations). Thus, the numbers in the x -axis give the relative location (out of the 400) in this sorted order. The y -axis gives the actual running time for each instance. Here, lower curves correspond to better performance.

All these figures show a similar clear trend with the exception of `ost003d` with 32 agents (discussed below). On the easy instances where little time is required (left of the figures), MDD-SAT is not the best. But, for the harder instances that need more time (right of the figures), MDD-SAT clearly outperform all the other solvers.

Intuitively, one might think that the search-based solvers will have an advantage in these domains since they contain many open spaces (low combinatorial difficulty) while the MDD-SAT approach will suffer here as it will need to generate a large number of formulae (as the domains are large). This might be true for the easy instances. Nevertheless, the effectiveness of MDD-SAT was clearly seen on the harder instances where generating the formulae and the external time to activate the architecture of the SAT solver seemed to pay off. This trend was also seen in the case of small densely occupied grids discussed above.

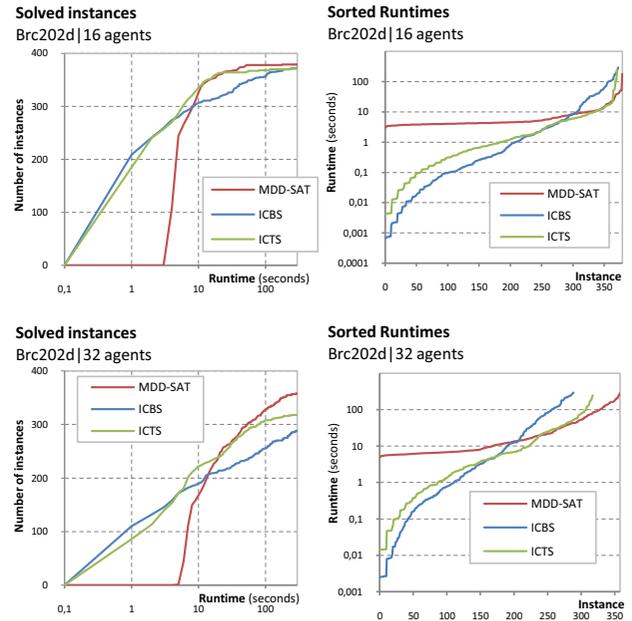


Figure 10. Results for dragon age map `brc202d` with 16 and 32 agents. The left part shows the number of instances (y -axis) a solver manages to solve in the given timeout (x -axis). The right part shows all the runtimes for a given solver sorted in the ascending order.

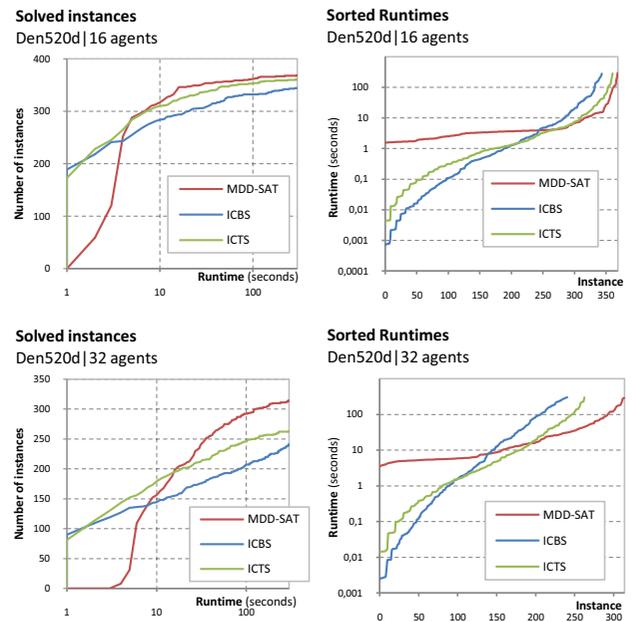


Figure 11. Results for dragon age map `den520d` with 16 and 32 agents. MDD-SAT is the best option on hard instances with more agents.

The `ost003d` map with 32 agents is the only case where MDD-SAT was outperformed by ICTS. This is probably due to the specific structure of `ost003d` which has a number of isolated open spaces. This gives an advantage to ICTS with relatively many agents (32) as conflicts mostly occur at the exits/doors of the open areas. ICTS handles this on a per-agent cost basis while the other solvers are less effective here.

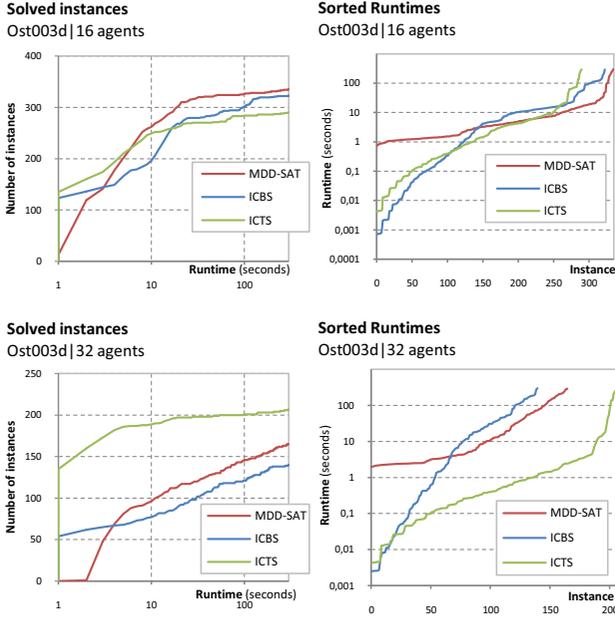


Figure 12. Results for dragon age map *ost003d* with 16 and 32 agents. Although MDD-SAT performs as best with 16 agents, it gets outperformed in the case with 32 agents by ICTS. This case shows that there is no universal winner among the tested algorithms.

The entire set of experiments show a clear trend. For the easy instances when a small amount of time is given the search-based algorithm may be faster. But, given enough time MDD-SAT is the correct choice, even in the large maps where it has an initial disadvantage. One of the reasons for this is modern SAT solvers have the ability to learn and improve their speed during the process of answering a SAT question. But, this learning needs sufficient time and large search trees to be effective. By contrast, search algorithms do not have this advantage.

5.3 Size of the Formulae

Concrete runtimes for 10 instances of *ost003d* are given in Table 2. MDD-SAT solves the hardest instance (#1) while other solvers ran out of time. The right part of the table illustrates the cumulative size of the formulae generated during the solving process. Although the map is much larger than the square grids, the size of formulae is comparable to the densely occupied grid (see Figure 1). This is because ξ_0 is a good lower bound of the optimal cost in the sparse maps.

The observation from this experiments is that the large underlying graph does not necessarily imply generating large Boolean formulae in the MDD-SAT solving process. Though in harder scenarios (where start and goals are far apart) large formulae are eventually generated but still do not represent any significant disadvantage for the MDD-SAT solver according to presented measurements. We observed that generating large formulae takes considerable portion of the total runtime (up to 10%-30%) within the MDD-SAT solver. Hence efficient implementation of this part of the solver has significant impact on the overall performance.

6 Summary and Conclusions

We introduced the first state-of-the-art SAT-based solver for the sum-of-costs variant of MAPF. The resulting enhanced encoding, called MDD-SAT migrates ideas from the search-based methods to use with SAT solvers, was experimentally compared to the state-of-the-art search-based solvers over a variety of domains - we tested 4-connected grids with random obstacles and large maps from computer games. We have seen that MDD-SAT is a better option in hard scenarios while the search-based solvers may perform better in easier cases.

Nevertheless, as previous authors mentioned [17, 5] there is no universal winner and each of the approaches has pros and cons and thus might work best in different circumstances. For example, ICTS was best on *ost003d* with 32 agents. This calls for a deeper study of various classes of MAPF instances and their characteristics and how the different algorithms behave across them. Not too much is known at present to the MAPF community on these aspects.

There are several factors behind the performance of the SAT-based approach: clause learning, constraint propagation, good implementation of the SAT solver. On the other hand, the SAT solver does not understand the structure of the encoded problem which may downgrade the performance. Hence, we consider that implementing techniques such as learning directly into the dedicated MAPF solver may be a future direction. Finally, migrating of other ideas from both classes of approaches might further improve the performance.

MAPF	Ost003d (seconds)			m	MDD-SAT, 16 agents	
	16 agents, distance=168				Distance	Variables
	MDD-SAT	ICBS	ICTS			
1	101.4	N/A	N/A	8	758.0	1 169.7
2	12.8	9.7	2.4	64	34 648.7	120 961.1
3	13.2	4.4	2.4	128	932 440.9	9 128 568.8
4	3.8	0.6	1.2			
5	13.5	9.6	3.2			
6	22.7	10.7	N/A	MDD-SAT, 32 agents		
7	N/A	N/A	N/A	Distance	Variables	Clauses
8	36.9	49.6	2.5	8	2 377.6	3 751.3
9	12.0	2.6	1.4	64	571 915.1	3 672 249.3
10	N/A	N/A	N/A	128	5 163 157.0	49 201 960.0

Table 2. Runtime for 10 instances (left) and the average size of the MDD-SAT formulae for *ost003d* (right)

7 Acknowledgements

This research was supported by the Israel Ministry of Science and the Czech Ministry of Education Youth and Sports through a joint grant given to Roni Stern, Ariel Felner and Pavel Surynek.

REFERENCES

- [1] G. Audemard, J. Lagniez, and L. Simon, ‘Improving glucose for incremental SAT solving with assumptions: Application to MUS extraction’, in *Theory and Applications of Satisfiability Testing - SAT 2013*, pp. 309–317, (2013).
- [2] G. Audemard and L. Simon, ‘Predicting learnt clauses quality in modern SAT solvers’, in *IJCAI*, pp. 399–404, (2009).
- [3] O. Bailleux and Y. Boufkhad, ‘Efficient CNF encoding of boolean cardinality constraints’, in *CP*, pp. 108–122, (2003).
- [4] A. Biere, A. Biere, M. Heule, H. van Maaren, and T. Walsh, *Handbook of Satisfiability: Volume 185 Frontiers in Artificial Intelligence and Applications*, IOS Press, 2009.

- [5] E. Boyarski, A. Felner, R. Stern, G. Sharon, D. Tolpin, O. Betzalel, and S. Shimony, 'ICBS: improved conflict-based search algorithm for multi-agent pathfinding', in *IJCAI*, pp. 740–746, (2015).
- [6] L. Cohen, T. Uras, and S. Koenig, 'Feasibility study: Using highways for bounded-suboptimal mapf', in *SOCS*, pp. 2–8, (2015).
- [7] B. de Wilde, A. ter Mors, and C. Witteveen, 'Push and rotate: a complete multi-agent pathfinding algorithm', *JAIR*, **51**, 443–492, (2014).
- [8] K. Dresner and P. Stone, 'A multiagent approach to autonomous intersection management', *JAIR*, **31**, 591–656, (2008).
- [9] E. Erdem, D. G. Kisa, U. Oztok, and P. Schueller, 'A general formal framework for pathfinding problems with multiple agents', in *AAAI*, (2013).
- [10] M. Goldenberg, A. Felner, R. Stern, G. Sharon, N. Sturtevant, R. Holte, and J. Schaeffer, 'Enhanced partial expansion A*', *JAIR*, **50**, 141–187, (2014).
- [11] M. Järvisalo, D. Le Berre, O. Roussel, and L. Simon, 'The international SAT solver competitions', *AI Magazine*, **33**(1), (2012).
- [12] M. M. Khorshid, R. C. Holte, and N. R. Sturtevant, 'A polynomial-time algorithm for non-optimal multi-agent pathfinding', in *Symposium on Combinatorial Search (SOCS)*, (2011).
- [13] Justyna Petke, *Bridging Constraint Satisfaction and Boolean Satisfiability*, Artificial Intelligence: Foundations, Theory, and Algorithms, Springer, 2015.
- [14] G. Röger and M. Helmert, 'Non-optimal multi-agent pathfinding is solved (since 1984).', in *SOCS*, (2012).
- [15] M. Ryan, 'Constraint-based multi-robot path planning', in *ICRA*, pp. 922–928, (2010).
- [16] G. Sharon, R. Stern, A. Felner, and N. Sturtevant, 'Conflict-based search for optimal multi-agent pathfinding', *Artif. Intell.*, **219**, 40–66, (2015).
- [17] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant, 'Conflict-based search for optimal multi-agent pathfinding', *Artif. Intell.*, **219**, 40–66, (2015).
- [18] G. Sharon, R. Stern, M. Goldenberg, and A. Felner, 'The increasing cost tree search for optimal multi-agent pathfinding', *Artificial Intelligence*, **195**, 470–495, (2013).
- [19] J. Silva and I. Lynce, 'Towards robust CNF encodings of cardinality constraints', in *CP*, pp. 483–497, (2007).
- [20] D. Silver, 'Cooperative pathfinding', in *AIIDE*, pp. 117–122, (2005).
- [21] C. Sinz, 'Towards an optimal CNF encoding of boolean cardinality constraints', in *CP*, pp. 827–831, (2005).
- [22] A. Srinivasan, T. Ham, S. Malik, and R. Brayton, 'Algorithms for discrete function manipulation', in *ICCAD*, pp. 92–95, (1990).
- [23] T. Standley, 'Finding optimal solutions to cooperative pathfinding problems.', in *AAAI*, pp. 173–178, (2010).
- [24] Nathan R. Sturtevant, 'Benchmarks for grid-based pathfinding', *Computational Intelligence and AI in Games*, **4**(2), 144–148, (2012).
- [25] P. Surynek, 'An optimization variant of multi-robot path planning is intractable', in *AAAI*, (2010).
- [26] P. Surynek, 'Towards optimal cooperative path planning in hard setups through satisfiability solving', in *PRICAI*, 564–576, (2012).
- [27] P. Surynek, 'Compact representations of cooperative path-finding as SAT based on matchings in bipartite graphs', in *ICTAI*, pp. 875–882, (2014).
- [28] P. Surynek, 'A simple approach to solving cooperative path-finding as propositional satisfiability works well', in *PRICAI*, pp. 827–833, (2014).
- [29] P. Surynek, 'Simple direct propositional encoding of cooperative path finding simplified yet more', in *MICAI*, pp. 410–425, (2014).
- [30] P. Surynek, 'Reduced time-expansion graphs and goal decomposition for solving cooperative path finding sub-optimally', in *IJCAI*, pp. 1916–1922, (2015).
- [31] G. Wagner and H. Choset, 'Subdimensional expansion for multirobot path planning', *Artif. Intell.*, **219**, 1–24, (2015).
- [32] K. Wang and A. Botea, 'MAPP: a scalable multi-agent path planning algorithm with tractability and completeness guarantees', *JAIR*, **42**, 55–90, (2011).
- [33] J. Yu and S. LaValle, 'Planning optimal paths for multiple robots on graphs', in *ICRA*, pp. 3612–3617, (2013).
- [34] J. Yu and S. M. LaValle, 'Structure and intractability of optimal multi-robot path planning on graphs', in *AAAI*, (2013).

Fixed-Domain Reasoning for Description Logics

Sarah Gaggl and Sebastian Rudolph and Lukas Schweizer¹

Abstract. After decades of fruitful research, description logics (DLs) have evolved into a de facto standard in logic-based knowledge representation. In particular, they serve as the formal basis of the standardized and very popular web ontology language (OWL), which also comes with the advantage of readily available user-friendly modeling tools and optimized reasoning engines. In the course of the wide-spread adoption of OWL and DLs, situations have been observed where logically less skilled practitioners are (ab)using these formalisms as constraint languages adopting a closed-world assumption, contrary to the open-world semantics imposed by the classical definitions and the standards. To provide a clear theoretical basis and inferencing support for this often practically reasonable “off-label use” we propose an alternative formal semantics reflecting the intuitive understanding of such scenarios. To that end, we introduce the *fixed-domain semantics* and argue that this semantics gives rise to an interesting new inferencing task: *model enumeration*. We describe how the new semantics can be axiomatized in very expressive DLs. We thoroughly investigate the complexities for standard reasoning as well as query answering under the fixed-domain semantics for a wide range of DLs. Further, we present an implementation of a fixed-domain DL reasoner based on a translation into answer set programming (ASP) which is competitive with alternative approaches for standard reasoning tasks and provides the added functionality of model enumeration.

1 Introduction

IT practitioners facing the task of developing logic-based specifications tend to prefer knowledge representation formalisms which are standardized, widely adopted and that come with elaborate modeling tool support. A notable example for this is the Web Ontology Language OWL [32]. Ontology editors like Protégé [16] provide user-friendly interfaces and intuitive access to a complex and involved formalism.

This leads to situations where OWL is chosen over other formalisms, even if the application scenario does not match the typical usage of this language. For example, problems of a constraint-satisfaction type do not go well with OWL’s standard semantics allowing for models of arbitrary size. Consider graph 3-coloring as a short but representative constraint-satisfaction problem. It is easy to create some OWL axioms specifying the conditions on valid colorings of a given graph. Asking for colorability as such can then be cast into an OWL

consistency problem, a natural task for OWL reasoners. Yet, generating (all) concrete colorings already requires capabilities that an OWL reasoner cannot provide out of the box, namely that of extracting or enumerating models.

To overcome these shortcomings, we propose *fixed-domain reasoning* for DLs – a family of logics providing the logical underpinning of OWL and its sublanguages. This intuitive and simple approach considers DLs under a non-standard model-theoretic semantics, modifying the modelhood condition by restricting the domain to an explicitly given, fixed finite set. We investigate the combined complexity of reasoning in the presence of a given fixed domain for a wide range of description logics, for which we establish tight bounds for standard reasoning tasks as well as query answering for various query notions. While satisfiability checking in OWL under the classical semantics is N2EXPTIME-complete [15] and query answering is not even known to be decidable, we show that these problems under the *fixed-domain semantics* are merely NP-complete and Π_2^P -complete, respectively.

We note that the fixed-domain condition can be axiomatized in OWL. Still, employing the axiomatization and using available OWL reasoners would not allow for the non-standard reasoning task of model enumeration. Therefore, we propose a different approach and define a translation of *SROIQ* knowledge bases (the logical counterparts to OWL ontologies) into answer set programming (ASP, [4]), such that there is a one-to-one correspondence between the fixed-domain models of the considered knowledge base and the set of answer-sets of the obtained program. This allows us to use existing ASP solvers (see [5] for an overview) for fixed-domain reasoning – including standard as well as non-standard tasks. For the proposed translation, we provide an implementation and present preliminary evaluations on typical constraint-satisfaction-type problems. This not only demonstrates feasibility, but also suggests significant improvement compared to the axiomatized approach using highly optimized OWL reasoners.

2 Preliminaries

In this section, we provide the necessary background of description logics, Datalog queries, and answer-set programming. Despite obvious structural commonalities of Datalog queries and ASP, we introduce and treat them separately, since their purpose is different: the former is used as a query formalism over DL knowledge bases while the latter serves as the target formalism for a compilation process.

¹ Technische Universität Dresden, Computational Logic Group, e-mail: firstname.lastname@tu-dresden.de

Description Logics OWL 2 DL, the version of the Web Ontology Language we focus on, is defined based on description logics (DLs, [2, 26]). We briefly recap the description logic *SRQIQ* (for details see [13]). Let N_I , N_C , and N_R be finite, disjoint sets called *individual names*, *concept names* and *role names* respectively. These atomic entities can be used to form complex ones as displayed in Table 1.

A *SRQIQ knowledge base* \mathcal{K} is a tuple $(\mathcal{A}, \mathcal{T}, \mathcal{R})$ where \mathcal{A} is a *SRQIQ* ABox, \mathcal{T} is a *SRQIQ* TBox and \mathcal{R} is a *SRQIQ* RBox. Table 2 presents the respective axiom types available in the three parts.² We use $N_I(\mathcal{K})$, $N_C(\mathcal{K})$, and $N_R(\mathcal{K})$ to denote the sets of individual names, concept names, and role names occurring in \mathcal{K} , respectively.

Table 1. Syntax and semantics of role and concept constructors in *SRQIQ*, where a_1, \dots, a_n denote individual names, s a role name, r a role expression and C and D concept expressions.

Name	Syntax	Semantics
inverse role	s^-	$\{(x, y) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (y, x) \in s^{\mathcal{I}}\}$
universal role	u	$\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
top	\top	$\Delta^{\mathcal{I}}$
bottom	\perp	\emptyset
negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
nominals	$\{a_1, \dots, a_n\}$	$\{a_1^{\mathcal{I}}, \dots, a_n^{\mathcal{I}}\}$
univ. restriction	$\forall r.C$	$\{x \mid \forall y.(x, y) \in r^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$
exist. restriction	$\exists r.C$	$\{x \mid \exists y.(x, y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
Self concept	$\exists r.Self$	$\{x \mid (x, x) \in r^{\mathcal{I}}\}$
qualified number	$\leq n r.C$	$\{x \mid \#\{y \in C^{\mathcal{I}} \mid (x, y) \in r^{\mathcal{I}}\} \leq n\}$
restriction	$\geq n r.C$	$\{x \mid \#\{y \in C^{\mathcal{I}} \mid (x, y) \in r^{\mathcal{I}}\} \geq n\}$

Table 2. Syntax and semantics of *SRQIQ* axioms.

Axiom α	$\mathcal{I} \models \alpha$, if	
$r_1 \circ \dots \circ r_n \sqsubseteq r$	$r_1^{\mathcal{I}} \circ \dots \circ r_n^{\mathcal{I}} \subseteq r^{\mathcal{I}}$	RBox \mathcal{R}
$\text{Dis}(s, r)$	$s^{\mathcal{I}} \cap r^{\mathcal{I}} = \emptyset$	
$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$	TBox \mathcal{T}
$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$	ABox \mathcal{A}
$r(a, b)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$	
$a \doteq b$	$a^{\mathcal{I}} = b^{\mathcal{I}}$	
$a \neq b$	$a^{\mathcal{I}} \neq b^{\mathcal{I}}$	

The semantics of *SRQIQ* is defined via interpretations $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ composed of a non-empty set $\Delta^{\mathcal{I}}$ called the *domain* of \mathcal{I} and a function $\cdot^{\mathcal{I}}$ mapping individual names to elements of $\Delta^{\mathcal{I}}$, concept names to subsets of $\Delta^{\mathcal{I}}$, and role names to subsets of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. This mapping is extended to complex role and concept expressions (cf. Table 1) and finally used to define satisfaction of axioms (see Table 2). We say that \mathcal{I} *satisfies* a knowledge base $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ (or \mathcal{I} is a *model* of \mathcal{K} , written: $\mathcal{I} \models \mathcal{K}$) if it satisfies all axioms of \mathcal{A} , \mathcal{T} , and \mathcal{R} . We say that a knowledge base \mathcal{K} *entails* an axiom α (written $\mathcal{K} \models \alpha$) if all models of \mathcal{K} are models of α .

² The original definition of *SRQIQ* contained more RBox axioms (expressing transitivity, (a)symmetry, (ir)reflexivity of roles), but these can be shown to be syntactic sugar. Moreover, the definition of *SRQIQ* contains so-called *global restrictions* which prevents certain axioms from occurring together. These complicated restrictions, while crucial for the decidability of classical reasoning in *SRQIQ* are not necessary for fixed-domain reasoning considered here, hence we omit them for the sake of brevity.

Boolean Datalog Queries Here we briefly introduce syntax and semantics of Datalog queries over description logic knowledge bases. A *term* can be a variable from a countably infinite set V of variables, or an element of N_I . An *atom* has the form $p(t_1, \dots, t_n)$ where t_1, \dots, t_n are terms and p is a predicate of arity n from a set Π of predicates containing N_C (arity 1) and N_R (arity 2) and containing a special predicate *goal* of arity 0. A *Boolean Datalog query* is a set of first-order logic Horn rules of the form $\forall X.a_1 \wedge \dots \wedge a_k \rightarrow a$ where a_1, \dots, a_n, a are atoms, but the predicate of a is not from N_C or N_R . $X \subseteq V$ denotes the set of variables occurring in the atoms. Given a DL interpretation \mathcal{I} , and a Boolean Datalog query Q , an *extended model* for \mathcal{I} and Q is a first-order interpretation \mathcal{J} over $\Delta^{\mathcal{I}}$ that coincides with \mathcal{I} on the interpretation of N_I , N_C , and N_R and satisfies all the rules from Q . We say that Q *matches* \mathcal{I} and write $\mathcal{I} \models Q$ if $\mathcal{J} \models \text{goal}$ for every extended model \mathcal{J} for \mathcal{I} and Q . For a DL knowledge base \mathcal{K} , we say \mathcal{K} *entails* Q iff $\mathcal{I} \models Q$ for every model \mathcal{I} of \mathcal{K} . *Bounded arity Datalog queries* are classes of queries where the arity of the used predicates is bounded by some constant. A *Boolean conjunctive query* is a Boolean Datalog query with just one rule where a_1, \dots, a_n use only predicates from $N_C \cup N_R$ and $a = \text{goal}$. In that case, such a query can be equivalently written as the first-order formula $\exists X.a_1 \wedge \dots \wedge a_k$.

Answer-Set Programming We review the basic notions of answer set programming [22] under the stable model semantics [11], for further details we refer to [4, 8].

We fix a countable set \mathcal{U} of (*domain*) *elements*, also called *constants*; and suppose a total order $<$ over the domain elements. An *atom* is an expression $p(t_1, \dots, t_n)$, where p is a *predicate* of arity $n \geq 0$ and each t_i is either a variable or an element from \mathcal{U} . An atom is *ground* if it is free of variables. $B_{\mathcal{U}}$ denotes the set of all ground atoms over \mathcal{U} . A (*normal*) *rule* ρ is of the form

$$a \leftarrow b_1, \dots, b_k, \text{ not } b_{k+1}, \dots, \text{ not } b_m.$$

with $m \geq k \geq 0$, where a, b_1, \dots, b_m are atoms, and “*not*” denotes *default negation*. The *head* of ρ is the singleton set $H(\rho) = \{a\}$ and the *body* of ρ is $B(\rho) = \{b_1, \dots, b_k, \text{ not } b_{k+1}, \dots, \text{ not } b_m\}$. Furthermore, $B^+(\rho) = \{b_1, \dots, b_k\}$ and $B^-(\rho) = \{b_{k+1}, \dots, b_m\}$. A rule ρ is *safe* if each variable in ρ occurs in $B^+(\rho)$. A rule ρ is *ground* if no variable occurs in ρ . A *fact* is a ground rule with empty body. An (*input*) *database* is a set of facts. A (*normal*) *program* is a finite set of normal rules. For a program Π and an input database D , we often write $\Pi(D)$ instead of $D \cup \Pi$. For any program Π , let U_{Π} be the set of all constants appearing in Π . $Gr(\Pi)$ is the set of rules $\rho\sigma$ obtained by applying, to each rule $\rho \in \Pi$, all possible substitutions σ from the variables in ρ to elements of U_{Π} .

An *interpretation* $I \subseteq B_{\mathcal{U}}$ *satisfies* a ground rule ρ iff $H(\rho) \cap I \neq \emptyset$ whenever $B^+(\rho) \subseteq I$, $B^-(\rho) \cap I = \emptyset$. I *satisfies* a ground program Π , if each $\rho \in \Pi$ is satisfied by I . A non-ground rule ρ (resp., a program Π) is satisfied by an interpretation I iff I satisfies all groundings of ρ (resp., $Gr(\Pi)$). $I \subseteq B_{\mathcal{U}}$ is an *answer set* (also called *stable model*) of Π iff it is the subset-minimal set satisfying the *Gelfond-Lifschitz reduct* $\Pi^I = \{H(\rho) \leftarrow B^+(\rho) \mid I \cap B^-(\rho) = \emptyset, \rho \in Gr(\Pi)\}$. For a program Π , we denote the set of its answer sets by $\mathcal{AS}(\Pi)$.

We make use of further syntactic extensions, namely integrity constraints and count expressions, which both can be recast to ordinary normal rules as described in [8]. An *integrity constraint* is a rule ρ where $H(\rho) = \emptyset$, intuitively representing an undesirable situation; i.e. it has to be avoided that $B(\rho)$ evaluates positively. Count expressions are of the form $\#count\{l : l_1, \dots, l_i\} \bowtie u$, where l is an atom and $l_j = p_j$ or $l_j = not\ p_j$, for p_j an atom, $1 \leq j \leq i$, u a non-negative integer, and $\bowtie \in \{\leq, <, =, >, \geq\}$. The expression $\{l : l_1, \dots, l_n\}$ denotes the set of all ground instantiations of l , governed through $\{l_1, \dots, l_n\}$. We restrict the occurrence of count expressions in a rule ρ to $B^+(\rho)$ only. Intuitively, an interpretation satisfies a count expression, if $N \bowtie u$ holds, where N is the cardinality of the set of ground instantiations of l , $N = |\{l \mid l_1, \dots, l_n\}|$, for $\bowtie \in \{\leq, <, =, >, \geq\}$ and u a non-negative integer.

3 Models over Fixed Domains

In DLs, models can be of arbitrary cardinality. In many applications, however, the domain of interest is known to be finite. In fact, restricting reasoning to models of finite domain size (called *finite model reasoning*, a natural assumption in database theory), has already become the focus of intense studies in DLs [17, 6, 25, 27]. As opposed to assuming the domain to be merely finite (but of arbitrary, unknown size), we consider the case where the domain has an *a priori known cardinality* and use the term *fixed domain*. We refer to such models as *fixed-domain models* and argue that in many applications, this modification of the standard DL semantics represents a more intuitive definition of what is considered and expected as a *model* of some knowledge base.

Definition 1 (Fixed-Domain Semantics) *Given a non-empty finite set $\Delta \subseteq N_I$, called fixed domain, an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is said to be Δ -fixed (or just fixed, if Δ is clear from the context), if $\Delta^{\mathcal{I}} = \Delta$ and $a^{\mathcal{I}} = a$ for all $a \in \Delta$. Accordingly, for a DL knowledge base \mathcal{K} , we call an interpretation \mathcal{I} a Δ -model of \mathcal{K} , if \mathcal{I} is a Δ -fixed interpretation and $\mathcal{I} \models \mathcal{K}$. A knowledge base \mathcal{K} is called Δ -satisfiable if it has a Δ -model. We say \mathcal{K} Δ -entails an axiom α ($\mathcal{K} \models_{\Delta} \alpha$) if every Δ -model of \mathcal{K} is also a model of α .*

We briefly demonstrate the effects of the fixed-domain semantics as opposed to the finite-model semantics (with entailment \models_{fin}) and the classical semantics.

Example 2 *Let $\mathcal{K} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$ and $\Delta = \{a, b\}$ with $\mathcal{A} = \{A(a), A(b), s(a, b)\}$, $\mathcal{T} = \{\top \sqsubseteq \exists r.B, \top \sqsubseteq \leq 1 r^{-}.\top\}$, and $\mathcal{R} = \{\text{Dis}(s, r)\}$. First we note that \mathcal{K} has a Δ -model \mathcal{I} representable as $\mathcal{A}_{\mathcal{I}} = \{A(a), A(b), B(a), B(b), s(a, b), r(a, a), r(b, b)\}$, thus \mathcal{K} is satisfiable under all three semantics. Then $\alpha = \top \sqsubseteq \exists r.\exists r.B$ holds in all models of \mathcal{K} , therefore $\mathcal{K} \models \alpha$, $\mathcal{K} \models_{\text{fin}} \alpha$, and $\mathcal{K} \models_{\Delta} \alpha$. Opposed to this, $\beta = \top \sqsubseteq B$ merely holds in all finite models, whence $\mathcal{K} \models_{\text{fin}} \beta$ and $\mathcal{K} \models_{\Delta} \beta$, but $\mathcal{K} \not\models \beta$. Finally, $\gamma = \top \sqsubseteq \exists r.\text{Self}$ only holds in all Δ -models, thus $\mathcal{K} \models_{\Delta} \gamma$, but $\mathcal{K} \not\models_{\text{fin}} \gamma$ and $\mathcal{K} \not\models \gamma$.*

Extraction & Enumeration of Δ -Models When performing knowledge base satisfiability checking in DLs (the primary reasoning task usually considered), a model constructed

by a reasoner merely serves as witness to claim satisfiability, rather than as an accessible artifact. However, as mentioned before, our approach aims at scenarios where a knowledge base is a formal problem description for which each model represents one solution; in particular the domain is part of the problem description and hence fixed a-priori. Then, retrieval of one, several, or all models is a natural task, as opposed to merely checking model existence. With *model extraction* we denote the task of materializing an identified model in order to be able to work with it, i.e. to inspect it in full detail and reuse it in downstream processes. The natural extension of model extraction is to make *all* models explicit, performing *model enumeration*. Most existing DL reasoning algorithms attempt to successively construct a model representation of a given knowledge base. However, most of the existing tableaux reasoners do not reveal the constructed model, besides the fact that models might end up being infinite such that an explicit representation is impossible. Regarding enumeration, we state that this task is not supported – not even implicitly – by any state-of-the-art DL reasoner, also due to the reason that in the standard case, the number of models is typically infinite and often even uncountable. We will use the notions of model extraction and enumeration in the way described above. Note that a related task, called *model expansion*, is used in the general first-order case, e.g. in the work of Mitchell and Ternovska [18]. There, an initial (partial) interpretation representing a problem instance is expanded to ultimately find a model of the given theory.

Example 3 *We consider the 3-coloring problem for an undirected graph $G = (V, E)$, encoded in $\mathcal{K}_{3\text{col}} = (\mathcal{A}, \mathcal{T}, \mathcal{R})$, with $\mathcal{T} = \{N \sqsubseteq N_r \sqcup N_g \sqcup N_b, N_r \sqsubseteq \forall \text{edge}.(N_g \sqcup N_b), N_g \sqsubseteq \forall \text{edge}.(N_b \sqcup N_r), N_b \sqsubseteq \forall \text{edge}.(N_r \sqcup N_g), N_r \sqsubseteq \neg N_g, N_r \sqsubseteq \neg N_b, N_g \sqsubseteq \neg N_b\}$, $\mathcal{A} = \{N(v) \mid v \in V\} \cup \{\text{edge}(v, v') \mid \forall \{v, v'\} \in E\} \cup \{\neg \text{edge}(v, v') \mid \{v, v'\} \in 2^V \setminus E\}$, and $\mathcal{R} = \emptyset$. Let $\Delta = V$ be the imposed fixed domain. It is not hard to see that there is a one-to-one correspondence between the Δ -models of $\mathcal{K}_{3\text{col}}$ and the colorings of G .*

Axiomatization of Δ -Models When introducing a new semantics for some logic, it is worthwhile to ask if existing reasoners can be used. Indeed, it is easy to see that, assuming $\Delta = \{a_1, \dots, a_n\}$, adding the GCI $\top \sqsubseteq \{a_1, \dots, a_n\}$ as well as the set of inequality axioms containing $a_i \neq a_j$ with $i < j$ to \mathcal{K} will (up to isomorphism) rule out all models of \mathcal{K} , not having Δ as their domain. Denoting these additional axioms with \mathcal{FD}_{Δ} , we find that \mathcal{K} is Δ -satisfiable iff $\mathcal{K} \cup \mathcal{FD}_{\Delta}$ is satisfiable under the classical DL semantics and, likewise, $\mathcal{K} \models_{\Delta} \alpha$ iff $\mathcal{K} \cup \mathcal{FD}_{\Delta} \models \alpha$ for any axiom α . Consequently, any off-the-shelf *SROIQ* reasoner can be used for fixed-domain reasoning, at least when it comes to the classical reasoning tasks.

However, the fact that the currently available DL reasoners are not optimized towards reasoning with axioms of the prescribed type (featuring disjunctions over potentially large sets of individuals) and that available reasoners do not support model extraction and model enumeration led us to develop an alternative computational approach based on ASP.

4 Complexity Analysis

In this section we investigate complexities of classical reasoning tasks under the fixed-domain semantics. Note that, next

to the size of the considered knowledge base (and - in the case of query entailment - the size of the query) the size of the domain $|\Delta|$ contributes to the input size of the reasoning problems considered.

Standard Reasoning The combined complexity of standard reasoning in *SRIOQ* is known to be N2EXPTIME-complete, both for arbitrary models and finite models [15]. Restricting to fixed domains leads to a drastic drop in complexity. Contrarily, imposing fixed domains on (allegedly) inexpressive fragments such as *DL-Lite_{core}*, turns reasoning into a harder problem.

Let DL_{\min} be a minimalistic description logic that merely allows TBox axioms of the form $A \sqsubseteq \neg B$, with $A, B \in \mathbf{N}_C$. Moreover, only atomic assertions of the form $A(a)$ and $r(a, b)$ are admitted. We first demonstrate that satisfiability checking in DL_{\min} is NP-hard, allowing us to bequeath hardness up to more expressive DLs such as *SRIOQ*. Subsequently, we demonstrate that fixed-domain satisfiability checking in *SRIOQ* is in NP, thus obtaining NP-completeness for all languages between DL_{\min} and *SRIOQ*.

Lemma 4 *The combined complexity of checking fixed-domain satisfiability of a DL_{\min} knowledge base $\mathcal{K} = (\mathcal{A}, \mathcal{T})$ is NP-hard.*

Proof (Sketch) We obtain hardness by a reduction of the 3-colorability problem. Let $G = (V, E)$ be the input graph. Then, for each node $v_i \in V = \{v_1, \dots, v_n\}$ we introduce a concept name V_i , and encode the edges as disjointness axioms, such that $\mathcal{T} = \{V_i \sqsubseteq \neg V_j \mid \{v_i, v_j\} \in E\}$. The ABox \mathcal{A} consists of the assertions $V_i(a_i)$ for each $V_i \in \{V_1, \dots, V_n\}$. Now let $\Delta = \{r, g, b\}$, such that under any Δ -fixed interpretation \mathcal{I} , necessarily $a_i^{\mathcal{I}} \in \{r, g, b\}$, $1 \leq i \leq n$. Consequently, G has a 3-coloring, iff $\mathcal{K} = (\mathcal{A}, \mathcal{T})$ is Δ -satisfiable. The reduction is linear in the size of G . \square

Lemma 5 *The combined complexity of checking fixed-domain satisfiability of *SRIOQ* knowledge bases is in NP.*

Proof (Sketch) Let \mathcal{K} be a *SRIOQ* knowledge base and Δ be the fixed domain. To show membership, we note that after guessing a Δ -fixed interpretation \mathcal{I} , modelhood can be checked in polynomial time. For this we let \mathcal{C} contain all the concept expressions occurring in \mathcal{K} (including subexpressions). Furthermore, let \mathcal{R} contain all role expressions and role chains (including subchains) occurring in \mathcal{K} . Obviously, \mathcal{C} and \mathcal{R} are of polynomial size. Then, in a bottom-up fashion, we can compute the extension $C^{\mathcal{I}}$ of every element C of \mathcal{C} and the extension $r^{\mathcal{I}}$ of every element r of \mathcal{R} along the defined semantics. Obviously, each such computation step requires only polynomial time. Finally, based on the computed extensions, every axiom of \mathcal{K} can be checked – again in polynomial time. \square

Combining these propositions yields the following theorem.

Theorem 6 *Fixed-domain satisfiability checking in any language between DL_{\min} and *SRIOQ* is NP-complete.*

Note that this finding contrasts with the observation that fixed-domain reasoning in first-order logic is PSPACE-complete. We omit the full proof here, just noting that membership and hardness can be easily shown based on the fact that checking modelhood in FOL is known to be PSPACE-complete [31] and, for the membership part, keeping in mind that $\text{NPSpace} = \text{PSPACE}$ thanks to Savitch's Theorem [28]. This emphasizes the fact that, while the fixed-domain restriction turns reasoning in FOL decidable, restricting to *SRIOQ* still gives a further advantage in terms of complexity (assuming $\text{NP} \neq \text{PSPACE}$).

Query Entailment We next consider the complexity of query entailment for DLs. Again, we will notice a very uniform behavior over a wide range of DLs and query types. We will start by showing a hardness result for a very minimalistic setting.

Lemma 7 *The combined complexity of fixed-domain entailment of conjunctive queries from a DL_{\min} knowledge base is Π_2^P -hard.*

Proof We show hardness by providing a polynomial reduction from evaluation of quantified Boolean formulae of the form $\Phi = \forall p_1, \dots, p_\ell \exists q_1, \dots, q_m \varphi$ such that φ is a Boolean formula where the propositional symbols are from the set $\{p_1, \dots, p_\ell, q_1, \dots, q_m\}$. Note that w.l.o.g. we can assume Φ to be in conjunctive normal form, i.e. it has the shape $\bigvee L_1 \wedge \dots \wedge \bigvee L_n$ where the L_i are sets of negated or un-negated propositional symbols.

Given such a formula Φ , we now construct a DL_{\min} knowledge base \mathcal{K} , a domain Δ , and a conjunctive query Q (all of polynomial size) such that $\mathcal{K} \Delta_{\mathcal{K}}$ -entails Q if and only if Φ evaluates to true. We let Δ consist of elements d_t^{true} and d_t^{false} for all $t \in \{p_1, \dots, p_\ell, q_1, \dots, q_m\}$, and \mathcal{K} consist of the axioms:

- $\text{InClause}_L(d_t^{\text{true}})$ whenever $t \in L$ and $\text{InClause}_L(d_t^{\text{false}})$ whenever $\neg t \in L$
- $\text{compatible}(d_t^{\text{true}}, d_u^{\text{true}})$ and $\text{compatible}(d_t^{\text{false}}, d_u^{\text{false}})$ for all $\{t, u\} \subseteq \{p_1, \dots, p_\ell, q_1, \dots, q_m\}$
- $\text{compatible}(d_t^{\text{false}}, d_u^{\text{true}})$ and $\text{compatible}(d_t^{\text{true}}, d_u^{\text{false}})$ for all $\{t, u\} \subseteq \{p_1, \dots, p_\ell, q_1, \dots, q_m\}$ with $t \neq u$
- $\text{Select}(d_t)$, $C_t(d_t)$ for all $t \in \{p_1, \dots, p_\ell\}$
- $\text{Select}(d_t^{\text{true}})$ and $\text{Select}(d_t^{\text{false}})$ for all $t \in \{q_1, \dots, q_m\}$
- $C_t(d_t^{\text{true}})$, $C_t(d_t^{\text{false}})$ for all $t \in \{p_1, \dots, p_\ell, q_1, \dots, q_m\}$
- $C_t \sqcap C_u \sqsubseteq \perp$ for all $\{t, u\} \in \{p_1, \dots, p_\ell, q_1, \dots, q_m\}$ with $t \neq u$

Finally, we let Q be the conjunctive query using the variables x_{L_1}, \dots, x_{L_n} and consisting of the atoms $\text{InClause}_L(x_L)$, $\text{Select}(x_L)$ for all $L \in \{L_1, \dots, L_n\}$ as well as $\text{compatible}(x_L, x_{L'})$ for all $\{L, L'\} \in \{L_1, \dots, L_n\}$.

We now sketch the argument why the above claimed correspondence holds. By construction, the minimal $\Delta_{\mathcal{K}}$ -models \mathcal{I} for every $i \in \{1, \dots, m\}$ are exactly those where (next to the explicitly stated concept and role memberships) either $d_{p_i}^{\text{true}} \in \text{Select}^{\mathcal{I}}$ or $d_{p_i}^{\text{false}} \in \text{Select}^{\mathcal{I}}$ holds. Consequently Q is entailed, iff for each of these models (representing all possible truth assignments to p_1, \dots, p_ℓ), one literal from every clause

L_i can be selected such that (a) this selection is consistent (i.e., no contradicting literals are selected) and (b) whenever a literal w.r.t. p_1, \dots, p_ℓ is selected, it must be the one corresponding with the model's predefined truth assignment for these propositional symbols. However, this is the case exactly if Φ is valid. \square

We continue by showing that even for very expressive DLs and query languages, query entailment under the fixed domain semantics is still in the second level of the polynomial hierarchy.

Lemma 8 *The combined complexity of the fixed-domain entailment of bounded-arity Datalog queries from a $SR\mathcal{OIQ}$ knowledge base is in Π_2^P .*

Proof Satisfaction of a bounded-arity Datalog query in a database (or finite interpretation) is in NP: there are only polynomially many ground atoms that can be derived, hence, whenever the query is entailed, there is a ground proof tree of polynomial size which can be verified in polynomial time. Consequently, fixed-domain non-entailment of such a query Q from a $SR\mathcal{OIQ}$ knowledge base \mathcal{K} can be realized by (a) guessing an interpretation \mathcal{I} (b) verifying $\mathcal{I} \models \mathcal{K}$ in polynomial time (cf. the proof of Lemma 5) and (c) using an NP oracle to verify $\mathcal{I} \not\models Q$. Consequently, checking fixed-domain entailment is in $\text{coNP}^{\text{NP}} = \Pi_2^P$. \square

Bounded-arity Datalog queries over DLs are rather expressive, they subsume many of the prominent query classes in knowledge representation and databases, including (unions of) conjunctive queries, positive queries, (unions of) conjunctive 2-way regular path queries [7], positive 2-way regular path queries, (unions of) conjunctive nested 2-way regular path queries [3] and regular queries as defined in [24]. Combining the two propositions, we obtain the following theorem.

Theorem 9 *For any class of queries subsuming conjunctive queries and subsumed by bounded-arity Datalog queries and any DL subsuming DL_{\min} and subsumed by $SR\mathcal{OIQ}$, the combined complexity of fixed-domain query entailment is Π_2^P -complete.*

5 Practical Fixed-Domain Reasoning

In Section 3 we already claimed that available reasoners perform poorly on knowledge bases when axiomatizing the fixed-domain semantics, and we support this statement with an evaluation in the sequel (cf. Section 5.2). Thus, a more viable approach is required when considering practical reasoning. To this end, we propose an encoding of arbitrary $SR\mathcal{OIQ}$ knowledge bases into *answer set programs*. This allows us to use existing ASP machinery to perform both standard reasoning as well as the non-standard tasks *model extraction & enumeration* and query entailment quite elegantly.

5.1 ASP Encoding of DL Knowledge Bases

We now describe how standard and non-standard reasoning tasks w.r.t. the fixed-domain semantics can be encoded by ASP. Intuitively, the set of all Δ -interpretations defines

a search space, which can be traversed searching for Δ -models, guided by appropriate constraints. We thus propose a translation $\Pi(\mathcal{K}, \Delta)$ for any $SR\mathcal{OIQ}$ knowledge base \mathcal{K} ; i.e. $\Pi(\mathcal{K}, \Delta) = \Pi_{\text{gen}}(\Delta) \cup \Pi_{\text{chk}}(\mathcal{K})$, consisting of a generating part $\Pi_{\text{gen}}(\Delta)$ that defines all potential candidate interpretations, and a constraining part $\Pi_{\text{chk}}(\mathcal{K})$ that rules out interpretations violating axioms in \mathcal{K} .

Simplified Form of Knowledge Bases We first impose a few additional assumptions regarding the knowledge base \mathcal{K} . To start with, we assume that the ABox only refers to domain individuals from Δ and does not mention elements from $N_I(\mathcal{K}) \setminus \Delta$ (except in nominal concepts). This can be obtained by the following standard model-preserving transformations (let $a, b \in N_I(\mathcal{K}) \setminus \Delta$ while $c, d \in \Delta$):

$$\begin{array}{llll} C(a) & \rightsquigarrow & \{a\} \sqsubseteq C & a \doteq b & \rightsquigarrow & \{a\} \sqsubseteq \{b\} \\ r(a, b) & \rightsquigarrow & \{a\} \sqsubseteq \exists r. \{b\} & a \not\doteq b & \rightsquigarrow & \{a\} \sqsubseteq \neg \{b\} \\ r(c, b) & \rightsquigarrow & \exists r. \{b\}(c) & c \doteq b & \rightsquigarrow & \{b\}(c) \\ r(a, d) & \rightsquigarrow & \exists r^-. \{a\}(d) & c \neq b & \rightsquigarrow & \neg \{b\}(c) \end{array}$$

In the light of this, it is also clear that we can assume the ABox to be free of (in)equality statements: between individuals from Δ they are either tautological (and can be removed) or obviously contradictory (and could be replaced by $\top \sqsubseteq \perp$).

Next, we make the common assumption that all concept expressions and all roles occurring in the ABox are concept and role names, respectively (which is easy to establish by the use of auxiliary concepts). Further, we impose the assumption that \mathcal{K} does not contain any nominal concepts. This can be realized by replacing every occurrence of a nominal concept $\{a\}$ in \mathcal{K} with a fresh auxiliary atomic concept $O_{\{a\}}$, adding the axiom $\text{Sing}(O_{\{a\}})$ (which we use as a shortcut for the two axioms $\top \sqsubseteq \leq 1 u. O_{\{a\}}$ and $\top \sqsubseteq \geq 1 u. O_{\{a\}}$ in order to enforce that $O_{\{a\}}$ must be interpreted by a singleton set) to the TBox and, in case $a \in \Delta$, adding the assertion $O_{\{a\}}(a)$ to the ABox. Obviously, there is a one-to-one correspondence between (Δ -)models \mathcal{I} of the original and (Δ -)models \mathcal{J} of the transformed knowledge base via $a^{\mathcal{I}} = \delta$ iff $O_{\{a\}}^{\mathcal{J}} = \{\delta\}$, hence (Δ -)models of the original knowledge base are straightforward to reconstruct.

Our last requirement is that the knowledge base is in normalized form, obtained by a modified structural transformation $\Omega(\mathcal{K})$, akin to the one proposed in [21]. A TBox axiom is normalized, if it is of the form $\top \sqsubseteq \bigsqcup_{i=1}^n C_i$, where C_i is of the form $B, \forall r. B, \exists r. \text{Self}, \neg \exists r. \text{Self}, \geq nr. B$, or $\leq nr. B$, for B a literal concept (i.e., a concept name or a negated concept name), r a role, and n a positive integer. It can be shown that the obtained normalized knowledge base $\Omega(\mathcal{K})$ is a model-conservative extension of \mathcal{K} , i.e. every (Δ -)model of $\Omega(\mathcal{K})$ is a (Δ -)model of \mathcal{K} and every (Δ -)model of \mathcal{K} can be turned into a (Δ -)model of $\Omega(\mathcal{K})$ by finding appropriate interpretations for the concepts and roles introduced by Ω . Thereby it is straightforward to extract a model for \mathcal{K} , given a model of $\Omega(\mathcal{K})$. Table 3 depicts the normalization rules in detail. Thanks to the correspondences depicted here, we can for the rest of our considerations assume that the considered knowledge base is free of mentions of non- Δ individuals, free of nominals, free of (in)equality statements, and normalized; we call such a knowledge base *simplified*.

Table 3. Ω -Normalization of knowledge base axioms.

$\Omega(\mathcal{K})$	$= \bigcup_{\alpha \in \mathcal{R} \cup \mathcal{A}} \Omega(\alpha) \cup \bigcup_{C_1 \sqsubseteq C_2 \in \mathcal{T}} \Omega(\top \sqsubseteq \text{nnf}(\neg C_1 \sqcup C_2))$
$\Omega(\top \sqsubseteq \mathbf{C} \sqcup C')$	$= \Omega(\top \sqsubseteq \mathbf{C} \sqcup \alpha_{C'}) \cup \bigcup_{1 \leq i \leq n} \Omega(\top \sqsubseteq \dot{\neg} \alpha_{C'} \sqcup C_i)$ for C' of the form $C' = C_1 \sqcap \dots \sqcap C_n$ and $n \geq 2$
$\Omega(\top \sqsubseteq \mathbf{C} \sqcup \exists r.D)$	$= \Omega(\top \sqsubseteq \mathbf{C} \sqcup \geq 1 r.D)$
$\Omega(\top \sqsubseteq \mathbf{C} \sqcup \forall r.D)$	$= \Omega(\top \sqsubseteq \mathbf{C} \sqcup \forall r.\alpha_D) \cup \Omega(\top \sqsubseteq \dot{\neg} \alpha_D \sqcup D)$
$\Omega(\top \sqsubseteq \mathbf{C} \sqcup \geq n r.D)$	$= \Omega(\top \sqsubseteq \mathbf{C} \sqcup \geq n r.\alpha_D) \cup \Omega(\top \sqsubseteq \dot{\neg} \alpha_D \sqcup D)$
$\Omega(\top \sqsubseteq \mathbf{C} \sqcup \leq n r.D)$	$= \Omega(\top \sqsubseteq \mathbf{C} \sqcup \leq n r.\dot{\neg} \alpha_{\neg D}) \cup \Omega(\top \sqsubseteq \dot{\neg} \alpha_{\neg D} \sqcup \dot{\neg} D)$
$\Omega(D(s))$	$= \{\alpha_D(s)\} \cup \Omega(\top \sqsubseteq \dot{\neg} \alpha_D \sqcup \text{nnf}(D))$
$\Omega(r^-(s, t))$	$= \{r(t, s)\}$
$\Omega(r_1 \circ \dots \circ r_n \sqsubseteq r)$	$= \{r_1 \circ r_2 \sqsubseteq r_{(r_1 \circ r_2)}\} \cup \Omega(r_{(r_1 \circ r_2)} \circ r_3 \circ \dots \circ r_n \sqsubseteq r)$ for any RIA with $n > 2$
$\Omega(\beta)$	$= \{\beta\}$ for any other axiom β
$\alpha_C = \begin{cases} Q_C & \text{if } \text{pos}(C) = \text{true} \\ \neg Q_C & \text{if } \text{pos}(C) = \text{false} \end{cases}$, where Q_C is a fresh concept name unique for C .
$\text{pos}(\top)$	$= \text{false}$
$\text{pos}(A)$	$= \text{true}$
$\text{pos}(\exists r.Self)$	$= \text{true}$
$\text{pos}(C_1 \sqcap C_2)$	$= \text{pos}(C_1) \vee \text{pos}(C_2)$
$\text{pos}(\forall r.C_1)$	$= \text{pos}(C_1)$
$\text{pos}(\geq n r.C_1)$	$= \text{true}$
$\text{pos}(\perp)$	$= \text{false}$
$\text{pos}(\neg A)$	$= \text{false}$
$\text{pos}(\neg \exists r.Self)$	$= \text{false}$
$\text{pos}(C_1 \sqcup C_2)$	$= \text{pos}(C_1) \vee \text{pos}(C_2)$
$\text{pos}(\leq n r.C_1)$	$= \begin{cases} \text{pos}(\dot{\neg} C_1) & \text{if } n = 0 \\ \text{true} & \text{otherwise} \end{cases}$

Note: A is an atomic concept, $C_{(i)}$ are arbitrary concept expressions, \mathbf{C} is a possibly empty disjunction of concept expressions, D is **not** a literal concept. The function $\dot{\neg}$ is defined as $\dot{\neg}(\neg A) = A$ and $\dot{\neg}(A) = \neg A$ for some atomic concept A .

Candidate Generation Following the generate & test paradigm, we let $\Pi_{\text{gen}}(\Delta)$ be the program that generates (all) possible interpretations over Δ ; i.e. for each concept name A and role name r all possible extensions over Δ are generated (while the interpretation of the individual names is just the identity, exploiting our assumption that \mathcal{K} does not contain non- Δ individual names). For each concept name A and role name r occurring in \mathcal{K} , $\Pi_{\text{gen}}(\Delta)$ contains the following rules:

$$A(X) \leftarrow \top(X), \text{not } \bar{A}(X). \quad (1)$$

$$\bar{A}(X) \leftarrow \top(X), \text{not } A(X). \quad (2)$$

$$r(X, Y) \leftarrow \top(X), \top(Y), \text{not } \bar{r}(X, Y). \quad (3)$$

$$\bar{r}(X, Y) \leftarrow \top(X), \top(Y), \text{not } r(X, Y). \quad (4)$$

Thereby, slightly overloading notation, A and \bar{A} are unary predicates introduced for every concept name A in \mathcal{K} and r and \bar{r} are binary predicates introduced for every role name r . Moreover, we use \top as unary domain predicate and let $\Pi_{\text{gen}}(\Delta)$ contain the fact $\top(\delta)$ for each domain element $\delta \in \Delta$. Rules (1) and (2) guess a set of ground instances of $A(X)$ for any concept name A , while Rules (3) and (4) realize the same for role names r . Further, we have to axiomatize the universal role u as follows:

$$u(X, Y) \leftarrow \top(X), \top(Y). \quad (5)$$

Like this, an answer set \mathbf{A} of $\Pi_{\text{gen}}(\Delta)$ directly corresponds to an interpretation $\mathcal{I}_{\mathbf{A}} = (\Delta, \cdot^{\mathbf{A}})$ of \mathcal{K} over the fixed-domain

Δ as follows:

$$A^{\mathbf{A}} = \{\delta \mid A(\delta) \in \mathbf{A}\}, \text{ for all } A \in N_C(\mathcal{K}),$$

$$r^{\mathbf{A}} = \{(\delta, \delta') \mid r(\delta, \delta') \in \mathbf{A}\}, \text{ for all } r \in N_R(\mathcal{K}),$$

$$a^{\mathbf{A}} = a.$$

Axiom Encoding Beginning with assertions in \mathcal{A} , for each $A(a)$ and each role $r(a, b)$ we add the identical fact to $\Pi_{\text{chk}}(\mathcal{K})$.

Further, we turn each axiom $\alpha \in \mathcal{T} \cup \mathcal{R}$ into a constraint, ultimately ruling out those candidate interpretations not satisfying α , whence $\Pi_{\text{chk}}(\mathcal{K}) = \mathcal{A} \cup \Pi_{\text{chk}}(\mathcal{T}) \cup \Pi_{\text{chk}}(\mathcal{R})$. Since each $\alpha \in \mathcal{T}$ is of the form $\top \sqsubseteq \bigsqcup_{i=1}^n C_i$, we simply turn it into a negative constraint of the form $\prod_{i=1}^n \neg C_i \sqsubseteq \perp$, and add its direct translation to $\Pi_{\text{chk}}(\mathcal{T})$. Thus, for each $\top \sqsubseteq \bigsqcup_{i=1}^n C_i$ a constraint of the following form is obtained, where the individual translation $\text{trans}(C_i)$ of concepts C_i is defined in Table 4:

$$\leftarrow \text{trans}(C_1), \dots, \text{trans}(C_n). \quad (6)$$

Role assertions and role inclusion axioms are also turned into constraints, and we add their direct translation to $\Pi_{\text{chk}}(\mathcal{R})$. For each role inclusion $r \sqsubseteq s \in \mathcal{R}$, role disjointness $\text{Dis}(r, s) \in \mathcal{R}$ and role chain $s_1 \circ s_2 \sqsubseteq r \in \mathcal{R}$ this yields

$$\leftarrow r(X, Y), \text{not } s(X, Y). \quad (7)$$

$$\leftarrow s(X, Y), r(X, Y). \quad (8)$$

$$\leftarrow s_1(X, Y), s_2(Y, Z), \text{not } r(X, Z). \quad (9)$$

respectively, where we silently assume that an expressions of the form $r^-(V_1, V_2)$ represents the atom $r(V_2, V_1)$.

Theorem 10 *Let \mathcal{K} be a simplified \mathcal{SROIQ} knowledge base. Then, $\mathcal{I} \models_{\Delta} \mathcal{K}$ exactly if $\{A(\delta) \mid \delta \in A^{\mathcal{I}}, A \in N_C(\mathcal{K}) \cup \{\top\}\} \cup \{r(\delta, \delta') \mid (\delta, \delta') \in r^{\mathcal{I}}, r \in N_R(\mathcal{K}) \cup \{u\}\}$ is an answer set of $\Pi(\mathcal{K}, \Delta)$.*

This theorem establishes a tight one-to-one correspondence between Δ -models of a knowledge base and answer sets of its ASP translation, and allows us to employ ASP solving tools for fixed-domain DL reasoning. Most notably, in addition to the standard DL reasoning tasks, *model extraction* and *model enumeration* can be carried out without additional efforts, since both are natural tasks for answer set solvers. Moreover, all mentioned query formalisms can be straightforwardly expressed in a rule-based way, whence integration in our framework is immediate.

Table 4. Translation of Concept Expressions.

C	$trans(C)$
A	$not\ A(X)$
$\neg A$	$A(X)$
$\forall r.A$	$\{not\ A(Y_A), r(X, Y_A)\}$
$\forall r.\neg A$	$\{r(X, Y_A), A(Y_A)\}$
$\exists r.Self$	$not\ r(X, X)$
$\neg\exists r.Self$	$r(X, X)$
$\geq n\ r.A$	$\#count\{r(X, Y_A) : A(Y_A)\} < n$
$\geq n\ r.\neg A$	$\#count\{r(X, Y_A) : not\ A(Y_A)\} < n$
$\leq n\ r.A$	$\#count\{r(X, Y_A) : A(Y_A)\} > n$
$\leq n\ r.\neg A$	$\#count\{r(X, Y_A) : not\ A(Y_A)\} > n$

5.2 Implementation and Experiments

We implemented our translation based approach as an open-source tool – named **Wolpertinger**.³ The obtained logic programs can be evaluated with most modern ASP solvers. However, the evaluation was conducted using **Clingo** [9] for grounding and solving, since it is currently the most prominent solver leading the latest competitions [5].

In fact, the tool is used for solving nontrivial real-world configuration problems in an industry project with very good results. Yet, for legal reasons we cannot disclose any details.

Hence, we present preliminary evaluation results based on simple ontologies, encoding constraint-satisfaction-type combinatorial problems. Existing OWL ontologies typically used for benchmarking, e.g. SNOMED or GALEN [29, 23], do not fit our purpose, since they are modeled with the classical semantics in mind and often have little or no ABox information.

Our tests provide runtimes compared to the popular **HermiT** and **Konclude** reasoners [12, 30]. Both are full OWL 2 DL reasoners and are leading the latest competitions. Whereas a direct comparison would not be fair, the conducted tests shall merely show the feasibility of our approach in comparison to standard DL reasoners using the axiomatization. In particular we focus on model enumeration, for which we can not conduct any comparison with existing DL reasoners. The evaluation itself is conducted on a standard desktop machine (Unix operating system, 2.7 Ghz Intel Core i5 Processor, 8 GB memory and standard Java-VM settings).

Unsatisfiability We construct an unsatisfiable knowledge base $\mathcal{K}_n = (\mathcal{A}_n, \mathcal{T}_n, \emptyset)$, with $\mathcal{T}_n = \{A_1 \sqsubseteq \exists r.A_2, \dots, A_n \sqsubseteq \exists r.A_{n+1}, A_i \sqcap A_j \sqsubseteq \perp \mid 1 \leq i < j \leq n+1\}$ and $\mathcal{A}_n = \{A_1(a_1), \top(a_1), \dots, \top(a_n)\}$, together with the fixed-domain $\Delta = \{a_1, \dots, a_n\}$.

Inspired by common pigeonhole-type problems, we have \mathcal{K}_n enforce an r -chain of length $n+1$ without repeating elements, yet, having fixed Δ to n elements such a model cannot exist. Table 5 depicts the runtimes for detecting unsatisfiability of \mathcal{K}_n , for increasing n . The durations correspond to the pure solving time as stated by the tools (including grounding in the case of **Clingo**), and neglecting pre-processing time. As the figures suggest, \mathcal{K}_n is a potential worst-case scenario, where any of the tools is doomed to test all combinations. Whereas **Wolpertinger** is faster in claiming inconsistency in all cases up to \mathcal{K}_{10} , **HermiT** is slightly faster up from \mathcal{K}_{11} – both leaving **Konclude** behind. As \mathcal{K}_{11} causes already a massive increase, in the runs 7–10 we restricted the search space for \mathcal{K}_{11} by explicitly adding negative r -edges and thus allowing for a more fine grained evaluation. However, \mathcal{K}_{12} is already beyond a feasible time bound for all reasoners.

Table 5. Runtimes: Detecting Unsatisfiability of \mathcal{K}_n .

#	\mathcal{K}_n	Wolpertinger	HermiT	Konclude
1	5	< 0.01 s	0.48 s	0.04 s
2	6	< 0.01 s	0.67 s	0.07 s
3	7	0.04 s	0.94 s	0.26 s
4	8	0.33 s	1.81 s	1.79 s
5	9	3.72 s	9.52 s	16.19 s
6	10	68.53 s	87.88 s	152.37 s
7	11 ^a	111.43 s	203.67 s	342.90 s
8	11 ^b	350.13 s	301.15 s	516.60 s
9	11 ^c	438.70 s	403.69 s	669.47 s
10	11 ^d	582.90 s	491.34 s	878.46 s
11	11	1 095.49 s	1 027.33 s	1 682.41 s

Table 6. Runtimes: Sudoku Satisfiability.

#	Size	Wolpertinger	HermiT	Konclude
1	6 × 6 - 0	1.04 s	6.32 s	2.75 s
2	9 × 9 - 46	5.68 s	0.93 s	11.37 s
3	9 × 9 - 42	5.78 s	1.06 s	20.68 s
4	9 × 9 - 38	6.96 s	1.47 s	60.17 s
5	9 × 9 - 28	6.87 s	27.48 s	> 30 min
6	9 × 9 - 0	6.73 s	> 30 min	> 30 min

Table 7. Runtimes: Enumerating 9×9-Sudoku Instances.

#	Models	Time(Total)	Time(Solving)
1	100	6.73 s	0.11 s
2	1 000	7.16 s	0.33 s
3	10 000	9.06 s	2.39 s
4	100 000	29.27 s	22.53 s
5	1 000 000	225.40 s	218.56 s

Model Extraction and Enumeration With Table 6, we next provide some figures for model extraction and partial enumeration (retrieving a given number of Δ -models). To this

³ <https://github.com/wolpertinger-reasoner/Wolpertinger>

end, we created a knowledge base modeling fully and correctly filled Sudokus, beginning with a 6×6-instance, consisting of 64 individuals, 13 concept names and 1 role name. Then testing on a 9×9-instance featuring 108 named individuals. Whereas **HerMiT** & **Konclude** are still able to claim satisfiability for the 6×6-instance, invoking a satisfiability test on the 9×9-instance (axiomatized knowledge base), no answer was given within 30 minutes. In both cases **Wolpertinger** detects satisfiability with reasonable runtimes. Therefore, we again restricted the search space in run 2-5 by having 46 cells pre-filled, respectively 42, 38 and 28.

For model enumeration, we used the knowledge base for the 9×9-instance and turning the task into generating new Sudoku instances. Table 7 depicts the figures, where it can be observed that, besides a constant time of around 6 seconds required for grounding, requesting 10^6 models is reasonably efficient.

6 Conclusion and Future Work

For OWL ontologies which represent constraint-type problems, the fixed-domain semantics allows to confine modelhood of interpretations to models of the right form. Although OWL still imposes some restrictions regarding expressivity (e.g., by restricting the arity of the used predicates to 1 and 2), we argue that quite large and involved problem scenarios can be modeled by OWL ontologies. Clearly, more comprehensive evaluations of our system with respect to such ontologies remain as imperative issue. Moreover, translations of fixed-domain reasoning problems into other formalisms are conceivable, including pure CSP languages or even SAT, which would have to be implemented and compared against the ASP approach.

Regarding our translation from DL to ASP, we want to point out that there is related work on computing finite models for general FO theories using ASP [10], where an incremental approach is implemented using iASP, in order to successively increase the size of the domain over which a model shall be constructed – ultimately proving finite satisfiability. Although different in its motivation, it should be possible to compare both translations if we consider the FO translation of some DL knowledge base and the cardinality of the given fixed domain as initial domain size.

Another interesting strand of research would be to consider extensions of the source formalism, e.g. by non-monotonic features. As ASP itself is a non-monotonic logic programming formalism, rule-based extensions of OWL – monotonic [14, 20] or nonmonotonic [19, 1] – should be straightforward to accommodate. On yet another note, we plan to incorporate typical ontology engineering tasks such as explanation and axiom pinpointing into our ASP-based framework.

ACKNOWLEDGEMENTS

This work is supported by DFG in the Research Training Group QuantLA (GRK 1763). Moreover, we are grateful for all the valuable feedback from our colleagues and the anonymous reviewers, which helped greatly to improve this work.

REFERENCES

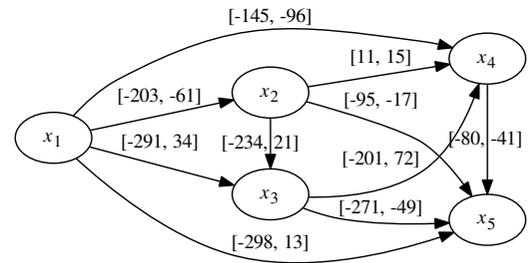
- [1] José Júlio Alferes, Matthias Knorr, and Terrance Swift, ‘Query-Driven Procedures for Hybrid MKNF Knowledge Bases’, *ACM Transactions on Computational Logic*, **14**(2), 16:1–16:43, (2013).
- [2] Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, and Peter Patel-Schneider, *The Description Logic Handbook: Theory, Implementation, and Applications*, Cambridge University Press, second edn., 2007.
- [3] Meghyn Bienvenu, Diego Calvanese, Magdalena Ortiz, and Mantas Simkus, ‘Nested regular path queries in description logics’, in *Proc. of the 14th Conference on Principles of Knowledge Representation and Reasoning (KR 2014)*, eds., Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter. AAAI Press, (2014).
- [4] Gerhard Brewka, Thomas Eiter, and Mirosław Trzuszczynski, ‘Answer set programming at a glance’, *Commun. ACM*, **54**(12), 92–103, (2011).
- [5] Francesco Calimeri, Martin Gebser, Marco Maratea, and Francesco Ricca, ‘Design and results of the 5th answer set programming competition’, *Artif. Intell.*, **231**, 151–181, (2016).
- [6] Diego Calvanese, ‘Finite model reasoning in description logics’, in *Proc. of the International Workshop on Description Logics (DL 1996)*, eds., Lin Padgham, Enrico Franconi, Manfred Gehrke, Deborah L. McGuinness, and Peter F. Patel-Schneider, volume WS-96-05 of *AAAI Technical Report*, pp. 25–36. AAAI Press, (1996).
- [7] Diego Calvanese, Thomas Eiter, and Magdalena Ortiz, ‘Regular path queries in expressive description logics with nominals’, in *Proc. of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009)*, ed., Craig Boutilier, pp. 714–720, (2009).
- [8] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub, ‘Answer Set Solving in Practice’, *Synthesis Lectures on Artificial Intelligence and Machine Learning*, **6**, 1–238, (2012).
- [9] Martin Gebser, Benjamin Kaufmann Roland Kaminski, Max Ostrowski Torsten Schaub, and Marius T. Schneider, ‘Potassco: The Potsdam Answer Set Solving Collection’, *AI Communications*, **24**(2), 107–124, (2011).
- [10] Martin Gebser, Orkunt Sabuncu, and Torsten Schaub, ‘Finite model computation via answer set programming’, in *Proc. of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, ed., Toby Walsh, pp. 2626–2631. IJCAI/AAAI, (2011).
- [11] Michael Gelfond and Vladimir Lifschitz, ‘Classical negation in logic programs and disjunctive databases’, *New Generation Comput.*, **9**(3/4), 365–386, (1991).
- [12] Birte Glimm, Ian Horrocks, Boris Motik, Giorgos Stoilos, and Zhe Wang, ‘HerMiT: an OWL 2 reasoner’, *Journal of Automated Reasoning*, **53**(3), 245–269, (2014).
- [13] Ian Horrocks, Oliver Kutz, and Ulrike Sattler, ‘The Even More Irresistible *SRQIQ*’, in *Proc. of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR 2006)*, eds., Patrick Doherty, John Mylopoulos, and Christopher A. Welty, pp. 57–67. AAAI Press, (2006).
- [14] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin N. Groszof, and Mike Dean, *SWRL: A Semantic Web Rule Language*, W3C Member Submission, 21 May 2004.
- [15] Yevgeny Kazakov, ‘*RIQ* and *SRQIQ* are harder than *SHQIQ*’, in *Proc. of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR 2008)*, eds., Gerhard Brewka and Jérôme Lang, pp. 274–284. AAAI Press, (2008).
- [16] Holger Knublauch, Mark A. Musen, and Alan L. Rector, ‘Editing Description Logic Ontologies with the Protégé OWL Plugin’, in *Proc. of the International Workshop on Description Logics (DL 2004)*, eds., Volker Haarslev and Ralf Möller, volume 104 of *CEUR Workshop Proceedings*. CEUR-WS.org, (2004).
- [17] Carsten Lutz, Ulrike Sattler, and Lidia Tendera, ‘The com-

- plexity of finite model reasoning in description logics’, *Information and Computation*, **199**(1-2), 132–171, (May 2005).
- [18] David G. Mitchell and Eugenia Ternovska, ‘A framework for representing and solving NP search problems’, in *Proc. of the 20th National Conference on Artificial Intelligence and the 17th Innovative Applications of Artificial Intelligence Conference (AAAI 2005)*, eds., Manuela M. Veloso and Subbarao Kambhampati, pp. 430–435. AAAI Press / The MIT Press, (2005).
- [19] Boris Motik and Riccardo Rosati, ‘Reconciling description logics and rules’, *Journal of the ACM*, **57**(5), (2010).
- [20] Boris Motik, Ulrike Sattler, and Rudi Studer, ‘Query Answering for OWL DL with Rules’, *Journal of Web Semantics*, **3**(1), 41–60, (2005).
- [21] Boris Motik, Rob Shearer, and Ian Horrocks, ‘Hypertableau Reasoning for Description Logics’, *Artificial Intelligence Research*, **36**, 165–228, (2009).
- [22] Ilkka Niemelä, ‘Logic programs with stable model semantics as a constraint programming paradigm’, *Ann. Math. Artif. Intell.*, **25**(3-4), 241–273, (1999).
- [23] Alan Rector and Ian Horrocks, ‘Experience building a large, re-usable medical ontology using a description logic with transitivity and concept inclusions’, in *Proc. of the Workshop on Ontological Engineering*, (1997).
- [24] Juan L. Reutter, Miguel Romero, and Moshe Y. Vardi, ‘Regular Queries on Graph Databases’, in *Proc. of the 18th International Conference on Database Theory (ICDT 2015)*, eds., Marcelo Arenas and Martín Ugarte, volume 31 of *LIPICs*, pp. 177–194. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, (2015).
- [25] Riccardo Rosati, ‘Finite Model Reasoning in DL-Lite’, in *Proc. of the 5th European Semantic Web Conference (ESWC 2008)*, eds., Sean Bechhofer, Manfred Hauswirth, Jörg Hoffmann, and Manolis Koubarakis, volume 5021 of *LNCS*, p. 215. Springer, (2008).
- [26] Sebastian Rudolph, ‘Foundations of Description Logics’, in *Reasoning Web. 7th International Summer School 2011, Tutorial Lectures*, eds., Axel Polleres, Claudia d’Amato, Marcelo Arenas, Siegfried Handschuh, Paula Kroner, Sascha Ossowski, and Peter F. Patel-Schneider, volume 6848 of *LNCS*, pp. 76–136. Springer, (2011).
- [27] Sebastian Rudolph, ‘Undecidability results for database-inspired reasoning problems in very expressive description logics’, in *Proc. of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR 2016)*, (2016).
- [28] Walter J. Savitch, ‘Relationships Between Nondeterministic and Deterministic Tape Complexities’, *J. Comput. Syst. Sci.*, **4**(2), 177–192, (April 1970).
- [29] Kent A. Spackman, Keith E. Campbell, and Roger A. Côté, ‘SNOMED RT: a reference terminology for health care’, in *AMIA 1997, American Medical Informatics Association Annual Symposium*. AMIA, (1997).
- [30] Andreas Steigmiller, Thorsten Liebig, and Birte Glimm, ‘Konclude: System description’, *Journal of Web Semantics*, **27**, 78–85, (2014).
- [31] Larry J. Stockmeyer, *The Complexity of Decision Problems in Automata Theory and Logic*, Ph.D. dissertation, Massachusetts Institute of Technology, 1974.
- [32] W3C OWL Working Group, *OWL 2 Web Ontology Language: Document Overview*, W3C Recommendation, 2009.

On Redundancy in Simple Temporal Networks¹

Jae Hee Lee² and Sanjiang Li³ and Zhiguo Long⁴ and Michael Sioutis⁵

Abstract. The Simple Temporal Problem (STP) has been widely used in various applications to schedule tasks. For dynamical systems, scheduling needs to be efficient and flexible to handle uncertainty and perturbation. To this end, modern approaches usually encode the temporal information as an STP instance. This representation contains redundant information, which can not only take a significant amount of storage space, but also make scheduling inefficient due to the non-concise representation. In this paper, we investigate the problem of simplifying an STP instance by removing redundant information. We show that such a simplification can result in a unique minimal representation without loss of temporal information, and present an efficient algorithm to achieve this task. Evaluation on a large benchmark dataset of STP exhibits a significant reduction in redundant information for the involved instances.



(a) An STN extracted from a job-shop problem instance.

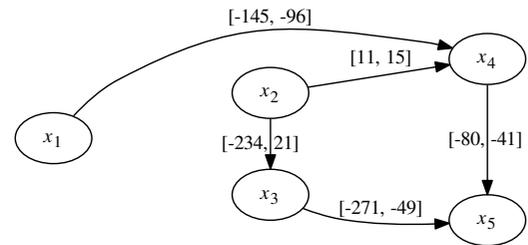
1 INTRODUCTION

The ability to reason about temporal information is necessary for intelligent agents that plan their actions to achieve their goals optimally. As such, temporal reasoning has been an active research area in Artificial Intelligence [6, 2, 16, 3, 17].

Among the different temporal representation frameworks, some of the most well-known ones are based on relations between time points or time intervals. Prominent examples of such representation frameworks include the Simple Temporal Problem [5], Allen's Interval Algebra [1], and Point Algebra [20]. In this paper, we will focus on the Simple Temporal Problem and be concerned with redundant information in instances of this problem.

The *Simple Temporal Problem* (STP) encodes the quantitative difference between two variables representing time points. An STP constraint $(x [a, b] y)$ associates an interval $[a, b]$ with variables x and y to represent the lower and upper bounds of the difference, i.e. $a \leq y - x \leq b$. An STP instance is called a Simple Temporal Network (STN), and consists of a set of variables and a set of STP constraints involving those variables. This can be represented as a labelled directed graph (see Figure 1). A solution of an STN is an assignment of time points to the variables such that all of the corresponding STP constraints are satisfied by the assignment.

There are often redundant constraints in STNs. Figure 1a shows an STN extracted from the job-shop scheduling problem benchmark dataset used in [14]. (We note that in all our examples we have modified the original STNs by reducing the bounds in their constraints to make the examples easier to follow; however, all qualitative properties of the STNs have been retained). We observe that



(b) An STN obtained by removing all redundant constraints in the STN in Figure 1a. The two STNs are equivalent by Theorem 6.

Figure 1. Two equivalent STNs.

the STN is densely structured and contains redundant constraints that can be removed without affecting the set of solutions. For example, the constraint $(x_2 [-95, -17] x_5)$ is redundant and can be removed, because combining the constraint between x_2 and x_4 and the constraint between x_4 and x_5 implies a tighter constraint, viz., $(x_2 [-69, -26] x_5)$. Redundant constraints are not limited to the aforementioned trivial case; although the constraint between x_3 and x_4 cannot be directly inferred from any other two constraints, it can be inferred by combining more constraints (cf. Example 6 in Section 3). Identifying such redundant constraints efficiently is one of the main goals of this paper.

After identifying all redundant constraints in an STN, the question arises whether we can remove all of them, while maintaining the solution set unchanged. The answer is not straightforward as the redundant constraints can depend on each other, in the sense that removing

¹ The authors have contributed equally.

² QCIS, FEIT, UTS, Australia, email: JaeHee.Lee@uts.edu.au

³ QCIS, FEIT, UTS, Australia, email: Sanjiang.Li@uts.edu.au

⁴ QCIS, FEIT, UTS, Australia, email: Zhiguo.Long@student.uts.edu.au

⁵ CRIL, University of Artois, France, email: sioutis@cril.fr

one redundant constraint can make another constraint non-redundant. In this paper we characterize STNs that retain the solution sets after removing all redundant constraints. As an example, the STN in Figure 1a and the STN in Figure 1b that results from removing all redundant constraints of the former STN share the same solution set.

In the literature, the problem of identifying and removing redundant information has been discussed in the context of logic formulas [8, 10] and of qualitative constraint networks (QCNs) [9, 18]. For QCNs, it was shown in [9] that an all-different network defined over a distributive subclass of relations of RCC8 [15] or of Point Algebra [20] has a unique subset of non-redundant constraints that characterises a network that is equivalent to the original one. Efficient algorithms have been developed to identify such subsets [9, 18]. Since QCNs are defined over a finite set of qualitative relations, whereas STNs are defined on quantitative relations, the techniques developed in [9, 18] are not immediately applicable to STNs.

For STP constraints, [11] and [4] identify another notion of redundancy, so-called *dominance*, for inferring the range of a variable. By identifying and removing such redundancy, a particular property of an STN, called the *dispatchability* [11], which is helpful for generating solutions of an STN online, may be retained. By contrast, in this paper we will show that, by removing redundant constraints, the structure of an STN is simplified, while the solution set remains the same.

The remainder of the paper is organised as follows. In Section 2, we recall some basic concepts and formally define redundant constraints. Then we prove some properties of redundant constraints in STNs in Section 3, and use those properties to devise an algorithm for identifying and removing redundant constraints in Section 4. Section 5 evaluates our approach by using benchmark datasets. Finally, Section 6 concludes the paper.

2 PRELIMINARIES

The *simple temporal problem* (STP) is a constraint satisfaction problem where each constraint is a set of linear inequalities of the form

$$a \leq y - x \leq b, \quad (1)$$

where a, b are constants and x, y are variables defined on a continuous domain representing time points. The constraint in (1) is abbreviated as $(x [a, b] y)$. As (1) is equivalent to $-b \leq x - y \leq -a$, we also obtain $(y [-b, -a] x)$.

Algebraic operations on STP constraints are defined as follows. The *intersection* of two STP constraints defined on variables x, y yields a new constraint over x, y that represents the conjunction of the constraints. It is defined as $(x [a_1, b_1] y) \cap (x [a_2, b_2] y) := (x [c, d] y)$, where $[c, d] = [a_1, b_1] \cap [a_2, b_2]$.

The *composition* of an STP constraint over variables x, z and another STP constraint z, y yields a new STP constraint over x, y that is inferred from the other two constraints and is defined as $(x [a_1, b_1] z) \otimes (z [a_2, b_2] y) := (x [c, d] y)$, where $c = a_1 + a_2$ and $d = b_1 + b_2$.

Note that for STP constraints, the composition and intersection are associative and, as noted in [5], composition distributes over non-empty intersection, i.e., $(x [a, b] y) \otimes ((y [c, d] z) \cap (y [e, f] z)) = ((x [a, b] y) \otimes (y [c, d] z)) \cap ((x [a, b] y) \otimes (y [e, f] z))$.

Definition 1. An instance of STP is called a *simple temporal network* (STN) and is a tuple (V, Γ) , where $V = \{x_1, \dots, x_n\}$ is a set of variables and Γ is a set of STP constraints defined on V .

We assume that all variables in V appear in Γ , thus, we will simply use Γ to refer to an STN and not explicitly mention V . Furthermore, we have that there is only one constraint between x and y .

An STN naturally induces a graph in the following sense.

Definition 2. The *constraint graph* $G = (V, E)$ of an STN Γ is an undirected graph, where the set V of vertices consists of the variables in Γ and the set E of edges consists of constrained unordered pairs of variables in Γ , i.e.,

$$E = \{\{x, y\} \mid x, y \in V, x \neq y, (x [a, b] y) \in \Gamma\}.$$

For simplicity, we write xy for an edge in place of $\{x, y\}$.

Let $G = (V, E)$ be the constraint graph of an STN Γ . For variables x, y with $xy \in E$, we write $\Gamma(x, y)$ to refer to the constraint between x and y in Γ . Thus, if $\Gamma(x, y) = (x [a, b] y)$, then $\Gamma(y, x) = (y [-b, -a] x)$. Note that in Figure 1a we used a labelled *directed* graph to illustrate an STN, where for any edge $xy \in E$ with $\Gamma(x, y) = (x [a, b] y)$ there is exactly one directed edge (x, y) in the illustration with the corresponding interval $[a, b]$.

Definition 3. A *solution* of an STN Γ is an assignment of time points to the variables in Γ such that all constraints in Γ are satisfied; in this case we say that Γ is *consistent*. Given two STNs Γ and Γ' defined on the same set of variables we write $\Gamma' \models \Gamma$, if every solution of Γ' is a solution of Γ . If $\Gamma' \models \Gamma$ and $\Gamma \models \Gamma'$ then we say that they are equivalent and write $\Gamma' \equiv \Gamma$. We also write $\Gamma \models (x [a, b] y)$ if every solution of Γ satisfies $(x [a, b] y)$.

We observe that $\{(x [a, b] y)\} \models (x [c, d] y)$ if and only if $[a, b] \subseteq [c, d]$. Therefore, if $\{(x [a, b] y)\} \models (x [c, d] y)$, we will say that $(x [a, b] y)$ *refines* $(x [c, d] y)$, written as $(x [a, b] y) \subseteq (x [c, d] y)$. We call an STN Γ' a *refinement* of Γ , if Γ' and Γ are defined on the same set of variables and if for every constraint $(x [a, b] y) \in \Gamma$ there exists a constraint $(x [a', b'] y) \in \Gamma'$ that refines $(x [a, b] y)$.

Definition 4 (Minimality). Let Γ be a consistent STN and let x and y be variables in Γ . Then an STP constraint $(x [a, b] y)$ is said to be *minimal* in Γ , if $\Gamma \models (x [a, b] y)$ and $(x [a, b] y)$ is the smallest constraint with respect to \subseteq .

A refinement Γ^m of Γ is said to be *the minimal network* of Γ , if for all $x, y \in V$ with $x \neq y$ there is a constraint between x and y in Γ^m that is minimal in Γ .

Note that Γ^m is uniquely defined, and since $\Gamma^m \models \Gamma$ and $\Gamma \models \Gamma^m$, we have $\Gamma^m \equiv \Gamma$. Figure 2a shows the minimal network of the STN Γ in Figure 1a, and it is also the minimal network of the STN in Figure 1b. We note that if two STNs Γ_1 and Γ_2 are equivalent, then they have the same minimal network: Suppose $\Gamma_1^m(x, y) \neq \Gamma_2^m(x, y)$ are minimal in Γ_1 and Γ_2 , respectively, then $\Gamma_1^m(x, y) \cap \Gamma_2^m(x, y)$ is minimal in both Γ_1 and Γ_2 and is properly contained in either $\Gamma_1^m(x, y)$ or $\Gamma_2^m(x, y)$, contradicting our assumption.

Definition 5. Let $G = (V, E)$ be the constraint graph of an STN Γ . A sequence $P := (x_{i_0}x_{i_1}, x_{i_1}x_{i_2}, \dots, x_{i_{k-1}}x_{i_k})$ of edges in E is called a *path* from x_{i_0} to x_{i_k} . The length of P is the number of edges in P and is denoted by $|P|$. A path P is called a *cycle*, if $x_{i_k} = x_{i_0}$. By $\Gamma(P)$ we denote the set $\{\Gamma(x_{i_0}, x_{i_1}), \Gamma(x_{i_1}, x_{i_2}), \dots, \Gamma(x_{i_{k-1}}, x_{i_k})\}$ of constraints over P . The successive composition of constraints over P with respect to Γ is defined as

$$\bigotimes \Gamma(P) := \Gamma(x_{i_0}, x_{i_1}) \otimes \Gamma(x_{i_1}, x_{i_2}) \otimes \dots \otimes \Gamma(x_{i_{k-1}}, x_{i_k}).$$

A concatenation of two paths $P_1 := (x_{i_0}x_{i_1}, \dots, x_{i_{k-1}}x_{i_k})$ and $P_2 := (x_{i_k}x_{i_{k+1}}, \dots, x_{i_{\ell-1}}x_{i_\ell})$, denoted by $P_1 + P_2$, is the path $(x_{i_0}x_{i_1}, \dots, x_{i_{\ell-1}}x_{i_\ell})$. We write $\Pi(x, y, E)$ for the set of all paths from x to y on set E of edges.

Example 1. In Figure 1b, $P = (x_1x_4, x_4x_2, x_2x_3)$ is a path of length three in $\Pi(x_1, x_3, E)$, and $C = (x_2x_4, x_4x_5, x_5x_3, x_3x_2)$ is a cycle in $\Pi(x_2, x_2, E)$. The composition of the constraints over P , which is $\otimes \Gamma(P)$, is then $(x_1[-145, -96]x_4) \otimes (x_4[-15, -11]x_2) \otimes (x_2[-234, 21]x_3)$. Note that $(x_4[-15, -11]x_2)$ is equivalent to $(x_2[11, 15]x_4)$.

The following lemma states that the algebraic operations \otimes and \cap are sufficient to calculate the minimal network of an STN. More precisely, the minimal constraint between two variables x and y is equal to the intersection of the compositions of the constraints over the paths in $\Pi(x, y, E)$.

Lemma 1. Let Γ be an STN and E be the set of edges of the constraint graph of Γ . Let Γ^m be the minimal network of Γ . Then for all $x, y \in V$ with $x \neq y$ we have

$$\Gamma^m(x, y) = \bigcap_{P \in \Pi(x, y, E)} \otimes \Gamma(P).$$

Proof. See [5, Section 3]. \square

For constraint satisfaction problems chordal (or triangulated) constraint graphs have been identified as a class for which efficient algorithms exist [18].

Next, we provide a characterization of a chordal graph equivalent to that in [18], which is more suited for the purpose of the paper.

Definition 6. (Chordal Graph) A graph $G = (V, E)$ is said to be *chordal*, if for any edge $xy \in E$ and any path P from x to y on E with $|P| \geq 2$ there exists $z \in V$ with $xz, zy \in E$ such that $P = P_1 + P_2$, where P_1 is a path from x to z and P_2 a path from z to y .

Example 2. The graph in Figure 1a is chordal as it is complete. We can make a non-chordal graph chordal by triangulating it. For example, the graph in Figure 1b is not chordal, but we can make it chordal by adding edge x_3x_4 and obtain the graph in Figure 2b.

In the following lemma we characterize a refinement (of an STN) whose constraint graph is chordal. Such a refinement has a constraint network that is less dense than that of the minimal network while sharing some nice properties with the minimal network.

Definition 7. Given an STN Γ , a refinement Γ^Δ of Γ is said to be a *chordal minimal network* of Γ w.r.t. graph G^Δ , if

1. $G^\Delta = (V, E^\Delta)$ is the constraint graph of Γ^Δ and is chordal with $E^\Delta \supseteq E$;
2. $\Gamma^\Delta(x, y) = \Gamma^m(x, y)$ for all $xy \in E^\Delta$.

Example 3. The minimal network Γ^m of Γ satisfies trivially the conditions for Γ^Δ , as its constraint graph is complete and thus chordal. The STN in Figure 2b has a chordal constraint graph that contains the constraint graph of the STN in Figure 1b. Its constraints are minimal.

For two equivalent STNs Γ_1 and Γ_2 suppose $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ are their constraint graphs, respectively. If $G^\Delta = (V, E^\Delta)$ is a chordal graph with $E^\Delta \supseteq E_1$ and $E^\Delta \supseteq E_2$, then the

chordal minimal network Γ^Δ of Γ_1 w.r.t. G^Δ is also a chordal minimal network of Γ_2 w.r.t. G^Δ , as Γ_1 and Γ_2 have the same minimal network.

The following characterization of Γ^Δ follows from its definition and Lemma 1.

Lemma 2. Let Γ and Γ^Δ , as well as E and E^Δ , be specified as in Definition 7. Then for all $x, y \in V$ with $xy \in E^\Delta$ we have

$$\Gamma^\Delta(x, y) = \bigcap_{P \in \Pi(x, y, E)} \otimes \Gamma(P).$$

3 REDUNDANCY IN SIMPLE TEMPORAL NETWORKS

In an STN some constraints can be redundant as they can be inferred from the rest of the constraints of the STN. In the remainder of this paper we assume that the STNs are consistent.

Definition 8. Let Γ be an STN. Then $\Gamma(x, y)$ is said to be *redundant* in Γ , if $\Gamma \setminus \{\Gamma(x, y)\} \models \Gamma(x, y)$. We say Γ is *prime*, if it does not contain any redundant constraints. A subset Γ' of Γ is called a *prime subnetwork* of Γ , if Γ' is prime and equivalent to Γ . The set of non-redundant constraints in Γ , denoted by Γ^c , is the *core* of Γ .

To obtain a prime subnetwork, a naive algorithm would first determine whether there exists a redundant constraint in Γ . If it does not find any, then Γ will be returned. Otherwise, for a constraint $\Gamma(x, y)$ that is redundant in Γ the algorithm sets $\Gamma \leftarrow \Gamma \setminus \{\Gamma(x, y)\}$ and repeats the procedure recursively.

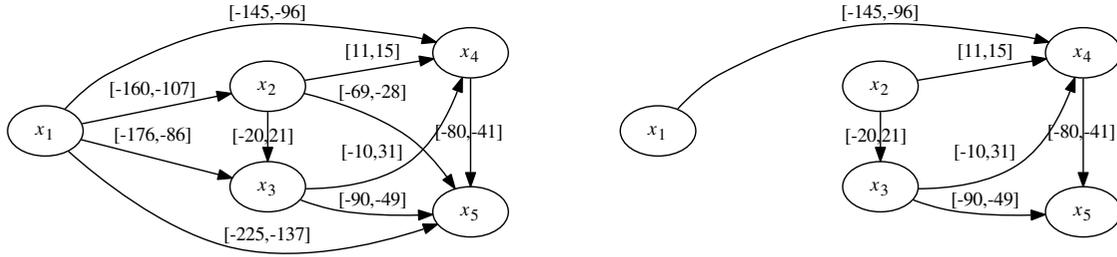
Determining the redundancy of a constraint $\Gamma(x, y)$ can be done by computing the minimal network $\tilde{\Gamma}^m$ of $\tilde{\Gamma} := \Gamma \setminus \{\Gamma(x, y)\}$ and checking whether $\tilde{\Gamma}^m(x, y)$ refines $\Gamma(x, y)$. Since there are $O(n^2)$ constraints in each of the $O(n^2)$ recursions, and since calculating a minimal network can be done in $O(n^3)$ (cf. [5]), the algorithm runs in $O(n^7)$. The algorithm can be improved to $O(n^5)$ by not revisiting constraints that were identified as non-redundant previously, because a non-redundant constraint in an STN Γ is also not redundant in any STN that is contained in Γ .

We observe that any prime subnetwork of Γ contains the core Γ^c as a subset of constraints. If the core Γ^c is a prime subnetwork of Γ , then it is the unique prime subnetwork. However, Γ^c is not necessarily a prime subnetwork of Γ , because removing a redundant constraint c of Γ can cancel the redundancy of another constraint, i.e. the latter constraint would remain redundant in Γ if the former constraint had not been removed. Thus, there can be more than one prime subnetworks of Γ and simultaneously removing all redundant constraints in Γ can lead to an STN that is not to equivalent to Γ , as the following simple example illustrates.

Example 4. Suppose that the STN consists of the constraints $(x [0, 1] y)$, $(x [2, 3] z)$, and $(y [2, 2] z)$, which translate to $0 \leq y - x \leq 1$, $2 \leq z - x \leq 3$, and $z - y = 2$ respectively. Then, it is easy to see that $(x [0, 1] y)$ and $(x [2, 3] z)$ are both redundant because $(x [0, 1] y) = (x [2, 3] z) \otimes (z [-2, -2] y)$ and $(x [2, 3] z) = (x [0, 1] y) \otimes (y [2, 2] z)$. However, removing both $(x [0, 1] y)$ and $(x [2, 3] z)$ will leave the single constraint $(y [2, 2] z)$ as the core, which obviously has a different solution set from the original STN.

For some other STNs the core is equivalent to the original STN.

Example 5. The STN in Figure 1b is the core of the STN in Figure 1a. Moreover, these two STNs are equivalent, because the minimal network of both of them is the STN shown in Figure 2a.



(a) The minimal network of the STNs from Figure 1a and from 1b. (b) An STN whose constraint graph is chordal and contains the constraint graph of the STN from Figure 1b. It is equivalent to the STN from Figure 1b, and its constraints are minimal.

Figure 2. STNs equivalent to the STNs in Figure 1.

We observe that in Example 4, the presence of constraint $(y[2, 2]z)$ makes the other two constraints dependent on each other with respect to their redundancy. Such STNs are *degenerated*.

Definition 9. Let Γ be an STN and Γ^Δ a chordal minimal network of Γ . A constraint $(x[a, b]y)$ is said to be *degenerated*, if $a = b$. STN Γ is said to be *degenerated* if there is $xy \in E^\Delta$ such that $\Gamma^\Delta(x, y) = (x[a, b]y)$ with $a = b$.

It is worth noting that if an STN is degenerated, say $(x[a, a]y)$, then we can easily fix it by either rewriting the constraint as $(x[a - \epsilon, a + \epsilon]y)$, or by removing constraints involving y and updating the constraints $\Gamma(x, z)$ using the rule $\Gamma(x, z) \cap (\Gamma(x, y) \otimes \Gamma(y, z))$.

In the following, we will show that for non-degenerated STNs, the core is the unique minimal subset of constraints that is equivalent to the original STN. To this end, we first characterize redundant constraints as the intersection of constraints over certain paths.

Lemma 3. Let Γ be an STN and $G = (V, E)$ its constraint graph. Then for all $x, y \in V$ with $xy \in E$ the following are equivalent:

- (i) $\Gamma(x, y)$ is redundant in Γ ;
- (ii) $\Gamma(x, y) \supseteq \bigcap_{P \in \Pi(x, y, E \setminus \{xy\})} \bigotimes \Gamma(P)$.

Proof. (i) \Rightarrow (ii). Let $\Gamma_0 := \Gamma \setminus \{\Gamma(x, y)\}$. Then, since $\Gamma(x, y)$ is redundant in Γ we have $\Gamma_0 \models \Gamma(x, y)$ thus $(\Gamma_0)^m \models \Gamma(x, y)$, where $(\Gamma_0)^m$ is the minimal network of Γ_0 . Hence

$$\Gamma(x, y) \supseteq (\Gamma_0)^m(x, y). \quad (2)$$

By Lemma 1 we know that

$$(\Gamma_0)^m(x, y) = \bigcap_{P \in \Pi(x, y, E \setminus \{xy\})} \bigotimes \Gamma(P). \quad (3)$$

From (2) and (3) it follows that

$$\Gamma(x, y) \supseteq \bigcap_{P \in \Pi(x, y, E \setminus \{xy\})} \bigotimes \Gamma(P).$$

(ii) \Rightarrow (i). For all paths between x and y on $E \setminus \{xy\}$, we have $\Gamma \models \Gamma(P)$ and thus $\Gamma \models \bigotimes \Gamma(P)$. Therefore

$$\Gamma \models \bigcap_{P \in \Pi(x, y, E \setminus \{xy\})} \bigotimes \Gamma(P).$$

By applying our assumption (ii) we then have $\Gamma \models \Gamma(x, y)$, and we showed (i). \square

Example 6. The constraint $\Gamma(x_3, x_4) = (x_3[-201, 72]x_4)$ in the STN Γ from Figure 1a is redundant in Γ , because $(x_3[-201, 72]x_4) \supseteq (x_3[-10, 31]x_4) = \Gamma(x_3, x_2) \otimes \Gamma(x_2, x_4) \cap \Gamma(x_3, x_5) \otimes \Gamma(x_5, x_4) \supseteq \bigcap_{P \in \Pi(x_3, x_4, E \setminus \{x_3x_4\})} \bigotimes \Gamma(P)$.

As noted earlier Γ^Δ shares some nice properties with the minimal network of Γ , thus allowing a characterization of the redundant constraints in Γ^Δ .

Proposition 4. Let Γ and Γ^Δ as well as E and E^Δ be specified as in Definition 7. Then for all $x, y \in V$ with $xy \in E^\Delta$ the following are equivalent:

- (i) $\Gamma^\Delta(x, y)$ is redundant in Γ^Δ
- (ii) $\Gamma^\Delta(x, y) = \bigcap_{P \in \Pi(x, y, E^\Delta \setminus \{xy\})} \bigotimes \Gamma^\Delta(P)$
- (iii) $\Gamma^\Delta(x, y) = \bigcap_{xz, zy \in E^\Delta} \Gamma^\Delta(x, z) \otimes \Gamma^\Delta(z, y)$

Proof. See Appendix A. \square

Example 7. Consider the minimal network in Figure 2a as Γ^Δ for Γ in Figure 1a. Then we have $\Gamma^\Delta(x_3, x_4) = (x_3[-10, 31]x_4) = (\Gamma^\Delta(x_3, x_1) \otimes \Gamma^\Delta(x_1, x_4)) \cap (\Gamma^\Delta(x_3, x_2) \otimes \Gamma^\Delta(x_2, x_4)) \cap (\Gamma^\Delta(x_3, x_5) \otimes \Gamma^\Delta(x_5, x_4))$.

There is a connection between redundant constraints in Γ and in Γ^Δ .

Proposition 5. Let Γ and Γ^Δ as well as E and E^Δ be specified as in Definition 7. Furthermore, let Γ be not degenerated. Then for all $x, y \in V$ with $xy \in E^\Delta$ the following are equivalent:

- (i) $\Gamma(x, y)$ is redundant in Γ
- (ii) $\Gamma^\Delta(x, y)$ is redundant in Γ^Δ
- (iii) $\Gamma^\Delta(x, y) = \bigcap_{\substack{P \in \Pi(x, y, E) \\ |P| \geq 2}} \bigotimes \Gamma(P)$

Proof. See Appendix A. \square

Example 8. Taking the STN Γ in Figure 1a and the STN Γ^Δ in Figure 2a as an example of the above proposition, we can see that the constraint $\Gamma(x_3, x_4) = (x_3[-201, 72]x_4)$ is redundant in Γ while $\Gamma^\Delta(x_3, x_4) = (x_3[-10, 31]x_4)$ is redundant in Γ^Δ .

The following theorem affirms that for a non-degenerated STN the core resulting from removing redundant constraint simultaneously is the unique prime subnetwork of the STN.

Theorem 6. *Let Γ be a non-degenerated STN. Then the core Γ^c of Γ is equivalent to Γ and the unique prime subnetwork of Γ .*

Proof. The proof is similar to that for RCC8 constraints (see [9, Theorem 29]). Suppose r_1, r_2, \dots, r_n are the redundant constraints of Γ . Let $\Gamma_0 := \Gamma$ and $\Gamma_i := \Gamma_{i-1} \setminus \{r_i\}$ for $1 \leq i \leq n$. Note that Γ_n is precisely Γ^c , the core of Γ .

We prove the claim by contradiction. Suppose k with $k < n$ is the largest integer such that $\Gamma_k \equiv \Gamma$. Let $r_{k+1} = \Gamma(x, y)$, thus $\Gamma(x, y)$ is redundant in Γ . Further, let Γ^Δ be a chordal minimal network of Γ . Then, on the one hand, we know by Proposition 5 that $\Gamma^\Delta(x, y)$ is redundant in Γ^Δ . On the other hand, since Γ^Δ is also a chordal minimal network of Γ_k , by Proposition 5 we also have that $\Gamma(x, y)$ is redundant in Γ_k . Therefore, $\Gamma_{k+1} = \Gamma_k \setminus \{r_{k+1}\} \equiv \Gamma_k \equiv \Gamma$, contradicting our assumption that $k < n$ is the largest integer such that Γ_k is equivalent to Γ . Therefore, $k = n$ and Γ^c is equivalent to Γ , thus it is a prime subnetwork of Γ that is uniquely defined. \square

With this result, we can use the core as a simplification of the original STN and obtain a maximal reduction in the size of representation, without changing the semantic essence.

4 EFFICIENT ALGORITHM FOR CALCULATING THE CORE

The result below follows directly from Propositions 5 and 4, and allows for efficient identification of redundant constraints in STNs.

Proposition 7. *Let Γ and Γ^Δ as well as E and E^Δ be specified as in Definition 7. Furthermore, let Γ be not degenerated. Then, for all $x, y \in V$ with $xy \in E^\Delta$ the following are equivalent:*

- (i) $\Gamma(x, y)$ is redundant in Γ ;
- (ii) $\Gamma^\Delta(x, y) = \bigcap_{x,z,y \in E^\Delta} \Gamma^\Delta(x, z) \otimes \Gamma^\Delta(z, y)$.

With the aid of Proposition 7, we propose Algorithm 1 to efficiently calculate the core of a non-degenerated STN Γ , which is also the prime subnetwork of Γ by Theorem 6. In this algorithm, we first construct a chordal graph $G^\Delta = (V, E^\Delta)$ by triangulating the constraint graph $G = (V, E)$ of Γ (line 3). Then, we construct the chordal minimal network Γ^Δ w.r.t. G^Δ (line 4). After that, we iteratively retrieve the core of Γ edge by edge (lines 5–11). By Proposition 7, to identify the redundancy of a constraint involving an edge $xy \in E$, we only need to check if the constraint corresponding to edge xy in E^Δ equals the intersection of the compositions of all paths of length two from x to y in E^Δ (lines 8–11).

Example 9. To calculate the core of the STN Γ in Figure 1a, Algorithm 1 first triangulates the constraint graph $G = (V, E)$ of Γ . In this case, as G is complete, any triangulation of G is still G . Then, by calculating the minimal network, Γ^Δ is obtained, which is the STN shown in Figure 2a. For each edge $x_i x_j$ in E , the algorithm checks if the constraint $\Gamma^\Delta(x_i, x_j)$ coincides with the intersection

Algorithm 1: CORE(Γ)

Input : A non-degenerated STN Γ .
Output: Γ^c , the core of Γ .

- 1 $\Gamma^c \leftarrow \emptyset$;
- 2 $G = (V, E) \leftarrow$ the constraint graph of Γ ;
- 3 $G^\Delta = (V, E^\Delta) \leftarrow$ TRIANGULATE(G);
- 4 $\Gamma^\Delta \leftarrow$ CHORDALMINIMALNETWORK(Γ, G^Δ);
- 5 **while** $E \neq \emptyset$ **do**
- 6 $xy \leftarrow E.POP()$;
- 7 $I \leftarrow (x[-\infty, \infty]y)$;
- 8 **foreach** z with $(x, z), (z, y) \in E^\Delta$ **do**
- 9 $I \leftarrow I \cap (\Gamma^\Delta(x, z) \otimes \Gamma^\Delta(z, y))$;
- 10 **if** $\Gamma^\Delta(x, y) \neq I$ **then**
- 11 $\Gamma^c \leftarrow \Gamma^c \cup \Gamma(x, y)$;
- 12 **return** Γ^c .

$\bigcap_{x_i x_k, x_k x_j \in E^\Delta} \Gamma^\Delta(x_i, x_k) \otimes \Gamma^\Delta(x_k, x_j)$. The result shows that $\Gamma^\Delta(x_1, x_2)$, $\Gamma^\Delta(x_1, x_3)$, $\Gamma^\Delta(x_1, x_5)$, $\Gamma^\Delta(x_2, x_5)$, and $\Gamma^\Delta(x_3, x_4)$ are such constraints. Therefore, we know that the corresponding constraints in Γ are redundant, and we obtain the core Γ^c in Figure 1b by removing them.

In what follows, we analyze the time complexity of Algorithm 1 in terms of the number of triangles t in the graph G^Δ . Note that a sparsely structured graph can contain much fewer triangles than a complete graph of the same number of vertices. For each execution of the while loop (lines 5–13) the algorithm checks the triangles in G^Δ that contain xy as an edge. Hence, the while loop checks each triangle in G^Δ at most three times and, thus, runs in $O(t + |E|)$ time. To obtain a sparse chordal graph G^Δ , we can use the *maximum cardinality search* algorithm [19] and a simple *fill-in* procedure [12]. In particular, the maximum cardinality search algorithm visits the vertices of G in an order such that, at any point, a vertex is visited that has the largest number of visited neighbours. Consequently, a vertex ordering α is produced. Then, the fill-in procedure considers the vertices in G one by one with respect to the vertex ordering α , and connects each pair of vertices in the neighbourhood of the vertex at hand with an undirected *fill* edge (if an edge is not already present). The obtained graph G^Δ is then a triangulation of G [7]. The maximum cardinality search algorithm has the nice property that it does not lead to any fill edges if the graph is already chordal. The entire operation of triangulating G in the aforementioned manner is linear in the size of G^Δ , viz., $O(|V| + |E^\Delta|)$. Further, to construct Γ^Δ we can use the state-of-the-art algorithm P³C [13], which runs in $O(t)$ time. Therefore, Algorithm 1 runs in $O(t + |V| + |E^\Delta|)$ time. We note that the running time of Algorithm 1 depends on the structure of the constraint graphs: for complete graphs an upper bound is $O(n^3)$; however, if the constraint graph is a tree, then Algorithm 1 runs in $O(n)$. For determining the prime subnetworks, this is a significant improvement over a naive algorithm that runs in $O(n^5)$ (cf. Section 3).

5 EVALUATION

In this section, we evaluate the previous theoretical results using a large benchmark dataset comprising over a thousand of STNs of various nature. Note that by the analysis that took place in the previous section, the performance of Algorithm 1 is strongly dependent on the P³C algorithm for constructing G^Δ , which has been shown to be

Table 1. Results on redundancy reduction

dataset	#STNs	#deg. ^a	properties ^b			reduction ^c			
			V	E	\bar{D}	μ	min	max	σ
Chordal-1	250	18	1 000	75 840–499 490	0.532	97.43%	94.41%	98.99%	1.25
Chordal-2	130	11	214–3 125	22 788–637 009	0.509	96.92%	94.61%	98.27%	0.98
Scale-free-1	130	48	1 000	1 996–67 360	0.039	10.27%	0.10%	55.23%	13.09
Scale-free-2	160	18	250–1 000	2 176–3 330	0.025	3.30%	0.37%	17.93%	3.74
New York	170	0	108–3 906	113–6 422	0.006	0.01%	0.00%	0.19%	0.04
Diamonds	130	0	111–2 751	111–2 751	0.006	0.00%	0.00%	0.00%	0.00
Job-shop	400	5	17–1 321	32–110 220	0.142	49.45%	0.00%	73.69%	16.91
HTN	121	0	500–625	748–1 599	0.007	0.02%	0.00%	0.15%	0.04

^a The number of degenerated STNs.

^b |V|, |E|, and \bar{D} : the number of vertices, the number of edges, and the average density of the constraint graphs of STNs respectively.

^c μ , min, max, and σ : the average, the minimal, the maximal, and the standard deviation value of reduction rate respectively.

very efficient in [13]. In light of that, and as we are more interested in obtaining results on redundancy reduction for STNs, we will not report on the computing time of our implementation.

Regarding the benchmark dataset, we employed the dataset of 1491 STNs used in the work of Planken et al. in [14]. The basic properties of that dataset are presented in Table 1. More details about the various natures of the dataset can be found in [14, Section 4]. The STNs vary from random scale-free networks and parts of the road network of New York City, to STNs generated from hierarchical task networks (HTNs) and job-shop scheduling problems. Note that Chordal-1 and Scale-free-1 contain STNs of a fixed number of variables, while Chordal-2 and Scale-free-2 contain STNs of a variant number of variables. It is worth noting that a small percentage of STNs are degenerated. To be exact, 100/1491 of the STNs are degenerated. The number of degenerated STNs varies per nature of dataset. For example, only 5 out of the 400 STNs of the job-shop scheduling problems dataset (Job-shop) are degenerated. On the other hand, over a third of the STNs of the 1000-variable scale-free STNs dataset are degenerated. In any case, all degenerated STNs were easily and minimally repaired by introducing a small weight ϵ to each degenerated interval $[a, a]$, hence, modifying each such interval to $[a - \epsilon, a + \epsilon]$. We also observe that Chordal-1 and Chordal-2, which are STNs whose constraint graphs are chordal, are the densest ones, followed by the STNs of the job-shop scheduling problems dataset, viz., Job-shop, and the scale-free networks datasets, viz., Scale-free-1 and Scale-free-2. On the contrary, STNs derived from the road network of New York City (New York), HTNs (HTN), and diamond-shaped networks (Diamonds) are extremely sparse, to the point of resembling trees, and are thus almost devoid of redundancy (as we will see in the evaluation to follow).

Regarding the results of our evaluation, Table 1 provides a detailed description of redundancy reduction per nature of dataset. Given the fact that chordal STNs are the densest ones, it is not surprising to obtain reduction rates lying between 94.41% to 98.99%, with a reduction rate of about 97% on average. With respect to the job-shop scheduling problems, we obtained an average reduction rate of 49.45%, while the reduction rate among the STNs reaches as high as 73.69%. Further, we obtain a 10.27% average reduction rate for scale-free STNs of a fixed number of variables, and a 3.3% average reduction rate for scale-free STNs of a variant number of variables. Similarly to the case of the job-scheduling problems, some of the STNs have less redundant constraints because they are already sparse. Finally, as expected due to their extreme sparseness, no significant reduction rate is obtained for any of the STNs derived

from the road network of New York City, the HTNs, or the diamond-shaped networks. For instance, all the constraints in the dataset of diamond-shaped networks are non-redundant.

As we strongly hinted earlier, the density of the constraint graph of a given STN correlates with the reduction that we can achieve in the number of constraints of that STN. The question arises of how big the correlation between densities and reduction rates really is, at least with respect to the dataset at hand. To answer this question, we use the Spearman’s rank correlation coefficient [21], a non-parametric test that is used to measure the degree of association between two variables. It assesses how well the relationship between two variables x and y can be described using a monotonic function. Like other correlation coefficients, this one varies between -1 and $+1$, with 0 implying no correlation. Correlations of -1 or $+1$ imply an exact monotonic relationship. Positive correlations imply that as x increases, so does y . Negative correlations imply that as x increases, y decreases. By using the average density \bar{D} and the average reduction rate μ as our variables, along with their respective values as provided in Table 1, we can obtain a Spearman’s rank correlation coefficient of 0.99, which demonstrates that each of our two variables is almost a perfect monotone function of the other. This result suggests that a strong correlation between densities and reduction rates exists, despite the various natures of the benchmark dataset.

6 CONCLUSION AND FUTURE WORK

In this paper, we investigated the redundancy problem of STNs. In particular, we showed that for any non-degenerated STN Γ , there is a unique minimal subset Γ^c of its constraints (i.e. the core of Γ) such that all of the constraints in Γ^c are not redundant in Γ , and Γ^c has the same set of solutions as Γ . We proposed an efficient algorithm to calculate the core, which runs in time linear in the number of triangles in a chordal graph that results from triangulating the constraint graph of Γ . Our experiments with a large benchmark dataset of STNs unveiled a vast number of redundant constraints, which suggests that the cores of the STNs exhibit a significant reduction in redundancy in practice. With respect to our dataset, we also identified a strong correlation between densities and reduction rates. For future work, it would be interesting to investigate whether that statistical dependence between our measures will hold if more datasets of various nature were to be considered.

ACKNOWLEDGEMENTS

We would like to thank the referees for their comments. This work was partially supported by the Alexander von Humboldt foundation, ARC DP120104159, and a PhD grant from the University of Artois and region Nord-Pas-de-Calais.

A PROOFS

Proposition 4. *Let Γ and Γ^Δ as well as E and E^Δ be specified as in Definition 7. Then for all $x, y \in V$ with $xy \in E^\Delta$ the following are equivalent:*

- (i) $\Gamma^\Delta(x, y)$ is redundant in Γ^Δ
- (ii) $\Gamma^\Delta(x, y) = \bigcap_{P \in \Pi(x, y, E^\Delta \setminus \{xy\})} \bigotimes \Gamma^\Delta(P)$
- (iii) $\Gamma^\Delta(x, y) = \bigcap_{xz, zy \in E^\Delta} \Gamma^\Delta(x, z) \otimes \Gamma^\Delta(z, y)$

Proof. (i) \Rightarrow (ii). Suppose $\Gamma^\Delta(x, y)$ is redundant in Γ^Δ . Then, by applying Lemma 3, we know that

$$\Gamma^\Delta(x, y) \supseteq \bigcap_{P \in \Pi(x, y, E^\Delta \setminus \{xy\})} \bigotimes \Gamma^\Delta(P).$$

On the other hand, $\Gamma^\Delta(x, y)$ is minimal in Γ^Δ . Therefore

$$\Gamma^\Delta(x, y) \subseteq \bigcap_{P \in \Pi(x, y, E^\Delta \setminus \{xy\})} \bigotimes \Gamma^\Delta(P).$$

and we showed (ii).

(ii) \Rightarrow (iii). Let P be a path between x and y on $E^\Delta \setminus \{xy\}$. Then, because G^Δ is chordal, there exists a $z' \in V$ with $(x, z'), (z', y) \in E^\Delta$, such that P is a concatenation of paths P_1 and P_2 , where P_1 is between x and z' and P_2 is between z' and y . Since $\Gamma^\Delta(x, z')$ and $\Gamma^\Delta(z', y)$ are minimal in Γ^Δ , we have

$$\bigotimes \Gamma^\Delta(P_1) \supseteq \Gamma^\Delta(x, z')$$

and

$$\bigotimes \Gamma^\Delta(P_2) \supseteq \Gamma^\Delta(z', y).$$

Hence,

$$\begin{aligned} \bigotimes \Gamma^\Delta(P) &= \bigotimes \Gamma^\Delta(P_1) \otimes \bigotimes \Gamma^\Delta(P_2) \\ &\supseteq \Gamma^\Delta(x, z') \otimes \Gamma^\Delta(z', y) \\ &\supseteq \bigcap_{xz, zy \in E^\Delta} \Gamma^\Delta(x, z) \otimes \Gamma^\Delta(z, y). \end{aligned}$$

Because the preceding statement holds for any path P between x and y on $E^\Delta \setminus \{xy\}$, we have

$$\bigcap_{P \in \Pi(x, y, E^\Delta \setminus \{xy\})} \bigotimes \Gamma^\Delta(P) \supseteq \bigcap_{xz, zy \in E^\Delta} \Gamma^\Delta(x, z) \otimes \Gamma^\Delta(z, y)$$

and by applying our assumption (ii), we have

$$\Gamma^\Delta(x, y) \supseteq \bigcap_{xz, zy \in E^\Delta} \Gamma^\Delta(x, z) \otimes \Gamma^\Delta(z, y).$$

The other inclusion

$$\Gamma^\Delta(x, y) \subseteq \bigcap_{xz, zy \in E^\Delta} \Gamma^\Delta(x, z) \otimes \Gamma^\Delta(z, y)$$

follows from the minimality of $\Gamma^\Delta(x, y)$ in Γ^Δ . Thus we showed (iii).

(iii) \Rightarrow (i). We first observe that for all $x, y, z \in V$ with $xy, xz, zy \in E^\Delta$

$$\Gamma^\Delta \setminus \{\Gamma^\Delta(x, y)\} \models \{\Gamma^\Delta(x, z), \Gamma^\Delta(z, y)\}$$

and

$$\{\Gamma^\Delta(x, z), \Gamma^\Delta(z, y)\} \models \Gamma^\Delta(x, z) \otimes \Gamma^\Delta(z, y)$$

and therefore

$$\Gamma^\Delta \setminus \{\Gamma^\Delta(x, y)\} \models \Gamma^\Delta(x, z) \otimes \Gamma^\Delta(z, y).$$

Thus we have

$$\Gamma^\Delta \setminus \{\Gamma^\Delta(x, y)\} \models \bigcap_{xz, zy \in E^\Delta} \Gamma^\Delta(x, z) \otimes \Gamma^\Delta(z, y)$$

By applying our assumption (iii) we then have

$$\Gamma^\Delta \setminus \{\Gamma^\Delta(x, y)\} \models \Gamma^\Delta(x, y)$$

and we showed (i). \square

Non-degenerated STNs have the following property.

Lemma 8. *Let Γ and Γ^Δ as well as E and E^Δ be specified as in Definition 7. Suppose Γ is not degenerated. Then there is a fixed $\varepsilon > 0$ such that for any cycle $C \in \Pi(x, x, E)$ we have that*

$$\bigotimes \Gamma(C) \supseteq (x [-\varepsilon, \varepsilon] x)$$

Proof. We first note that there exists an $\varepsilon > 0$ such that for any $x \in V$ and for any path $P \in \Pi(x, x, E^\Delta)$ of length 3, i.e. P is a triangle, we have $\bigotimes \Gamma^\Delta(P) \supseteq (x [-\varepsilon, \varepsilon] x)$. This is because there are only finitely many triangles in E^Δ and the composition of non-degenerated constraints is again non-degenerate. Then, because G^Δ is chordal, for any cycle $C \in \Pi(x, x, E)$ with $xy \in C$ there exists a $z \in V$ with $xz, zy \in E^\Delta$ such that $C = P_1 + P_2 + yx$ with $P_1 \in \Pi(x, z, E)$ and $P_2 \in \Pi(z, y, E)$. Then

$$\begin{aligned} \bigotimes \Gamma(C) &= \bigotimes \Gamma(P_1) \otimes \bigotimes \Gamma(P_2) \otimes \Gamma(y, x) \\ &\supseteq \bigotimes \Gamma^\Delta(P_1) \otimes \bigotimes \Gamma^\Delta(P_2) \otimes \Gamma^\Delta(y, x) \\ &\supseteq \Gamma^\Delta(x, z) \otimes \Gamma^\Delta(z, y) \otimes \Gamma^\Delta(y, x) \\ &\supseteq (x [-\varepsilon, \varepsilon] x) \end{aligned}$$

\square

Example 10. The STN Γ_2 in Figure 2b is not degenerated. Consider the paths $C_1 = (x_2x_4)$, $C_2 = (x_4x_3, x_3x_5, x_5x_4)$, $C_3 = (x_4x_3, x_3x_2)$. Note that $C = C_1 + C_2 + C_3$ and C_2 are cycles. Then $\bigotimes \Gamma_2(C) = \bigotimes \Gamma_2(C_1) \otimes \bigotimes \Gamma_2(C_2) \otimes \bigotimes \Gamma_2(C_3) = (x_2[11, 15]x_4) \otimes (x_4[-80, 41]x_4) \otimes (x_4[-52, 30]x_2) = (x_2[-121, 86]x_2) \supseteq (x_2[-41, 45]x_2) = \bigotimes \Gamma_2(C_1) \otimes \bigotimes \Gamma_2(C_3) \supseteq (x_2[-\varepsilon_C, \varepsilon_C]x_2)$, where we can choose $\varepsilon_C = 41$.

Proposition 5. *Let Γ and Γ^Δ as well as E and E^Δ be specified as in Definition 7. Furthermore, Γ is not degenerated. Then for all $x, y \in V$ with $xy \in E^\Delta$ the following are equivalent:*

- (i) $\Gamma(x, y)$ is redundant in Γ
- (ii) $\Gamma^\Delta(x, y)$ is redundant in Γ^Δ

$$(iii) \Gamma^\Delta(x, y) = \bigcap_{\substack{P \in \Pi(x, y, E) \\ |P| \geq 2}} \bigotimes \Gamma(P)$$

Proof. (i) \Rightarrow (ii). Because $\Gamma^\Delta \models \Gamma$, we have

$$\Gamma^\Delta \setminus \{\Gamma^\Delta(x, y)\} \models \Gamma \setminus \{\Gamma(x, y)\},$$

and since, by our assumption, $\Gamma \setminus \{\Gamma(x, y)\} \equiv \Gamma$, we have

$$\Gamma^\Delta \setminus \{\Gamma^\Delta(x, y)\} \models \Gamma.$$

Note that, $\Gamma^m \models \Gamma^\Delta$ and $\Gamma^\Delta \models \Gamma$. Then, because $\Gamma^m \equiv \Gamma$, we also have $\Gamma \models \Gamma^\Delta$. Therefore $\Gamma \equiv \Gamma^\Delta$ and we finally have

$$\Gamma^\Delta \setminus \{\Gamma^\Delta(x, y)\} \models \Gamma^\Delta,$$

and showed (ii).

(ii) \Rightarrow (iii). Suppose $\Gamma^\Delta(x, y)$ is redundant in Γ^Δ . Then, by Proposition 4 we know that

$$\Gamma^\Delta(x, y) = \bigcap_{xz, zy \in E^\Delta} \Gamma^\Delta(x, z) \otimes \Gamma^\Delta(z, y).$$

By applying Lemma 2 we then have

$$\begin{aligned} & \Gamma^\Delta(x, y) \\ &= \bigcap_{xz, zy \in E^\Delta} \left(\bigcap_{P_1 \in \Pi(x, z, E)} \bigotimes \Gamma(P_1) \right) \otimes \left(\bigcap_{P_2 \in \Pi(z, y, E)} \bigotimes \Gamma(P_2) \right) \end{aligned}$$

Since STP constraints are distributive (composition distributes over intersections) we have

$$\Gamma^\Delta(x, y) = \bigcap_{\substack{xz, zy \in E^\Delta \\ P_1 \in \Pi(x, z, E) \\ P_2 \in \Pi(z, y, E)}} \bigotimes \Gamma(P_1 + P_2) \quad (4)$$

On the other hand, because G^Δ is chordal, P is a path between x and y on E with $|P| \geq 2$ if and only if there exists a $z \in V$ with $(x, z), (z, y) \in E^\Delta, z \neq x, z \neq y$, such that P is a concatenation of paths P_1 and P_2 , where P_1 is between x and z and P_2 is between z and y . Hence we have

$$\bigcap_{\substack{P \in \Pi(x, y, E) \\ |P| \geq 2}} \bigotimes \Gamma(P) = \bigcap_{\substack{xz, zy \in E^\Delta \\ P_1 \in \Pi(x, z, E) \\ P_2 \in \Pi(z, y, E)}} \bigotimes \Gamma(P_1 + P_2) \quad (5)$$

From (4) and (5) it follows

$$\Gamma^\Delta(x, y) = \bigcap_{\substack{P \in \Pi(x, y, E) \\ |P| \geq 2}} \bigotimes \Gamma(P)$$

and we showed (iii).

(iii) \Rightarrow (i): Suppose

$$\Gamma^\Delta(x, y) = \bigcap_{\substack{P \in \Pi(x, y, E) \\ |P| \geq 2}} \bigotimes \Gamma(P)$$

We first note that we can partition the set of paths between x and y on E with $|P| \geq 2$ into two subsets \mathcal{P}_1 and \mathcal{P}_2 , where

$$\mathcal{P}_1 := \{P \in \Pi(x, y, E) \mid xy \in P, |P| \geq 2\},$$

$$\mathcal{P}_2 := \{P \in \Pi(x, y, E) \mid xy \notin P, |P| \geq 2\}.$$

Hence

$$\Gamma^\Delta(x, y) = \bigcap_{P \in \mathcal{P}_1 \cup \mathcal{P}_2} \bigotimes \Gamma(P). \quad (6)$$

We note that \mathcal{P}_1 consists of paths that are of the form $C_1 + (xy)$, $(xy) + C_2$, or $C_1 + (xy) + C_2$, where C_1 and C_2 are cycles on E with $|C_1| \geq 2$ and $|C_2| \geq 2$. Then, since Γ is not degenerated, by Lemma 8, there is an $\varepsilon > 0$ such that $\bigotimes \Gamma(C_1) \supseteq (x[-\varepsilon, \varepsilon]x)$, $\bigotimes \Gamma(C_2) \supseteq (y[-\varepsilon, \varepsilon]y)$, and we have, without loss of generality, for $P = C_1 + \{xy\} + C_2$

$$\begin{aligned} \bigotimes \Gamma(P) &= \bigotimes \Gamma(C_1 + \{xy\} + C_2) \\ &= \bigotimes \Gamma(C_1) \otimes \Gamma(x, y) \otimes \bigotimes \Gamma(C_2) \\ &\supseteq (x[-\varepsilon, \varepsilon]x) \otimes \Gamma(x, y) \\ &= (x[a - \varepsilon, b + \varepsilon]y), \end{aligned}$$

where $\Gamma(x, y) = (x[a, b]y)$. Hence,

$$\bigcap_{P \in \mathcal{P}_1} \bigotimes \Gamma(P) \supseteq (x[a - \varepsilon, b + \varepsilon]y). \quad (7)$$

Thus by (6) and (7) we have

$$\begin{aligned} \Gamma^\Delta(x, y) &= \bigcap_{P \in \mathcal{P}_1 \cup \mathcal{P}_2} \bigotimes \Gamma(P) \\ &= \bigcap_{P \in \mathcal{P}_1} \bigotimes \Gamma(P) \cap \bigcap_{P \in \mathcal{P}_2} \bigotimes \Gamma(P) \\ &\supseteq (x[a - \varepsilon, b + \varepsilon]y) \cap \bigcap_{P \in \mathcal{P}_2} \bigotimes \Gamma(P) \\ &\supseteq (x[a, b]y) \cap \bigcap_{P \in \mathcal{P}_2} \bigotimes \Gamma(P) \\ &= (x[a, b]y) \cap \bigcap_{P \in \mathcal{P}_1} \bigotimes \Gamma(P) \cap \bigcap_{P \in \mathcal{P}_2} \bigotimes \Gamma(P) \\ &= \Gamma(x, y) \cap \bigcap_{P \in \mathcal{P}_1 \cup \mathcal{P}_2} \bigotimes \Gamma(P) \\ &= \Gamma(x, y) \cap \Gamma^\Delta(x, y) \\ &= \Gamma^\Delta(x, y) \end{aligned}$$

Consequently we have

$$\begin{aligned} & (x[a - \varepsilon, b + \varepsilon]y) \cap \bigcap_{P \in \mathcal{P}_2} \bigotimes \Gamma(P) \\ &= (x[a, b]y) \cap \bigcap_{P \in \mathcal{P}_2} \bigotimes \Gamma(P). \end{aligned}$$

By setting $(x[c, d]y) := \bigcap_{P \in \mathcal{P}_2} \bigotimes \Gamma(P)$ we then have

$$(x[a - \varepsilon, b + \varepsilon]y) \cap (x[c, d]y) = (x[a, b]y) \cap (x[c, d]y),$$

which yields $[\max(a - \varepsilon, c), \min(b + \varepsilon, d)] = [\max(a, c), \min(b, d)]$ which holds only if $a \leq c$ and $b \geq d$, i.e. $(x[a, b]y) \supseteq (x[c, d]y)$. Therefore

$$\Gamma(x, y) \supseteq \bigcap_{P \in \mathcal{P}_2} \bigotimes \Gamma(P).$$

Since

$$\bigcap_{P \in \mathcal{P}_2} \bigotimes \Gamma(P) = \bigcap_{P \in \Pi(x, y, E) \setminus \{xy\}} \bigotimes \Gamma(P)$$

we have

$$\Gamma(x, y) \supseteq \bigcap_{P \in \Pi(x, y, E) \setminus \{xy\}} \bigotimes \Gamma(P)$$

and by Lemma 3 we have that $\Gamma(x, y)$ is redundant in Γ . \square

REFERENCES

- [1] James F. Allen, ‘Maintaining knowledge about temporal intervals’, *Commun. ACM*, **26**, 832–843, (1983).
- [2] Roman Barták, Robert A. Morris, and K. Brent Venable, ‘An introduction to constraint-based temporal reasoning’, *Synthesis Lectures on Artificial Intelligence and Machine Learning*, **8**(1), 1–121, (2014).
- [3] John L. Bresina, Ari K. Jnsson, Paul H. Morris, and Kanna Rajan, ‘Activity planning for the mars exploration rovers’, in *ICAPS*, pp. 40–49, (2005).
- [4] Patrick R. Conrad, Julie A. Shah, and Brian C. Williams, ‘Flexible execution of plans with choice’, in *ICAPS*, (2009).
- [5] Rina Dechter, Itay Meiri, and Judea Pearl, ‘Temporal constraint networks’, *Artif. Intell.*, **49**(1-3), 61–95, (1991).
- [6] Michael Fisher, Dov M. Gabbay, and L. Vila, *Handbook of temporal reasoning in artificial intelligence*, number 1 in Foundations of artificial intelligence, Elsevier, Amsterdam ; Boston, 1st edn., 2005.
- [7] Delbert Fulkerson and Oliver Gross, ‘Incidence matrices and interval graphs.’, *Pacific J. Math.*, **15**(3), 835–855, (1965).
- [8] Georg Gottlob and Christian G. Fermüller, ‘Removing redundancy from a clause’, *Artif. Intell.*, **61**(2), 263 – 289, (1993).
- [9] Sanjiang Li, Zhiguo Long, Weiming Liu, Matt Duckham, and Alan Both, ‘On redundant topological constraints’, *Artif. Intell.*, **225**, 51–76, (2015).
- [10] Paolo Liberatore, ‘Redundancy in logic I: CNF propositional formulae’, *Artif. Intell.*, **163**(2), 203–232, (2005).
- [11] Nicola Muscettola, Paul H. Morris, and Ioannis Tsamardinou, ‘Reformulating temporal plans for efficient execution’, in *KR*, pp. 444–452, (1998).
- [12] Seymour Parter, ‘The use of linear graphs in gauss elimination’, *SIAM review*, **3**(2), 119–130, (1961).
- [13] Léon Planken, Mathijs de Weerd, and Roman van der Krogt, ‘P³C: A new algorithm for the simple temporal problem’, in *ICAPS*, pp. 256–263, (2008).
- [14] Léon Planken, Mathijs de Weerd, and Roman van der Krogt, ‘Computing all-pairs shortest paths by leveraging low treewidth’, *J. Artif. Intell. Res.*, **43**, 353–388, (2012).
- [15] David A. Randell, Zhan Cui, and Anthony G. Cohn, ‘A spatial logic based on regions and connection’, in *KR*, pp. 165–176, (1992).
- [16] Francesca Rossi, Kristen Brent Venable, and Neil Yorke-Smith, ‘Uncertainty in soft temporal constraint problems: A general framework and controllability algorithms for the fuzzy case’, *J. Artif. Intell. Res.*, 617–674, (2006).
- [17] Julie A Shah and Brian C Williams, ‘Fast dynamic scheduling of disjunctive temporal constraint networks through incremental compilation’, in *ICAPS*, pp. 322–329, (2008).
- [18] Michael Sioutis, Sanjiang Li, and Jean-François Condotta, ‘Efficiently characterizing non-redundant constraints in large real world qualitative spatial networks’, in *IJCAI*, pp. 3229–3235, (2015).
- [19] Robert Endre Tarjan and Mihalis Yannakakis, ‘Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs’, *SIAM J. Comput.*, **13**(3), 566–579, (1984).
- [20] Marc B. Vilain and Henry A. Kautz, ‘Constraint propagation algorithms for temporal reasoning’, in *AAAI*, pp. 377–382, (1986).
- [21] Daniel Zwillinger and Stephen Kokoska, *CRC standard probability and statistics tables and formulae*, CRC Press, 1999.

On Metric Temporal Description Logics

Víctor Gutiérrez-Basulto¹ and Jean Christoph Jung¹ and Ana Ozaki²

Abstract. We introduce *metric* temporal description logics (*mTDLs*) as combinations of the classical description logic \mathcal{ALC} with (a) LTL^{bin} , an extension of the temporal logic LTL with succinctly represented intervals, and (b) metric temporal logic MTL, extending LTL^{bin} with capabilities to quantitatively reason about time delays. Our main contributions are algorithms and tight complexity bounds for the satisfiability problem in these *mTDLs*: For *mTDLs* based on (fragments of) LTL^{bin} , we establish complexity bounds ranging from EXPTIME to 2EXPSpace. For *mTDLs* based on (fragments of) MTL interpreted over the naturals, we establish complexity bounds ranging from EXPSpace to 2EXPSpace.

1 Introduction

Classical Description logics (DLs) are fragments of first-order logic aiming at the representation of and reasoning about knowledge. The importance of DLs lies in the fact that they are, arguably, the prime formalism to encode ontologies, e.g., they underpin the web ontology language OWL 2, the medical ontology SNOMED CT and the thesaurus of the US national cancer institute. It has been observed that in many domains where ontologies are used an implicit or explicit notion of *time* plays a central role [25]. As an instance, many terms in the medical domain are described making reference to temporal patterns; for example, the description of the autoimmune disease diabetes must specify that it might lead to glaucoma in the future. On the other hand, DLs were initially developed with the aim of capturing *static* knowledge. To address this shortcoming, over the last 20 years a plethora of temporal DLs (TDLs), extensions of DLs with an explicit temporal component, have been proposed [5, 25].

The most popular approach to constructing TDLs is to combine classical DLs with temporal logics such as LTL and CTL, and to provide a two-dimensional product-like semantics [29, 15, 25]—one dimension for time and the other for DL quantification. Temporal DLs of this kind support the definition of terms using, e.g., the temporal operators ‘at the next/previous point’ or ‘somewhere in the future/past’. As an example, in TDLs based on CTL we can use $\exists \text{has.Diabetes} \sqsubseteq E \diamond \exists \text{develops.Glaucoma}$ to say that ‘a patient with diabetes may develop glaucoma in the future’. The importance of TDLs based on LTL and CTL is witnessed by the vast amount of research conducted on the topic in the last decade; in particular, TDLs using expressive as well as lightweight DLs, with different levels of interaction between the components, have been investigated [31, 7, 25, 13, 17, 8, 18, 19]. Moreover, this sort of TDLs have been already successfully used in applications, e.g., to describe conceptual models capturing the evolution of databases over time [8]. However, their temporal constructs does not seem to always adequately cater for the needs of ontology designers and users. Indeed,

temporal primitives such as ‘eventually in the future’ might not be sufficiently precise for temporal conceptual modeling in an ontology. As an instance, in the medical domain ontological modelling often requires reference to concrete durations. Consider, for example, lyme disease: Affected patients develop a rash within 3-32 days after the infection. Since the infection can only occur after exposure to ticks, the concrete temporal interval of 3-32 days can be used to rule out lyme disease as a cause of rash.

Observe that, although it is possible to express eventuality within an interval by ‘unfolding’ all the timepoints represented in an interval, allowing intervals in the language with end-points in *binary* would result in an exponentially more succinct statement. For instance, for expressing that a patient “develops a rash eventually within 3-32 days”, in TDLs based on plain LTL, the ontology designer has to write $\bigcirc^3(\exists \text{has.Rash}) \sqcup \dots \sqcup \bigcirc^{32}(\exists \text{has.Rash})$, whereas the same can be expressed elegantly, and more succinctly by $\diamond_{[3,32]} \exists \text{has.Rash}$ in the logics studied in this paper. Despite the need of this feature, TDLs based on temporal logics succinctly capturing time intervals (to the best of our knowledge) have not yet been considered in the literature. It is important to note that TDLs based on Halpern and Shoham’s (*HS*) interval logic of Allen’s relations have been recently investigated [3, 9]. However, these TDLs are orthogonal to the ones investigated here because they are interval-based logics, i.e., intervals (instead of time points) are the basic time units.

The purpose of this paper is to initiate the study of metric TDLs (*mTDLs*) allowing for quantitative temporal reasoning. In particular, we are interested in TDLs merging qualitative temporal assertions together with quantitative constraints so as to get the benefits of the qualitative and quantitative abstraction levels. To this end, we consider TDLs based on (a) LTL^{bin} , the extension of LTL with succinctly represented intervals, and (b) the real-time metric temporal logic MTL, extending LTL^{bin} with capabilities to quantitatively reason about time delays. We look at TDLs that might prove useful for applications: we consider the traditional DL \mathcal{ALC} , make the (most-general) constant domain assumption, and apply temporal operators to concepts and, in the second part of the paper, to TBox statements. We do not apply temporal operators to roles since this typically leads to undecidability [25]. Our main contributions are algorithms for the satisfiability problem and complexity bounds.

Our study starts with *mTDLs* based on LTL^{bin} and temporal operators applied only to concepts. For example, in $LTL^{\text{bin}}_{\mathcal{ALC}}$ we can use

$$\exists \text{exposedTo.Tick} \sqsubseteq \square_{[3,32]} (\exists \text{has.Rash} \rightarrow \diamond_{[0,3]} \exists \text{gets.LymeDTest})$$

to say that ‘persons exposed to ticks whom develop a rash within 3-32 days after that must be tested for Lyme disease within three days.’

We have argued above that $LTL^{\text{bin}}_{\mathcal{ALC}}$ is not more expressive than $LTL_{\mathcal{ALC}}$; however, the (exponential!) translation does not give tight complexity bounds. More specifically, the translation yields a 2EX-

¹ University of Bremen, Germany; {victor.jeanjung}@tzi.de

² University of Liverpool, UK; anaozaki@liverpool.ac.uk

	propositional	combined with \mathcal{ALC} temporal operators on concepts	combined with \mathcal{ALC} temporal operators on concepts and TBoxes
LTL	PSPACE [30]	EXPTIME [25]	EXPSpace [15]
LTL ^{bin}	EXPSpace [1]	EXPSpace [Thm. 1]	2EXPSpace [Thm. 5]
LTL ^{0,∞}	PSPACE [24]	EXPTIME [Thm. 2]	EXPSpace [Thm. 7]
MTL	EXPSpace [1]	2EXPSpace [Thm. 3]	2EXPSpace [Thm. 6]
MTL ^{0,∞}	PSPACE [24]	EXPSpace [Thm. 4]	EXPSpace [Thm. 8]

Table 1: Overview of previous and **new** complexity results.

PTime upper bound because satisfiability in $LTL_{\mathcal{ALC}}$ can be done in EXPTIME [25]. In contrast, in our first main result, we develop an algorithm based on quasimodels in order to obtain EXPSpace-completeness for satisfiability in $LTL_{\mathcal{ALC}}^{\text{bin}}$. Recall that satisfiability in LTL^{bin} is also EXPSpace-complete [1], thus, the combination with \mathcal{ALC} is “for free.”

As the next step, we consider as temporal component metric temporal logic MTL [23], which extends LTL^{bin} by explicitly associating to each state a timestamp, allowing then for quantitative reasoning of time-delays. MTL has been intensively studied in the last 20 years; in particular, different semantics, syntactic restrictions and underlying time domains have been considered, for an overview see [27]. We will consider here MTL over the naturals with so-called *pointwise semantics*. Under this semantics we can see states as *observations*, say of a real-time system, that have an explicit discrete timestamp and consequently think of the time difference between two consecutive observations as a time-delay. For example, in $MTL_{\mathcal{ALC}}$ we use

$$\text{PhDStudent} \sqcap \neg \text{pays.Fees} \sqsubseteq \bigcirc_{[1,3]} (\exists \text{gets.Reminder} \sqcap (\diamond_{[0,7]} \exists \text{pays.Fees} \sqcup (\neg \text{access.Lab} \mathcal{U}_{[7,\infty)} \exists \text{pays.Fees})))$$

to express that, if the system observes that a PhD student has not paid the fees, then it should issue a reminder in the next system cycle (that is, observation) which is necessarily occurring within the next three time units; moreover, the student then should pay the fees within seven time units (for example days) or she does not have access to the lab until she pays.

Based on the aforementioned result for $LTL_{\mathcal{ALC}}^{\text{bin}}$ and the limited interaction of the dimensions, one could conjecture that the complexity of $MTL_{\mathcal{ALC}}$ is not higher than that of the components, and therefore EXPSpace-complete. Surprisingly, we show that this is not the case by establishing a 2EXPSpace lower bound, which is later shown to be tight.

We then turn our attention to the case where temporal operators are additionally applied to TBoxes. As for the basic case, we start by looking at $LTL_{\mathcal{ALC}}^{\text{bin}}$. Most interestingly, it can be shown that the aforementioned 2EXPSpace-hardness result can be lifted to $LTL_{\mathcal{ALC}}^{\text{bin}}$ temporal TBox satisfiability. Matching upper bounds for $LTL_{\mathcal{ALC}}^{\text{bin}}$ and $MTL_{\mathcal{ALC}}$ follow from the translation to the qualitative case $LTL_{\mathcal{ALC}}$, where TBox satisfiability is known to be EXPSpace-complete [25].

Based on similar observations for the propositional case [24], we finally looked at the restrictions $LTL_{\mathcal{ALC}}^{0,\infty}$ and $MTL_{\mathcal{ALC}}^{0,\infty}$ where intervals are only of the form $[0, c]$ or $[c, \infty]$. This is still expressive enough to succinctly model, for instance, time limits. We show that, indeed, the quasimodel technique can be leveraged to show that this leads to better complexity in many cases.

An overview of existing and new results is given in Table 1.

Missing proofs are provided in an extended version, available at www.informatik.uni-bremen.de/tcki/research/papers/GJO16.pdf.

2 Preliminaries

Intervals. We use standard notation for (open and closed) intervals, e.g., $[c_1, c_2]$ is the set of all $n \in \mathbb{N}$ with $c_1 \leq n < c_2$. It is thus clear what is meant with $n \in I$ and $I \subseteq I'$ for intervals I, I' .

$LTL_{\mathcal{ALC}}^{\text{bin}}$ **syntax.** $LTL_{\mathcal{ALC}}^{\text{bin}}$ is a TDL based on LTL and the classical DL \mathcal{ALC} . Let \mathbb{N}_C and \mathbb{N}_R be countably infinite sets of *concept* and *role names*, respectively. $LTL_{\mathcal{ALC}}^{\text{bin}}$ -concepts C, D are formed according to the rule:

$$C, D ::= A \mid \neg C \mid C \sqcap D \mid \exists r.C \mid \bigcirc C \mid CU_I D$$

where $A \in \mathbb{N}_C$, $r \in \mathbb{N}_R$, and I is an *interval* of the form $[c_1, c_2]$ or $[c_1, \infty)$ with $c_1, c_2 \in \mathbb{N}$ given in *binary*. We use standard Boolean and temporal abbreviations: $C \sqcup D$, $\forall r.C$, \top , $\diamond_I C$, and $\square_I C$ for $\neg(\neg C \sqcap \neg D)$, $\neg \exists r. \neg C$, $A \sqcup \neg A$, $\top \mathcal{U}_I C$, and $\neg \diamond_I \neg C$, respectively. We omit intervals of the form $[0, \infty)$ and write $CU D$ instead of $CU_{[0,\infty)} D$, and use the subscript $\cdot c$ to refer to intervals of the form $[c, c]$.

An $LTL_{\mathcal{ALC}}^{\text{bin}}$ TBox is a finite set of *concept inclusions* (CIs) $C \sqsubseteq D$ with C, D $LTL_{\mathcal{ALC}}^{\text{bin}}$ -concepts. We use $C \equiv D$ to refer to the two concept inclusions $C \sqsubseteq D$ and $D \sqsubseteq C$. The *size* of a TBox \mathcal{T} (a concept C) is the number of symbols required to write \mathcal{T} (C).

$LTL_{\mathcal{ALC}}^{\text{bin}}$ **semantics.** The semantics of $LTL_{\mathcal{ALC}}^{\text{bin}}$ is given in terms of *interpretations*, that is, structures $\mathfrak{J} = (\Delta^{\mathfrak{J}}, (\mathcal{I}_n)_{n \in \mathbb{N}})$, where each \mathcal{I}_n is a classical DL interpretation with domain $\Delta^{\mathfrak{J}}$: we have $A^{\mathcal{I}_n} \subseteq \Delta^{\mathfrak{J}}$ and $r^{\mathcal{I}_n} \subseteq \Delta^{\mathfrak{J}} \times \Delta^{\mathfrak{J}}$. We usually write $A^{\mathfrak{J},n}$ and $r^{\mathfrak{J},n}$ instead of $A^{\mathcal{I}_n}$ and $r^{\mathcal{I}_n}$, respectively. For instance, $d \in A^{\mathfrak{J},n}$ means that in the interpretation \mathfrak{J} , the object d is an instance of the concept name A at *time point* n . The stipulation that all time points share the same domain $\Delta^{\mathfrak{J}}$ is called the *constant domain assumption* (meaning that objects are not created or destroyed over time), and it is the most general choice in the sense that increasing, decreasing, and varying domains can all be reduced to it [15].

We now define the semantics of $LTL_{\mathcal{ALC}}^{\text{bin}}$ -concepts. To this end, we extend the mapping $\cdot^{\mathfrak{J},n}$ from concept names to complex $LTL_{\mathcal{ALC}}^{\text{bin}}$ -concepts as follows:

$$\begin{aligned} (\neg C)^{\mathfrak{J},n} &= \Delta^{\mathfrak{J}} \setminus C^{\mathfrak{J},n}, \\ (C \sqcap D)^{\mathfrak{J},n} &= C^{\mathfrak{J},n} \cap D^{\mathfrak{J},n}, \\ (\exists r.C)^{\mathfrak{J},n} &= \{d \in \Delta^{\mathfrak{J}} \mid \exists e \in C^{\mathfrak{J},n} \text{ with } (d, e) \in r^{\mathfrak{J},n}\}, \\ (\bigcirc C)^{\mathfrak{J},n} &= \{d \in \Delta^{\mathfrak{J}} \mid d \in C^{\mathfrak{J},n+1}\}, \\ (CU_I D)^{\mathfrak{J},n} &= \{d \in \Delta^{\mathfrak{J}} \mid \exists k > n : d \in D^{\mathfrak{J},k} \wedge k - n \in I \\ &\quad \wedge \forall m \in (n, k) : d \in C^{\mathfrak{J},m}\}. \end{aligned}$$

An interpretation \mathfrak{J} is a *model* of a concept C if $C^{\mathfrak{J},0} \neq \emptyset$; it is a model of a CI $C \sqsubseteq D$, written $\mathfrak{J} \models C \sqsubseteq D$, if $C^{\mathfrak{J},n} \subseteq D^{\mathfrak{J},n}$, for all $n \in \mathbb{N}$. We call \mathfrak{J} a *model of a TBox* \mathcal{T} , written $\mathfrak{J} \models \mathcal{T}$, if $\mathfrak{J} \models \alpha$ for all $\alpha \in \mathcal{T}$. Note that TBoxes are interpreted *globally* in the sense that all CIs must be satisfied at every time point.

Reasoning problem. We are interested in the reasoning problem of *satisfiability relative to global TBoxes* (throughout the paper only called *satisfiability*), that is, given an $\text{LTL}_{\mathcal{ALC}}^{\text{bin}}$ -concept C and TBox \mathcal{T} , decide whether C and \mathcal{T} have a common model.

Sequences. Throughout the paper we use sequences with the following notation. For a (possibly infinite) sequence $\sigma = \sigma(0)\sigma(1)\dots$, we write $\sigma^{\leq n}$ and $\sigma^{>n}$ for the *head* $\sigma(0)\sigma(1)\dots\sigma(n)$ and *tail* $\sigma(n+1)\sigma(n+2)\dots$ of σ , respectively. We also write $\sigma^{>i.\leq j}$ for the subsequence $\sigma(i+1)\dots\sigma(j)$ of σ . For a finite sequence σ_1 and a sequence σ_2 , we denote with $\sigma_1 * \sigma_2$, or just $\sigma_1\sigma_2$ if no confusion is possible, the *concatenation* of σ_1 and σ_2 . As usual, we define $\sigma^1 = \sigma$, $\sigma^{n+1} = \sigma\sigma^n$ and $\sigma^\omega = \sigma\sigma\sigma\dots$

3 $\text{LTL}_{\mathcal{ALC}}^{\text{bin}}$

We aim at devising algorithms and establishing tight complexity bounds for the satisfiability problem. We first concentrate on developing an algorithm for satisfiability in $\text{LTL}_{\mathcal{ALC}}^{\text{bin}}$, yielding a tight EXPSpace upper bound. The lower bound is a consequence of the following: (i) allowing the abbreviation \circ^n (meaning n consecutive ‘next’ operators) with n encoded in binary in LTL makes satisfiability checking EXPSpace-hard [1, 2] and (ii) \circ^n can be expressed in $\text{LTL}_{\mathcal{ALC}}^{\text{bin}}$ with \diamond_n .

In the second part of this section, we show that satisfiability in the restriction $\text{LTL}_{\mathcal{ALC}}^{0,\infty}$ of $\text{LTL}_{\mathcal{ALC}}^{\text{bin}}$ to intervals of the form $[0, c]$ or $[c, \infty)$ is complexity-wise better-behaved. In particular, it is EXPTIME-complete and thus not harder than in \mathcal{ALC} .

The main structure underlying our decision procedure are so-called *quasimodels*, which have been used for studying the satisfiability in various TDLs [31, 15, 7, 17]. In a nutshell, a quasimodel is an abstraction of an interpretation $\mathfrak{I} = (\Delta^\mathfrak{I}, (\mathcal{I}_n)_{n \in \mathbb{N}})$ in which each (possibly infinite) \mathcal{I}_n is replaced by a *quasistate*, that is, a *finite* set of *types*.

We show that quasimodels exhibit a monotonic behavior and apply regularity arguments to show membership in EXPSpace and EXPTIME, respectively. We show that to check satisfiability it suffices to consider quasimodels of the form:

$$S(0)S(1)\dots S(n)^\omega, \quad (1)$$

with $S(i) \supseteq S(i+1)$, for all $0 \leq i < n$, and n *double-* and *single-exponentially* bounded in the size of C and \mathcal{T} for $\text{LTL}_{\mathcal{ALC}}^{\text{bin}}$ and $\text{LTL}_{\mathcal{ALC}}^{0,\infty}$, respectively.

Note that a similar regularity condition holds for LTL in the sense that every satisfiable LTL formula has a regular model like (1) (with $S(i)$ propositional valuations) [26]. The main difference is that n is exponentially-bounded (satisfiability is thus PSPACE) and that a larger suffix could be the regular part repeating infinitely; in $\text{LTL}_{\mathcal{ALC}}^{\text{bin}}$, due to monotonicity, $S(n)$ is the only periodic set.

Throughout the section, we assume without loss of generality that the TBox \mathcal{T} is of the form $\{\top \sqsubseteq \mathcal{C}_\mathcal{T}\}$ and denote with $\text{sub}(C, \mathcal{T})$ the set of all subconcepts of C and $\mathcal{C}_\mathcal{T}$.

3.1 Full $\text{LTL}_{\mathcal{ALC}}^{\text{bin}}$

We start with introducing some required notation. Denote with $\text{cl}(C, \mathcal{T})$ the closure under single negations of the set:

$$\text{sub}(C, \mathcal{T}) \cup \{DUE \mid DU_{[c,\infty)}E \in \text{sub}(C, \mathcal{T})\}. \quad (\ddagger)$$

As usual, a *type* for C and \mathcal{T} is a subset $t \subseteq \text{cl}(C, \mathcal{T})$ such that:

- $D \in t$ iff $\neg D \notin t$, for all $\neg D \in \text{cl}(C, \mathcal{T})$;
- $D \sqcap E \in t$ iff $\{D, E\} \subseteq t$, for all $D \sqcap E \in \text{cl}(C, \mathcal{T})$;
- $C_\mathcal{T} \in t$.

We will use $\text{tp}(C, \mathcal{T})$ to denote the set of all types for C and \mathcal{T} and $\sharp_{C, \mathcal{T}}$ to denote the number of types, $|\text{tp}(C, \mathcal{T})|$.

We now describe a set of types that appropriately abstracts a classical description logic interpretation \mathcal{I}_n . A *quasistate* for C and \mathcal{T} is a set $Q \subseteq \text{tp}(C, \mathcal{T})$ of types such that:

- if $t \in Q$ and $\exists r.D \in t$, then there is $t' \in Q$ such that $\{D\} \cup \{\neg E \mid \neg \exists r.E \in t\} \subseteq t'$.

We next show how to temporally relate types in different quasistates; most importantly, regarding how temporal formulas of the form $\circ D$ and $DU_I E$ are captured. Let $\bar{t} = t(0)t(1)\dots \in \text{tp}(C, \mathcal{T})$ be a (possibly infinite) sequence of types. We say that \bar{t} *realizes* $DU_I E$ if there is $m \in I$ such that $E \in t(m)$ and, for all $0 < l < m$, we have $D \in t(l)$. From here on, we use $S = S(0)S(1)\dots$ to denote an infinite sequence of quasistates for C and \mathcal{T} . A *run* $r = r(0)r(1)\dots$ *through* S is a sequence of types for C and \mathcal{T} such that for all $n \geq 0$:

- (R1) $r(n) \in S(n)$;
- (R2) $\circ D \in r(n)$ iff $D \in r(n+1)$, for all $\circ D \in \text{cl}(C, \mathcal{T})$;
- (R3) $DU_I E \in r(n)$ iff $r^{\geq n}$ realizes $DU_I E$, for all $DU_I E \in \text{cl}(C, \mathcal{T})$.

Intuitively, a run is a sequence of types which characterizes the temporal evolution of a domain element.

We now have the ingredients to formally define a quasimodel. A *quasimodel* for C and \mathcal{T} is a pair (S, \mathfrak{R}) with \mathfrak{R} a set of runs through S such that:

- (Q1) $C \in t$ for some $t \in S(0)$; and
- (Q2) for all $t \in S(n)$, $n \geq 0$ there is a run $r \in \mathfrak{R}$ such that $r(n) = t$.

Intuitively, (Q1) ensures that C is witnessed at time point 0, and (Q2) ensures that each type has an appropriate temporal evolution through the quasimodel. We show in the appendix that concept satisfiability is characterized by the existence of a quasimodel for C and \mathcal{T} :

Lemma 1. *There is a model of C and \mathcal{T} iff there is a quasimodel for C and \mathcal{T} .*

This characterization, however, does not serve yet as the basis of an algorithm as both S and \mathfrak{R} are infinite. In the next step, we show that quasimodels can be assumed to have a certain regular shape. Henceforth, let K denote the largest constant occurring in C and \mathcal{T} (or 1 if none exist), and let $\ell_1 = (\sharp_{C, \mathcal{T}})^K + K$. We then have the following normal form of quasimodels.

Lemma 2. *There is a quasimodel for C and \mathcal{T} iff there is a quasimodel (S, \mathfrak{R}) for C and \mathcal{T} where S is of the form*

$$S = S_0^{n_0} \dots S_{m-1}^{n_{m-1}} S_m^\omega$$

for quasistates S_0, \dots, S_m with $S_i \supseteq S_{i+1}$, $0 \leq i < m$, and numbers $n_0, \dots, n_{m-1} < \ell_1$.

The proof of Lemma 2 (cf. appendix) proceeds in two steps. (i) We first show that we can extend any quasimodel such that a quasistate at time $i+1$ is contained in the quasistate at time i , that is, $S_{i+1} \subseteq S_i$, $i \geq 0$. (ii) We then show that if ℓ_1 consecutive quasistates coincide, that is, $S(i) = S(i+1) = \dots = S(i+\ell_1)$ for some $i \geq 0$, then

we can assume that all subsequent quasistates coincide as well, that is, $S(j) = S(i)$ for all $j \geq i$.

Obviously, the (strict!) containment condition on the S_i in Lemma 2 implies that m is at most $\sharp_{C,\mathcal{T}}$ since the $S(i)$ are non-empty sets of types. Moreover, note that, due to ℓ_1 , the length of the initial irregular part of S is double-exponentially bounded in the size of C and \mathcal{T} . Lemmas 1 and 2 give thus rise to the following non-deterministic procedure for checking concept satisfiability.

1. Non-deterministically choose $m < \sharp_{C,\mathcal{T}}$ non-empty sets of types $S_0 \supseteq \dots \supseteq S_m$ and a sequence n_0, \dots, n_{m-1} of binary numbers such that $n_i < \ell_1$ for all $0 \leq i < m$.
2. Verify that the sequence S defined as

$$S = S_0^{n_0} \dots S_{m-1}^{n_{m-1}} S_m^\omega$$

can be extended to a quasimodel for C and \mathcal{T} , that is, check that:

- (a) each S_i , $0 \leq i \leq m$, is a quasistate;
- (b) there is a $t \in S_0$ with $C \in t$;
- (c) for each $i \geq 0$ and $t \in S(i)$, there is a run r through S such that $r(i) = t$.

The procedure is obviously correct (given Lemmas 1 and 2), but involves a non-effective step: in 2(c), infinitely many tests have to be performed. It thus remains to show how to effectively execute 2(c). To this end, we show, in Lemma 3, that it suffices to check 2(c) for all $i \leq \sharp_{C,\mathcal{T}} \cdot \ell_1$, a double exponential number; then, in Lemma 4, we identify a certain regular form of runs, lending itself to implementation. For both Lemmas, let S be as in Lemma 2.

Lemma 3. *If the condition in 2(c) is satisfied for all $i \leq \sharp_{C,\mathcal{T}} \cdot \ell_1$, then it is satisfied for all $i \geq 0$.*

Proof. Similar to the proof of (ii) in Lemma 2. \square

Lemma 4. *If there is a r run through S with $r(i) = t$, for some $i \geq 0$, then there is a run r' through S which satisfies $r'(i) = t$ and is of the shape*

$$r' = s(0) \dots s(k_1) * (s'(0) \dots s'(k_2))^\omega$$

for types $s(0), \dots, s(k_1), s'(0), \dots, s'(k_2)$ and $k_1 \leq i + (\sharp_{C,\mathcal{T}})^K$, and $k_2 \leq |\text{cl}(C, \mathcal{T})| \cdot (\sharp_{C,\mathcal{T}})^K$.

Proof. We are going to use the following Claim.

Claim. If r is a run through S and $r^{\geq p-K, \leq p} = r^{\geq q-K, \leq q}$ for some $p < q$, then $r' = r^{\leq p} * r^{> q}$ is a run through S .

Proof of the Claim. We need to show that Conditions (R1) to (R3) hold for r' . Condition (R1) is an immediate consequence of the construction of r' and S . For (R2), we only need to check that for every $\circ D \in \text{cl}(C, \mathcal{T})$, $\circ D \in r'(n)$ iff $D \in r'(n+1)$, for all $n \geq 0$. This follows from the fact that $r(p) = r(q)$ and $r \in \mathfrak{R}$.

For (R3), we check concepts of the form $DU_I E \in \text{cl}(C, \mathcal{T})$. Note that since $r^{\geq p-K, \leq p} = r^{\geq q-K, \leq q}$, we have $r'^{\geq p-K} = r^{\geq q-K}$. Then, for $n \geq p-K$,

$$DU_I E \in r'(n) \text{ iff } r'^{\geq n} \text{ realizes } DU_I E.$$

From now on assume $n < p-K$. If $I = [c_1, c_2]$ then, since $c_1, c_2 \in [0, K]$, we cannot exceed p . Then, (R3) holds.

Now, consider $I = [c_1, \infty)$, where $c_1 \in [0, K]$. As $r'^{\geq p-K} = r^{\geq q-K}$, if already $r'^{\geq n, \leq p}$ realizes $DU_I E$, then $DU_I E \in r'(n)$. Otherwise, assume that $r'^{\geq n, \leq p}$ does not realize $DU_I E$. Then, for $n < p-K$, we have that $DU_I E \in r(n)$ iff $D \in r(n), \dots, r(p)$ and $DU_E \in r(p) = r'(p)$.

As $r'^{\leq p} = r^{\leq p}$ and $r'(p) = r(q)$, we have $DU_I E \in r'(n)$ iff $D \in r'(n), \dots, r'(p)$ and $r'^{\geq p}$ realizes DU_E . Then, for $n < p-K$, $DU_I E \in r'(n)$ iff $r'^{\geq n}$ realizes $DU_I E$. That is, (R3) holds.

This finishes the proof of the Claim.

An n -sequence of types is just a finite sequence of types $s(0) \dots s(n-1)$. For $k \in \mathbb{N} \cup \{\infty\}$, we say that an n -sequence $s(0) \dots s(n-1)$ appears k times in r if there are k distinct $j \geq 0$ such that $r(j+l) = s(l)$, for all $0 \leq l < n$.

Let r be a run through S with $r(i) = t$ and choose $k_1 \geq i$ minimal such that every K -sequence appearing in $r^{\geq k_1}$ appears infinitely often. We argue that it is without loss of generality that, between i and k_1 , every K -sequence appears at most once: by the Claim, we can cut sequences that appear more than once. As there are at most $(\sharp_{C,\mathcal{T}})^K$ such K -sequences, we can assume that $k_1 \leq i + (\sharp_{C,\mathcal{T}})^K$.

Now, let U be the set of $CU_I D \in r(k_1)$ that are not realized in $r^{\geq k_1, \leq k_1+K}$, and choose $n \geq k_1$ minimal such that $r^{\geq k_1, \leq k_1+K} = r^{\geq n, \leq n+K}$ and each $CU_I D \in U$ is realized in $r^{\geq k_1, < n}$. Let m_1 be minimal such that $r^{\geq k_1, \leq m_1}$ realizes some $DU_I E \in U$. Reasoning as above using the Claim yields that we can assume without loss of generality that $m_1 \leq k_1 + (\sharp_{C,\mathcal{T}})^K$. Let now $m_2 > m_1$ be minimal such that $r^{\geq k_1, \leq m_2}$ realizes two $CU_I D \in U$. As before, we can show that without loss of generality $m_2 \leq k_1 + 2(\sharp_{C,\mathcal{T}})^K$. Continuing this reasoning, we can conclude that, without loss of generality, $n \leq k_1 + |\text{cl}(C, \mathcal{T})| \cdot (\sharp_{C,\mathcal{T}})^K$.

It is now routine to verify that $r' = r^{< k_1} * (r^{\geq k_1, < k_2})^\omega$, with $k_2 = n - k_1$, is a run through S . \square

We are now in a position to show that the above procedure can be implemented using only (non-deterministic) exponential space. Obviously, the sets S_0, \dots, S_m and the numbers n_0, \dots, n_{m-1} can be stored in exponential space. Moreover, steps 2(a) and 2(b) can clearly be checked in exponential space. For 2(c), Lemma 3 implies that at most $\sharp_{C,\mathcal{T}} \cdot \sharp_{C,\mathcal{T}} \cdot \ell_1$, that is, double-exponentially many, pairs (t, i) have to be considered, but only one at a time. Finally, (the proof of) Lemma 4 enables the following algorithm for checking the existence of a run. First, guess binary numbers k_1, k_2 as in Lemma 4; then, guess a run in the form of the lemma. For the latter, proceed in a ‘‘sliding window’’ fashion: keep K consecutive types and verify (R1)-(R3) for the first type in the sequence, then drop that type, guess the next type, and continue. For detecting the loop, store the sequence $r(k_1), \dots, r(k_1 + K)$ and verify that it appears again at $r(k_1 + k_2), \dots, r(k_1 + k_2 + K)$ and, moreover, that each $DU_I E \in r(k_1)$ is realized before k_2 . We conclude the desired result.

Theorem 1. *Satisfiability in $\text{LTL}_{\text{ALC}}^{\text{bin}}$ is EXPSPACE-complete.*

3.2 $\text{LTL}_{\text{ALC}}^{0,\infty}$

In this section, we consider $\text{LTL}_{\text{ALC}}^{0,\infty}$, a well-behaved, yet expressive, fragment of $\text{LTL}_{\text{ALC}}^{\text{bin}}$ in which intervals can only be of the form $[0, c]$ or $[c, \infty)$. This sort of intervals is useful to set maximum (deadline) points and minimum (initial) ones. For example, the CI

$\text{PhDStudent} \sqcap \exists \text{defends.Thesis} \sqsubseteq \diamond_{[0,4]} \exists \text{submits.RevisedThesis}$

says that ‘PhD students who defend their thesis must submit a revised version within 4 weeks’.

We show that allowing intervals of this restricted form does not increase the complexity of satisfiability compared to \mathcal{ALC} or $\text{LTL}_{\mathcal{ALC}}$, for both of which concept satisfiability is EXPTIME-complete [25]. We concentrate again on the upper bound since EXPTIME-hardness follows from satisfiability in \mathcal{ALC} [12].

Our algorithm relies again on quasimodels; however, we will slightly adapt the definition of types to address the restricted intervals. As a consequence, it will suffice to consider quasimodels of the form (1) where n is only *single-exponentially* bounded, finally yielding an EXPTIME decision procedure.

We first adapt the notion of a type. Instead of (‡), we define $\text{cl}(C, \mathcal{T})$ as the closure under single negations of $\text{sub}(C, \mathcal{T})$ extended with

$$\{DU_{[0,c]}E, DU_{[c,\infty)}E \mid c \in [0, K], DU_I E \in \text{sub}(C, \mathcal{T})\}.$$

Based on this, it is straightforward to show that Lemma 2 remains true for $\text{LTL}_{\mathcal{ALC}}^{0,\infty}$.

Note that there are now double-exponentially many types which typically prohibits an EXPTIME decision procedure based on type elimination [28]. However, it is easy to see that a type t appearing in some quasimodel satisfies the following property.

(P) For every $DU_{I_1}E, DU_{I_2}E \in \text{cl}(C, \mathcal{T})$ with $I_1 \subseteq I_2$, we have $DU_{I_1}E \in t$ implies $DU_{I_2}E \in t$.

To see (P), fix some quasimodel (S, \mathfrak{R}) for C and \mathcal{T} and assume that $DU_{I_1}E \in t$ for some $t \in S(n)$, $n \geq 0$. By Condition (Q2), there is a run $r \in \mathfrak{R}$ such that $r(n) = t$. As $DU_{I_1}E \in t$, by Condition (R3), $r^{\geq n}$ realizes $DU_{I_1}E$. Since $I_1 \subseteq I_2$, $r^{\geq n}$ also realizes $DU_{I_2}E$. By Condition (R3) again, $DU_{I_2}E \in r(n)$.

Thus, it suffices to consider only types that satisfy (P), whose number $\sharp_{C, \mathcal{T}}$ is bounded by $(2 \cdot K)^{2|\text{sub}(C, \mathcal{T})|}$, that is, exponential. From now on assume w.l.o.g. that $\text{tp}(C, \mathcal{T})$ is the set of types in which (P) holds. The next lemma shows that we can assume that our quasimodels reach a periodic quasistate after at most exponentially many quasistates.

Lemma 5. *There is a quasimodel for C and \mathcal{T} iff there is a quasimodel (S, \mathfrak{R}) for C and \mathcal{T} of the form*

$$S = S_0 \dots S_{n-1}(S_n)^\omega,$$

for quasistates S_0, \dots, S_n with $S_i \supseteq S_{i+1}$, $0 \leq i < n \leq \sharp_{C, \mathcal{T}}$.

The proof proceeds in two steps, as in Lemma 2. In the first step we modify our quasimodel so that each quasistate at time $i + 1$ is contained in the quasistate at time i . But now, in the second step, we show that if *two* consecutive quasistates coincide, that is, $S(i) = S(i + 1)$ for some $i \geq 0$, then we can assume that all subsequent quasistates coincide as well, that is, $S(j) = S(i)$ for all $j \geq i$.

Based on Lemma 5, we now present an algorithm that performs type elimination, similar to what has been done for $\text{LTL}_{\mathcal{ALC}}$ [25].

Define $\rho(n) = \min\{\sharp_{C, \mathcal{T}}, n\}$, for all $n \geq 0$, and, moreover, define an operation “ -1 ” on intervals as follows: $[0, c] - 1 = [0, c - 1]$ and $[c + 1, \infty) - 1 = [c, \infty)$, for all $c \geq 0$, and $[0, \infty) - 1 = [0, \infty)$. We say that types t and t' are *compatible* if the following holds:

- $\circ D \in t$ iff $D \in t'$, for all $\circ D \in \text{cl}(C, \mathcal{T})$; and
- $DU_I E \in t$ iff either (the sequence) tt' realizes $DU_I E$, or $\{D, DU_{I-1}E\} \subseteq t'$, for all $DU_I E \in \text{cl}(C, \mathcal{T})$.

The algorithm starts with sets

$$S_0, \dots, S_{n-1}, S_n$$

where $n = \sharp_{C, \mathcal{T}}$ and each S_i is initially set to $\text{tp}(C, \mathcal{T})$. We then exhaustively eliminate types t from some S_i , $0 \leq i \leq n$ if t violates one of the following conditions:

- (T1) for all $\exists r. D \in t$, there is $t' \in S_i$ such that $\{D\} \cup \{-E \mid \neg \exists r. E \in t\} \subseteq t'$;
- (T2) there is $t' \in S_{\rho(i+1)}$ such that t and t' are compatible;
- (T3) if $i > 0$, there is $t' \in S_{i-1}$ such that t' and t are compatible;
- (T4) for all $DU_I E \in t$, there is $k \geq 0$ and a sequence

$$t_1 \in S_{\rho(i+1)}, \dots, t_k \in S_{\rho(i+k)}$$

such that $t_0 \dots t_k$ (with $t_0 = t$) realizes $DU_I E$, and t_l and t_{l+1} are compatible, for all $0 \leq l < k$.

Before giving details on how to implement the conditions, especially (T4), we finish the description of the algorithm and show correctness. The algorithm stops when no further types can be eliminated. It returns ‘satisfiable’ if there is a surviving $t \in S_0$ with $C \in t$, and ‘unsatisfiable’, otherwise.

Lemma 6. *The algorithm returns ‘satisfiable’ iff there is a quasimodel for C and \mathcal{T} .*

Proof. For (\Rightarrow), let S_0^*, \dots, S_n^* be the result of the type elimination procedure. Define (S^*, \mathfrak{R}) with $S^* = S_0^* \dots S_{n-1}^*(S_n^*)^\omega$ and \mathfrak{R} as the set of all sequences r of types such that, for all $i \geq 0$:

1. $r(i) \in S^*(i)$;
2. $r(i)$ and $r(i + 1)$ are compatible; and
3. $DU_I E \in r(i)$ iff $r^{\geq i}$ realizes $DU_I E$, for all $DU_I E \in \text{cl}(C, \mathcal{T})$.

We now argue that (S^*, \mathfrak{R}) is a quasimodel. By (T1), the sets S_0^*, \dots, S_n^* generated by the algorithm are quasistates, so S^* is a sequence of quasistates. By assumption, there is $t \in S_0^*$ with $C \in t$, which gives us (Q1). By definition of \mathfrak{R} , we have that every $r \in \mathfrak{R}$ is a run through S^* (see (R1)-(R3)). Then, for (Q2), we only need to see that for every $t \in S^*(j)$ there is $r \in \mathfrak{R}$ such that $r(j) = t$, $j \in \mathbb{N}$. Let $r' = r^{\leq j}$ be a sequence of types such that all consecutive types in r' are compatible and $r'(j) = t$. By (T3) such sequence exists. We now extend this run using (T2) and (T4). Assume that there is no $DU_I E \in r'(j)$. Then by (T2) there is $t' \in S^*(j + 1)$ such that t and t' are compatible. So we extend r' with t' . Now assume there is $DU_I E \in r'(j)$. By (T4) there is a minimal sequence $t_0 t_1 \dots t_k$ of types that realizes all $DU_I E \in r'(j)$. We extend r' with t_1 . Continuing with this process one can ensure the existence of an infinite sequence satisfying all conditions of the sequences in \mathfrak{R} .

For the other direction (\Leftarrow), assume there is a quasimodel, which is without loss of generality of the form $S' = S'_0 \dots S'_{n-1}(S'_n)^\omega$, by Lemma 5. Let S_0^*, \dots, S_n^* be the result of the type elimination. It is routine to verify that $S'_i \subseteq S_i^*$, $0 \leq i \leq n$ by showing that no type in S'_i violates (T1)-(T4). Clearly, each type satisfies (T1), since (S', \mathfrak{R}) is a quasimodel. Moreover, conditions (T2)-(T4) are consequences of the existence of runs through each type.

Observe finally that, by (Q1), there is some $t \in S'_0$, thus $t \in S_0^*$, with $C \in t$, that is, the algorithm returns ‘satisfiable’. \square

It is not hard to see that the algorithm runs in exponential time. The maintained sets of types have initially exponential size and in every step some type is eliminated. Conditions (T1)-(T3) can clearly be checked in exponential time. Finally, (T4) can be cast as a reachability problem, which can be solved in polynomial time, in the following (exponentially sized) graph: vertices (t, i) for all $t \in S_i$ and edges between (t, i) and $(t', \rho(i + 1))$ iff t and t' compatible. We thus conclude:

Theorem 2. *Satisfiability in $\text{LTL}_{\mathcal{ALCC}}^{0,\infty}$ is EXPTIME-complete.*

4 $\text{MTL}_{\mathcal{ALCC}}$

In this section, we investigate a TDL that emerges from combining the real-time logic MTL (over the naturals) and \mathcal{ALCC} . $\text{MTL}_{\mathcal{ALCC}}$ -concepts are formed according to the following rule

$$C, D ::= A \mid \neg C \mid C \sqcap D \mid \exists r.C \mid \circ_I C \mid \text{CU}_I D,$$

where $A \in \mathbf{N}_C$, $r \in \mathbf{N}_R$, and I is an interval.

Note that $\text{MTL}_{\mathcal{ALCC}}$ -concepts are formed just like $\text{LTL}_{\mathcal{ALCC}}^{\text{bin}}$ -concepts except for the constructor \circ_I . The main difference between $\text{LTL}_{\mathcal{ALCC}}^{\text{bin}}$ and $\text{MTL}_{\mathcal{ALCC}}$ lies in their semantics: $\text{MTL}_{\mathcal{ALCC}}$ is a timed extension of $\text{LTL}_{\mathcal{ALCC}}^{\text{bin}}$, in other words, each interpretation in $(\mathcal{I}_n)_{n \in \mathbb{N}}$ explicitly refers to its time (think of it as the reading of a 'fictitious discrete clock') allowing to quantitatively reason about time delays.

Formally, a *timed interpretation* \mathcal{J} is a tuple $(\Delta^{\mathcal{J}}, (\mathcal{I}_n)_{n \in \mathbb{N}}, \tau)$ with $\tau : \mathbb{N} \rightarrow \mathbb{N}$ a mapping with $\tau(n) < \tau(n+1)$, for all $n \in \mathbb{N}$, which specifies that the n -th interpretation happens to be at time point $\tau(n)$. Note that there might be *gaps* between two interpretations, e.g., when $\tau(3) = 8$ and $\tau(4) = 10$, then there is no interpretation at time point 9. Intuively, we view $(\mathcal{I}_n, \tau(n))_{n \geq 0}$ as a *sequence of observations*, for instance, in a real-time system, and then understand the difference $\tau(n+1) - \tau(n)$ as the time delay between observations n and $n+1$.

The interpretation function $\cdot^{\mathcal{J},n}$ is lifted to complex concepts as in Section 2 for the constructors \neg , \sqcap , and $\exists r.C$. For \circ_I and U_I , it is defined as follows:

$$\begin{aligned} (\circ_I C)^{\mathcal{J},n} &= \{d \mid d \in C^{\mathcal{J},n+1} \wedge \tau(n+1) - \tau(n) \in I\}, \\ (\text{CU}_I D)^{\mathcal{J},n} &= \{d \mid \exists k > n : d \in D^{\mathcal{J},k} \wedge \tau(k) - \tau(n) \in I \\ &\quad \wedge \forall m \in (n, k) : d \in C^{\mathcal{J},m}\}. \end{aligned}$$

One could expect that, just like for $\text{LTL}_{\mathcal{ALCC}}^{\text{bin}}$, the complexity of satisfiability in $\text{MTL}_{\mathcal{ALCC}}$ is not higher than in the components; in particular, EXPSpace-complete as in MTL [1]. Surprisingly, we prove that there is an exponential jump in the complexity; the main reason for such an increase is that, due to slightly different semantics, the independence of elements in each \mathcal{I}_n is lost.

Theorem 3. *Satisfiability in $\text{MTL}_{\mathcal{ALCC}}$ is 2EXPSpace-complete.*

We prove here only the lower bound. The upper bound will follow from a more general result, see Theorem 6 in Section 5.

Proof. We reduce the word problem of a double-exponentially space-bounded deterministic Turing machine. Fix that TM $\mathfrak{A} = (Q, \Sigma, \Gamma, \delta, q_0, F)$ with $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{l, r\}$ and assume that \mathfrak{A} is 2^{2^n} -space bounded on inputs of length n . Let $Q' = Q \cup \{\bar{q}\}$, $k = |\Gamma \times Q'| + 1$ and fix some bijection $\pi : [1, k-1] \rightarrow \Gamma \times Q'$. We are going to use the following symbols:

- **Tape**, to mark the tape cells;
- $A_{a,q}$, $a \in \Gamma$, $q \in Q'$, to label cells with a symbol a and a state q ; \bar{q} expresses that the head is somewhere else.

Recall that we use the abbreviations \diamond_i and \circ_i instead of $\diamond_{[i,i]}$ and $\circ_{[i,i]}$, and just \diamond instead of $\diamond_{[0,\infty)}$. For inputs of length n , we will construct a TBox \mathcal{T}_n , whose basic ingredients are the following concept inclusions:

$$\text{Tape} \sqsubseteq \diamond_{[0,k]} \text{Tape} \sqcap \square_{[0,k-1]} \neg \text{Tape} \quad (2)$$

$$\text{Tape} \sqsubseteq \circ_{[0,k-1]} \top \quad (3)$$

$$\circ_i \top \equiv A_{\pi(i)}, \quad \text{for all } i \in [1, k-1] \quad (4)$$

Intuitively, using CI (2), we enforce that every k -th time point is labeled with **Tape**. By CI (3), we express that, if **Tape** is observed, the next observation is due within 1 to $k-1$ time points, but there is a choice. Finally, using CI (4), we *globally* mark all domain elements in a world, depending on the delay of the next observation, with some $A_{a,q}$, that is, information about state and tape symbol.

It remains to show how to synchronize consecutive configurations. Basically, the technique goes back to the following well-known lemma [21, Lemma 3.3], which is based on [20, Lemma 4.1] itself.

Lemma 7. *For each $n \geq 1$, there is a satisfiable formula φ_n in propositional temporal logic extended with \circ^n , n in binary, of size $O(n)$, and some $M \geq 0$ such that $\varphi_n \models \circ^m p_2$ iff $m = M + j \cdot 2^n \cdot 2^{2^n}$, for some $j \geq 0$.*

Using (the proof of) this well-known result, one can define a concept C_n that satisfies an analogous property, namely

$$\mathcal{T}_n \models C_n \sqsubseteq \diamond_m P_2 \quad \text{iff} \quad m = j \cdot k \cdot 2^n \cdot 2^{2^n}, \text{ for some } j.$$

We use this concept C_n (without giving details on the shape of C_n) to describe the remaining relevant parts in \mathcal{T}_n . We include the following concept inclusions:

- For $a \in \Gamma$ and \bar{q} :

$$\text{Tape} \sqcap A_{a,\bar{q}} \sqsubseteq \exists r.(C_n \sqcap \neg P_2 \mathcal{U}(P_2 \sqcap \bigsqcup_{q' \in Q'} A_{a,q'})) \quad (5)$$

- For $a \in \Gamma$, $q \in Q$, and $\delta(q, a) = (-, b, -)$:

$$\text{Tape} \sqcap A_{a,q} \sqsubseteq \exists r.(C_n \sqcap \neg P_2 \mathcal{U}(P_2 \sqcap A_{b,\bar{q}})) \quad (6)$$

- For $a \in \Gamma$, $q \in Q$, and $\delta(q, a) = (q', -, r)$:

$$\text{Tape} \sqcap A_{a,q} \sqsubseteq \diamond_k \exists r.(C_n \sqcap \neg P_2 \mathcal{U}(P_2 \sqcap \bigsqcup_{b \in \Gamma} A_{b,q'})) \quad (7)$$

- For $a \in \Gamma$, $q \in Q$, and $\delta(q, a) = (q', -, l)$:

$$\text{Tape} \sqcap \diamond_k A_{a,q} \sqsubseteq \exists r.(C_n \sqcap \neg P_2 \mathcal{U}(P_2 \sqcap \bigsqcup_{b \in \Gamma} A_{b,q'})) \quad (8)$$

Let $N = k \cdot 2^n \cdot 2^{2^n}$. Intuitively, CI (5) states that a world labeled with \bar{q} is labeled with the same symbol in the next configuration, that is, N tape cells later. CI (6) ensures that, if a world is labeled with (a, q) , then the corresponding world N tape cells later is labeled with b when $\delta(q, a) = (-, b, -)$; the corresponding state is \bar{q} as the head moves left or right. Finally, CIs (7) and (8) make sure that the head is moved according to the transition. It remains to ensure that the non-head worlds are labeled with \bar{q} . For this, one has to take into account the environment of a cell, as illustrated by the following CI:

$$\begin{aligned} \text{Tape} \sqcap A_{a_1,q_1} \sqcap \diamond_k (\neg P_2 \sqcap A_{a_2,q_2} \sqcap \diamond_k (\neg P_2 \sqcap A_{a_3,q_3})) \sqsubseteq \\ \diamond_k \exists r.(C_n \sqcap \neg P_2 \mathcal{U}(P_2 \sqcap \bigsqcup_{b \in \Gamma} A_{b,\bar{q}})), \end{aligned}$$

if $q_1 = q_2 = q_3 = \bar{q}$ or $\delta(q_1, a_1) = (-, -, l)$ or $\delta(q_3, a_3) = (-, -, r)$.

The remaining cases are similar. In particular, at cells close to the left or right border of a configuration, it suffices to take a smaller environment into account.

Now, let $w = a_1 \cdots a_n$ be some input word for \mathfrak{A} . Define a concept C_w by taking:

$$\begin{aligned} C_w = \text{Tape} \sqcap C_n \sqcap A_{a_1,q_0} \sqcap \prod_{i=1}^{n-1} \diamond_{ik} A_{a_{i+1},\bar{q}} \sqcap \\ \diamond_{(n-1)k} (A_{\bar{q},\bar{q}} \mathcal{U} P_2) \sqcap \diamond \bigsqcup_{a \in \Gamma, q \in F} A_{a,q}. \end{aligned}$$

Intuitively, Tape ensures a computation is initiated, C_n ensures that the tape is separated into configurations, A_{a_1, q_0} and the big conjunction enforces that the input word is written on the tape and $A_{b, \bar{q}} \mathcal{U}P_2$ ensures that the remaining cells are labeled with blank b and are non-head states. Finally, the last conjunct expresses that a final state is reachable. Based on the construction, it is not hard to verify the following claim, which finishes the reduction.

Claim. \mathfrak{A} accepts w of length n if there is a model of C_w and \mathcal{T}_n . \square

4.1 $\text{MTL}_{\mathcal{ALC}}^{0, \infty}$

Restricting the intervals to the form $[0, c]$ and $[c, \infty)$ leads to better complexity also for $\text{MTL}_{\mathcal{ALC}}$; however, not to EXPTIME as for $\text{LTL}_{\mathcal{ALC}}^{0, \infty}$. To see this, we sketch here how to adapt the reduction used in the previous theorem to get an EXPSpace-lower bound. A matching upper bound follows from Theorem 8 below.

Recall that CIs (2)-(4) provide the central idea of the reduction. While (2) and (3) are already in $\text{MTL}_{\mathcal{ALC}}^{0, \infty}$, we replace (4) with CIs

$$\neg \diamond_{[0, i]} \top \sqcap \prod_{l=i+2}^{k-1} \neg A_{\pi(l)} \sqsubseteq A_{\pi(i+1)}, \quad \text{and} \quad (9)$$

$$A_{\pi(i)} \sqsubseteq \neg \diamond_{[0, i-1]} \top, \quad (10)$$

for all $0 \leq i < k-1$. Intuitively, (9) expresses that if there is a gap of at least i (realized by $\neg \diamond_{[0, i]} \top$) and all $A_{\pi(l)}$ for $l > i+1$ are not satisfied, that is, there is no larger gap, conclude $A_{\pi(i+1)}$. Together with (10), this implies that, again, a unique $A_{a, q}$ is satisfied for all domain elements in a world. Note that, as there is a fixed Turing machine with an EXPSpace-hard word problem, k is fixed and we do not require succinct encoding here.

The remainder of the above proof deals with synchronizing information between consecutive configurations. While the concept C_n can certainly not be defined in $\text{MTL}_{\mathcal{ALC}}^{0, \infty}$, we can use the succinct intervals to communicate between tape cells that are exponentially far away. For instance, we can mark every $N = k \cdot 2^n$ -th time point with a concept name X by $X \sqsubseteq \diamond_{[0, N]} X \sqcap \square_{[0, N-1]} \neg X$. We thus get:

Theorem 4. *Satisfiability in $\text{MTL}_{\mathcal{ALC}}^{0, \infty}$ is EXPSpace-complete.*

5 Temporal TBoxes

We now take a look at the case where temporal operators can also be applied to concept inclusions in the TBox, which adds means for expressing dynamics of global information, e.g., in norms.

5.1 Temporal TBoxes in $\text{LTL}_{\mathcal{ALC}}^{\text{bin}}$

Temporal $\text{LTL}_{\mathcal{ALC}}^{\text{bin}}$ -TBoxes are defined by the following grammar:

$$\varphi, \psi ::= C \sqsubseteq D \mid \neg \varphi \mid \varphi \wedge \psi \mid \bigcirc \varphi \mid \varphi \mathcal{U}_I \psi,$$

where C, D are $\text{LTL}_{\mathcal{ALC}}^{\text{bin}}$ -concepts, I an interval. We define the truth relation $\mathfrak{J}, n \models \varphi$ (with \mathfrak{J} an interpretation and $n \in \mathbb{N}$ a time point) by starting with $\mathfrak{J}, n \models C \sqsubseteq D$ iff $C^{\mathfrak{J}, n} \subseteq D^{\mathfrak{J}, n}$, and extending it to the complex TBox formulas analogously to Section 2; e.g., $\mathfrak{J}, n \models \bigcirc \varphi$ iff $\mathfrak{J}, n+1 \models \varphi$. \mathfrak{J} is a model of a temporal $\text{LTL}_{\mathcal{ALC}}^{\text{bin}}$ -TBox φ if $\mathfrak{J}, 0 \models \varphi$.

We are concerned with the problem of *temporal TBox satisfiability*, that is, the problem of deciding whether a given temporal TBox φ has a model. Note that, in contrast to Section 3, a concept is not part of the input because there is a model of a concept C and a temporal TBox φ if and only if the temporal TBox $\neg(\top \sqsubseteq \neg C) \wedge \varphi$ is satisfiable.

Temporal TBoxes are useful to set the dynamics of protocols or norms. For example, the temporal $\text{LTL}_{\mathcal{ALC}}^{\text{bin}}$ -TBox

$$\diamond_{[4, 4]} (\text{PhDStud} \sqcap \exists \text{defends.Thesis} \sqcap \neg \exists \text{has.ConfPub} \sqsubseteq \bigcirc \exists \text{takes.ValidationExam})$$

says that after 4 years there will be a norm stating that all PhD students who defend their thesis and do not have a conference publication will need to take a validation exam the year after that.

The first result here is that the complexity of temporal TBox satisfiability is exponentially higher than for concept satisfiability relative to global TBoxes; notably, the lower bound is a consequence of Theorem 3 above.

Theorem 5. *Satisfiability of temporal $\text{LTL}_{\mathcal{ALC}}^{\text{bin}}$ -TBoxes is 2EXPSpace-complete.*

Membership in 2EXPSpace is a consequence of the following: (i) satisfiability of temporal $\text{LTL}_{\mathcal{ALC}}$ -TBoxes is EXPSpace-complete [25], and (ii) any $\text{LTL}_{\mathcal{ALC}}^{\text{bin}}$ temporal TBox can be translated into an equivalent though exponentially larger $\text{LTL}_{\mathcal{ALC}}$ -TBox, by expanding the succinctly encoded intervals.

For the lower bound, we reduce the satisfiability problem for $\text{MTL}_{\mathcal{ALC}}$, which is 2EXPSpace-hard cf. Theorem 3. Introduce a fresh concept name Gap , which intuitively models the ‘‘gaps’’ between consecutive observations in $\text{MTL}_{\mathcal{ALC}}$, and define the map \cdot^\dagger inductively by taking:

$$\begin{aligned} A^\dagger &= A \\ (\neg C)^\dagger &= \neg(C^\dagger) \\ (C \sqcap D)^\dagger &= C^\dagger \sqcap D^\dagger \\ (\exists r.C)^\dagger &= \exists r.C^\dagger \\ (\bigcirc_I C)^\dagger &= \text{Gap} \mathcal{U}_I (\neg \text{Gap} \sqcap C^\dagger) \\ (C \mathcal{U}_I D)^\dagger &= (\text{Gap} \sqcup C^\dagger) \mathcal{U}_I (\neg \text{Gap} \sqcap D^\dagger) \\ (C \sqsubseteq D)^\dagger &= (\neg \text{Gap} \sqcap C^\dagger \sqsubseteq D^\dagger) \end{aligned}$$

It is routine to verify that:

Lemma 8. *An $\text{MTL}_{\mathcal{ALC}}$ -concept C and TBox \mathcal{T} are satisfiable iff the following temporal $\text{LTL}_{\mathcal{ALC}}^{\text{bin}}$ -TBox is satisfiable:*

$$\begin{aligned} &\neg(\top \sqsubseteq \text{Gap} \sqcup \neg C^\dagger) \wedge \\ &\square \bigwedge_{\alpha \in \mathcal{T}} \alpha^\dagger \wedge \square(\top \sqsubseteq \text{Gap} \vee \top \sqsubseteq \neg \text{Gap}) \wedge \square \diamond(\top \sqsubseteq \neg \text{Gap}). \end{aligned}$$

5.2 Temporal TBoxes in $\text{MTL}_{\mathcal{ALC}}$

The syntax of *temporal $\text{MTL}_{\mathcal{ALC}}$ -TBoxes* is obtained from the syntax of $\text{LTL}_{\mathcal{ALC}}^{\text{bin}}$ -TBoxes by just replacing $\bigcirc \varphi$ with $\bigcirc_I \varphi$. The semantics is adapted accordingly as discussed in Section 4; for instance

$$\mathfrak{J}, n \models \bigcirc_I \varphi \quad \text{iff} \quad \mathfrak{J}, n+1 \models \varphi \quad \text{and} \quad \tau(n+1) - \tau(n) \in I.$$

Theorem 6. *Satisfiability of temporal $\text{MTL}_{\mathcal{ALC}}$ -TBoxes is 2EXPSpace-complete.*

The lower bound is inherited from Theorem 3. For the upper bound, we lift the mapping \cdot^\dagger given in the proof of Theorem 5 to temporal $\text{MTL}_{\mathcal{ALCC}}\text{-TBoxes}$. For a temporal $\text{MTL}_{\mathcal{ALCC}}\text{-TBox}$ φ , define φ^\dagger inductively as follows:

$$\begin{aligned}(\varphi \wedge \psi)^\dagger &= \varphi^\dagger \wedge \psi^\dagger \\ (\neg\varphi)^\dagger &= \neg(\varphi^\dagger) \\ (\bigcirc_I \varphi)^\dagger &= (\text{T} \sqsubseteq \text{Gap}) \mathcal{U}_I (\text{T} \sqsubseteq \neg \text{Gap} \wedge \varphi^\dagger) \\ (\varphi \mathcal{U}_I \psi)^\dagger &= (\text{T} \sqsubseteq \text{Gap} \vee \varphi^\dagger) \mathcal{U}_I (\text{T} \sqsubseteq \neg \text{Gap} \wedge \psi^\dagger)\end{aligned}$$

The following Lemma, which is proved similar to Lemma 8, together with the fact that satisfiability of temporal $\text{LTL}_{\mathcal{ALCC}}^{\text{bin}}\text{-TBoxes}$ can be checked in 2EXPSpace concludes the upper bound.

Lemma 9. *A temporal $\text{MTL}_{\mathcal{ALCC}}\text{-TBox}$ φ is satisfiable iff $\varphi^\dagger \wedge \square(\text{T} \sqsubseteq \text{Gap} \vee \text{T} \sqsubseteq \neg \text{Gap}) \wedge \diamond(\text{T} \sqsubseteq \neg \text{Gap})$ is satisfiable.*

5.3 Restriction to intervals $[0, c]$, $[c, \infty)$

We have seen in Theorem 2 that the restriction to intervals of the form $[0, c]$, $[c, \infty)$ leads to better complexity in the case of (classical) satisfiability. We show here that this in fact also applies to temporal TBoxes. In fact, the observations made in Section 3.2 apply here as well and it is fairly straightforward to extend it to this more general setting. The upper bound is then obtained by adapting a strategy that has been used for monodic first-order temporal logic, $\mathcal{QTL}_{\mathcal{U}}^{\text{in}}$ in [15, Theorem 11.30].

Due to this proximity, we sketch only the necessary changes. We need to extend the definition of a type to reflect the information about the TBox formulas as follows. For a TBox formula φ , denote with $\text{sub}(\varphi)$ the set of all subformulas of φ together with all subconcepts appearing in some of these subformulas; in particular, $\text{sub}(\varphi)$ can contain both a concept inclusion $C \sqsubseteq D$ and a concept C . Similar to Section 3.2, $\text{cl}(\varphi)$ is the closure under single negation of $\text{sub}(\varphi)$ extended with the set

$$\{\alpha \mathcal{U}_{[0,c]}\beta, \alpha \mathcal{U}_{[c,\infty)}\beta \mid c \in [0, K], \alpha \mathcal{U}_I \beta \in \text{sub}(\varphi)\},$$

where K is the largest constant in φ and α, β could be concepts or TBox formulae. Now, a type is a subset $t \subseteq \text{cl}(\varphi)$ such that:

- $\alpha \in t$ iff $\neg\alpha \notin t$, for all $\neg\alpha \in \text{cl}(\varphi)$;
- $\psi \wedge \chi \in t$ iff $\{\psi, \chi\} \subseteq t$, for all $\psi \wedge \chi \in \text{cl}(\varphi)$;
- $D \sqcap E \in t$ iff $\{D, E\} \subseteq t$, for all $D \sqcap E \in \text{cl}(\varphi)$;
- $C \sqsubseteq D \in t$ and $C \in t$ implies $D \in t$.

As argued in Section 3.2, we only need to consider those (exponentially many) types, which satisfy property **(P)**, appropriately lifted to include TBox formulas. A *quasistate* for φ is a set of types with the additional requirement that the types *agree on the TBox formulas*, that is, $\psi \in t$ iff $\psi \in t'$ for types t, t' in the same quasistate, and all TBox subformulas ψ . After lifting also the run condition **(R3)** to apply to TBox formulas, the notion of a quasimodel remains (almost) identical: a *quasimodel* for φ is a pair (S, \mathfrak{R}) such that:

- (Q1)** $\varphi \in t$ for some $t \in S(0)$; and
- (Q2)** for all $t \in S(n)$, $n \geq 0$ there is a run $r \in \mathfrak{R}$ such that $r(n) = t$.

As before, the existence of a quasimodel for φ characterizes satisfiability of φ . Moreover, if there is a quasimodel, then there is a quasimodel (S, \mathfrak{R}) of the regular form

$$S = S(0) \dots S(n-1)S(n) \dots S(n+m-1)^\omega$$

with $n \leq \#\text{qs}_\varphi$ and $m \leq |\text{sub}(\varphi)| \cdot \#\text{qs}_\varphi \cdot (\#\varphi)^2 + \#\text{qs}_\varphi$, where $\#\varphi$ and $\#\text{qs}_\varphi$ denote the number of types and quasistates for φ , respectively. Thus, both the length n of the initial part and the length m of the cycle are double exponentially bounded.

Based on this, one can devise the following algorithm, similar to [15, Lemma 11.30] and the algorithm for the proof of Theorem 1: guess numbers n, m within the mentioned bounds, and step by step the sequence S keeping always only two consecutive quasistates. While guessing the sequence, verify on the fly that each type in $S(i)$ has a compatible type in $S(i+1)$, and vice versa. At time point n , store $S(n)$ and continue m more steps until reaching $S(m+n) = S(n)$. Moreover, verify that all $\alpha \mathcal{U}_I \beta$ appearing in some type in $S(n)$ are realized on the way to $S(n+m)$. It should be clear that this can be done in (non-deterministic) exponential-space, yielding:

Theorem 7. *Satisfiability of temporal $\text{LTL}_{\mathcal{ALCC}}^{0,\infty}\text{-TBoxes}$ is EXPSpace -complete.*

As a consequence of Lemma 9, we additionally obtain:

Theorem 8. *Satisfiability of temporal $\text{MTL}_{\mathcal{ALCC}}^{0,\infty}\text{-TBoxes}$ is EXPSpace -complete.*

6 Conclusions and Future Work

In this paper, we have launched the study of *metric* TDLs allowing for quantitative temporal reasoning, and established a fairly complete landscape of the complexity of satisfiability for $\text{LTL}_{\mathcal{ALCC}}^{\text{bin}}$ and $\text{MTL}_{\mathcal{ALCC}}$ (over the naturals). Most interestingly, we have shown that the ability to reason explicitly about timestamps of observations brings additional computational complexity. In particular, the complexity of concept satisfiability is then the same as that of temporal TBox satisfiability, c.f. Table 1.

As immediate future work, we will investigate TDLs based on MTL with *continuous-semantics* (over the reals). For some applications, the continuous-semantics seems to be more appropriate in the sense that a real-time system is continuously observed instead of only when an event or action happens. The change from pointwise to continuous semantics is not for free since full MTL becomes undecidable; however, several decidable fragments have been already identified [27]. We plan to build on these results and study TDLs based on decidable fragments of MTL with continuous-semantics.

We will also look at quantitative TDLs in the context of *ontology-based data access (OBDA)* [14] over temporal databases. We believe that the present paper lays important foundations for understanding the *combined complexity* of the *query answering problem* with *mTDLs*. However, for *data complexity*, i.e., when only the data is considered as part of the input, TBoxes with succinctly represented intervals can be used for free. In this case, an interesting problem is to consider data timestamped with intervals, succinctly representing its validity time. In this scenario, it would be fruitful to study restrictions of *mTDLs* based on ‘data-tractable’ DLs such as those in *DL-Lite* [4] or \mathcal{EL} [11], whose temporal extensions to access temporal (timestamped) data have been recently investigated [10, 6, 16, 22]. However, none of these works studies interval encoding of timestamps.

REFERENCES

- [1] R. Alur and T. A. Henzinger, ‘Real-time logics: Complexity and expressiveness’, *Inf. Comput.*, **104**(1), 35–77, (1993).
- [2] R. Alur and T. A. Henzinger, ‘A really temporal logic’, *J. ACM*, **41**(1), 181–204, (1994).
- [3] A. Artale, D. Bresolin, A. Montanari, G. Sciavicco, and V. Ryzhikov, ‘DL-Lite and interval temporal logics: a marriage proposal’, in *Proc. of ECAI-14*, pp. 957–958, (2014).
- [4] A. Artale, D. Calvanese, R. Kontchakov, and M. Zakharyashev, ‘The DL-Lite family and relations’, *J. Artif. Intell. Res. (JAIR)*, **36**, 1–69, (2009).
- [5] A. Artale and E. Franconi, ‘A survey of temporal extensions of description logics’, *Ann. Math. Artif. Intell.*, **30**(1-4), 171–210, (2000).
- [6] A. Artale, R. Kontchakov, A. Kovtunova, V. Ryzhikov, F. Wolter, and M. Zakharyashev, ‘First-order rewritability of temporal ontology-mediated queries’, in *Proc. of IJCAI-15*, pp. 2706–2712, (2015).
- [7] A. Artale, R. Kontchakov, C. Lutz, F. Wolter, and M. Zakharyashev, ‘Temporalising tractable description logics’, in *Proc. of TIME-07*, (2007).
- [8] A. Artale, R. Kontchakov, V. Ryzhikov, and M. Zakharyashev, ‘A cookbook for temporal conceptual data modelling with description logics’, *ACM Trans. Comput. Log.*, **15**(3), 25, (2014).
- [9] A. Artale, R. Kontchakov, V. Ryzhikov, and M. Zakharyashev, ‘Tractable interval temporal propositional and description logics’, in *Proc. of AAI-15*, pp. 1417–1423, (2015).
- [10] A. Artale, R. Kontchakov, F. Wolter, and M. Zakharyashev, ‘Temporal description logic for ontology-based data access’, in *Proc. of IJCAI-13*, (2013).
- [11] F. Baader, S. Brandt, and C. Lutz, ‘Pushing the EL envelope’, in *Proc. of IJCAI-05*, pp. 364–369, (2005).
- [12] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, eds. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
- [13] F. Baader, S. Ghilardi, and C. Lutz, ‘LTL over description logic axioms’, *ACM Trans. Comput. Log.*, **13**(3), 21, (2012).
- [14] M. Bienvenu and M. Ortiz, ‘Ontology-mediated query answering with data-tractable description logics’, in *Proc. of Reasoning Web*, pp. 218–307, (2015).
- [15] D. M. Gabbay, A. Kurucz, F. Wolter, and M. Zakharyashev, *Many-dimensional modal logics: theory and applications*, volume 148, North Holland, 2003.
- [16] V. Gutiérrez-Basulto, J. C. Jung, and R. Kontchakov, ‘Temporalized \mathcal{EL} ontologies for accessing temporal data: Complexity of atomic queries’, in *Proc. of IJCAI-16*, (2016).
- [17] V. Gutiérrez-Basulto, J. C. Jung, and C. Lutz, ‘Complexity of branching temporal description logics’, in *Proc. of ECAI-12*, (2012).
- [18] V. Gutiérrez-Basulto, J. C. Jung, and T. Schneider, ‘Lightweight description logics and branching time: a troublesome marriage’, in *Proc. of KR-14*, (2014).
- [19] V. Gutiérrez-Basulto, J.C. Jung, and T. Schneider, ‘Lightweight temporal description logics with rigid roles and restricted TBoxes’, in *Proc. of IJCAI-15*, pp. 3015–3021, (2015).
- [20] J. Y. Halpern and M. Y. Vardi, ‘The complexity of reasoning about knowledge and time. i. lower bounds’, *J. Comput. Syst. Sci.*, **38**(1), 195–237, (1989).
- [21] I. M. Hodkinson, R. Kontchakov, A. Kurucz, F. Wolter, and M. Zakharyashev, ‘On the computational complexity of decidable fragments of first-order linear temporal logics’, in *Proc. of TIME-ICTL 2003*, pp. 91–98, (2003).
- [22] R. Kontchakov, L. Pandolfo, L. Pulina, V. Ryzhikov, and M. Zakharyashev, ‘Temporal and spatial OBDA with many-dimensional Halpern-Shoham logic’, in *Proc. of IJCAI-16*, (2016).
- [23] R. Koymans, ‘Specifying real-time properties with metric temporal logic’, *Real-Time Systems*, **2**(4), 255–299, (1990).
- [24] C. Lutz, D. Walther, and F. Wolter, ‘Quantitative temporal logics over the reals: PSpace and below’, *Inf. Comput.*, **205**(1), 99–123, (2007).
- [25] C. Lutz, F. Wolter, and M. Zakharyashev, ‘Temporal description logics: A survey.’, in *Proc. of TIME-07*, pp. 3–14, (2008).
- [26] Z. Manna and P. Wolper, ‘Synthesis of communicating processes from temporal logic specifications’, *ACM Trans. Program. Lang. Syst.*, **6**(1), 68–93, (1984).
- [27] J. Ouaknine and J. Worrell, ‘Some recent results in metric temporal logic’, in *Proc. of FORMATS-08*, pp. 1–13, (2008).
- [28] V. Pratt, ‘Models of program logics’, in *Proc. of 20th Annual Symposium on Foundations of Computer Science*, pp. 115–122, (1979).
- [29] K. Schild, ‘Combining terminological logics with tense logic’, in *Proc. of EPIA*, pp. 105–120. Springer, (1993).
- [30] A. P. Sistla and E. M. Clarke, ‘The complexity of propositional linear temporal logics’, *J. ACM*, **32**(3), 733–749, (1985).
- [31] F. Wolter and M. Zakharyashev, ‘Temporalizing description logics’, in *Proc. of FroCoS-98*, pp. 379–402. Studies Press/Wiley, (1999).

Plan-Based Narrative Generation with Coordinated Subplots

Julie Porteous¹, Fred Charles² and Marc Cavazza³

Abstract. Despite recent progress in plan-based narrative generation, one major limitation is that systems tend to produce a single plotline whose progression entirely determines the narrative experience. However, for certain narrative genres such as serial dramas and soaps, multiple interleaved subplots are expected by the audience, as this tends to be the norm in real-world, human-authored narratives. Current narrative generation techniques have overlooked this important requirement, something which could improve the perceived quality of generated stories. To this end, we have developed a flexible plan-based approach to multiplot narrative generation, that successfully generates narratives conforming to different subplot profiles, in terms of the number of subplots interleaved and the relative time spent on each presentation. We have identified specific challenges such as: distribution of virtual characters across subplots; length of each subplot presentation; and transitioning between subplots.

In this paper, we overview this approach and describe its operation in a prototype Interactive Storytelling (IS) System set in the serial drama genre. Results of experiments with the system demonstrate its usability. Furthermore, results of a user study highlight the potential of the approach, with clear user preference for presentations that feature interleaved multiple subplots.

1 Introduction

Interactive Storytelling (IS) has emerged as a popular application of AI techniques in particular Planning to new media entertainment, where planning is in charge of preserving story consistency despite variable initial options or in-story interactions (for example work such as [37, 1, 26, 5, 24, 11]). Hence IS represents an interesting new application of planning: one that challenges received wisdom about optimality and knowledge representation, as the shape of plan trajectories determines narrative structure. [21]). This emphasis on story “backbone” consistency has sometimes resulted in departure from the properties of real-world narratives, which far from reflecting a canonical structure, often interleave various sub-plots featuring the same characters, for instance in popular TV dramas, as analyzed by [35]. Nevertheless an enduring challenge remains, which is to improve user acceptance of these automatically generated narratives.

Interestingly, studies of popular dramas and TV series (an important target genre for narrative generation and IS) suggest that individual episodes, even if self-contained, feature multiple sub-plots. For example, the series MASH [29] had double plot-lines per episode whilst more recent series such as SEINFELD [31] and ER [27] feature even more [33]. Further, the way in which different subplots are

interleaved has been shown to be a factor in user acceptance, with individual scenes tending to be short, providing only a slight bit of progression in a given plotline and moving quickly among plots [35].

Previous research has attempted to make narratives most realistic despite the constraints imposed by single-plot single-goal generation. The main methods have sought to improve the presentation of the story at discourse level or to generate narratives with multiple goals. However, none has really explored the generation of coordinated subplots in an effort to reproduce the authoring of real-world drama.

Hence, we were motivated to develop an approach to narrative generation which provided just such a coordination mechanism. Our solution achieves narrative generation and multiplot coordination in a single pass via forward heuristic search through partially ordered subgoals. It represents an extension to our successful approach to single subplot narrative generation which uses authored subgoals to incrementally build up the narrative plan by composition of smaller plan segments, themselves generated using a base planner [22]. We aim to demonstrate that the same landmark model that has been successful to coordinate the pace and trajectories of narrative plans can be extended to represent multiple sub-plots. The coordination of subplots would thus be framed as another high-level landmark ordering problem. This will be managed through heuristic search over the landmark graph, using heuristic knowledge on plot duration, ratios and switching which can be acquired from media studies literature. In this paper we concentrate on narrative generation within our IS system, however we emphasise that a key strength of this incremental forward narrative generation approach is the flexibility it provides to respond to user interaction with the system.

Thus this work addresses an important creativity challenge facing automated narrative generation techniques: to generate output that shares the properties of human-created content but without a heavy authoring burden. Our contribution is the extension of plan-based single subplot narrative generation from action descriptions to the generation of narratives featuring multiple interleaved subplots from a combination of action descriptions and high-level properties of subplots. Clearly these high-level properties – generic filmic knowledge – requires some authoring but we argue that our approach constitutes little additional authoring overhead and greatly increases realism and acceptance (as illustrated by our user study).

Our approach to multiplot narrative generation is fully implemented and integrated within an IS system set in a medical hospital drama (where multiple sub-plots tend to be the rule [35]). This system is used throughout the paper for illustration and was also used in a series of experiments which we report in the paper. Fig.1 provides an overview of the system architecture.

The paper is organised as follows: the next section covers the discussion of related work. The narrative framework and the require-

¹ Teesside University, email: j.porteous@tees.ac.uk

² Teesside University, email: f.charles@tees.ac.uk

³ University of Kent, email: m.o.cavazza@kent.ac.uk

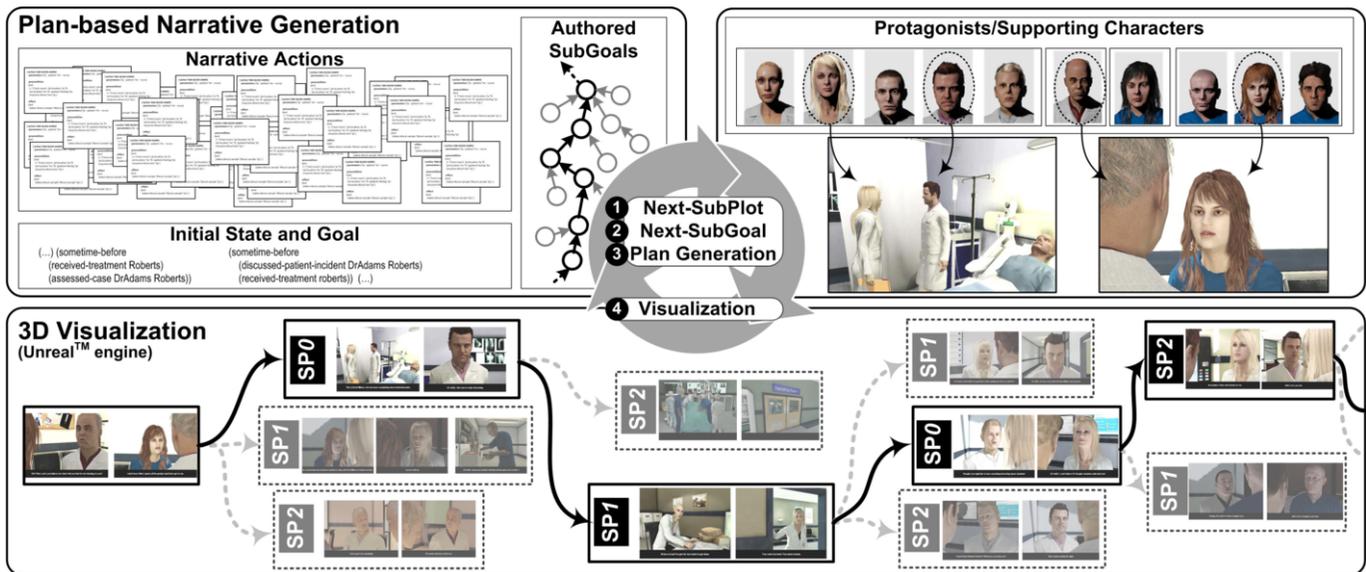


Figure 1. Overview of Multiple Subplot Generation. The system operates in a “plan-execution” loop with narratives generated and visualised incrementally. Each iteration proceeds through the following steps: (1) the selection of the next subplot for the narrative to transition to; (2) for that subplot the selection of the next subgoal to use to generate the next phase of the narrative towards; (3) plan-based generation of the next phase of the narrative using the selected subgoal as goal; (4) as narrative segments are generated they are visualized using the UDK[®] game engine and shown to the viewer.

ments for multiple subplot narratives are detailed in section 3). Authoring of the narrative domain model is discussed in section 4. The narrative generation procedure is detailed in section 5 and is illustrated with an example (section 6). Results of experiments and user study are reported in sections 7 and 8. In section 9 we conclude and assess the potential of the approach.

2 Related Work

There are many examples of the use of planning for generation of narrative but none to date looking at interleaving of multiple plotlines. A number of these plan-based approaches to narrative generation use authored goals to help shape output narratives and in our work we have drawn inspiration from them. In particular, [23] introduced a notion of author goals and used them to extend a partial order planner in an approach referred to as “complexifying” of the planning process. Subsequently, [21] used authored constraints to help structure narrative as part of a dynamic run-time mechanism that selected constraints for generation of narrative variants. Their approach exploited forward state space planning (essential for supporting interaction), used a standard representation language (PDDL3.0), and provided an author-friendly interface to specify narrative plan dynamics.

The interleaving of multiple subplots can be seen as a discourse level mechanism, for example, used to maintain user interest. Some related research has tackled discourse level aspects by firstly generating narrative and then tackling discourse in a post-processing phase. Examples of this include, [6, 2], for the introduction of suspense, and [14], for cinematic representation. However, our approach generates and coordinates multiplot narratives in a “single pass”: something which is arguably more principled and more robust than presenting separate aspects of plot at discourse level.

There are some narrative approaches that have adopted a drama manager approach [15, 36, 18, 32]. Whilst this approach could be used for generation of multiplot narratives we argue that this lacks the generative power and flexibility leveraged by planning along with the

author friendly mechanism for declarative specification of narrative structuring control knowledge.

Some work has looked at generation of narratives consisting of multiple quests. For example, [16] use planning for off-line adaptation of authored narratives which include multiple “quests” in the context of game plot lines with focus on plotline adaptation to create new plausible quest sequences. The Crystal Island interactive narrative of [17] featured multiple quest subplots in an approach to user goal recognition aiming to combine multiple plot elements to create rich customized stories in the domain of microbiology with the user playing the role of detective solving a science mystery. [13] presented an approach to the generation of side quests for role-playing games to enhance players sense of agency. We observe that whilst these systems sought to generate multiple quests there was no requirement to interleave them or to conform to the conventions of a particular genre.

Also related are cinematic conventions governing editing and continuity “... to tell a story coherently and clearly, to map out the chain of characters’ actions in an undistracting way ...” [4, 34]. One of the advantages of plan-based narrative generation is its power to support narrative causality both at the local and global levels and to ensure coherent character behaviour [38].

3 Coordinated Multiplot Narrative Requirements

Narratives in the domain of television serial drama tend to be built up from a number of interleaved single subplots, where the resulting output multiplot narrative is composed of chains of *segments* each drawn from the individual subplots [33]. Here, as a target for generating such narratives, we identify a core set of requirements for individual subplot and multiplot coordinated narratives. This constitutes generic reusable filmic knowledge, sourced from leading film studies [3, 33, 35] and genre analysis. The rationale for selection of these requirements is given below and are summarised in Fig. 3. These requirements are also summarised in Fig. 2.

SINGLE	Characters	Protagonist with additional characters: allies (support) and antagonists (obstruct).
	Structure	Narrative segments: <i>introduction</i> <i>obstruction</i> → <i>resolution</i> (one or multiple) <i>exposition</i> (one or multiple)
MULTIPLE	Characters	Different Protagonist for each subplot Characters can appear across subplots
	Structure	Composed of two or more single subplots Target total for number of segments Target ratio of time on each subplot Subplot segments distributed evenly
	Interleave	Well-formed multiple subplot narrative is: <i>a sequence of narrative segments with adjacent segments from different subplots</i>

Figure 2. Summary of requirements for narratives with **single** subplots and **multiple** interleaved subplots. See section 3 for further detail of rationale.

3.1 Individual Subplots

Analyses from Smith [33] and Bordwell [3] provide insights into the roles of characters and structure of plot in individual subplots for our target genre: at the centre of each individual subplot is one character – the protagonist – a goal driven individual who encounters obstacles in the pursuit of their goal; individual subplot structure requires some initial introduction to the protagonist and their goal, followed by one (or multiple) obstructions to them in the pursuit of their goal, leading to some final resolution. Accordingly we require the following outline structure for individual subplots:

- *introduction*: to introduce the protagonist and their goal
- one or multiple occurrences of:
 - *obstruction*: the protagonist encounters some form of obstacle which prevents them from directly achieving their goal
 - *resolution*: either the obstacle is overcome or goal is achieved
- *exposition*: a feature of narratives in this genre is that individual subplots can, at any stage, include phases of “dispersed exposition” [35], where additional information is given to the audience, e.g. plot re-caps or further insight into characters. Hence we allow for subplots to include one or multiple episodes of exposition.

Importantly, this allows for longer length subplot narratives to be built up via composition of multiple phases of obstruction and resolution, interleaved with one or more exposition phases.

Narrative Segments

For clarity, we introduce the notion of narrative “segments” to refer to the different phases of individual subplots that are coordinated to form multiplot narratives – in other words ground instances of *introduction*, *obstruction*, *resolution* and *exposition* as described above.

3.2 Multiple Interleaved Subplots

Multiple subplot narratives are composed from the interleaving of segments from a number of individual subplots. Hence each of these subplots display the properties relating to character and structure described above. However the subplots are not independent because virtual characters can appear in different roles across subplots and the consequences of events in one subplot can impact on the unfolding of the narrative in other subplot(s). As an example consider an episode of ER [27] (episode 15:17) which features one subplot with

a protagonist, Dr Carter, struggling in the emergency department after a break, whilst in another subplot with a different protagonist, Dr Banfield who is struggling to conceive. Dr Carter appears as support. Hence we require that each subplot has a unique protagonist but otherwise characters can appear freely across subplots.

Genre analysis of the medical dramas SCRUBS, HOUSE and ER [30, 28, 27] revealed varying numbers of subplots interleaved in episodes, ranging from two upwards. Further, there was no clear pattern of the relative proportion of each subplot: sometimes the subplots were evenly split whereas in others some subplots were more dominant. Hence, to best support these features of human-authored dramas we developed an approach that was flexible with respect to these structural properties. Thus we allow for the relative proportion of different subplots to be specified as part of the input parameters which are used as targets for generation of a given narrative.

Finally, from the genre analysis was observed an even distribution of subplot segments across entire episodes rather than concentrated in different phases which is consistent with observations of rapid quick movement between subplots [35].

Thus rules are required for managing plot switching that can be readily integrated with the overall narrative generative framework in a flexible manner to provide support for all the major features encountered in human-authored dramas. Hence our framework is flexible with respect to structural properties such as the number of subplots to be interleaved, the rate of switching between them, the relative proportion of the narrative per subplot and the overall duration of the narrative. Consequently, we allow for these requirements (summarised Fig. 2) to be specified as part of narrative generator input.

The final requirement of interleaved multiplot narratives is that they are “well formed” which we take to be sequences of narrative segments where *adjacent* segments are from *different* subplots⁴. As an example, a well-formed three subplot narrative (for subplots s0, s1 and s2), could consist of a 12-segment sequence, with 3, 3 and 6 segments from subplots s0, s1 and s2 respectively:

s1 | s2 | s0 | s2 | s1 | s2 | s0 | s2 | s1 | s2 | s0 | s2

Note that, as shown in the system architecture Fig. 1, since narrative generation in the system works in an incremental “plan-execute” loop, a single narrative isn’t generated in the traditional sense, rather as narrative plan segments are generated they are visualised to the audience. This incremental approach allows for greater flexibility in the presence of user interaction as detailed in [22] and has been designed to be compatible with various implementations, whether story variability derived from initial parameters (as in the NETWORKING system [20] used in this work) or is due to user interaction [9].

4 Authoring for Coordinated Subplot Generation

The multiplot generation approach we present in this paper extends our plan-based approach to single subplot narrative generation to multiplot narratives. Hence it requires an input domain model consisting of pre- and post-condition actions augmented with authored subgoals which are used to control narrative structure.

For our medical drama domain, the narrative planning actions are those that characterize the genre, such as conflicts over patient diagnosis, treatment, professional rivalries, battles to save patients, romance, domestic conflicts and support for friends and so on. The process of modelling these narrative actions is as detailed in [22, 20].

The approach also requires the domain model to include authored subgoals to provide control over the structure of the narrative as it is

⁴ We note this places certain restrictions on the relative target number of segments for different subplots. We assume this is met.

generated. These subgoals can be authored using an intuitive visual user interface (as discussed in [21]) which helps ameliorate some of the authoring overhead. Below we consider further the authoring of the subgoals and their role in coordinated multiplot generation.

4.1 Narrative Segment Subgoals

Our approach is to require that the narrative planning domain model is augmented to include authored narrative subgoals in the style of [23, 22]. These are partial descriptions of interesting states of the narrative world that can be used to generate segments of the narrative. The minimal representational assumption on these subgoals are that they are sets of domain facts.

As an example, consider the narrative segment which introduces a subplot where the protagonist is struggling with pressure of work as shown in Fig. 3. The subgoal here is *shown-pressure-of-work* and the plan generated for this goal ensures that the pressure of work being experienced by Dr Adams (protagonist) has been introduced in the narrative. This example illustrates a form of disjunctive specification of the subgoal (as in [8]). The subgoal is loosely specified, thus making no requirement on how this goal is achieved and thus allowing a range of different plans to be able to achieve the goal depending on the state of the narrative world. The figure illustrates 2 different alternative narrative segments: one generated when the supporting character Dr Miller is happy and a different one when she is angry.

This form of disjunctive specification of subgoals is important since subplots aren't independent and consequences of actions in one subplot can impact on later segments of other subplots. The small example shown in Fig. 3 also illustrates this: suppose if Dr Miller is also the protagonist of a different subplot, then the progress of that subplot can impact depending on whether she has been *harassed-by-colleague* leaving her angry, or happy as a result of *arranged-date*.

4.2 Partial Order over Subgoals

In order to generate such segments, we adopt the approach of [21] and use authored subgoals to represent appropriate narrative situations that can be used to control structure during narrative generation. As illustration, Fig. 4 shows some subgoals and their orders for a single patient treatment subplot, modelled using the PDDL3 modal operators, *sometime-before*, *sometime* and *at-end* (comments show them numbered, sg1-sg4 and goal). For the purposes of generating a single subplot the subgoal (*taken-case DrGreen Roberts*) can be used to generate a narrative segment introducing the doctors' intentions for the patient, whilst the (*exposition*) subgoal (*discussed-ethics DrGreen DrCook*) generates a segment that gives the audience information about the doctors' views. We note that even this small partially ordered collection of subgoals allows for the generation of a range of differently structured subplots of varying numbers of interleaved segments. For example, the sequence of subgoals *sg1|sg2|sg3|sg4* could be used to incrementally generate a subplot consisting of 5 segments (narratives generated using predicates such as (*taken-case DrGreen Roberts*) as goals), whereas the sequence *sg5|ex1|sg6* could be used to generate a very different subplot with 3 segments, including a segment of exposition, for subgoal (*discussed-ethics DrGreen DrCook*), where insights are given into DrGreen's ethical views.

In addition we also use partial orders over the authored subgoals as a mechanism for structuring narrative content, for example to ensure that the protagonists goals are introduced early in the subplot and that obstacles encountered by the protagonist happen prior to them being overcome. This structuring information can be captured via an

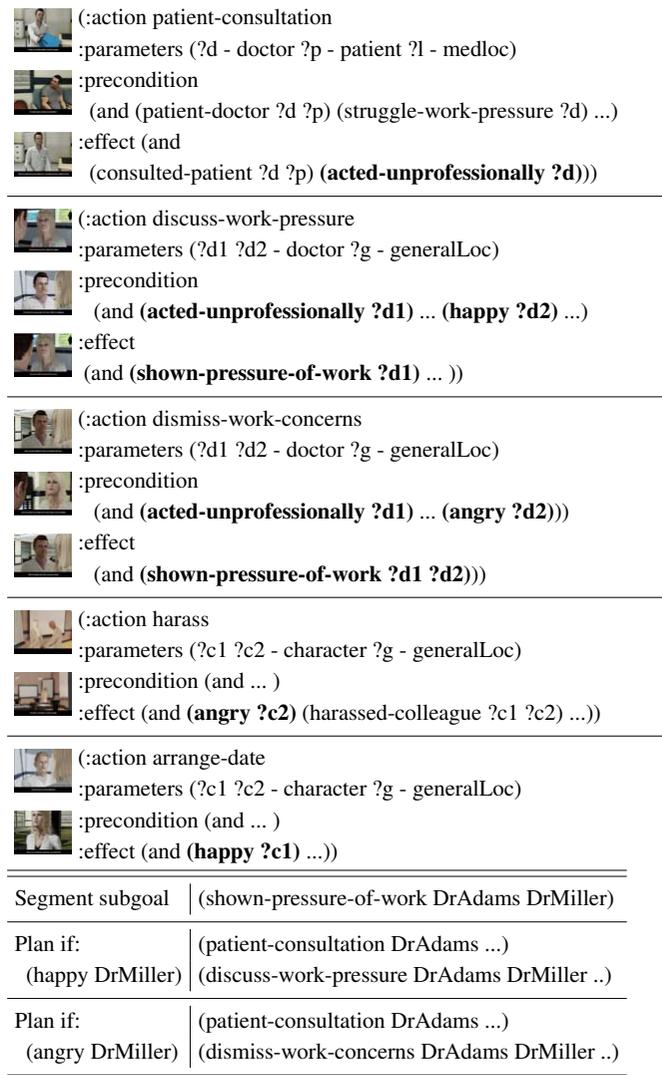


Figure 3. Narrative modelling example: segment subgoal loosely specified (disjunctive specification [8]) allowing for different plans to be generated depending on state of the narrative world: if Dr Miller is happy then her response to Dr Adam's failure to cope with pressures of work (sympathetic discussion) is very different to if she is angry (dismisses his concerns).

intuitive visual interface, as in [21], and then it can be automatically instantiated to ground domain predicates and PDDL3 *sometime* and *sometime-before* modal operators at run time. This provides a much more user friendly mechanism for authoring this information than within the individual domain actions themselves.

5 Narrative Generation

Our approach to multiplot narrative generation is based on an incremental heuristic search through the space of partially ordered authored subgoals for each of the subplots. Within each loop of the search the following are selected: the next subplot to generate a narrative segment for (ie which subplot to "transition" to), and which subgoal on the frontier of the orders for that subplot to use to generate that narrative segment. Then the narrative is generated forwards

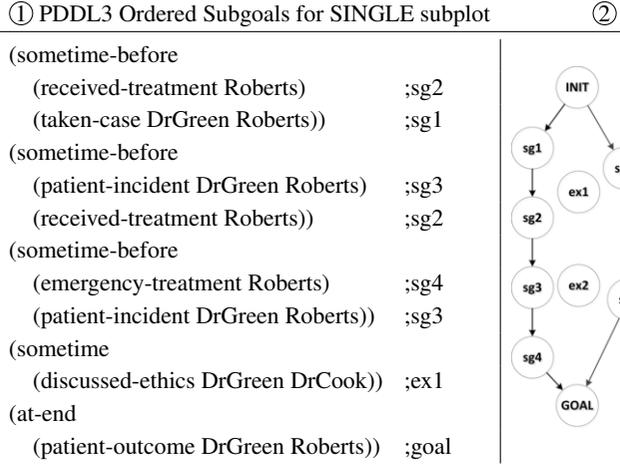


Figure 4. ① PDDL3.0 specification of subgoal orders for a single subplot using the sometime-before and sometime modal operators ② Graphical representation of the partial order showing: all subgoals are ordered before the goal; $\{sg1, \dots, sg4\}$ and $\{sg5, \dots, sg6\}$ are totally ordered (sometime-before modal operator); ex1, ex2 can occur any time (sometime modal operator).

by generating that segment.

One challenge is to achieve even distribution of subplots over the length of the narrative whilst ensuring that adjacent segments are from different subplots (well-formed). Hence the next subplot selected to transition to is the subplot with the most remaining target segments and different to the current.

Another challenge is to select the next subgoal for segment generation that offers the best possibility of matching the input target lengths for the whole narrative. To assess this, the length of the narrative generated so far is considered along with the remaining targets for all subplots and the possibility afforded by each of the frontier subgoals for the selected subplot.

5.1 Multiplot Generation Algorithm

An outline of our multiplot narrative generation algorithm is shown in Fig. 5. The input is (A, I, T, L, G, S) as follows: A , a set of pre- and post-condition planning actions; I , the initial state of the narrative world; T , a set of target segment counts, where $t \in T$ gives the target for each subplot; L , the target length of the output narrative in terms of the total number of segments it contains; G , a set of goal conditions; S , a partial order over subgoals for each subplot.

A narrative plan is a sequence of actions that maps the initial state into a state where all G are true. The aim is that narratives match the target length, L , contain the target number of segments for each subplot, T , and are well-formed (true if adjacent segments are from different subplots).

As shown in Fig. 5, GENERATE-MULTILOT builds up narratives incrementally. In each loop the first step is selecting the next subplot to switch to, procedure NEXT-SUBPLOT [line 6 and below], then to select the next subgoal to use for generating the next narrative segment, procedure NEXT-SUBGOAL [line 7 and below]. Once a subgoal has been selected this is used to generate the next segment of the narrative using the current state of the narrative world as the initial state and the selected sg as goal, GENERATE-PLAN [line 8]. This small portion of narrative is generated using the classical planner METRIC-FF [12] which is embedded in the system. At this point the generated segment is visualised to the user [line 9], via staging in a 3D world

```

1: procedure GENERATE-MULTILOT( $A, I, T, L, G, S$ )
2:    $C \leftarrow I$ 
3:    $Narr, last-sp, seg, sg, sp \leftarrow \{\}$ 
4:    $\forall s \in S : count_s \leftarrow 0, N \leftarrow 0$ 
5:   while  $G \neq \{\}$  do ▷ Loop
6:      $sp \leftarrow \text{NEXT-SUBPLOT}(S, last-sp)$ 
7:      $sg \leftarrow \text{NEXT-SUBGOAL}(S, sp, count_{sp}, T)$ 
8:      $seg \leftarrow \text{GENERATE-PLAN}(C, A, sg)$ 
9:     VISUALIZE-SEGMENT( $seg$ )
10:     $C \leftarrow \text{ADVANCE-STATE}(C, seg)$  ▷ Apply plan actions
11:     $G \leftarrow \text{UPDATE-GOALS}(C)$ 
12:     $Narr \leftarrow Narr \bullet seg$ 
13:     $count_{sp}++, T_{sp}--, last-sp \leftarrow sp, N++$ 
14:  end while ▷ Until all goals solved
15: end procedure
16: return  $Narr$ 
  
```

Figure 5. Outline Algorithm. Multiple subplot narratives are generated incrementally (while loop line 5). In each loop the next subplot to switch to is selected, and from that the next subgoal to use to structure the next segment of the narrative. As each segment is generated it is visualized to the audience (staged in a 3D world). For more details see text.

(as shown in Fig. 1). Then the current state of the narrative world is advanced by application of the narrative actions in seg , ADVANCE-STATE [line 10] and the overall narrative, $Narr$, is extended forwards by concatenation of seg . The main loop continues until the termination condition: a narrative state with all goal conditions are achieved.

Note that narrative generation operates in a “plan-execute” loop (as shown in Fig 1), with narrative segments generated and then visualized incrementally. Hence backtracking isn’t possible but this is managed via loose sub-goal specification (as detailed in section 4). It allows for narrative continuation regardless of narrative state: something of particular importance in interactive systems [22].

Procedure NEXT-SUBPLOT

The subplot sp for which the following are both true:

1. the most segments still remaining to be interleaved into the narrative i.e. the greatest difference between target count, T_{sp} , and the count of segments, $count_{sp}$ already in $Narr$
2. sp is different to the previous loop: $last-sp \neq sp$.

Procedure NEXT-SUBGOAL

Subgoal, sg , is the one that minimizes $D=L-(N+N_h)$ where, L is the target length for the narrative, N is the number of segments in the narrative generated so far and N_h denotes the number of segments still to be generated for all subplots.

The value N_h is calculated for each unvisited frontier subgoal in the partial order S as the total of the following distances: for subplot sp , the distance of the path from sg to $goal$ in S_{sp} ; and for each other subplot, the remaining target distance in T (decremented as segments are added [line 13]).

For subgoal sg , if the distance D doesn’t provide the potential to match the target length L , there are two considerations:

1. if $D < (L - (N + N_h))$: any *exposition* subgoals in the partial order S_{sp} are also considered now with the subgoal returned by NEXT-SUBGOAL drawn at random from the set of all exposition subgoals and sg . This enables matching the overall target length by adding additional exposition relevant to the subplot theme.
2. if $D > (L - (N + N_h))$: it is also possible to reduce the subplot length by shortening the path to match the desired length target. However, in practice, for experiments with our prototype this wasn’t required (see section 7.1).

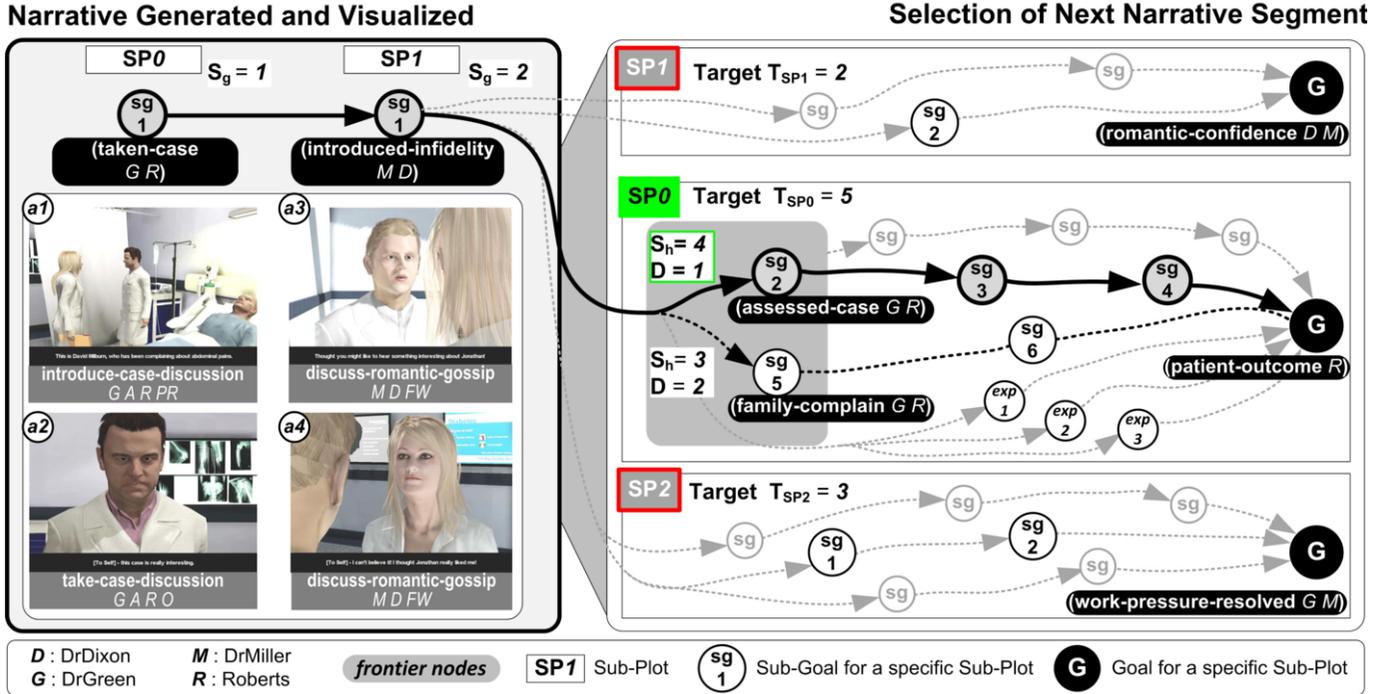


Figure 6. Example of Multiplot Generation. The figure shows the situation after the first two narrative segments have been generated and visualized (the actions labelled A1, A2 for SP0 and A3, A4 for SP1 in the box on the left hand side). At this point the narrative generator must select: 1) which subplot to transition to; and 2) which subgoal to use to generate the next segment. In the figure subplot SP0 (highlighted green) is selected as the subplot to transition to. For selection of the next segment, the frontier narrative subgoals SG2 and SG5 are considered, SG2 provides the closest match to the target narrative length but it doesn't match exactly so the exposition subgoals are also considered with one of these selected (ties broken randomly). Further detail: see section 6.

6 Narrative Generation Example

As illustration consider the example shown in Fig. 6, with three subplots (labelled SP0-SP2), with goals *patient-outcome*, *work-pressure-resolved* and *romantic-conflict-resolved*, target length, $L=12$ and initial target subplot counts, $T=\{6, 3, 3\}$. The figure shows the situation after two iterations of the algorithm with the first two segments of the narrative generated using the following subgoals:

(taken-case DrGreen Roberts) ;; SP0-SG1
(introduced-infidelity DrMiller DrDixon) ;; SP1-SG1

and illustrated with thumbnails of the actions visualization (labelled a1-a4). At this point target counts for the 3 subplots have been adjusted to T to $\{5, 2, 3\}$. Then the subplot for the next narrative segment is selected: in this case SP0 since it has the largest target length (currently $T = \{5, 2, 3\}$) and is different to the previous segment (SP1). The next subgoal is then selected from the frontier of the partial order for this subplot in S . The subgoals considered are:

(assessed-case DrGreen Roberts) ;; SP0-SG2
(family-complain DrGreen Roberts) ;; SP0-SG5

The subgoal that offers the best potential to generate a narrative that meets the target length, L , is then selected based on: the length of the narrative generated so far, N and the combined length of all the segments still to be generated on each of the subplots, N_h . This means considering the length of the path to the goal for each of the frontier narrative nodes and combining that with the target number required on each of the other subplots and then minimizing the distance $D=L-(N+N_h)$. Hence for subplot SP0, for the situation in Fig. 6, where the

current targets for SP1 and SP2 are 2 and 3 respectively, $L=12$ and $N = 2$ then the values for N_h and D are:

Subgoal	$N_h=Path(sp0) + T_{sp1}+T_{sp2}$	$D=L-(N+N_h)$
SP0-SG2	4 + (2 + 3)	12 - (2 + 9) = 1
SP0-SG5	3 + (2 + 3)	12 - (2 + 8) = 2

and since we are minimizing D , subgoal SP0:SG2 is the best. However, at this point, since no values of D equal 0 (i.e. none exactly match the target length), exposition subgoals are also considered. Hence also considered are:

(discuss-ethics DrGreen Roberts) ;; SP0-EX1
(background-patient Roberts) ;; SP0-EX2
(background-doctor DrGreen) ;; SP0-EX3

with the choice being made at random: in this case, selection of subgoal sp0-ex3 as shown in Fig. 6 and that is used as the next goal for incremental narrative generation. This process continues till a state of the narrative world is reached where all the subplot goals are true.

7 Experimental Evaluation

Our prototype interactive narrative was used in the experiments. The narrative domain model for our virtual hospital environment has 15 different locations and 8 doctors, 5 nurses, 4 patients, 4 relatives. For the experiments a test set of narrative planning instances were generated, scaled from 2-4 subplots, with target narrative lengths, L , of 8, 12 and 20 (± 2) segments respectively and random assignment of integer target segment counts, T , across subplots. Subplot goals



Figure 7. Generated Narrative Thumbnails (narrative targets of 3 subplots, target length 12, target segment counts {6,3,3}) The alteration of colours gives a simple overview of the balance and pace between subplot in each different narrative variant. Hence ① has 6 segments on medical ethics subplot and 3 on romance and work pressure; ② 6 on patient treatment; ③ 6 on work pressure. All demonstrate: well formed subplot interleaving; even subplot distribution across duration of narrative. Note how the different subplots of a narrative can interact: for example, for narrative ① the “romance” subplot (orange) can impact on the “clinical” subplot (blue) as follows “acts unprofessionally due to romantic deceptions”.

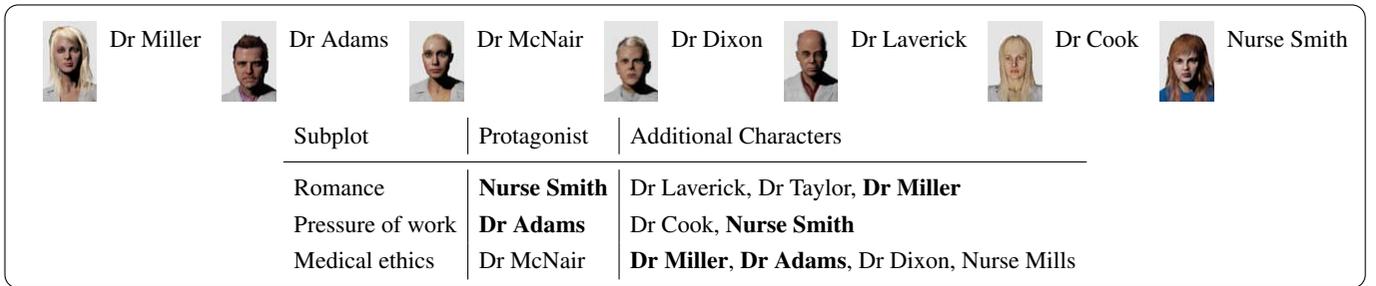


Figure 8. Illustration of Character “Floating” over Subplots. Here we consider the example narrative ① that was shown as thumbnails in Fig. 7. The characters appearing in the narrative are shown across the top and from the thumbnails it can be seen that the characters Nurse Smith and Dr Adams appear as protagonist in one subplot and as a supporting character in another (for Nurse Smith this is the romance and pressure of work subplots, and for Dr Adams the pressure of work and medical ethics subplots). Further, the character Dr Miller appears as support in two of the subplots.

were randomly selected from the set of facts which were tagged as: work pressure, romance, medical ethics or patient treatment.

7.1 Matching Target Narrative Properties

It is important that generated narratives match the target narrative properties and to assess this we generated narratives for the set of test narrative problem instances. As an illustration of these narratives, Fig. 7 shows some visualization summaries as thumbnails. In addition, Fig. 8 gives an illustration of the interdependence of the different subplots with the appearance of characters in different subplots and in different roles.

With regard to matching target multiplot length, our approach achieved 100% fit to integer target counts. This performance can be explained in part due to properties of the chains of ordered subgoals which are kept small in order to provide flexibility: short subplots are possible as are longer segments via composition of shorter chains (e.g. just the small fragment shown in Fig. 4 can yield subplots from length 3 segments (path through subgoal SGX) to 10 (combination of paths through subgoals SG1, EX1 and SG5)).

All generated narratives featured well-formed interleaved subplots, (i.e. adjacent segments are from different subplots). This property is guaranteed by the requirement of procedure NEXT-SUBPLOT that $last-sp \neq sp$, providing that the relative subplot target lengths are within the following bound: the difference between the number of segments in the longest subplot and the combined lengths of all others is ≤ 1 . Our problem set was generated for target counts within this bound.

Results also showed that the approach achieves the desired even distribution of subplots over the whole of the narrative. As illustration, below are tabulated the mean separation between segments from the *same* subplot (i.e. count of the intervening segments) for 2, 3 and 4 subplot narratives with target counts as shown.

#Subplots (Targets)	2 ($T=\{4,4\}$)	3 ($T=\{3,3,6\}$)	4 ($T=\{5,5,5,5\}$)
Mean Separation	1:1	3:3:1	3:3:3:3

Finally we note that narrative generation with well-formed subplots retains the generative power of single-plot generation (a strength of planning [26]).

7.2 Narrative Generation Performance

The average times for generation of narrative *segments* for our test problem set were 0.75, 1.4 and 3.9 seconds respectively for the 2, 3 and 4 subplot instances.

The increase in generation time is a consequence of the increase in the overall length of the narrative as additional subplots are required to be interleaved. These timings are acceptable for use in our Interactive Narrative since (see Fig. 1), narrative generation and “execution” (i.e. presentation on a 3D stage) operate in parallel, with narrative generation taking place during the presentation of the previous segment. Given that the average duration of narrative segment presentation is 25 seconds, this allows ample time for generation to be accommodated.

8 User Evaluation

We staged user experiments to evaluate how users would perceive multi-plot narratives compared to the standard single-goal, single-plot narrative in the same domain. Thirty adults participated in the evaluation: compensated for their time with an online retailer voucher worth €28. A consistent protocol was used across assessment of User Preference 8.1 and understanding 8.2: all narrative presentations and questions were delivered via an online questionnaire with order of presentation controlled across subjects.



Figure 9. User preference for Interleaved vs Non-Interleaved narrative presentations: it can be seen that the user group expressed a very clear preference for the interleaved multiplot version when asked “which narrative presentation did you prefer?”.

8.1 User Preference

Our aim was to explore whether user perception of system generated interleaved narratives reflected what they’re used to seeing in human-authored dramas and took user preference as a proxy for this. To this end we adapted the method of [19] (text to 3D visualization), and asked participants to compare different narratives with the same content presented in different order (and with the same semantics) as follows: (i) subplots interleaved using the approach introduced in the paper; (ii) non-interleaved subplots ordered $\{\{s_{1_1}, \dots, s_{1_n}\}, \{s_{2_1}, \dots, s_{2_n}\}, \dots, \{s_{n_1}, \dots, s_{n_n}\}\}$.

Participants were randomly allocated to groups to control between subjects for: interleaved vs non-interleaved; and order of watching videos. Users viewed a total of 4 presentations (2 variations of 2 narratives each with 3 subplots) to avoid cognitive overload by introducing too many characters.

The online questionnaire asked participants to: “Please explain your reasons for preferring the presentation” and users entered free text responses which were judged to relate to subplots if explicitly mentioned (or similar e.g. storylines, storythreads). The results are shown in Fig. 9, and show that the majority, 72%, chose the interleaved presentation. In addition, participants were also asked whether they enjoyed the presentations and given the opportunity to provide free text responses: 80% gave positive responses and of those, 65% volunteered the subplots as the reason. The following give a flavour of the participant responses:

“There were multiple story lines all taking place in parallel.”
“Stories are told at once rather than one by one, ... more dramatic“
“The layout .. was more dramatic. As and issue was raised then it moved on and didn’t resolve it straight away ...“
“they have a few stories intertwined so the viewer doesnt get bored there is always something new to focus on“
“The storylines where mixed with each other rather than continuing until they were finished so the events felt more realistic.“

8.2 Narrative Understanding

We were keen to show that the interleaving of subplots didn’t harm story understanding. To demonstrate this we used the QUEST approach where narratives are represented as conceptual graphs that are used to rate the relative quality of comprehension questions [7]. The QUEST evaluation consists of a task for users to complete in order to demonstrate story understanding. It was originally developed for text understanding and has been widely used for IS following [7] by using dynamically generated Q/A pairs. It has been used for the same purpose in [10, 26, 14]). Our intention in using QUEST was to demonstrate that plot interleaving did not impair story understanding.

For the study participants were randomly assigned to groups to watch either interleaved or non-interleaved presentations of two narrative (order of narrative viewing was controlled). Afterwards they

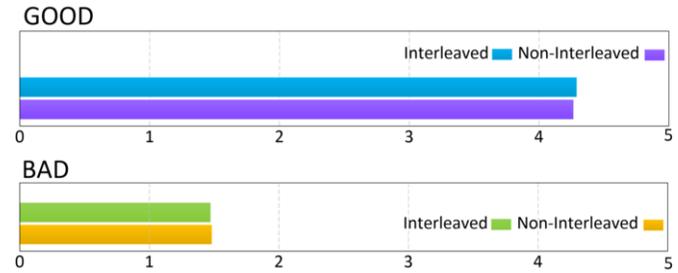


Figure 10. Results of QUEST evaluation: mean responses for GOOD and BAD question-answer (QA) pairs for user groups who watched interleaved or non-interleaved narrative presentations. Understandability is indicated if scores for GOOD QA pairs are high and low for BAD pairs. As anticipated, the results show no difference in story understandability between the different variants (see text for further details).

were asked to assign goodness of answer (GOA) values to question-answer (QA) pairs and then the correlation with the predicted quality given by the QUEST graphs was assessed. For example, a sample pair from our experiments was: Q: “What did Dr. Thompson do about his patients treatment?” A: “He changed the treatment.” and where GOA depends on narrative content i.e. whether treatment changed.

Each user rated the GOA of six QA pairs, selected from the QUEST model, with a value from 1 (very bad) to 5 (very good). Ratings were compared against measures of reachability and arc distance in the QUEST graph for the narrative with expected values for the GOA with 5 (very good) for those with arc distance 1, 4 for those with arc distance 2, and so on, with 1 (very bad) expected for QA pairs that were unreachable in the QUEST graph. After [25] we partitioned the question space into “good” and “bad” QA pairs, where in a rating system of 1–5 good and bad pairs have a system predicted GOA greater or less than neutral respectively and with understandability indicated if scores for bad pairs were low and high for good.

The results were as follows: mean values for good QA pairs of 4.25 and 4.27 out of 5 (for interleaved and non-interleaved respectively); and for bad pairs 1.47 and 1.44. These results indicate understandability and also no difference between interleaved and non-interleaved presentations since in both cases t-test shows no significant difference. This is a promising result supporting our expectation that there is no loss of story understanding when narrative subplots are interleaved.

9 Conclusions

We have extended our earlier landmark approach to plan-based narrative generation to enable it to take into account subplots within the same framework. It should be emphasised that this approach is compatible with previous PDDL based representation of narrative actions and doesn’t require bespoke narrative representation.

In the evaluation we demonstrated that the approach is able to generate narratives that conform to different subplot profiles, specified in terms of the number of subplots interleaved and the relative time spent on each presentation. Results of our user evaluation also supported our prediction that generated narratives correspond to user expectations of the genre with the majority of users preferring presentations with interleaved subplots, and with the majority of positive comments attributing this to the subplots.

Overall, these results demonstrate the real potential of the approach to automatically generate interleaved narratives that match user expectations of human-authored narratives and which are clearly preferred by the users.

REFERENCES

- [1] Ruth Aylett, João Dias, and Ana Paiva, 'An Affectively Driven Planner for Synthetic Characters', in *Proc. of 16th Int. Conference on Automated Planning and Scheduling (ICAPS)*, (2006).
- [2] B-C Bae and R. Michael Young, 'A Use of Flashback and Foreshadowing for Surprise/Arousal in Narrative Using a Plan-Based Approach', in *Proc. of 1st Int. Conference on Interactive Digital Storytelling (ICIDS)*, (2008).
- [3] David Bordwell, *Narration in the Fiction Film*, Madison: University of Wisconsin, 1985.
- [4] David Bordwell and Kristin Thompson, *Film Art: An Introduction*, McGraw-Hill Higher Education (9th Edition), 2010.
- [5] Marc Cavazza, David Pizzi, Fred Charles, Thuriid Vogt, and Elisabeth André, 'Emotional input for character-based interactive storytelling', in *Proc. of the 8th Int. Joint Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, IFAAMAS, (2009).
- [6] Yun-Gyung Cheong and R Michael Young, 'Suspenser: A Story Generation System for Suspense', *Transactions on Computational Intelligence and Artificial Intelligence in Games*, (2014).
- [7] David B. Christian and R. Michael Young, 'Comparing Cognitive and Computational Models of Narrative Structure', in *Proc. of the 19th National Conference on AI (AAAI)*, (2004).
- [8] B. C. Gazen and C. Knoblock, 'Combining the Expressiveness of UCPOP with the Efficiency of Graphplan', in *Proc. of the European Conf. on Planning (ECP)*, (1997).
- [9] Stephen Gilroy, Julie Porteous, Fred Charles, Marc Cavazza, Eyal Soreq, Gal Raz, Limor Ikar, Ayelet Or-Borichov, Udi Ben-Arie, Ilana Klovatch, and Talma Hendler, 'A Brain-Computer Interface to a Plan-based Narrative', in *Proc. of the 23rd Int. Joint Conf. on Artificial Intelligence (IJCAI-13)*, (2013).
- [10] Stephen W. Gilroy, Julie Porteous, Fred Charles, and Marc Cavazza, 'Exploring Passive User Interaction for Adaptive Narratives', in *Proc. of the 17th Int. Conf. on Intelligent User Interfaces (IUI-12)*, (2012).
- [11] P. Haslum, 'Narrative Planning: Compilations to Classical Planning', *Journal of AI Research*, **44**, 383–395, (2012).
- [12] J. Hoffmann and B. Nebel, 'The FF Planning System: Fast Plan Generation through Heuristic Search', *Journal of AI Research*, **14**, 253–302, (2001).
- [13] Tomas Hromada, Martin Černý, Michal Bída, and Cyril Brom, 'Generating Side Quests from Building Blocks', in *Proc. of the 8th Int. Conf. on Interactive Digital Storytelling (ICIDS)*, (2015).
- [14] Arnav Jhala and R Michael Young, 'Cinematic visual discourse: Representation, generation, and evaluation', *Computational Intelligence and AI in Games, IEEE Transactions on*, **2**(2), 69–81, (2010).
- [15] Michael Lebowitz, 'Story-telling as planning and learning', *Poetics*, **14**(6), 483–502, (1985).
- [16] Boyang Li and Mark O. Riedl, 'An Offline Planning Approach to Game Plotline Adaptation', in *Proc. of the 6th National Conference on AI and Interactive Digital Entertainment (AIIDE)*, (2010).
- [17] Bradford Mott, Sunyoung Lee, and James Lester, 'Probabilistic goal recognition in interactive narrative environments', in *Proc. of the 21st National Conference on Artificial Intelligence (AAAI)*, (2006).
- [18] Mark J. Nelson and Michael Mateas, 'Search-Based Drama Management in the Interactive Fiction Anchorhead', in *Proc. of the 1st AI and Interactive Digital Entertainment Conference (AIIDE)*, (2005).
- [19] Brian O'Neill and Mark Riedl, 'Dramatis: A Computational Model of Suspense', in *Proc. of the 28th National Conference on AI (AAAI)*, (2014).
- [20] J. Porteous, F. Charles, and M. Cavazza, 'NetworkING: using Character Relationships for Interactive Narrative Generation', in *Proc. of 12th Int. Conf. on Autonomous Agents and MultiAgent Systems (AAMAS)*, IFAAMAS, (2013).
- [21] J. Porteous, J. Teutenberg, D. Pizzi, and M. Cavazza, 'Visual Programming of Plan Dynamics using Constraints and Landmarks', in *Proc. of the 21st Int. Conf. on Automated Planning and Scheduling (ICAPS)*, (2011).
- [22] Julie Porteous, Marc Cavazza, and Fred Charles, 'Applying Planning to Interactive Storytelling: Narrative Control using State Constraints', *ACM Transactions on Intelligent Systems and Technology (ACM TIST)*, **1**(2), 1–21, (2010).
- [23] M. Riedl, 'Incorporating Authorial Intent into Generative Narrative Systems', in *Proc. of the AAAI Spring Symposium: Intelligent Narrative Technologies II*, (2009).
- [24] M. Riedl, CJ Saretto, and R.M. Young, 'Managing interaction between users and agents in a multi-agent storytelling environment', in *Proc. of the 2nd Int. Conf. on Autonomous Agents and MultiAgent Systems (AAMAS)*, IFAAMAS, (2003).
- [25] M. Riedl and R.M. Young, 'Open-world planning for story generation', in *Proc. of the Int. Joint Conference on AI (IJCAI)*, (2005).
- [26] Mark O. Riedl and R. Michael. Young, 'Narrative Planning: Balancing Plot and Character', *Journal of AI Research*, **39**, 217–267, (2010).
- [27] ER, 1994. (TV Series) Warner Brothers, 1994-2009.
- [28] HOUSE, 2004. (TV series, Universal, 2004-2012).
- [29] MASH, 1972. (TV series) 20th Century Fox, 1972-1983.
- [30] SCRUBS, 2001. (TV Series), ABC Studios, 2001-2010.
- [31] Seinfeld, 1989. (TV series) West-Shapiro Productions and Castle Rock Entertainment, 1989-1998.
- [32] Manu Sharma, Santiago Ontañón, Manish Mehta, and Ashwin Ram, 'Drama Management and Player Modeling for Interactive Fiction Games', *Computational Intelligence*, **26**(2), (2010).
- [33] Greg M. Smith, 'Plotting a TV Show About Nothing: Patterns of Narration in Seinfeld', in *Creative Screenwriting*, volume 2(1), (1995).
- [34] T. J. Smith, D. T. Levin, and J. Cutting, 'A Window on Reality: Perceiving Edited Moving Images', *Current Directions in Psychological Science*, **21**, (2012).
- [35] Kristin Thompson, *Storytelling in Film and Television*, Harvard University Press, 2003.
- [36] Peter Weyhrauch, *Guiding interactive drama*, 1997.
- [37] R. M. Young, 'Creating Interactive Narrative Structures: The Potential for AI Approaches', in *AAAI Spring Symposium in Artificial Intelligence and Entertainment*. AAAI Press, (2000).
- [38] R. Michael Young, 'Notes on the use of plan structures in the creation of interactive plot', in *Proc. of the AAAI Fall Symposium on Narrative Intelligence*, (1999).

Hybrid Gaussian and von Mises Model-Based Clustering

Sergio Luengo-Sanchez¹ and Concha Bielza and Pedro Larrañaga

Abstract. Data collected about a phenomenon often measures its magnitude and direction. The most common approach to clustering this data assumes that directional data can be modeled as Gaussian. However, directional data has special properties that conventional statistics cannot handle. To deal with them, other approaches like the von Mises distribution must be applied. In this paper we present a new model based on mixtures of Bayesian networks to simultaneously cluster both linear and directional data.

1 Introduction

In a wide range of scientific fields, angle measurement is required to represent information about a phenomenon. Usually, this data comes together with its magnitude. Examples are in meteorology with wind direction and speed measurements [10], rhythmometry, medicine or demography [3, 4].

Typically, when this data is collected, an exploratory analysis is performed to reveal patterns. Cluster analysis partitions data into groups of homogeneous observations. A probabilistic clustering approach is model-based clustering [14, 31, 32]. Model-based clustering assumes that data was generated by a statistical model. Finite mixtures of Gaussians are the most commonly used distribution in model-based clustering because they can approximate any non-directional multivariate density given enough components [45].

However, mixtures of Gaussians which are based on classical statistics, are not suitable for clustering directional data because they cannot handle its periodicity. For example, given angles 1° and 359° , the linear mean would be 180° . This points in the opposite direction to the angular mean angle which is 0° . To address this problem, a popular choice for the component distribution when data is positioned on the surface of a sphere or hypersphere is the von Mises-Fisher (vMF) distribution [2, 17]. The vMF distribution is the circular analogue of the multivariate normal distribution whose covariance matrix is a multiple of the identity matrix [26]. This model outperforms others based on linear distributions for problems such as text categorization and gene expression analysis [2, 47]. Another common choice when data is multimodal and is distributed on a torus or hypertorus are the mixtures of von Mises (vM) distributions. Mixtures of bivariate [29] and multivariate [25] vM distributions have been applied successfully in bioinformatics to characterize the structure of proteins.

Because mixtures of Gaussian, vM or vMF distributions partially solve the problem of simultaneously clustering directional and linear data, several distributions have been proposed to cluster cylindrical data, that is, a linear and a directional variable together [9, 16, 28, 38]. They all represent the joint probability density function of a univariate Gaussian and a univariate vM distribution. However, data is usu-

ally multidimensional and consists of a large number of linear and directional variables.

In our work we introduce a hybrid model inspired by a recent directional extension of the naive Bayes classifier [23], mixing multivariate Gaussian and multivariate vMF distributions. We adapt this model to learn a mixture where each component is the product of a multivariate Gaussian and several independent vM distributions. This is a learning from incomplete data problem, where variables are observed and the cluster membership is a hidden variable [22]. We exploit conditional independence assumptions encoded by the Bayesian network to factorize the joint probability distributions. This decomposition enables efficient model learning.

The rest of this paper is organized as follows. Section 2 introduces background material. Section 3 describes a clustering algorithm for naive Bayes, that is, a Bayesian network with structural constraints, where data are directional only. Under the naive Bayes assumption, Section 4 extends the previous model to the hybrid case mixing Gaussian and vM distributions. Additionally, the hybrid case is further improved by learning the Bayesian network structure from Gaussian variables to relax the topology of the Gaussian part. The approaches are evaluated and results are discussed in Section 5. Section 6 outlines the conclusions and future research.

2 Background

2.1 Von Mises distribution

The univariate vM distribution for an angle $\theta \in [0, 2\pi]$ defines a probability density function over points on a circle with a radius of one according to

$$f_{vM}(\theta; \mu, \kappa) = \frac{e^{\kappa \cos(\theta - \mu)}}{2\pi I_0(\kappa)}, \quad (1)$$

where $\mu \in [0, 2\pi]$ is the mean direction, $\kappa \geq 0$ is the concentration parameter and $I_0(\kappa)$ is the modified Bessel function of the first kind and order zero

$$I_0(\kappa) = \frac{1}{\pi} \int_0^\pi e^{\kappa \cos \theta} d\theta. \quad (2)$$

Parameters μ and $1/\kappa$ are analogous to the normal distribution μ and σ^2 , so the vM distribution is also known as the circular normal distribution.

2.2 Model-based clustering

Consider the finite set $\mathbf{X} = \{X_1, X_2, \dots, X_L\}$ of variables and let $\mathbf{x} = \{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N\}$ be a dataset where each \mathbf{x}^i assigns a value to all variables in \mathbf{X} . The goal of model-based clustering is to recover the statistical model that generated \mathbf{x} . Finite mixture models provide a formal setting for model-based clustering. In finite mixture models,

¹ Universidad Politécnica de Madrid, Spain, email: sluengo@fi.upm.es

each cluster is represented by a probability distribution. The linear superposition of the above distributions generates the finite mixture density function

$$f(\mathbf{x}; \boldsymbol{\pi}) = \sum_{k=1}^K \pi_k f_k(\mathbf{x}; \boldsymbol{\theta}_k), \quad (3)$$

where the mixing proportions π_k are nonnegative and sum to 1, $\boldsymbol{\theta}_k$ are the parameters for cluster k , and K is the number of mixture components. In our case, f_k denotes a joint probability distribution over \mathbf{x} encoded as a Bayesian network.

A Bayesian network [22, 35] is a directed acyclic graph that represents the probabilistic relationships among variables in \mathbf{X} . It consists of a pair $B = (S, \boldsymbol{\theta})$. The first component, S , is the graph structure whose vertices correspond to the variables X_1, X_2, \dots, X_L . The second component, $\boldsymbol{\theta}$, represents the parameters. We use $\mathbf{Pa}_l = \{U_{1l}, U_{2l}, \dots, U_{Tl}\}$ to denote the parents of node X_l in S , where U_{tl} is its t -th parent. Structure S encodes the local Markov property, i.e., each variable X_l is independent of its non-descendants given its parents \mathbf{Pa}_l . Hence, the joint probability distribution can be factorized as

$$f(\mathbf{x}) = \prod_{l=1}^L f(X_l | \mathbf{Pa}_l; \boldsymbol{\theta}). \quad (4)$$

Naive Bayes (NB) [13] is the simplest Bayesian network structure and one of the most extended models for classification. All variables are assumed to be conditionally independent given the class variable. Accordingly, the class variable, or cluster variable Z in our case, is the only parent of all variables in the graph and no more arcs are allowed. Although there are few real-world cases where this strong assumption about the conditional dependencies holds, is simple, its accuracy is competitive and it has a small generalization error. One of its advantages is that the probability distribution of Z is efficiently computed when the data is complete because the maximum likelihood estimator (MLE) or maximum a posteriori (MAP) methods can be applied to estimate parameters. In a clustering problem, however, cluster Z is a latent or hidden variable. Thus, this is a parameter learning problem with missing data where the values of the class variable are unknown. When working with missing data, we need to estimate the optimum parameters of the model at the same time as we try to hypothesize the cluster of each instance. Because of its simplicity and efficiency the expectation-maximization (EM) algorithm [12] is the most popular method for dealing with this problem.

EM addresses the missing data problem selecting a starting point, which is either an initial set of parameters or an initial assignment to the cluster variable. Once we have a parameter set, we can apply inference to complete the data or, conversely, once we have the complete data, we can estimate the set of parameters from MLE. Thus, there are two separate steps: use parameters to complete the data (expectation step)

$$Q_i(z^i) = p(z^i | \mathbf{x}^i; \boldsymbol{\theta}) = \frac{f(\mathbf{x}^i | z^i; \boldsymbol{\theta}) p(z^i; \boldsymbol{\theta})}{\sum_z f(\mathbf{x}^i | z; \boldsymbol{\theta}) p(z; \boldsymbol{\theta})}, \quad (5)$$

and then estimate a new set of parameters from the complete data (maximization step)

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{i=1}^N \sum_{z^i} Q_i(z^i) \log \frac{f(\mathbf{x}^i, z^i; \boldsymbol{\theta})}{Q_i(z^i)}. \quad (6)$$

Both steps are repeated iteratively, and the likelihood improves until convergence to a local maximum.

In clustering with Bayesian networks, the EM algorithm only optimizes one of the components of the pair $B = (S, \boldsymbol{\theta})$, namely the parameters $\boldsymbol{\theta}$. The structure S must be preset and fixed. However, finding the underlying structure of the data has advantages such as discovering dependence relations between variables and efficient factorization. There are several successful methods for structure and parameters learning when data is complete [18, 22]. If data is complete, a heuristic search [11, 22, 46] for an optimal structure may compare the scores [1, 39] of the current model and the model built after adding or removing arcs between variables. When data is incomplete, this search is no longer viable because the network score does not decompose, and inference is needed at every step of the learning process to evaluate the model. Structural EM [15, 36] is a method based on EM to learn structure and parameters when data is incomplete.

Structural EM introduces structural learning as an additional step in the EM algorithm. It starts with a given initial structure S and a set of parameters $\boldsymbol{\theta}$. Then, it iterates between a pair of steps. First, the parameters are maximized according to the standard EM algorithm because it is cheaper than searching for a better model. When it converges, the algorithm searches for the best structure completing data with the output of the expectation step. Both steps are repeated until convergence. Any general-purpose heuristic search algorithm for structural learning can be applied. A common choice for the score to be maximized is the Bayesian information criterion (BIC) [39] because, if the search procedure always finds a better structure at each iteration, BIC guarantees that the score increases monotonically. BIC is a measure that adds a penalty to the log-likelihood \hat{L} based on the number of model parameters v . This score is used to choose the clustering model (parameterization, structure and number of clusters) to ensure that the selected models are not too complex. BIC is computed as

$$BIC = 2\hat{L} - v \log(N), \quad (7)$$

where N is the size of the dataset.

3 Clustering directional data

Directional data clustering has been addressed previously in the literature based on the EM framework and vM distribution. Several studies have modeled directional data with mixtures of univariate vM distribution [7, 30, 33]. The use of the EM algorithm to cluster mixtures of bivariate vM distributions is investigated in [29]. The maximization step was tackled by means of numerical optimization because the MLE does not have a closed-form solution. A multivariate vM mixture model is studied in [27] proposing an approximation of the intractable normalizing constant when data is highly concentrated. This approach computes MLE according to the method of moments and the EM. However, the likelihood function may not always be monotonically increasing because it is using an approximation, although it does usually stabilize to some local maximum.

In this section, we first introduce a mixture of Bayesian networks for clustering data when variables are only directional, i.e., $\mathbf{Y} = Y_1, Y_2, \dots, Y_M$. In fact, the goal is to learn the parameters $\boldsymbol{\theta}$ of Bayesian networks given that they have a NB structure S . Figure 1 shows the graphical structure of the proposed model for each mixture component where, assuming that directional data is distributed according to the vM distribution, nodes represent vM variables and arcs encode the dependencies.

We choose the NB structure because its factorization can solve the above clustering problems for multivariate vM distributions. To

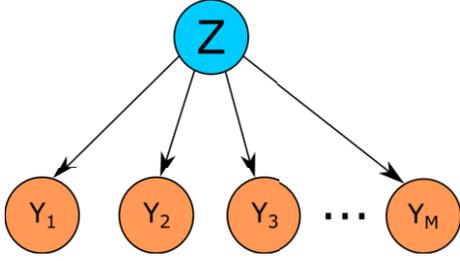


Figure 1. Graphical structure S for the naive Bayes model. The latent variable Z is the parent of all the variables, ruling out all other arcs. Thus, given Z , all the variables are conditionally independent of each other.

exploit the benefits of NB factorization, however, data must be complete. This is not the case in clustering because Z is hidden. Therefore, we need to apply the EM algorithm. First, we compute the expected values of Z according to the expectation step (Equation (5)). This completes the data so the joint distribution can be factorized to

$$f(\mathbf{Y}, Z; \boldsymbol{\theta}) = p(Z; \boldsymbol{\theta}) \prod_{m=1}^M f(Y_m | Z; \boldsymbol{\theta}), \quad (8)$$

which is a product of conditional probabilities such that each variable of the model contributes a factor of that product. This representation shows that NB naturally extends to M -dimensional data, which is one of the benefits of factorization.

Once the expectation step has been calculated, parameter estimation is carried out in the maximization step. Substituting the joint probability distribution of Equation (8) in Equation (6) results in

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{i=1}^N \sum_{z^i} Q_i(z^i) \left[\sum_{m=1}^M \log f(y_m^i | z^i; \boldsymbol{\theta}) + \log p(z^i; \boldsymbol{\theta}) \right]. \quad (9)$$

Thus, the sum of the log-likelihood of each variable must be maximized for each cluster to compute the MLE of $\boldsymbol{\theta}$ in the mixture. Representing MLE as a summation simplifies parameter estimation so that each variable is optimized locally, i.e., independently of the others. The biggest advantage of this property is that MLE equations are closed-form and are computed efficiently for each variable Y_m of cluster k as [8]

$$\begin{aligned} \hat{\pi}_k &= \frac{\sum_{i=1}^N Q_i(k)}{N} \\ \hat{\mu}_k^m &= \arctan \left(\frac{\sum_{i=1}^N Q_i(k) \sin(y_m^i)}{\sum_{i=1}^N Q_i(k) \cos(y_m^i)} \right) \\ \hat{\kappa}_k^m &= A^{-1} \left(\frac{\sum_{i=1}^N Q_i(k) \cos(y_m^i - \hat{\mu}_k^m)}{\sum_{i=1}^N Q_i(k)} \right), \end{aligned} \quad (10)$$

where $A(\hat{\kappa}_k^m) = \frac{I_1(\hat{\kappa}_k^m)}{I_0(\hat{\kappa}_k^m)}$. An accurate approximation for A^{-1} is presented in [6].

This approach based on exploiting independence constraints avoids the numerical optimization needed for the mixtures of bivariate vM distributions. It also ensures that likelihood increases monotonically in each EM step until convergence to a local maximum even though data is not concentrated. However, NB does not capture information about the dependence between variables inside each cluster.

4 Hybrid Gaussian and von Mises clustering

Some practical scenarios involve several linear and directional variables. For example, geomagnetic and ionospheric signals are apparently associated with earthquake prediction [24, 42, 43, 44]. Thus, a dataset for this field of study is composed of measure such as magnitude and coordinates of an earthquake, speed of solar wind, ionospheric total electron content, magnetic activity, etc. A hybrid model for jointly clustering multivariate linear and multivariate directional variables can be achieved by means of mixtures of Bayesian networks. In this section we present a hybrid Gaussian and von Mises clustering from a preset NB structure. Then, we improve this model by learning the graph structure among Gaussian variables during the clustering process.

4.1 Conditionally independent variables

In Section 3, NB dependence constraints among variables were exploited to factorize the joint probability as a product where each factor corresponded to one variable. When they were combined using the EM algorithm to estimate the cluster parameters, the parameters of each variable were maximized independently of the others, resulting in closed-form equations. The advantages provided by the factorization of the NB structure on directional data are now extrapolated to achieve a model for multidimensional hybrid data.

Given a set of linear $\mathbf{X} = X_1, X_2, \dots, X_L$ and directional $\mathbf{Y} = Y_1, Y_2, \dots, Y_M$ variables, a NB structure S (see Figure 2) and the expected values of Z computed according to the expectation step, the joint probability distribution factorizes as

$$f(\mathbf{X}, \mathbf{Y}, Z; \boldsymbol{\theta}) = p(Z; \boldsymbol{\theta}) \prod_{l=1}^L f(X_l | Z; \boldsymbol{\theta}) \prod_{m=1}^M f(Y_m | Z; \boldsymbol{\theta}). \quad (11)$$

This introduces a new product of conditional probabilities of Gaussians with respect to Equation (8). Consequently, the model generalizes to higher dimensions and handles any number of linear and directional variables.

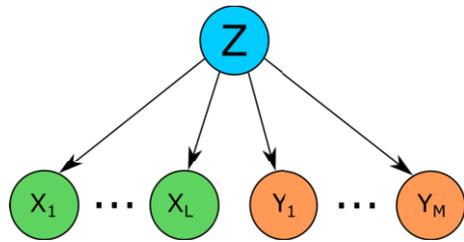


Figure 2. Graphical structure S for the NB hybrid model. Hybrid Gaussian and vM NB consists of a graphical structure where green nodes are Gaussian variables, orange nodes are vM variables and Z is the parent of all the variables.

The maximization step is computed by substituting the joint probability with the above factorization (11):

$$\begin{aligned} \hat{\boldsymbol{\theta}} &= \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \sum_{i=1}^N \sum_{z^i} Q_i(z^i) \\ &\left[\sum_{l=1}^L \log f(x_l^i | z^i; \boldsymbol{\theta}) + \sum_{m=1}^M \log f(y_m^i | z^i; \boldsymbol{\theta}) + \log p(z^i; \boldsymbol{\theta}) \right] \end{aligned} \quad (12)$$

Because of the NB structure assumption, parameter estimation involves the maximization of sums of log-likelihoods. Therefore, the parameters of Gaussian and vM variables are estimated locally. Hence, the maximization step for vM variables can be computed according to Equation (10) and for Gaussian variables according to the well-known equations

$$\begin{aligned}\hat{\mu}_k^l &= \frac{\sum_{i=1}^N Q_i(k) x_i^l}{\sum_{i=1}^N Q_i(k)} \\ \hat{\sigma}_k^l &= \sqrt{\frac{\sum_{i=1}^N Q_i(k) (x_i^l - \hat{\mu}_k^l)^2}{\sum_{i=1}^N Q_i(k)}}.\end{aligned}\quad (13)$$

4.2 General hybrid model

Strong constraints were imposed on the structure of the Bayesian network for the above models. However, when the structure is pre-set and fixed, some beneficial properties of the Bayesian networks are lost. Discovering the graph topology provides information about the relations of dependence between variables and may improve the model's accuracy. The Structural EM algorithm [15, 36] defines a flexible approach to clustering, automatically learning the structure of the network during the clustering process.

The proposed multivariate model aims to fit hybrid data, so some relations between variables must be constrained in the learning structure step of the Structural EM algorithm to exploit factorization efficiently as we did in previous sections. First, we assume independence between Gaussian and vM variables given the parent node Z . Second, vM variables should be conditionally independent of each other to achieve closed-form equations for the maximization step. As a result, a new scenario is set where Gaussian dependencies are freely learned by Structural EM (without constraints), the structure of vM variables is fixed and dependencies between Gaussian and vM variables are ruled out (Figure 3).

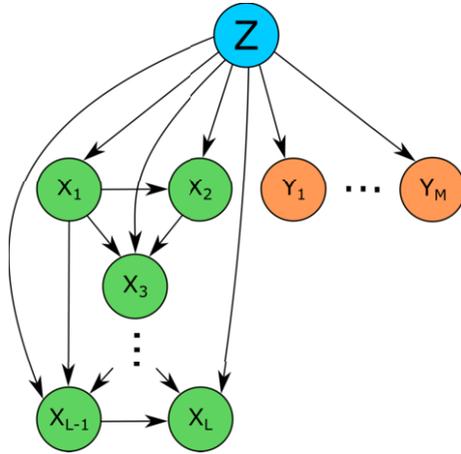


Figure 3. An example of the graphical structure S for the general hybrid model. The structure of Gaussian variables is learned during the clustering process. vM variables are independent given Z which is the parent of all the variables. There is no dependence between Gaussian and vM variables.

The first step of Structural EM algorithm, EM computation, is addressed as in previous sections by computing the expected values of Z according to the expectation step, we find that a new distribution

is factorized as

$$f(\mathbf{X}, \mathbf{Y}, Z; \boldsymbol{\theta}) = p(Z; \boldsymbol{\theta}) \prod_{l=1}^L f(X_l | \mathbf{Pa}_l, Z; \boldsymbol{\theta}) \prod_{m=1}^M f(Y_m | Z; \boldsymbol{\theta}), \quad (14)$$

which is quite similar to the factorization shown in Equation (11). The difference lies in the decomposition of the conditional probability distribution of Gaussian variables because other Gaussian variables may be their parents (\mathbf{Pa}_l). In (14), $f(X_l | \mathbf{Pa}_l, Z; \boldsymbol{\theta})$ is a linear Gaussian, i.e., a linear combination of its Gaussian parents

$$f(X_l | \mathbf{Pa}_l, Z = k; \boldsymbol{\theta}) = \mathcal{N}(\beta_{0k} + \boldsymbol{\beta}_k \mathbf{Pa}_l, (\sigma_k^l)^2), \quad (15)$$

where $\boldsymbol{\beta}_k$ is the vector of coefficients of the linear Gaussian for cluster k . When the only parent of a Gaussian variable is Z , then $\beta_{0k} = \mu_k$.

Parameter estimation is tackled by the maximization step substituting the joint probability distribution by its factorization (14):

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \operatorname{argmax}_{\boldsymbol{\theta}} \sum_{i=1}^N \sum_{z^i} Q_i(z^i) \\ &\left[\sum_{l=1}^L \log f(x_l^i | \mathbf{Pa}_l^i, z^i; \boldsymbol{\theta}) + \sum_{m=1}^M \log f(y_m^i | z^i; \boldsymbol{\theta}) + \log p(z^i; \boldsymbol{\theta}) \right].\end{aligned}\quad (16)$$

As in previous cases, due to the independence assumption represented by the structure, MLE entails maximizing a sum of log-likelihoods to locally estimate vM and Gaussian variables without Gaussian parents according to Equations (10) and (13). Nevertheless, some Gaussian variables have Gaussian parents on which the computation of their MLE depends. For these variables, we set

$$\mathbb{E}_D[X] = \sum_{i=1}^N Q_i(k) x^i,$$

and we get the MLE of $\hat{\boldsymbol{\beta}}$ coefficients from the following system of equations:

$$\begin{aligned}\mathbb{E}_D[X_l] &= \hat{\beta}_{0k} \mathbb{E}_D[\mathbf{1}] + \hat{\beta}_{1k} \mathbb{E}_D[U_{1l}] + \cdots + \hat{\beta}_{Tk} \mathbb{E}_D[U_{Tl}] \\ \mathbb{E}_D[X_l \cdot U_{1l}] &= \hat{\beta}_{0k} \mathbb{E}_D[U_{1l}] + \hat{\beta}_{1k} \mathbb{E}_D[U_{1l} \cdot U_{1l}] + \cdots \\ &\quad + \hat{\beta}_{Tk} \mathbb{E}_D[U_{1l} \cdot U_{Tl}] \\ &\vdots \\ \mathbb{E}_D[X_l \cdot U_{Tl}] &= \hat{\beta}_{0k} \mathbb{E}_D[U_{Tl}] + \hat{\beta}_{1k} \mathbb{E}_D[U_{1l} \cdot U_{Tl}] + \cdots \\ &\quad + \hat{\beta}_{Tk} \mathbb{E}_D[U_{Tl} \cdot U_{Tl}].\end{aligned}\quad (17)$$

Once the coefficients are known, the variance of X_l is computed as

$$(\hat{\sigma}_k^l)^2 = \frac{\sum_{i=1}^N Q_i(k) (x_i^l - \hat{\beta}_{0k} - \hat{\boldsymbol{\beta}}_k \mathbf{Pa}_l^i)^2}{\sum_{i=1}^N Q_i(k)}. \quad (18)$$

The expectation and maximization steps iterate until convergence.

The EM algorithm outputs complete data and a set of parameters $\boldsymbol{\theta}$. Structural EM applies this outcome to learn the structure of the Bayesian network. When complete data is available, heuristic search algorithms optimize the score locally due to the decomposability property. Thus, part of the network topology can be optimized, while the rest remains unchanged. We exploit this point to search the structure for the Gaussian variables. We choose the BIC score to search for the best structure because it guarantees that the algorithm always converges in a local maximum.

5 Experiments and results

In this section, we report two experiments with different goals. On the one hand, we numerically evaluate all the proposed models by clustering artificial datasets and measuring the accuracy of the estimated parameters. On the other hand, we introduce an application of the general hybrid model for neuroscience to cluster neuronal dendritic spines.

5.1 Artificial datasets

To achieve a deeper insight into the suitability of the above models for clustering tasks, we evaluate them numerically. We study the performance of the vM and general hybrid models comparing their goodness of fit and their accuracy against the most common choice for multivariate directional data modeling, the Gaussian mixture model. This study should highlight the differences of applying a linear distribution in place of a directional distribution for directional data modeling. For all the experiments, data was simulated to find out beforehand the component of the mixture that generated each instance and the model parameters for comparison with the outcome of the experiment. For each experiment we rebooted the algorithm 10 times, changing the initial parameterization each time. We saved the model that maximized the BIC score.

5.1.1 Von Mises model

In the first place, we evaluated the goodness of fit of the model based on mixtures of NB for vM variables which we compare with the Gaussian mixture model. To do this, we simulated data from three clusters and two variables $\Theta \sim vM(\mu_\Theta, \kappa_\Theta)$, $\Phi \sim vM(\mu_\Phi, \kappa_\Phi)$. We set the concentration parameter κ to low values so clusters overlap. We analyzed the goodness of fit of both models depending on the sample size ($N = 30, 300$).

Table 1. Comparison of parameter estimation between vM and Gaussian models changing the sample size. Each cluster is denoted by Cl., followed by its number. For the Gaussian mixture model κ was computed as $1/\sigma^2$.

We use boldface to denote the value of the distribution that best approximates each parameter for each cluster.

Variable	Parameters	Original		
		Cl. 1	Cl. 2	Cl. 3
Θ	μ_Θ	0	$\pi/2$	π
	κ_Θ	1	1	1
Φ	μ_Φ	0	$\pi/2$	π
	κ_Φ	2	2	3

N = 30							
Variable	Parameters	vM Clustering			Gaussian Clustering		
		Cl. 1	Cl. 2	Cl. 3	Cl. 1	Cl. 2	Cl. 3
Θ	$\hat{\mu}_\Theta$	-0.53	1.68	2.77	0.79	2.1	5.57
	$\hat{\kappa}_\Theta$	3.89	2.94	1.44	2.44	1.26	5.66
Φ	$\hat{\mu}_\Phi$	0.54	0.77	3.19	0.79	1.71	6.06
	$\hat{\kappa}_\Phi$	2.89	2.22	3.18	2.87	0.3	100

N = 300							
Variable	Parameters	vM Clustering			Gaussian Clustering		
		Cl. 1	Cl. 2	Cl. 3	Cl. 1	Cl. 2	Cl. 3
Θ	$\hat{\mu}_\Theta$	-0.36	1.59	2.87	5.03	1.62	3.25
	$\hat{\kappa}_\Theta$	1.4	1.34	0.58	1.13	1.06	0.29
Φ	$\hat{\mu}_\Phi$	0.08	1.27	3.25	4.87	1.48	1.84
	$\hat{\kappa}_\Phi$	1.85	1.47	3.33	0.66	0.5	0.92

Results from Table 1 show that the mixtures of NB for vM variables yield better results for estimating the mean of the distributions,

especially when the mean is 0, than the Gaussian mixture model, which fails due to the special properties of the directional data. When the sample size increases, the proposed model further improves the estimation of κ values.

Then, we evaluated the performance of the clustering algorithm by changing the number of clusters ($K = 3, 5, 10$) and variables ($M = 10, 25, 50$). Modifying the number of variables provides information about the accuracy of the model when data is concentrated or sparse. Varying the number of clusters in a bounded and fixed space we measure the performance of the method as more clusters overlap. For the experiment, complete data was available, i.e., variables and cluster labels were known. We started by hiding the cluster label of all instances and clustering the data. We crisply assigned each instance to the cluster with maximum membership probability. As a result, each instance belonged to one group. Then, we compared the real label with label provided by the clustering algorithm to get its hit rate. The accuracy of the proposed model was compared against the Gaussian mixture model, see Table 2.

Table 2. Hit rate of vM vs Gaussian mixture models. We simulated 100 instances from each cluster. The best results are denoted in boldface.

N. Cl./N. Var.	vM clustering			Gaussian clustering		
	10	25	50	10	25	50
3	99%	100%	100%	94.6%	99.6%	68.33%
5	97%	100%	100%	47.2%	59.2%	100%
10	56.2%	99.1%	100%	38.7%	40.4%	38.3%

Analyzing Table 2 we find that mixtures of vM distributions improve their accuracy as the number of variables increases. This is because clusters are further apart and consequently easily separated in higher dimensions. The opposite applies when the number of clusters grows. In this case the clusters overlap. Therefore, the boundaries between them are not clearly defined, and clustering algorithms are less accurate. However, the Gaussian mixture model behaves differently. Even though data sparsity increases when the number of variables is 50, the accuracy of Gaussian variables decays for 3 and 10 clusters with respect to the case when there are 25 variables. For all cases vM clustering achieves better results than Gaussian mixture models.

5.1.2 General hybrid model

To evaluate the general hybrid model, we adapted the above experiments to hybrid data. We started by validating the goodness of fit and the structure learned by the model. To do this, we manually defined a Bayesian network with five Gaussian nodes and two vM nodes (Figure 4). As before the number of clusters is 3. We simulated 100 instance of this Bayesian network for each cluster. Then, we applied the general hybrid model to learn the model parameters and the structure from Gaussian variables.

We measured the distance between the original and the learned structure according to the Hamming distance, i.e., the number of changes in a BN structure needed to turn it into another. The operations are add an arc, drop an arc or revert arc. Figure 4 shows that we only need to add one arc ($X_5 \rightarrow X_3$) to achieve the original structure, so the Hamming distance was one and the structure was an accurate approximation.

Table 3 shows the results of parameter estimation. First, we observe that X_3 has one parameter less because the learned structure missed an arc with respect to the original structure. The elimination of the coefficient β_5 is offset by the remaining coefficients $\hat{\beta}_0$ and $\hat{\beta}_1$. Despite this fact, the value of $\hat{\sigma}$ accurately approximates the original

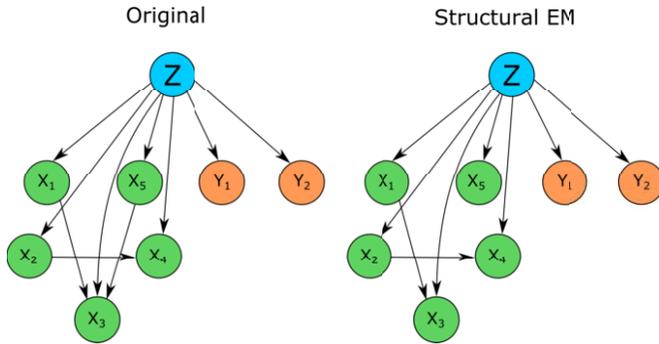


Figure 4. Original structure of the BN and structure learned by the general hybrid model. The Structural EM approximates the original structure quite well but drops the arc from X_5 to X_3 .

value for that variable. Also note that good approximations were obtained for most of the estimated parameters, except in some cases like the mean of X_1 for cluster 2 and the mean of X_2 for cluster 3. Of particular note are the good results for the directional variables, especially for the means.

Table 3. Parameter estimation of general hybrid model with respect to the original model

Variable	Parameters	Original		
		Cl. 1	Cl. 2	Cl. 3
X_1	β_0	0	1	0
	σ	1	2.27	2.27
X_2	β_0	0	0	1
	σ	1.4	2	2
X_3	$\beta_0, \beta_1, \beta_5$	0.04, 1.05, 0.11	-0.65, 0.19, 1.47	0.0, 0.29, 0.96
	σ	1.4	0.75	1.24
X_4	β_0, β_2	-0.01, 0.77	-0.01, 0.11	0.87, 0.1
	σ	1.67	1.38	1.37
X_5	β_0	-0.01	0.99	0.04
	σ	2.3	0.99	1.03
Y_1	μ	0	$\pi/2$	π
	κ	1	1	1
Y_2	μ	0	$\pi/2$	π
	κ	2	2	3

Variable	Parameters	General hybrid model		
		Cl. 1	Cl. 2	Cl. 3
X_1	$\hat{\beta}_0$	0.08	-0.1	-0.26
	$\hat{\sigma}$	1.05	2.41	2.35
X_2	$\hat{\beta}_0$	-0.12	0.56	0.14
	$\hat{\sigma}$	1.34	2.09	2.01
X_3	$\hat{\beta}_0, \hat{\beta}_1$	0.29, 0.90	-0.06, -0.64	0.01, 1.52
	$\hat{\sigma}$	1.56	0.79	1.30
X_4	$\hat{\beta}_0, \hat{\beta}_2$	0.18, 0.77	0.05, 0.02	0.85, 0.00
	$\hat{\sigma}$	1.77	1.36	1.29
X_5	$\hat{\beta}_0$	-0.14	-0.08	-0.06
	$\hat{\sigma}$	2.31	1.07	1.05
Y_1	$\hat{\mu}$	0.11	1.43	2.91
	$\hat{\kappa}$	0.92	0.77	1.53
Y_2	$\hat{\mu}$	-0.19	1.56	3.13
	$\hat{\kappa}$	1.21	2.05	2.71

Next, we look at the performance of the general hybrid model by changing the proportional number of Gaussian and vM variables, as well as the number of clusters. We simulated three different datasets to evaluate the model and compare it with multivariate Gaussian mixture models. The first dataset had an equal number of linear and directional variables and consisted of five Gaussian and five vM variables. The second dataset had more linear variables: 15 Gaussian and

5 vM variables. The third dataset had 5 Gaussians and 15 vM variables. Again we hid the cluster label of the instances for data clustering.

Table 4. Hit rate of general hybrid and Gaussian mixture models. We simulate 100 instances for each cluster. We change the number of variables for the data. First we analyze 5 Gaussian and 5 vM, then 15 Gaussian and 5 vM and finally 5 Gaussian and 15 vM.

N. Cl./N. Var.	General hybrid clustering			Gaussian clustering		
	5-5	15-5	5-15	5-5	15-5	5-15
3	99.6%	100%	100%	99%	99.6%	100%
5	95.4%	100%	100%	89.2%	99.8%	99.6%
10	94.6%	99.8%	100%	81.9%	99.2%	95.5%

According to Table 4 general hybrid model overcomes Gaussian mixture model in all the proposed scenarios. General hybrid model yields better results when there is an equal number of linear and directional variables and a low dimensional space. However, when there are more linear variables than directional variables, the Gaussian mixture models turns competitive and almost tie our model. In the last trial, when the number of directional variables surpass the number of linear variables, the proposed model slightly outperforms the Gaussian mixture model. General hybrid model obtained better BIC score in all the cases.

5.2 Clustering of dendritic spines

Dendritic spines are small membranous protusions. They are receptors of excitatory synapses placed on the surface of some neuronal dendrites [34]. They have captured the attention of neuroscientists because their morphology has been associated with brain functionality. For example, it has been claimed that thin spines contribute to learning, while the biggest and steady spines are linked to the memory process. Disturbances of their morphology or density have been related to mental disorders such as schizophrenia, dementia or mental retardation [21]. Therefore, the clustering of dendritic spines is attracting interest in neuroscience. A traditionally accepted categorization is described in [37], proposing four groups (Figure 5). There is also debate about whether morphologies constitute a continuum instead of discrete classes [19].

We present an application of the general hybrid model to cluster dendritic spines according to their morphology. For the experiment we used a set of 500 triangular meshes representing the surface of three-dimensional dendritic spines reconstructed from pyramidal neurons extracted from the cingular cortex of a human male (aged 40). Spines were provided by the Cajal Cortical Circuits Lab (UPM-CSIC). For details about spine acquisition, see [5].

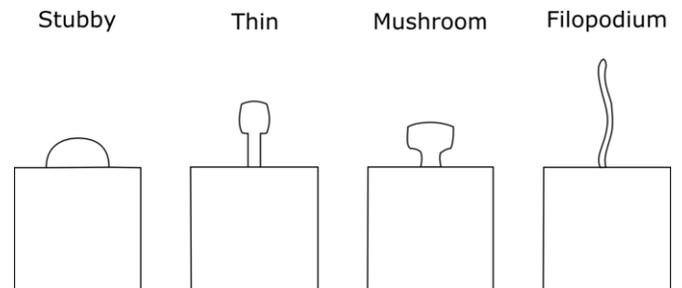


Figure 5. Traditional classification of spines proposed in [37], adapted from [40].

Meshes had to be previously transformed into data characterizing the morphology of the spines. This task was addressed using the multiresolutional Reeb graph (MRG) [20, 41] technique which constructs a graph from a 3D geometrical model to describe its topology. MRG partitions a triangular mesh into regions according to a function $\alpha(\cdot)$. In our case, this function was the geodesic distance because it is invariant to translation and rotation. Geodesic distance was computed from the mean point of the total surface that is in contact with the dendrite to each vertex of the mesh. The domain of $\alpha(\cdot)$ was divided into seven regions. For each region, we measured morphological characteristics, i.e. length, growth direction, eccentricity, flatness and size of the region (see Figure 6). There are a total of 35 linear variables and 30 directional variables.

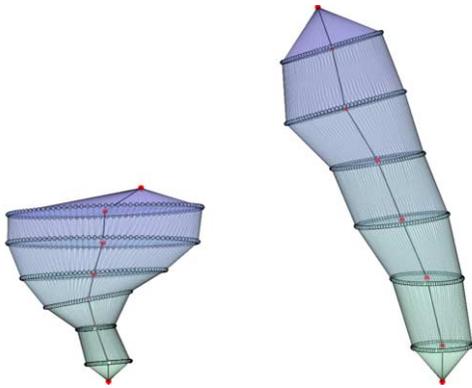


Figure 6. Examples of spines after computing and dividing the MRG it into regions. For each region, we measure morphological characteristics.

Since this experiment serves merely to illustrate an application of the proposed model and does not represent any valid neuroscientific result, we then jittered data with Gaussian and von Mises noise of zero mean. We ran the general hybrid model several times, modifying the number of clusters from two to ten. As a result, we managed to maximize the BIC score and AIC score for three clusters and seven clusters respectively. We analyzed exclusively the results provided by BIC score because its number of clusters is closest to the number of categories in the traditional classification.

To characterize clusters and compare them with the classification in [37], we performed a Welch t-test for linear variables and a Watson-Williams test for directional variables. We observed that almost all linear variables are significantly different between cluster 2 and the other two clusters. However, cluster 1 and cluster 3 only differ in so far as cluster 3 has a small neck at the base of the spine. We checked which cluster takes the maximum and minimum value for each measured feature to characterize the spine.

Thus, cluster 1 presents the shortest and flattest regions. Besides, all the regions are of the same size. This description fits the stubby class. Cluster 2 shows the longest and most elongated regions. Additionally, the size of the regions increases from the base to the top and the growth of the regions is less straight. Hence, this cluster groups filopodium and thin classes. Cluster 3 has short regions and a small base. This cluster grows backwards (in a $\frac{3\pi}{2}$ direction) while the other two clusters grow to the left (in a π direction). It apparently matches the mushroom class.

The Structural EM also provides some interesting information about the dependencies represented by the graph topology. For example, we find that the length of the next region depends on the length

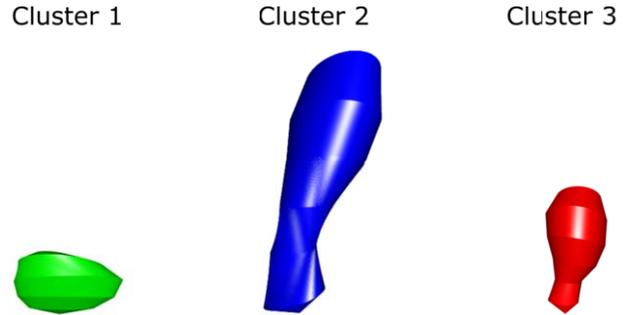


Figure 7. Examples of spines for each of the clusters. The spine representing cluster 1 is shaded green, the spine representing cluster 2 is shaded blue and the spine representing cluster 3 is shaded red.

of the previous regions. Also the size of the regions is related to the size of previous regions. We also observe connections between the eccentricity of the region and its length. These dependencies may be relevant for the electrophysiological behavior of the spine.

6 Conclusion

This paper investigated models for clustering hybrid (linear and directional) data. Although the most common approach for modeling this data is by means of Gaussian mixture models, directional data has some special properties that rule out the use of classical statistics. Assuming that directional data are Gaussian sometimes leads to poor approximations. In this paper, we reviewed previous models for clustering multivariate von Mises and multivariate hybrid data: current methods for clustering multivariate directional data are constrained to concentrated data and involve numerical optimization, whereas we did not find any specific clustering models for multivariate hybrid data.

This is why we proposed clustering models for multivariate directional and hybrid data based on Bayesian networks. To be precise, we exploited the benefits of factorization provided by the naive Bayes structure to get closed-form equations for the expectation-maximization algorithm. Additionally, we also improved the hybrid model by learning the graph structure from the linear variables according to the Structural EM framework.

We evaluated the proposed models against multivariate Gaussian distributions. The results provided by our models are better than the outcome of the Gaussian mixture model in almost all scenarios where directional data is involved. Besides, we applied a hybrid Structural EM algorithm to cluster dendritic spines with the aim of illustrating real applications of the model.

Future research includes the extension of the hybrid model to cover other distributions like von Mises-Fisher, Kent or discrete nodes, as well as relations of dependence between Gaussian and directional variables.

ACKNOWLEDGEMENTS

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness through the Cajal Blue Brain (C080020-09; the Spanish partner of the Blue Brain initiative from EPFL) and TIN2013-41592-P projects, by the Regional Government of Madrid through the S2013/ICE-2845-CASI-CAM-CM project, and by the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 604102 (Human Brain Project).

REFERENCES

- [1] H Akaike, 'A new look at the statistical model identification', *IEEE Transactions on Automatic Control*, **19**(6), 716–723, (1974).
- [2] A Banerjee, IS Dhillon, J Ghosh, and S Sra, 'Clustering on the unit hypersphere using von Mises-Fisher distributions', *Journal of Machine Learning Research*, **6**, 1345–1382, (2005).
- [3] E Batschelet, *Circular Statistics in Biology*, Academic Press London, 1981.
- [4] E Batschelet, D Hillman, M Smolensky, and F Halberg, 'Angular-linear correlation coefficient for rhythmometry and circannually changing human birth rates at different geographic latitudes', *International Journal of Chronobiology*, **1**(55), 183–202, (1973).
- [5] R Benavides-Piccione, I Feraud-Espinosa, V Robles, R Yuste, and J DeFelipe, 'Age-based comparison of human dendritic spine structure using complete three-dimensional reconstructions', *Cerebral Cortex*, **23**(8), 1798–1810, (2012).
- [6] DJ Best and NI Fisher, 'The BIAS of the maximum likelihood estimators of the von Mises-Fisher concentration parameters', *Communications in Statistics-Simulation and Computation*, **10**(5), 493–502, (1981).
- [7] S Calderara, R Cucchiara, and A Prati, 'Detection of abnormal behaviors using a mixture of von Mises distributions', in *IEEE Conference on Advanced Video and Signal Based Surveillance, 2007*, pp. 141–146, (2007).
- [8] S Calderara, A Prati, and R Cucchiara, 'Mixtures of von Mises Distributions for People Trajectory Shape Analysis', *IEEE Transactions on Circuits and Systems for Video Technology*, **21**(4), 457–471, (2011).
- [9] J A Carta, P Ramírez, and C Bueno, 'A joint probability density function of wind speed and direction for wind energy analysis', *Energy Conversion and Management*, **49**(6), 1309–1320, (2008).
- [10] JA Carta, P Ramirez, and S Velazquez, 'A review of wind speed probability distributions used in wind energy analysis: Case studies in the Canary Islands', *Renewable and Sustainable Energy Reviews*, **13**(5), 933–955, (2009).
- [11] GF Cooper and E Herskovits, 'A Bayesian method for the induction of probabilistic networks from data', *Machine Learning*, **9**(4), 309–347, (1992).
- [12] AP Dempster, NM Laird, and DB Rubin, 'Maximum likelihood from incomplete data via the EM algorithm', *Journal of the Royal Statistical Society. Series B*, **1**, 1–38, (1977).
- [13] RO Duda, PE Hart, and DG Stork, *Pattern Classification*, John Wiley & Sons, 2012.
- [14] C Fraley and AE Raftery, 'Model-based clustering, discriminant analysis, and density estimation', *Journal of the American Statistical Association*, **97**(458), 611–631, (2002).
- [15] N Friedman, 'Learning belief networks in the presence of missing values and hidden variables', in *ICML*, volume 97, pp. 125–133, (1997).
- [16] R Gatto and SR Jammalamadaka, 'The generalized von Mises distribution', *Statistical Methodology*, **4**(3), 341–353, (2007).
- [17] Kurt H and Bettina G, 'movMF: An R package for fitting mixtures of von Mises-Fisher distributions', *Journal of Statistical Software*, **58**(1), 1–31, (2014).
- [18] D Heckerman, D Geiger, and DM Chickering, 'Learning Bayesian networks: The combination of knowledge and statistical data', *Machine Learning*, **20**, 197–243, (1995).
- [19] A Herrera-Arellano, J Miranda-Sánchez, P Ávila-Castro, S Herrera-Álvarez, JE Jiménez-Ferrer, A Zamilpa, R Román-Ramos, H Ponce-Monter, and J Tortoriello, 'Clinical effects produced by a standardized herbal medicinal product of Hibiscus sabdariffa on patients with hypertension. A randomized, double-blind, lisinopril-controlled clinical trial', *Planta Medica*, **73**(1), 6–12, (2007).
- [20] M Hilaga, Y Shinagawa, T Kohmura, and TL Kunii, 'Topology matching for fully automatic similarity estimation of 3D shapes', in *Proceedings of SIGGRAPH*, pp. 203–212. ACM, (2001).
- [21] B Jacobs, L Driscoll, and M Schall, 'Life-span dendritic and spine changes in areas 10 and 18 of human cortex: A quantitative Golgi study', *Journal of Comparative Neurology*, **386**(4), 661–680, (1997).
- [22] D Koller and N Friedman, *Probabilistic Graphical Models: Principles and Techniques*, MIT Press, 2009.
- [23] PL López-Cruz, C Bielza, and P Larrañaga, 'Directional naive Bayes classifiers', *Pattern Analysis and Applications*, **18**(2), 225–246, (2013).
- [24] JJ Love and JN Thomas, 'Insignificant solar-terrestrial triggering of earthquakes', *Geophysical Research Letters*, **40**, 1165–1170, (2013).
- [25] KV Mardia, G Hughes, CC Taylor, and H Singh, 'A multivariate von Mises distribution with applications to bioinformatics', *The Canadian Journal of Statistics*, **36**(1), 99–109, (2008).
- [26] KV Mardia and PE Jupp, *Directional Statistics*, John Wiley & Sons, 1999.
- [27] KV Mardia, JT Kent, Z Zhang, CC Taylor, and T Hamelryck, 'Mixtures of concentrated multivariate sine distributions with applications to bioinformatics', *Journal of Applied Statistics*, **39**(11), 2475–2492, (2012).
- [28] KV Mardia and TW Sutton, 'A model for cylindrical variables with applications', *Journal of the Royal Statistical Society. Series B*, **40**(2), 229–233, (1978).
- [29] KV Mardia, CC Taylor, and GK Subramaniam, 'Protein bioinformatics and mixtures of bivariate von Mises distributions for angular data', *Biometrics*, **63**, 505–512, (2007).
- [30] N Masseran, AM Razali, K Ibrahim, and MT Latif, 'Fitting a mixture of von Mises distributions in order to model data on wind direction in Peninsular Malaysia', *Energy Conversion and Management*, **72**, 94–102, (2013).
- [31] G McLachlan and KE Basford, *Mixture Models: Inference and Applications to Clustering*, Wiley, 1988.
- [32] V Melnykov and R Maitra, 'Finite mixture models and model-based clustering', *Statistics Surveys*, **4**, 80–116, (2010).
- [33] JA Mooney, PJ Helms, and IT Jolliffe, 'Fitting mixtures of von Mises distributions: A case study involving sudden infant death syndrome', *Computational Statistics and Data Analysis*, **41**(34), 505–513, (2003).
- [34] EA Nimchinsky, BL Sabatini, and K Svoboda, 'Structure and function of dendritic spines', *Annual Review of Physiology*, **64**, 313–353, (2002).
- [35] J Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers Inc., 1988.
- [36] JM Peña, JA Lozano, and P Larrañaga, 'An improved Bayesian structural EM algorithm for learning Bayesian networks for clustering', *Pattern Recognition Letters*, **21**(8), 779–786, (2000).
- [37] A Peters and IR Kaiserman-Abramof, 'The small pyramidal neuron of the rat cerebral cortex. The perikaryon, dendrites and spines', *American Journal of Anatomy*, **127**(4), 321–355, (1970).
- [38] X Qin, SJ Zhang, and DX Yan, 'A new circular distribution and its application to wind data', *Journal of Mathematics Research*, **2**(3), 12–17, (2010).
- [39] G Schwarz, 'Estimating the dimension of a model', *The Annals of Statistics*, **6**, 461–464, (1978).
- [40] N Spruston, 'Pyramidal neurons: Dendritic structure and synaptic integration', *Nature Reviews Neuroscience*, **9**(3), 206–221, (2008).
- [41] JWH Tangelder and RC Veltkamp, 'A survey of content based 3D shape retrieval methods', *Multimedia Tools and Applications*, **39**(3), 441–471, (2008).
- [42] JN Thomas, JJ Love, and MJS Johnston, 'On the reported magnetic precursor of the 1989 Loma Prieta earthquake', *Physics of the Earth and Planetary Interiors*, **173**(3), 207–215, (2009).
- [43] JN Thomas, JJ Love, MJS Johnston, and K Yumoto, 'On the reported magnetic precursor of the 1993 Guam earthquake', *Geophysical Research Letters*, **36**, (2009).
- [44] JN Thomas, JJ Love, A Komjathy, OP Verkhoglyadova, M Butala, and N Rivera, 'On the reported ionospheric precursor of the 1999 Hector Mine, California earthquake', *Geophysical Research Letters*, **39**, (2012).
- [45] DM Titterton, AFM Smith, and UE Makoy, *Statistical Analysis of Finite Mixture Distributions*, Wiley, 1985.
- [46] I Tsamardinos, LE Brown, and CF Aliferis, 'The max-min hill-climbing Bayesian network structure learning algorithm', *Machine Learning*, **65**(1), 31–78, (2006).
- [47] S Zhong and J Ghosh, 'A comparative study of generative models for document clustering', in *Proceedings of the Workshop on Clustering High Dimensional Data and Its Applications in SIAM Data Mining Conference*, (2003).

Adaptive Symbiotic Collaboration for Targeted Complex Manipulation Tasks

Rui Silva¹ and Francisco S. Melo² and Manuela Veloso³

Abstract.

This paper addresses the problem of human-robot collaboration in the context of manipulation tasks. In particular, we focus on tasks where a robot must perform some complex manipulation that is successfully completed only upon reaching some target pose provided by a human user. We propose an approach in which the robot explicitly reasons about its ability to complete the task and proactively requests the assistance of the human teammate when necessary. Our approach effectively trades-off the benefits arising from the human assistance with the cost of disturbing the user. We also propose an adaptation mechanism that enables the robot to adjust its behavior to the particular manner by which the human user responds to the requests made by the robot. We test our approach in a simple illustrative scenario and in two real interaction scenarios involving the Baxter robot.

1 Introduction

In this paper we address the general problem of human-robot collaboration, where a human user and a robot work together towards the successful completion of some predefined task. We are particularly interested in tasks where the robot is required to perform a complex motion involving the manipulation of an object which will act as the “interface” supporting the interaction with the human user. Examples of the tasks we envision include assisting the human user in dressing a piece of clothing [5, 8] or jointly preparing a drink [4].

Our contribution is aimed towards three distinguishing aspects of the class of scenarios described:

- The task is usually hard to model, which poses difficulties in the application of standard motion planners. To deal with this challenge, we adopt a motion representation that is particularly suited for *learning from demonstration* [1]. Explicitly teaching the robot the desired motion circumvents the need for explicit planning. At the same time, it provides a natural and intuitive interface for human users to program the robot to execute new tasks [18].
- Although the particular motion that the robot must perform to complete the desired task depends on the human user (for example, the dressing motion depends on the pose of the human user; pouring a drink into a cup depends on where the user places the cup), the *general shape of the trajectory* is approximately the same. Our approach relies on the notion of *motion primitives* [7],

which represent idealized motions to perform a given task. Motion primitives provide a flexible way of representing the desired trajectory while, at the same time, modulating the trajectory to adapt to changing task parameters (e.g., depending on the human user).

- Finally, when interacting with the robot, the human user seldom takes into consideration the motion limitations that the robot has. For example, in the dressing assistance task, the user may assume a pose that blocks the robot’s motion; or, in the drink pouring scenario, the human user may place the cup outside of the robot’s workspace. We follow recent work on *symbiotic autonomy* [16], where the robot acknowledges and explicitly reasons about its own limitations, and relies on the human assistance to overcome such limitations. In this process, the robot takes into consideration the burden imposed upon the human user, trading it off with the benefits obtained from the human intervention.

Summarizing, our contributions are threefold. First, we contribute a novel decision-theoretic framework enabling a robot to reason explicitly about its own limitations during task execution. Second, we exploit our framework in the context of *symbiotic autonomy* [16], allowing the the robot to consider the potential benefit of enrolling human assistance during task execution against the associated cost. Third, we combine our framework with an online adaptive mechanism that uses the outcome of the successive interactions between the robot and the user to refine the model of the user’s behavior and improve the quality of the interaction.

1.1 Related work

The topic of cooperative object manipulation has been widely studied in the literature, where the standard example involves joint lifting a heavy object [20]. In such collaborative tasks, the human user typically plays the role of the “leader”, guiding the execution of the task, and the robot adapts its execution to that of the leader. In such interaction paradigm, the robot must often *predict* the motion of the human and act accordingly, and several approaches have been proposed in which collaborative manipulation relies on the prediction of human motion [2, 10–12].

Recent work has, to some extent, shifted to the robot part of the responsibility and initiative in completing the task. In this line of work, the motion executed by the robot is planned so that it can easily be interpreted/predicted by the human user [4, 6, 13], thus facilitating the (implicit) coordination of the two elements. For example, Dragan et al. [4] investigate how the shape of the trajectory by the robot impacts task efficiency: trajectories that are more *legible* allow the human user to more effectively interact with the robot than trajectories that are more *functional*. A related idea is pursued in [22], where, in this case, the best motion is learned.

¹ INESC-ID and Instituto Superior Técnico, University of Lisbon and Carnegie Mellon University, e-mail: rui.teixeira.silva@tecnico.ulisboa.pt

² INESC-ID and Instituto Superior Técnico, University of Lisbon, e-mail: fmelo@inesc-id.pt

³ Carnegie Mellon University, e-mail: veloso@cs.cmu.edu

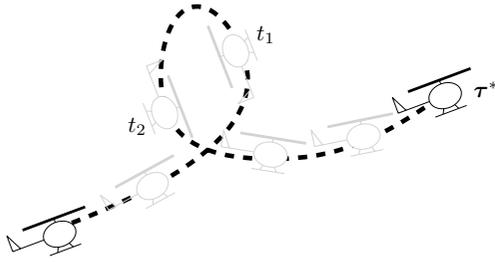


Figure 1. Illustration of a possible trajectory to perform a back flip. At some instant t_1 , the tail of the helicopter will stand below the cabin; afterwards, at some time instant $t_2 > t_1$, the tail will pass over the cabin.

Another line of work closely related with our own work investigates *adjustable autonomy* in human-robot and human-agent teams [3, 17, 19]. For example, Scerri et al. [17] introduce the notion of *transfer-of-control*, where agents interacting with humans adopt a decision-theoretic framework to reason about when to handle the task control to the human users. Sellner et al. [19] investigate the impact of sliding autonomy in the performance of a multi-robot team.

Also relevant for our work is recent research in *impromptu teams* or *ad hoc teams* [14, 21]. Ad hoc teamwork seeks to develop strategies that enable an agent to rapidly infer the strategy of its teammates and act accordingly, towards the joint completion of some common task. Although most work in this area addresses this problem from a high-level, decision-theoretic perspective, the challenges faced in this topic of research bare a close resemblance to those found in collaborative human-robot manipulation: the “ad hoc agent” must infer the goal of the teammate and adapt its actions accordingly.

In this paper we propose an approach in which the robot takes upon itself the initiative of solving the task, explicitly requesting the assistance of the human teammate when necessary. This approach, in which the robot plays an active role in the interaction, can be seen as an instantiation of the concept of *symbiotic autonomy* [16], recently introduced and explored in the context of robot navigation. We propose an approach in which the robot is able to autonomously trade-off the cost associated with “disturbing” the human user with the potential benefit arising from the human intervention. In our approach we use *probabilistic motion primitives*, recently introduced by Paraschos et al. [15] as a flexible representation for robot motions and explored in the context of interaction in [9].

2 Probabilistic Motion Primitives

In our approach we use *probabilistic motion primitives* to represent the motion that the robot should perform during task execution.

Probabilistic motion primitives (ProMPs) were introduced in [15] as a way to represent in a flexible way the motion necessary to complete some well-defined task. Consider, for example, a helicopter for which we want to represent the motion associated with performing a back flip. There are multiple ways by which a back flip can be performed, all of which, however, share several distinctive features. For example, independently of how the back flip is performed, at some point in time the tail of the helicopter will stand below the cabin, after which the tail will move above the cabin (see Fig. 1).

A ProMP can be seen as a representation of an “idealized trajectory” τ^* for the intended motion, where the different ways by which the task can be addressed are seen as *perturbations* of this idealized trajectory. Therefore, a ProMP takes the form of a probability dis-

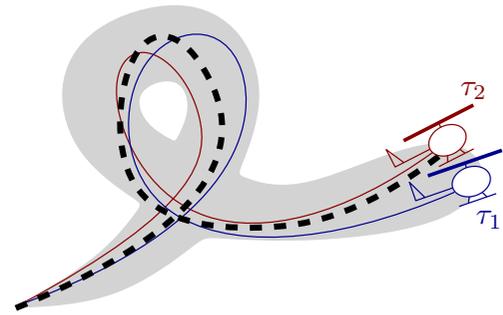


Figure 2. A ProMP that represents a helicopter back flip may assign positive probability to trajectories fulfilling the desired task (corresponding to the shaded area). Trajectories closer to the desired trajectory, herein represented as the dashed line, are assigned larger probability, which in this case could mean that $p_{\text{flip}}(\tau_2) > p_{\text{flip}}(\tau_1)$.

tribution over the space of trajectories. Trajectories “closer” to the idealized trajectory are assigned a larger probability, while trajectories that are further away are assigned smaller probability. The probability of a given trajectory describes how likely it is for an agent to perform such trajectory when performing the desired task.

Returning to our previous helicopter example, a ProMP to represent a back flip could assign a positive probability to trajectories in which, at some point in time, the helicopter tail moves from below the cabin to above the cabin, assigning larger probability to those closer to the desired trajectory. Figure 2 illustrates this idea: the shaded area visually represents the “space of trajectories” that are assigned positive probability. Moreover, in this illustration the trajectory τ_2 is close to the desired trajectory (the dashed line), which would imply that the ProMP would assign larger probability to τ_2 than to τ_1 .

We represent a trajectory as a sequence

$$\tau = \{\mathbf{y}(0), \mathbf{y}(1), \dots, \mathbf{y}(T)\},$$

where T denotes the trajectory length and $\mathbf{y}(t)$ is the pose of the robot at time-step t . We write \mathcal{T} to denote the space of trajectories and refer to a ProMP M as some distribution p_M over \mathcal{T} , where $p_M(\tau)$ denotes the probability of trajectory $\tau \in \mathcal{T}$ in the context of the desired task, as seen above.

For representational purposes, and following [15], we assume that the “idealized” trajectory associated with a ProMP can be constructed as the linear combination of a set of well-defined *trajectory features*, ϕ_k , $k = 1, \dots, K$; the trajectories pertaining to the ProMP thus take the general form

$$\mathbf{y}(t) = \sum_{k=1}^K \phi_k(t)w_k + \varepsilon(t) = \Phi^\top(t)\mathbf{w} + \varepsilon(t), \quad (1)$$

where $\{\varepsilon(t), t = 1, \dots, T\}$ is a noise sequence and $\mathbf{w} \in \mathbb{R}^K$ is a vector of parameters, w_k representing the weight of feature ϕ_k in the trajectory. This simplifying assumption establishes a correspondence between the space of trajectories \mathcal{T} and the space of parameters, and allows the probability distribution over trajectories that represents the ProMP to be expressed as a distribution over parameters, which is easier to represent and manipulate. Therefore, one can easily express operations involving the motion primitive in terms of familiar concepts and operations from probability theory. For example,

- Prior knowledge regarding the task or preference over the trajectories that best accomplish it can be expressed in the form of a prior

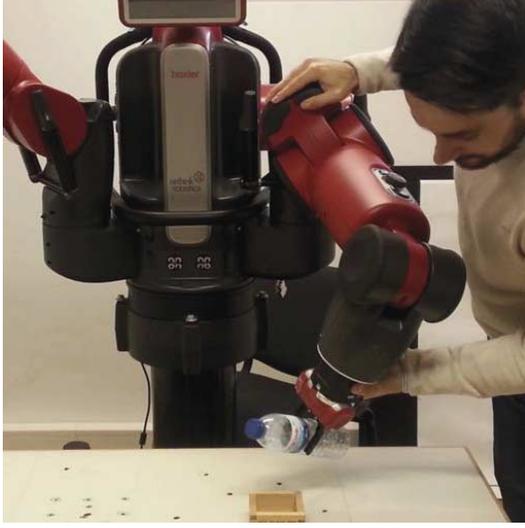


Figure 3. Kinesthetic teaching. In the image, the human teacher shows the robot the necessary motion to pour liquid into a cup.

distribution p_M over \mathbb{R}^K .

- Sample demonstrations of the desired trajectory provided by an expert can be integrated into the ProMP by a standard Bayesian update. For example, letting $\mathcal{D} = \{\tau_1, \dots, \tau_N\}$ denote a dataset containing several (independent) trajectories demonstrating the desired motion, we can *learn* from such demonstrations simply by updating the ProMP distribution to the posterior

$$\begin{aligned} p_M(\mathbf{w} | \mathcal{D}) &\propto p(\mathcal{D} | \mathbf{w})p_M(\mathbf{w}) \\ &= \prod_{n=1}^N p(\tau_n | \mathbf{w})p_M(\mathbf{w}), \end{aligned} \quad (2)$$

where $p(\tau_n | \mathbf{w})$ represents the likelihood of observing a trajectory τ_n when the idealized trajectory is represented by the parameter vector \mathbf{w} .

- Similarly, modulating the trajectory to reach a certain target pose \mathbf{y}^* at time-step t can easily be achieved by computing

$$p_{\text{reach}}(\mathbf{w} | \mathbf{y}(t) = \mathbf{y}^*) \propto p(\mathbf{y}(t) = \mathbf{y}^* | \mathbf{w})p_M(\mathbf{w}) \quad (3)$$

where, once again, $p(\mathbf{y}(t) = \mathbf{y}^* | \mathbf{w})$ represents the likelihood of attaining pose \mathbf{y}^* at time-step t when the idealized trajectory is represented by the parameter vector \mathbf{w} .

In the next section we describe the class of problems addressed in the paper, where a robot must perform a complex motion towards a target provided by a human user. We contribute a novel approach that enables the robot to reason about asking the human user for assistance, explicitly weighting the potential improvement in (task) performance that may result from such assistance against the cost of disturbing the human user. Our approach is designed to leverage the representational power of ProMPs while, at the same time, enable the robot to adjust to the particular user by learning how the latter responds to the requests made by the robot.

3 Collaborative Manipulation

We focus on tasks where a robot must perform some potentially complex manipulation that is successfully completed only upon reaching

some target pose provided by a human user. Examples of such scenarios include placing an object in a container held by the user, or assisting a human to put on some piece of clothing. In both examples, the robot must perform a complex motion that culminates with reaching a target pose—the position of the container or a part of the human body.

In all our examples we start by building a ProMP M from a set of trajectories acquired by kinesthetic demonstration: a human expert guides the robot along the whole motion from the initial position to the target position (see Fig. 3). We adopt a fully Bayesian approach, where the demonstrated trajectories are used to compute a posterior distribution p_M over parameters, as suggested by the ProMP manipulations discussed in Section 2.

Following [15] we use a parametric representation for the distribution p_M , and denote by θ the parameters of the distribution. We can rewrite the ProMP update equation for a single trajectory τ to bring forth the role of the distribution parameters to yield

$$p_M(\mathbf{w} | \tau, \theta) = \frac{1}{\eta} p(\tau | \mathbf{w})p_M(\mathbf{w} | \theta),$$

where η is a normalizing constant. The likelihood $p(\tau | \mathbf{w})$ depends directly on the noise sequence $\{\varepsilon(t), t = 1, \dots, T\}$ in (1). For simplicity, we take $\{\varepsilon(t), t = 1, \dots, T\}$ to be white Gaussian noise with zero mean and known covariance Σ_ε . Therefore, and since the sequence $\{\varepsilon(t), t = 1, \dots, T\}$ is uncorrelated in virtue of our whiteness assumption, the likelihood $p(\tau | \mathbf{w})$ can be written as

$$\begin{aligned} p(\tau | \mathbf{w}) &= \prod_{t=1}^T p(\mathbf{y}(t) | \mathbf{w}) \\ &= \prod_{t=1}^T \text{Normal}(\mathbf{y}(t) | \Phi^\top(t)\mathbf{w}, \Sigma_\varepsilon), \end{aligned}$$

and taking a Gaussian prior over \mathcal{W} , we have, for each trajectory τ ,

$$\begin{aligned} p_M(\mathbf{w} | \tau, \theta) &= \frac{1}{\eta} \prod_{t=1}^T \text{Normal}(\mathbf{y}(t) | \Phi^\top(t)\mathbf{w}, \Sigma_\varepsilon) \text{Normal}(\mathbf{w} | \mu_w, \Sigma_w), \end{aligned}$$

where $\theta = \{\mu_w, \Sigma_w\}$.

3.1 Task execution

Given a ProMP M , learned from a set of demonstrated trajectories, and a target pose \mathbf{y}^* , we can now modulate the trajectories of the ProMP M using (3) which, in light of our Gaussian assumption, translates into the standard updates for μ_w and Σ_w ,

$$\begin{aligned} \mu_w^{\text{new}} &= \mu_w + \mathbf{K}(\mathbf{y}^* - \Phi^\top(T)\mu_w) \\ \Sigma_w^{\text{new}} &= (\mathbf{I} - \mathbf{K}\Phi^\top(T))\Sigma_w, \end{aligned}$$

where the matrix \mathbf{K} is given by

$$\mathbf{K} = \Sigma_w \Phi(T) \mathbf{S}^{-1}$$

and

$$\mathbf{S} = \Phi^\top(T)\Sigma_w\Phi(T) + \Sigma_\varepsilon.$$

The updated Gaussian distribution, with parameters $\theta^{\text{new}} = \{\mu_w^{\text{new}}, \Sigma_w^{\text{new}}\}$, can be used to obtain a trajectory that leads the robot to the new target position.

However, for targets \mathbf{y}^* too distant from those observed in the demonstrations, the modulation may yield trajectories quite different from the desired one that may actually extend beyond the workspace of the robot. We introduce a *cost function*, $c_E : \mathcal{T} \rightarrow \mathbb{R}$, providing the robot with a quantitative measure of the quality of any trajectory and a way to explicitly reason about its ability to perform it.

We write $c_E(\tau)$ to denote the *execution cost* associated with a trajectory $\tau \in \mathcal{T}$ in light of the target task. The cost function c_E can account for task success, safe execution, robustness, or any other performance criteria that the task designer specifies. Then, given a target pose \mathbf{y}^* and a ProMP M , we can compute the *expected execution cost* of M given \mathbf{y}^* as

$$C_E(\mathbf{y}^*) = \mathbb{E}_\tau [c_E(\tau)] \triangleq \int c_E(\tau) p(\tau | \mathbf{y}^*, \boldsymbol{\theta}) d\tau, \quad (4)$$

where, as before, $\boldsymbol{\theta}$ represents the parameters of the ProMP distribution p_M . For the Gaussian setup considered, we can further break down the computation in (4). Abusing somewhat our notation, let $c_E(\mathbf{w})$ denote the (expected) execution cost associated with the parameter vector \mathbf{w} , i.e.,

$$\begin{aligned} c_E(\mathbf{w}) &= \int c_E(\tau) p(\tau | \mathbf{w}) d\tau \\ &= \int c_E(\Phi^\top \mathbf{w} + \boldsymbol{\varepsilon}) p(\boldsymbol{\varepsilon}) d\boldsymbol{\varepsilon} \end{aligned}$$

where we wrote $\boldsymbol{\varepsilon}$ to denote the noise sequence. Then

$$\begin{aligned} C_E(\mathbf{y}^*) &= \int c_E(\mathbf{w}) p(\mathbf{w} | \mathbf{y}^*, \boldsymbol{\theta}) d\mathbf{w} \\ &= \int c_E(\mathbf{w}) \text{Normal}(\mathbf{w} | \boldsymbol{\mu}_w^{\text{new}}, \boldsymbol{\Sigma}_w^{\text{new}}) d\mathbf{w}. \end{aligned}$$

3.2 Active collaborative manipulation

A natural possibility to address the difficulties posed by targets placed in inconvenient locations (in the sense discussed above) is to acknowledge such difficulties and request the human user for assistance in moving the target to a more convenient location. Suppose then that the robot has available a finite set of instructions, $\mathcal{A} = \{a_0, a_1, \dots, a_J\}$, that it can voice to the human user. Instructions a_1, \dots, a_J correspond to requests to the human user to move the target in a specific manner, while a_0 corresponds to the *null instruction* (request nothing from the human user). For example, in the scenario of Fig. 3 where the robot must pour a liquid in a cup, a possible request could be to place the cup closer to the robot.

We associate with each instruction $a \in \mathcal{A}$ a *request cost*, $c_R(a)$, that quantifies the burden imposed upon the user in executing the action associated with instruction a , and should be designed relatively to $c_E(\tau)$. Thus, assuming a $c_E(\tau)$ in $[0, 1]$, $c_R(a) = 0.1$ indicates that action a adds an additional 10% burden. Additionally, associated with each instruction $a \in \mathcal{A}$ and each target position \mathbf{y} , we consider a *transition model* that describes (probabilistically) the outcome of the user's action in terms of the target position. In particular, we write $p_{\text{target}}(\mathbf{y}' | \mathbf{y}, a)$ to denote the probability that, after requesting a , the target moves from \mathbf{y} to \mathbf{y}' . For action a_0 , $p_{\text{target}}(\mathbf{y}' | \mathbf{y}, a_0) = \delta(\mathbf{y}', \mathbf{y})$.

With the elements above, the total cost incurred by the robot upon performing request a at the target position \mathbf{y} and then moving to

reach the target in the resulting position is given by

$$\begin{aligned} C_M(a) &= c_R(a) + \mathbb{E}_{\mathbf{y}'} [C_E(\mathbf{y}') | \mathbf{y}, a] \\ &= c_R(a) + \int_{\mathbf{y}'} C_E(\mathbf{y}') p_{\text{target}}(\mathbf{y}' | \mathbf{y}, a) d\mathbf{y}'. \end{aligned}$$

Assuming that the instruction set \mathcal{A} always includes the null instruction a_0 (with a cost of $c_R(a_0) = 0$), the robot will select action

$$a^* = \arg \min_{a \in \mathcal{A}} C_M(a),$$

thus pro-actively requesting the user's assistance in the reaching task whenever the benefit from the user's aid effectively surpasses the burden imposed upon that user. The interplay between the two components of the cost C_M , namely c_E and c_R , determines in which circumstances it pays off to rely on human assistance: when c_R is small it is generally better to request, while for large c_R it is generally better to execute without human assistance.

3.3 An illustrative example

We now illustrate the framework for active collaborative manipulation introduced above in a simple 2D example. We used kinesthetic teaching to collect a number of projected trajectories, depicted in Fig. 4,⁴ and computed the associated ProMP parameters $\boldsymbol{\mu}_w$ and $\boldsymbol{\Sigma}_w$.

In this example scenario, we consider a simple execution cost that penalizes deviations from the known target area—the shaded region in Fig. 4(a). In fact, targets away from those reached in the demonstrations lead to trajectories with shapes significantly different from those demonstrated, as seen in Fig. 4(b). Therefore, in this example scenario, given a trajectory $\tau = \{\mathbf{y}(0), \dots, \mathbf{y}(T)\}$ we use

$$\begin{aligned} c_E(\tau) &= K_E (1 - p_M(\mathbf{y}(T) | \boldsymbol{\theta})) \\ &= K_E \left(1 - \int p(\mathbf{y}(T) | \mathbf{w}) p_M(\mathbf{w} | \boldsymbol{\theta}) d\mathbf{w} \right), \end{aligned}$$

where

$$\begin{aligned} p(\mathbf{y}(T) | \mathbf{w}) &= \text{Normal}(\mathbf{y}(T) - \Phi^\top(T)\mathbf{w} | 0, \boldsymbol{\Sigma}_\varepsilon), \\ p_M(\mathbf{w} | \boldsymbol{\theta}) &= \text{Normal}(\mathbf{w} | \boldsymbol{\mu}_w, \boldsymbol{\Sigma}_w), \end{aligned}$$

and K_E is a constant.

We consider an instruction set $\mathcal{A} = \{\emptyset, U, D, L, R\}$, corresponding to the null request (\emptyset) and requests to move the target up (U), down (D), left (L) and right (R), and a constant request cost $c_R(a) = K_R$, $a \in \mathcal{A} \setminus \{\emptyset\}$. Figure 5 compares, for different values of K_E and K_R the average performance obtained by (i) always trying to execute the trajectory most likely to lead to the target, without ever requesting for assistance (**Always execute**); (ii) always requesting assistance before executing (**Always request**); and (iii) using our approach.⁵

Several aspects are worth noting in Fig. 5. First of all, as expected, the total cost grows linearly with both K_E and K_R , as seen in the curves corresponding to the approach that always executes and the approach that always requests, respectively.

A second point worth noting is that, also as seen in Section 3.2, for small K_R , it is generally better to request, while for large K_R it is always better to execute without human assistance.

⁴ The trajectories were collected using the Baxter robot and the positions of the end effector were then projected in a 2D plane.

⁵ The results in Fig. 5 were obtained by sampling 1,000 targets uniformly at random from the robot's workspace, and averaging the performance of the different approaches in these 1,000 points. Human intervention after an instruction by the robot leads to an average displacement of the target of ~ 20 cm in the requested direction. Finally, we considered that there was no execution noise.

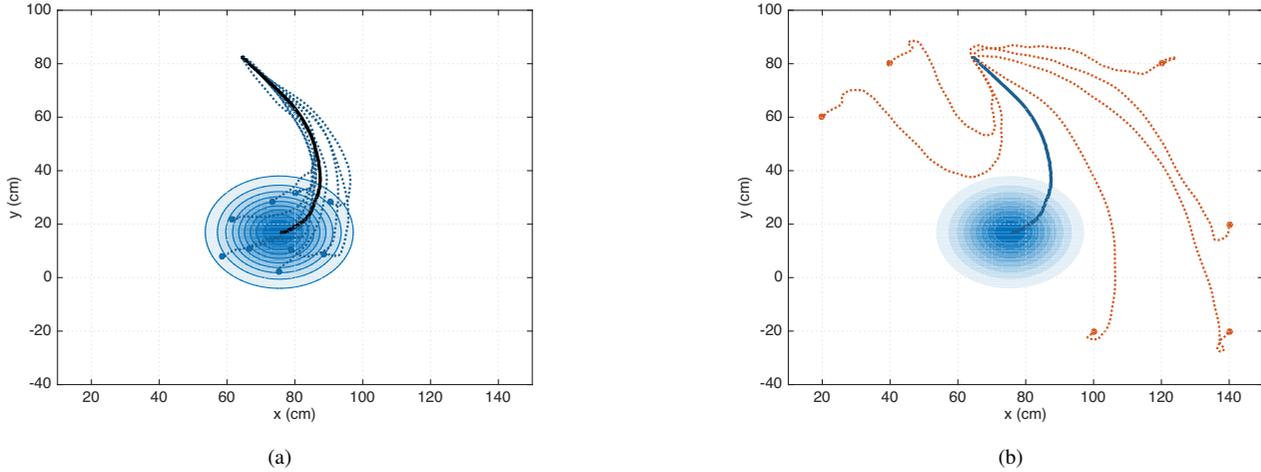


Figure 4. (a) 2D trajectories collected through kinesthetic teaching (dotted lines). The solid line represents the average trajectory, while the contours represent the distribution of targets estimated from the demonstrated trajectories. (b) Trajectories obtained by modulating the ProMP to targets away from the “known area”, represented by the contour. The solid line represents the “mean trajectory”.

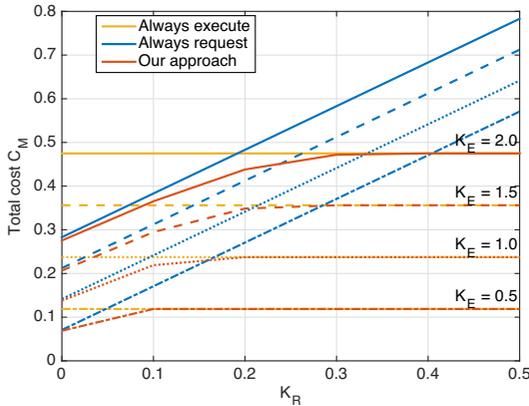


Figure 5. Performance of three approaches to the manipulation task that rely on human assistance at different levels. See text for details.

Finally, we note that our approach naturally outperforms the other two naïve approaches, striking the correct trade-off between the costs and benefits of requesting human assistance, the difference being largest in those situations where the latter two perform similarly. This observation can be explained by noting that, except in a “comfort region” around the “average target”—in which any human action moves the target away—human intervention may always be selected so as to bring the target closer to the comfort region. When the two naïve approaches perform similarly means that the larger execution cost incurred by the approach that always executes in areas far from the comfort region compensates the request cost incurred by the approach that always requests around the comfort zone. Our approach, nevertheless, takes the best of the two and is thus able to attain significant improvements with respect to the other two approaches.

4 Adapting to the human user

According our approach the robot selects the action

$$a^* = \arg \min_{a \in \mathcal{A}} C_M(a),$$

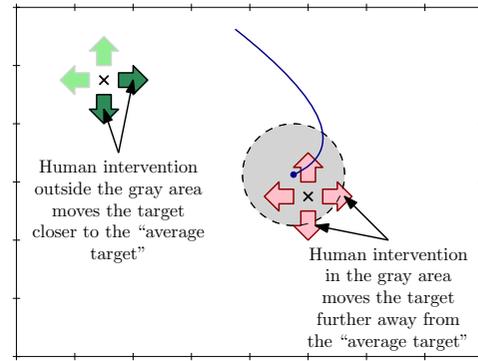


Figure 6. Diagram illustrating the impact of human intervention in different regions of the robot’s workspace.

where the cost C_M is given by

$$C_M(a) = c_R(a) + \mathbb{E}_y [C_E(\mathbf{y}') \mid \mathbf{y}, a].$$

As pointed out before, the cost C_M expresses the trade-off between the costs and benefits of recruiting human assistance in the execution of the desired task. However, computing C_M requires knowledge of the transition probabilities $p_{\text{target}}(\mathbf{y}' \mid \mathbf{y}, a)$, which essentially corresponds to a model of how the human user moves the target in response to the robot’s request. Such model can be learned from data obtained from people moving the target. However, there is still the possibility that, for one particular user, the model is incorrect—for example if the user has some physical limitation, it may not be able to perform some of the actions requested by the robot.

To illustrate the potential impact of a wrong model in the performance of the robot, we return to the example from Section 3.3 and determine what happens when the transition model $p_{\text{target}}(\mathbf{y}' \mid \mathbf{y}, a)$ is incorrect. In particular, let us suppose that a particular user cannot perform vertical motions. We set $K_R = 0.17$ and $K_E = 2$ and draw 50,000 target poses uniformly at random in the robot’s workspace.

The results are summarized in Table 1.

Table 1. Performance of different approaches with correct and incorrect models.

Approach	Avg. cost
Always execute	0.4764
Always request (incorrect model)	0.5184
Always request (correct model)	0.5023
Our approach (incorrect model)	0.4748
Our approach (correct model)	0.4405

The results in Table 1 show that, in the example, the approach that never requests assistance from the human user is generally superior to the approach that always requests. It is also worth noting that the performance of the approach that always requests does not significantly change by using an incorrect model. Both approaches always request human assistance, and the only difference lies on the selection of the instructions to be voiced. The difference in performance is explained mostly by those situations in which the best action is a request to move the target horizontally but the approach with the invalid model incorrectly requests the user to move the target vertically, and vice-versa.

The results also show that even with an incorrect model our approach is still able to outperform (even if by just a small margin) the naïve approaches. Finally, the results confirm that an incorrect model indeed impacts the performance of the robot—when using the correct model, our approach presents a significant advantage over all other approaches.

To address the possibility of an incorrect transition model, we introduce an adaptation mechanism that, after each action requested from the human user, updates the model to more faithfully translate the most recent evidence. In particular, we adopt a parameterized representation for the transition probabilities p_{target} and write $p_{\text{target}}(\mathbf{y}' | \mathbf{y}, \boldsymbol{\alpha})$ to denote the transition probabilities corresponding to the parameter $\boldsymbol{\alpha}$. We associate a parameter vector $\boldsymbol{\alpha}_a$ with each action $a \in \mathcal{A}$, implying that

$$p_{\text{target}}(\mathbf{y}' | \mathbf{y}, a) \equiv p_{\text{target}}(\mathbf{y}' | \mathbf{y}, \boldsymbol{\alpha}_a).$$

Then, by updating the parameters $\boldsymbol{\alpha}_a$ associated with each action $a \in \mathcal{A}$ the robot is able to incrementally adapt to the specific human user that it is currently interacting with.

We conclude by revisiting the example of Section 3.3 to illustrate the impact of adaptation in our approach.

4.1 Illustrative example revisited

We adopt a simple parametric model for the transition probabilities p_{target} . Namely, when the robot voices an instruction $a \in \mathcal{A} \setminus \{a_0\}$,

$$\mathbf{y}' = \mathbf{y} + \boldsymbol{\alpha}_a + \nu,$$

where $\boldsymbol{\alpha}_a$ is a displacement vector associated with action a and ν is a zero-mean Gaussian disturbance. Equivalently,

$$p_{\text{target}}(\mathbf{y}' | \mathbf{y}, \boldsymbol{\alpha}_a) = \text{Normal}(\mathbf{y}' - (\mathbf{y} + \boldsymbol{\alpha}_a), \boldsymbol{\Sigma}_\nu),$$

where $\boldsymbol{\Sigma}_\nu$ is the variance of the disturbance. The update of each parameter vector $\boldsymbol{\alpha}_a$ can be done more or less aggressively, depending on how much weight we want to assign to the robot's experience.

We set $K_R = 0.17$ and $K_E = 2$ and draw 5,000 target poses uniformly at random in the robot's workspace. After every 250 steps

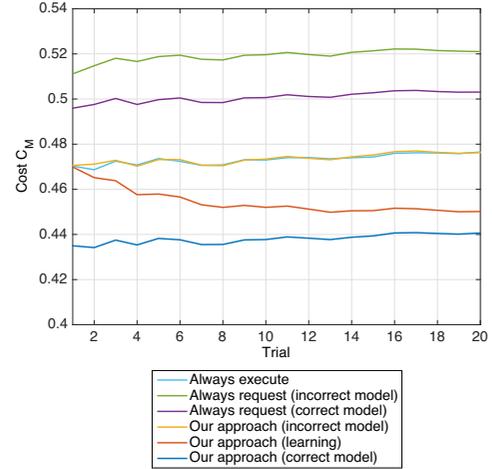


Figure 7. Average learning performance in the example of Section 3.3.

we update the parameters $\boldsymbol{\alpha}_a$, $a \in \mathcal{A}$, in a total of 20 update steps. Figure 7 depicts the obtained results, including the performance of the five approaches featured in Table 1, as well as the learning approach. As is clear from the plot, the learning approach initially exhibits a performance similar to our approach with an incorrect model. However, as more information from the user becomes available, the performance slowly converges towards that of a correct model.

5 Experimental results

We illustrate the application of our framework in two real-world tasks involving the Baxter robot. In both tasks, Baxter posed the requests to the human user through voice commands. In the first task, the robot pours a bottle to a cup placed on a table. The robot was provided with five demonstrations of the intended motion through kinesthetic teaching (as demonstrated in Fig. 3), with the cup placed in different positions in the table.⁶ These trajectories were used to compute a ProMP as described in the illustrative example of Section 3. At execution time, we use homography to determine the position of the cup in the table, and modulate the ProMP as in (3) to obtain the trajectory for that target. Figure 8(a) illustrates the execution of a pouring motion. If the bottleneck ends over the cup, the motion is considered a success and the robot incurs a small execution cost (depending on the distance to the center of the cup). If the bottleneck does not end over the cup, the motion is considered a failure, and the robot incurs a large penalty. For decision-making, a simple model predicting the probability of failure is used to compute the execution cost.

Note that, even if this task in itself is not very complex, the limited number of demonstrations prevents the robot from confidently generalizing the motion to the whole table, allowing us to study the effectiveness of our approach in handling the robot's limitations.

In the second task, the robot helps the user to put on a backpack. This task is significantly more complex, comprising two motions involving both of Baxter's arms (in a total of 14 DoF) and requiring some compliance from the user. The robot was provided with four demonstrations of the intended motion, with the user placed in different positions in front of the robot. We used as targets the hands of

⁶ Trajectories were stored in joint space.

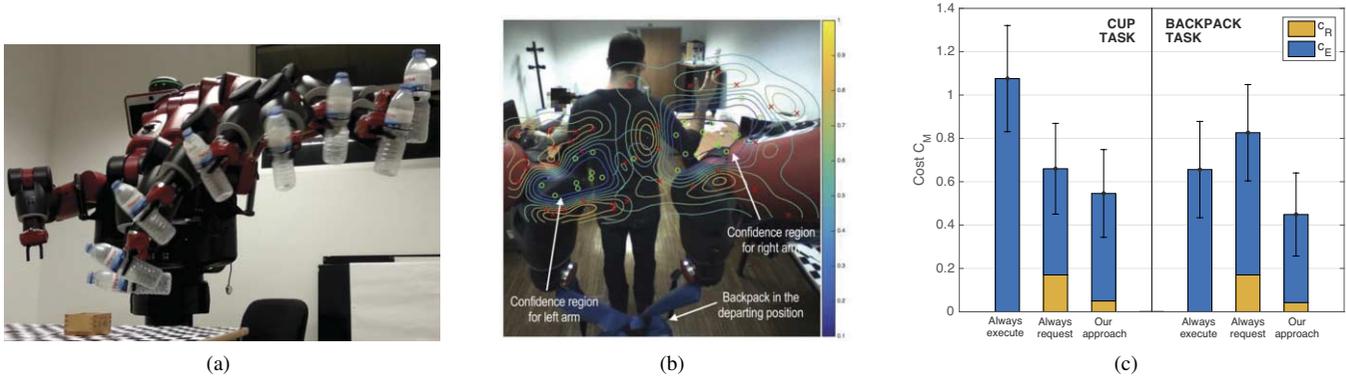


Figure 8. (a) Example of a pouring motion executed by Baxter. (b) Contour lines of the execution cost function c_E for the backpack task, superimposed on the robot’s view. (c) Comparison of the average performance of the three approaches. The results portrayed are averages over 17 trials (cup) and 16 trials (backpack) for each method. Error bars depict the standard error.

the user. At execution time, we again use homography to determine the position of the hands of the user in a vertical plane. The execution cost merely measures the success or failure of the motion. For decision-making, we use an SVM trained to predict success/failure given the target position to estimate the execution cost. Figure 8(b) shows the contours of the cost function for the backpack task.

We compare the performance of three approaches: (i) executing without ever requesting user assistance (**Always execute**); (ii) executing after always requesting user assistance (**Always request**); and (iii) ask human assistance when convenient (**Our approach**). Each approach was evaluated for a number of trials in each of the two tasks. In each trial, the robot is allowed to interact with a human user, where the target (cup or user) is placed in random positions in front of the robot. The robot proceeds as prescribed by the approach under consideration, and we evaluate the success of the corresponding motion. The results are summarized in Fig. 8(c).

Analyzing the results in detail, we note first that, as expected, the **Always execute** approach incurs no request cost. The **Always request** approach, on the other hand, incurs the largest request cost. However, overall, it is significantly better than the approach that always executes, which indicates that the request cost incurred by the former when requesting unnecessary user assistance still compensates the execution cost incurred by the latter with targets that it cannot safely reach. Our approach is the most cost-effective of all three, incurring less execution cost, request cost and overall cost than any of the other approaches. The results show that our approach successfully weights the robot’s limitations, enlisting user assistance when the latter can be of use, thus establishing a form of symbiotic autonomy where both user and robot assist one another for mutual benefit.

We illustrate in Figures 9(a)-9(c) the performance of the different approaches for different positions of the cup. Several interesting observations are in order. First of all, considering Fig. 9(a), it is possible to observe that the robot is able to successfully complete the tasks when the cup is in the vicinity of the area demonstrated (marked as the shaded region). When the cup was placed in a position relatively distant from that area, the modulation process is less accurate, leading the robot to “miss” the cup and fail the task.

Considering now Fig. 9(b), we note that by requesting the user to move the cup, a significant number of cup positions that the robot missed in the **Always execute** approach can now be successfully corrected into positions that the robot can successfully reach. However, for those positions that the robot was already able to reach, the request to the user is unnecessary, leading the robot to incur an un-

necessary request cost.

Finally, considering Fig. 9(c) it is clear that the robot requests only the user’s assistance in those positions that it cannot successfully reach. Some of these remain out of the reach of the robot, but a significant number of them (as already observed in the **Always request** approach) can now successfully be reached. These results illustrate that our approach is, indeed, able to successfully trade-off the cost of requesting the assistance from the human user and the potential benefit arising from such assistance. The results illustrated in Fig. 9 are summarized (in terms of cost) in the plot of Fig. 8(c). As expected, the approach **Always execute** incurs no request cost. Conversely, the average request cost for **Always request** is $K_R = 0.17$, since all trials incur exactly this cost. Our approach incurs smaller cost both in terms of execution and in terms of request.

6 Conclusions

In this paper we presented a novel approach that explores the concept of *symbiotic autonomy* in the context of collaborative manipulation tasks. Our approach enables a robot to reason explicitly about its limitation when executing a complex manipulation task with respect to a target provided by a human user, and reason about when the cost of disturbing the human user and requesting its assistance is compensated by the benefits arising from such assistance. Our approach relies on a model of the human responses to the robot’s request; when discrepancies are detected, our approach successively adjusts the model to a particular user, as the number of interactions increases. Our results show that our approach is effectively able to perform the desired trade-off, as well as adapt to gross errors in the user’s model.

Our work also opens interesting directions for future work. Our current decision process is myopic: it selects the best option at each time-step disregarding the potential future effect of the actions. However, by using a more evolved decision-theoretic approach, it is possible to further optimize the (long-term) performance of the robot. In fact, given the Markovian nature of the target displacement model used, it should be possible to formulate the decision problem of the agent using a decision-theoretic model such as a Markov decision process, for which solution techniques are readily available.

Acknowledgements

This work was partially supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UID/CEC/

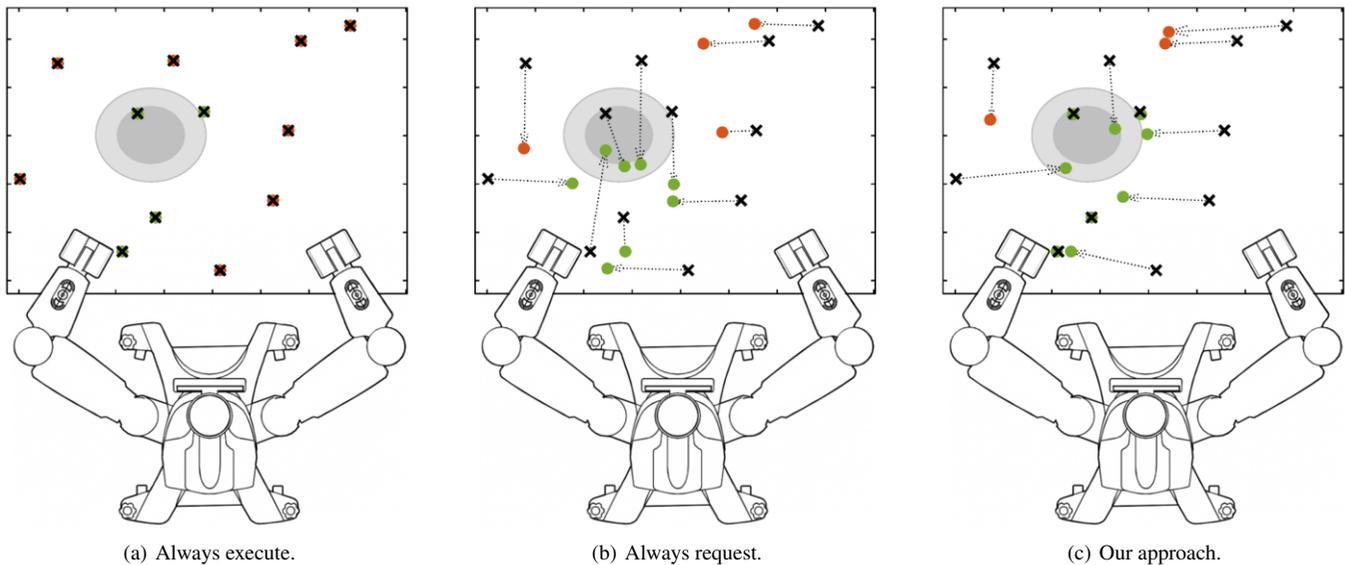


Figure 9. Visual representation of some of the tests performed with the Baxter robot in the pouring task. The gray area corresponds to the “comfort zone”, i.e., the area covered by the demonstrated trajectories. The black crosses (“x”) represent some of the positions where the cup was placed, and red and green dots represent whether that particular motion was successful or not (we consider a success whenever the bottle stops on top of the cup). For the approaches involving human motion, the dotted lines represent the motion performed by the human user upon request.

50021/2013 and the Carnegie Mellon Portugal Program and its Information and Communications Technologies Institute, under project CMUP-ERI/HCI/0051/2013. The first author acknowledges the PhD grant SFRH/BD/113695/2015.

References

- [1] B. Argall, S. Chernova, and M. Veloso, ‘A survey of robot learning from demonstration’, *Robotics and Auton. Systems*, **57**(5), 469–483, (2009).
- [2] B. Corteville, E. Aertbelien, H. Bruyninckx, J. De Schutter, and H. Van Brussel, ‘Human-inspired robot assistant for fast point-to-point movements’, in *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 3639–3644, (2007).
- [3] M. Dias, B. Kannan, B. Browning, E. Jones, B. Argall, M. Dias, M. Zinck, M. Veloso, and A. Stentz, ‘Sliding autonomy for peer-to-peer human-robot teams’, in *Proc. 10th Int. Conf. Intelligent Autonomous Systems*, pp. 332–341, (2008).
- [4] A. Dragan, S. Bauman, J. Forlizzi, and S. Srinivasa, ‘Effects of robot motion on human-robot collaboration’, in *Proc. 10th ACM/IEEE Int. Conf. Human-Robot Interaction*, pp. 51–58, (2015).
- [5] Y. Gao, H. Chang, and Y. Demiris, ‘User modelling for personalised dressing assistance by humanoid robots’, in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 1840–1845, (2015).
- [6] M. Gielniak, C.K. Liu, and A. Thomaz, ‘Generating human-like motion for robots’, *Int. J. Robotics Research*, **32**(11), 1275–1301, (2013).
- [7] A. Ijspeert, J. Nakanishi, and S. Schaal, ‘Learning attractor landscapes for learning motor primitives’, in *Adv. Neural Information Processing Systems*, volume 15, pp. 1547–1554, (2003).
- [8] S. Klee, B. Ferreira, R. Silva, J. Costeira, F. Melo, and M. Veloso, ‘Personalized assistance for dressing users’, in *Proc. 7th Int. Conf. Social Robots*, pp. 359–369, (2015).
- [9] G. Maeda, M. Ewerton, R. Lioutikov, H. Ben Amor, J. Peters, and G. Neumann, ‘Learning interaction for collaborative tasks with probabilistic movement primitives’, in *Proc. 14th IEEE-RAS Int. Conf. Humanoid Robots*, pp. 527–534, (2014).
- [10] Y. Maeda, T. Hara, and T. Arai, ‘Human-robot cooperative manipulation with motion estimation’, in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 2240–2245, (2001).
- [11] J. Mainprice and D. Berenson, ‘Human-robot collaborative manipulation planning using early prediction of human motion’, in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 299–306, (2013).
- [12] J. Mainprice, R. Hayne, and D. Berenson, ‘Predicting human reaching motion in collaborative tasks using inverse optimal control and iterative re-planning’, in *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 885–892, (2015).
- [13] J. Mainprice, E.A. Sisbot, L. Jaillet, J. Cortés, R. Alami, and T. Siméon, ‘Planning human-aware motions using a sampling-based costmap planner’, in *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 5012–5017, (2011).
- [14] F. Melo and A. Sardinha, ‘Ad hoc teamwork by learning teammates’ task’, *Autonomous Agents and Multi-Agent Systems*, **30**(2), 175–219, (2016).
- [15] A. Paraschos, C. Daniel, J. Peters, and G. Neumann, ‘Probabilistic movement primitives’, in *Adv. Neural Information Proc. Systems*, volume 26, pp. 2616–2624, (2013).
- [16] S. Rosenthal, J. Biswas, and M. Veloso, ‘An effective personal mobile robot agent through symbiotic human-robot interaction’, in *Proc. 9th Int. Joint Conf. Autonomous Agents and Multiagent Systems*, pp. 915–922, (2010).
- [17] P. Scerri, D. Pynadath, and M. Tambe, ‘Towards adjustable autonomy for the real world’, *J. Artificial Intelligence Res.*, **1**, 2–50, (2003).
- [18] S. Schaal, ‘Is imitation learning the route to humanoid robots?’, *Trends in Cognitive Sciences*, **3**(6), 233–242, (1999).
- [19] B. Sellner, F. Heger, L. Hiatt, R. Simmons, and S. Singh, ‘Coordinated multiagent teams and sliding autonomy for large-scale assembly’, *Proc. IEEE*, **94**(7), 1425–1444, (2006).
- [20] W. Sheng, A. Thobbi, and Y. Gu, ‘An integrated framework for human-robot collaborative manipulation’, *IEEE Trans. Cybernetics*, **45**(10), 2030–2041, (2015).
- [21] P. Stone, G. Kaminka, S. Kraus, and J. Rosenschein, ‘Ad hoc autonomous agent teams: Collaboration without pre-coordination’, in *Proc. 24th AAAI Conf. Artificial Intelligence*, pp. 1504–1509, (2010).
- [22] F. Stulp, J. Grizou, B. Busch, and M. Lopes, ‘Facilitating intention prediction for humans by optimizing robot motions’, in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pp. 1249–1255, (2015).

Attuning Ontology Alignments to Semantically Heterogeneous Multi-Agent Interactions

Paula Chocron^{1,2} and Marco Schorlemmer¹

Abstract.

In this paper we tackle the problem of semantic heterogeneity in multi-agent communication, i.e., when agents in a multi-agent system use different vocabularies for message passing, or might interpret shared vocabulary in varying ways. The problem of achieving meaningful communication in such semantically heterogeneous multi-agent interactions has been mainly tackled either by using ontology alignments to translate vocabularies, or by using methods that learn an alignment by observing how the utterance of particular terms affects the unfolding of an interaction. We propose solutions that combine these approaches and study how agents can use external alignments with possibly incomplete or erroneous mappings when communicating with each other in the context of a multi-agent interaction. We further show experimentally that with the experience gained through repeated interactions and by using simple learning techniques agents can find and repair those mappings of an ontology alignment that lead to unsuccessful interactions, thus improving the success rate of their future interactions.

1 INTRODUCTION

An important problem in the design and implementation of distributed systems is to guarantee a sufficiently good level of interoperability between separately engineered software components as for the whole system to function adequately with respect to the expected functionality it is supposed to deliver. A particularly critical problem is that of semantic heterogeneity, i.e., when the terms used in the exchange of information between system components are interpreted differently by each of them [14]. This problem has triggered a significant amount of research in the fields of databases, the semantic web, or multi-agent systems [9, 11, 8]. In this paper we will focus on the problem of semantic heterogeneity in multi-agent communication, having in mind open multi-agent systems for which an interaction model or protocol is specified, but whose agents might have differences in the vocabulary they use for message passing, or might interpret shared vocabulary in varying ways.

When faced with multi-agent systems whose agents use different vocabularies in their communicative acts, the immediate and most common solution is to resort to some semantic alignment technique, maybe supplemented with some process of vocabulary or alignment negotiation, so as for agents to determine how foreign vocabulary needs to be interpreted using the local one [11, 16, 22, 17]. The success of this solution is obviously very dependent on the quality of the

alignments that can be computed, which in turn is very much conditioned by the detail in which vocabularies are specified and to the external semantic resources that might be available. For ontologies with a rich taxonomic structure and with detailed axiomatic specifications of the intended meaning of entities and relations, one can take advantage of state-of-the-art ontology matching tools that exploit many different kind of techniques, going from simple syntactic matching all the way to formal logical reasoning [11]. But when faced with underspecified vocabularies, the alignments obtained may prove to be insufficient for achieving the semantic interoperability required for a multi-agent interaction to be successful.

An alternative approach that does not depend on any ontology, semantic alignment tool or external semantic resource, was described in [3], in which agents gradually learn from the experience of repeated interactions those mappings of their vocabularies that lead to successful multi-agent communication. In that approach, meaning is assumed to be only determined by the interaction context, and it is never explicitly communicated. Unfortunately, the convergence to a common vocabulary using this approach can be very slow, i.e. many repetitions of the same interaction need to be enacted to get reasonable expectations of success, since no other source of meaning besides the interaction is taken into account.

Consequently, it seems reasonable to attempt to improve upon these two complementary approaches by combining them, developing novel semantic alignment techniques for multi-agent communication that can take advantage from the strengths of both approaches. Such techniques would exploit the availability of the ontological knowledge associated to a vocabulary and the powerful ontology matching techniques that make use of it, but would also take into account the experience that agents accumulated of the actual use of their vocabulary in the concrete contexts of particular interactions and the outcomes of these.

In this paper we set out to show how agents that are to perform a task together as specified in an interaction model or protocol can learn dynamically a translation that is useful for their interaction, with the help of semantic alignments computed by some external ontology matcher. While relying on these semantic alignments can provide valuable information that eases the convergence to a meaningful translation, external alignments can also contain errors or be inadequate for the particular interaction context in which the agents are using them, something that could hinder successful communication.

We first propose a straightforward way of combining a previously computed, external ontology alignment between separate vocabularies with the alignment technique that learns the semantic relationship between vocabularies from the experience gained by repeated multi-agent interaction. We then move on to describe a more powerful technique that uses reinforcement learning techniques to han-

¹ Artificial Intelligence Research Institute (IIIA-CSIC), Bellaterra (Barcelona), Catalonia, Spain. e-mail: pchocron, marco@iiia.csic.es

² Universitat Autònoma de Barcelona, Bellaterra (Barcelona), Catalonia, Spain

dle low quality mappings of the external ontology alignment in the context of the interaction, and thus to improve the ratio of success in repeated interactions, by taking into account previous experience with these low quality mappings.

We compare experimentally the proposed techniques, showing that they both solve the drawbacks of using only one source of meaning (only ontology alignment or only interaction experience) and improve the success rate of multi-agent interactions. Our methods are independent of the how vocabularies are ontologically structured and of the internal structure of each agent. In addition, each agent computes its own alignment based on its own interaction experience, so that no shared framework for meaning negotiation is required.

2 ALIGNMENTS VS. INTERACTION EXPERIENCE

In this paper we focus on interactions between two agents a_1 and a_2 , who might use words from different vocabularies to communicate. A vocabulary is the finite set of words an agent is allowed to use in its messages, and we will write V_1 and V_2 for the vocabularies of our two agents. Each agent can organise its vocabulary in its own way, with additional structure that makes it a taxonomy of words, or even a fully fledged ontology specifying the intended meaning of the words of the vocabulary. In our work, however, we only need to assume that a vocabulary V is provided with some *similarity measure* $sim : V \times V \rightarrow [0, 1]$ between its words.³

We have mentioned two main approaches to tackling semantic heterogeneity in agent communication: those that rely on external alignments and those that learn from repeated interaction. In this section we will explain both of these techniques, formulating them as solutions to the problem of choosing how to interpret a foreign word in a received message. We focus on the situation in which agent a_1 receives a message with a word v_2 sent by agent a_2 when it was actually waiting for one from some known set U of expected words to receive. The agent therefore needs to choose a word $v_1 \in U$ that matches with the received one v_2 , in such a way that the interaction proceeds correctly. We will explain each of the two approaches as a technique to compute a probability distribution for each $v_1 \in U$ according to which agent a_1 can choose a possible match for v_2 .

2.1 Using External Vocabulary Alignments

One approach to achieving mutual understanding between agents that use different vocabularies is to use an external alignment, taking advantage of the multiple matching techniques that have been developed in the last decades. These techniques vary from sophisticated OWL ontologies matchers to syntactic similarity measures, and the choice between these possibilities will depend on the additional structure in the vocabularies of agents, the availability of the matching tools, and the access to the vocabulary and structure of the agents's interlocutors.

Definition 1 An alignment \mathcal{A} between two vocabularies V_1 and V_2 is a finite set of mappings between words in V_1 and V_2 . A mapping is defined as a quadruple $\langle v_1, v_2, n, r \rangle$, where $v_1 \in V_1$, $v_2 \in V_2$, $n \in (0, 1]$ is the degree of confidence on the mapping, and r is the kind of relation that holds between words. An alignment contains at most one tuple for each pair of words $\langle v_1, v_2 \rangle$. [5]

³ Adequate similarity measures will depend on the additional structure given to the vocabulary. For taxonomies, for instance, a choice could be the Wu-Palmer measure [24]. Even if no similarity measure is provided, we can always resort to the trivial one that assigns 1 to the identity and 0 otherwise.

In this work, we will consider alignments with only equivalence (\equiv) as the relation holding between words in the mappings. Given an alignment \mathcal{A} , if a mapping $\langle v_1, v_2, n, \equiv \rangle$ belongs to \mathcal{A} we will write $v_1 \equiv v_2$ and denote its confidence with $conf(v_1 \equiv v_2)$.

The quality of vocabulary alignments is typically measured in comparison with a *reference alignment*, for which values of *precision* and *recall* are computed. As it is commonly done, we do not take into account the confidence degrees in these measures.

Definition 2 Given an alignment \mathcal{A} , let \mathcal{A}' denote the set of mappings of \mathcal{A} for which we have removed the confidence degree, i.e., $\mathcal{A}' = \{ \langle v_1, v_2, r \rangle \mid \langle v_1, v_2, n, r \rangle \in \mathcal{A} \text{ for some } n \}$. The precision of an alignment \mathcal{A} with respect to a reference alignment \mathcal{B} is the fraction of the mappings in \mathcal{A}' that are also in \mathcal{B}' :

$$precision(\mathcal{A}, \mathcal{B}) = \frac{|\mathcal{A}' \cap \mathcal{B}'|}{|\mathcal{A}'|}$$

while its recall is the fraction of the mappings in \mathcal{B} that were found by \mathcal{A} :

$$recall(\mathcal{A}, \mathcal{B}) = \frac{|\mathcal{A}' \cap \mathcal{B}'|}{|\mathcal{B}'|}$$

The most straightforward approach to use an alignment to tackle the problem of semantic heterogeneity in agent communication is to use it directly for translating words in messages. If an agent receives the word v_2 while waiting for words in U , it will choose the word that matches with v_2 with highest confidence in the alignment; if there is no such word, it chooses one randomly. This approach will work well if the alignment is adequate for the task the agents are performing; however, low recall will mean more random choices, which can cause unsuccessful interaction, while low precision implies a higher probability of choosing incorrect matches, which can also cause an interaction to fail.

One way of mitigating the effect of low recall is to consider not only the mappings that are explicitly present in the alignment, but to take into account the additional structure vocabularies may have. This can be achieved by using the similarity measure between words of one vocabulary in order to choose a word that is close to a match. Consider a similarity threshold $\theta \in [0, 1]$. For each $v_1 \in U$, let $V_{\equiv v_2}(v_1)$ be the set of words that match with v_2 in the alignment and that are closer than θ to v_1 :

$$V_{\equiv v_2}(v_1) = \{v'_1 \in V_1 \mid v'_1 \equiv v_2 \text{ and } sim(v_1, v'_1) \geq \theta\}$$

To compute a probability distribution over the interpretations, we first assign a value to each possibility. In this case, the value of interpreting v_2 as v_1 is given by

$$\mathcal{V}_{alg}(v_1, v_2) = \begin{cases} \max_{v'_1} conf(v'_1 \equiv v_2) sim(v_1, v'_1) & \text{if } V_{\equiv v_2}(v_1) \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

where $v'_1 \in V_{\equiv v_2}(v_1)$.

To reduce the effects of low precision, a solution is to not trust the alignment completely, including an exploration parameter ξ_1 in the definition of a probability distribution over words in U .

Alignment Criterion (a1g).

Let $\hat{\mathcal{V}}_{alg}(v_1, v_2)$ be the normalised value of $\mathcal{V}_{alg}(v_1, v_2)$ for each $v_1 \in U$. Choose $v_1 \in U$ with probability:

$$p_{alg}(v_1) = \xi_1 \times \hat{\mathcal{V}}_{alg}(v_1, v_2) + (1 - \xi_1) \frac{1}{|U|}$$

The exploration parameter ξ_1 introduces a well known dilemma. Setting it to large values result in a very weak method when there are wrong mappings in the alignment, while with low values we can be losing useful information. Ideally, the parameter should depend on the precision of the alignment \mathcal{A} , but this is something agents are not expected to know in advance.

2.1.1 The Alignment Criterion in Action: A Running Example

In what follows we introduce an illustrative example using a simple travel agency scenario adapted from [2]. The complete vocabularies and specifications of the ontologies used by the agents can be found in [2]; we do not need them explicitly here.

Consider a Travel Agent (TA) offering two services: to book a flight for a given date and destination, or to provide information about the available hotels in a city. TA uses its own vocabulary, which is not necessarily shared by its clients; we consider a particular Customer (C) who uses a different language. To be able to interact with C, agent TA may use an ontology alignment provided by some external source. Table 1 shows a relevant fragment of the alignment provided by the matcher Falcon-AO [15] as reported in [2]. Consider a situation in which TA is waiting for agent C to specify if it wants a return flight or not, so $U = \{OneWay, RoundTrip\}$ at this state. Using the alignment criteria and the simple 0 – 1 similarity measure, if C sends $\{Single\}$, the travel agent will interpret it as $\{RoundTrip\}$.

$v_1 \in V_1$	$v_2 \in V_2$	Confidence
Return	Package	0.41
Single	RoundTrip	0.19
UnregCustomer	OneWay	0.03
Flight	Customer	0.01
destination	airlineCompany	0.99
carrier	to	0.99
departing	leavingDate	0.99
origin	from	0.99
returning	returnDate	0.76
hotelBookingsIn	city	0.30

Table 1: Extract of the alignment provided by Falcon-AO as reported by Atencia in [2]

2.2 Learning from Interaction Experience

A second approach to communicating with linguistically heterogeneous partners does not use any external resource, but instead considers meaning to be determined by the specification of interactions. Agents that perform the same task repeatedly interacting with the same partners can learn the meaning of words by simply observing the outcomes for different possibilities and choosing again the ones that gave good results. This technique was developed in [3], and in this section we reformulate it as a solution to a learning problem, which will be useful to introduce, in Section 3, the novel methods we propose.

In this approach, each agent has its own specification of the interaction it takes part in. In the last decades, the multi-agent community has discussed thoroughly the question of how to model agent interactions and communication languages, and many different techniques have been proposed [23]. In this work we will abstract these techniques to consider only a very simple message exchange mechanism, modelled as a finite-state automaton in which state transitions

are triggered by messages. To represent different outcomes of the interaction that all agents can recognise, we define a set of predicates called *state properties* to characterise final states.

Definition 3 Given two agents a_1 and a_2 , a vocabulary V and a set of state properties SP , an interaction model IM is defined as a tuple $\langle Q, q_0, \delta, F, \rho, speaks \rangle$ where Q is a finite set of states, $q_0 \in Q$ is the initial state, $F \subseteq Q$ is the set of final states, $\rho : F \rightarrow \mathcal{P}(SP)$ is a function assigning a subset of state properties to each final state, and $speaks : Q \rightarrow \{a_1, a_2\}$ is a function assigning to each state its sender agent, and $\delta : Q \times V \rightarrow Q$ is a partial function called the transition function.

Note that while we do not specify any particular turn-taking pattern, we do require that, for each state, all messages labelling transitions from this state share the same sender agent, who is determined with the *speaks* function.

While we assume that the language to specify state properties in SP is shared, agents may use different vocabularies for the messages they send to each other. Consequently we will denote with IM_1 the interaction model followed by agent a_1 using vocabulary V_1 , and with IM_2 the one followed by agent a_2 using vocabulary V_2 .

When the interaction is in a state $q \in Q$ for which $speaks(q) = a_1$, agent a_1 chooses a word from V_1 to utter according to IM_1 . If instead $speaks(q) = a_2$, a_1 will wait to receive a message from a_2 . Since the received word is from V_2 , it will need to interpret it in the context of that particular interaction state, following a transition according to IM_1 . That is, it will choose a word from the set of expected words for state q , given by $U(q) = \{v \in V_1 \mid \delta(q, v) \text{ is defined}\}$. Since a_2 does the same, an interaction between two agents can be defined as a sequence of uttered messages along with how they were interpreted. A successful interaction is one that leads both agents to final states with the same state properties.

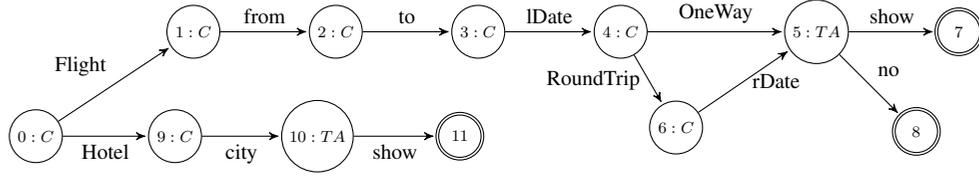
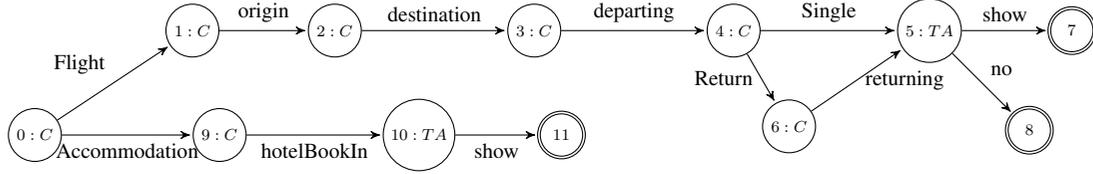
For the following definitions it will be useful to restrict interaction models to deterministic ones, and to recall that in this case an accepted string can be associated with only one sequence of states that are visited to produce it.

Definition 4 A successful interaction between interaction models IM_1, IM_2 is a finite sequence of pairs of words $\langle v_1, v_2 \rangle$ with $v_1 \in V_1$ and $v_2 \in V_2$ such that the projection of its first coordinates is a string accepted by IM_1 , the projection of its second coordinates is a string accepted by IM_2 , and both projections visit, in their respective interaction model, sequences of states with the same senders, reaching final states with the same state properties.

Successful interactions lead to an intuitive notion of alignment between two interaction models, which is composed of all tuples $\langle v_1, v_2 \rangle$ that belong to successful interactions. However, this alignment could have different interpretations for the same word, corresponding to different states. Agents will be interested, more specifically, in finding which interpretation they need to choose according to each state in order to interact successfully:

Definition 5 Let *int* be a successful interaction between IM_1 and IM_2 , and let *states* be the sequence of states visited in IM_1 determined by the projection of the first coordinates of *int*. Then all tuples $\langle q, v_1, v_2 \rangle$ obtained by adding the ordered items of *states* to the pairs in *int* belong to the pragmatic alignment from interaction model IM_1 to interaction model IM_2 .

Notice, first, that pragmatic alignments are defined from one interaction model to another one, and second, that unlike those in Defini-

Figure 1: Interaction Model IM_{TA} for the Travel AgentFigure 2: Interaction Model IM_C for the Customer

tion 1, mappings are parametrised by states. We will write $v_1 \simeq_q v_2$ if $\langle q, v_1, v_2 \rangle$ belongs to the pragmatic alignment from IM_1 to IM_2 .

Definition 6 Two interaction models IM_1 and IM_2 are structurally equivalent if all strings accepted by them separately are projections of a successful interaction between them.

Interaction models are structurally equivalent if they are equivalent modulo interpretation of words in messages, or in other words, interactions can always finish successfully if the correct mappings are chosen. In this work we will assume that IM_1 and IM_2 are structurally equivalent. While the methods we propose can be easily adapted to interaction models that have minor differences, they will not perform well when such differences are substantial; how to adapt heterogeneous protocols is a difficult problem that is out of the scope of this paper.

To interact successfully with each other, agents need to discover the pragmatic alignment between their interaction models. The method proposed in [3] that we explain in this section, as well as the novel ones presented in the next one, are techniques to let agents learn these pragmatic mappings automatically from repeated interaction. We will formulate these solutions using standard concepts and notation from Reinforcement Learning (see, e.g., [20]). As usual, we start by defining the learning model.

Since a_1 needs to learn which interpretation is good for a received word in a specific state, the states of the learning model will be pairs $\langle q, v_2 \rangle$, where $q \in Q$, $speaks(q) = a_2$, and v_2 is a word received. In that situation, a_1 can choose how to interpret v_2 from a the set of expected messages, therefore the set of actions for a state $\{q, v_2\}$ of the learning model are the words in $U(q)$. Let us make two remarks. First, we are abstracting the interaction states in which a_1 speaks, because it does not need to learn any interpretation in those, and the pragmatic alignment will be independent of the messages it utters. Second, the agents do not know the learning model a priori, since they ignore which messages their interlocutor can utter. We will use methods that do not require agents to know the model. Our objective is to estimate action values $\mathcal{V}(v_1 \simeq_q v_2)$, which represent the confidence in that v_2 should be interpreted as v_1 in q . In this section we present a simple solution: values for all mappings in an interaction are updated when the interaction ends, adding 1 if it succeeded, or 0 if it failed.

In each interaction they take part in, agents will keep a record of the mapped pairs as a sequence $(q_0, v_{10}, v_{20}), \dots, (q_n, v_{1n}, v_{2n})$.

When the interaction ends, the values of mapped pairs are updated in the following way.

$$\mathcal{V}_{exp}(v_1 \simeq_q v_2) = \begin{cases} \mathcal{V}_{exp}(v_1 \simeq_q v_2) + 1 & \text{if the interaction succeeded} \\ \mathcal{V}_{exp}(v_1 \simeq_q v_2) & \text{if the interaction failed} \end{cases}$$

Let $\#exp(q, v_2) = \sum_{v_1' \in U(q)} \mathcal{V}_{exp}(v_1' \simeq_q v_2)$, and consider an exploration parameter ξ_2 close to 1. The interpretation can be chosen according to the following criterion:

Experience Criterion (exp).

Let $e(q, v_1, v_2) = \frac{\mathcal{V}_{exp}(v_1 \simeq_q v_2)}{\#exp(q, v_2)}$. Choose $v \in U(q)$ with probability:

$$p_{exp}(v_1 \simeq_q v_2) = \begin{cases} \xi_2 e(q, v_1, v_2) + (1 - \xi_2) \frac{1}{|U(q)|} & \text{if } \#exp(q, v_2) > 0 \\ \frac{1}{|U(q)|} & \text{if } \#exp(q, v_2) = 0 \end{cases}$$

This method divides the matching decisions in two phases. While there is not enough information from the experience, the agent maps randomly; once interactions start to be successful, it repeats good choices.

Let us make two remarks. First, the exploration parameter ξ_2 is included to consider situations in which the configuration of the interaction protocols can make two mappings be correct in one state. Second, since the outcome of the interaction is only known once it finished, this mechanism is affected by the *credit assignment problem*: learning from unsuccessful interactions is difficult, because it is not known which of the mappings was wrong.

2.2.1 The Experience Criterion in Action

Consider again the Travel Agency example introduced in Section 2.1.1, but now suppose Travel Agent TA has specified the tasks it can perform with the interaction model IM_{TA} in Figure 1. The letter in each state represent its speaker agent, while transitions are only labeled with the content of messages. Consider $SP = \{success, failure, book, info\}$ and the following state property function: $\rho(7) = \{success, book\}$, $\rho(8) = \{failure, book\}$, $\rho(11) = \{success, info\}$.

The Customer Agent C, from its side, follows interaction model IM_C which is structurally equivalent to IM_{TA} . An example of a successful interaction between these interaction models is given by the sequence $\langle Hotel, Accommodation \rangle, \langle hotelBookIn, city \rangle, \langle show, show \rangle$. This implies, for example, that in the pragmatic alignment from IM_{TA} to IM_C , $city \simeq_9 hotelBookIn$.

Using criterion exp , the TA will first go through a learning phase, in which the interactions will be mostly unsuccessful since it is choosing mappings randomly. However, since the interaction has few interpretations choices and each of them with few options, it should not take long to find correct mappings.

3 COMBINING ALIGNMENTS WITH INTERACTION EXPERIENCE

In this section we propose methods than combine an external source of meaning with the interaction context. We consider again a_1 interacting repeatedly with a_2 ; now, in addition, a_1 has access to an external alignment \mathcal{A} between V_1 and V_2 .

A central concern when using \mathcal{A} is that, since it was produced by an external resource, it does not necessarily agree with the pragmatic alignment between IM_1 and IM_2 . This raises the question of how vocabulary alignments relate to pragmatic ones; taking into account that the last ones are parametrised by states. A straightforward definition of the precision and recall measures when compared to two interaction models considers as reference alignment all pairs in all successful interactions, considering correct all the mappings that are useful when interacting.

There is a situation in which a mapping in an external alignment results particularly harmful for the interaction. The problem arises when a mapping $v_1 \equiv v_2$ belongs to \mathcal{A} and $v_1 \in U(q)$, but $v'_1 \simeq_q v_2$ does not belong to the pragmatic alignment between the interaction models of both agents. In the travel agency example, this happens in state 5, because $RoundTrip \equiv Single$ and $RoundTrip \in U(5)$, but $RoundTrip \not\equiv_5 Single$ because it does not lead to any successful interaction. When the alignment is followed, most of the times $RoundTrip$ will be chosen as an interpretation for $Single$, causing the interaction to fail. We will refer to this kind of mappings as *misleading*. A misleading mapping can be *repaired* by making its value lower than other possibilities, so that it is not chosen anymore.

The following is a very straightforward combination of the alignment with the interaction experience:

Alignment and Experience Criterion (**alg-exp**).

Choose $v_1 \in U(q)$ with probability:

$$p_{alg-exp}(v_1 \simeq_q v_2) = \begin{cases} p_{exp}(v_1 \simeq_q v_2) & \text{if } \#exp(q, v_2) > 0 \\ p_{alg}(v_1 \simeq_q v_2) & \text{if } \#exp(q, v_2) = 0 \end{cases} \quad 4$$

This method affects only the exploratory part of the learning in the exp criterion; successful interactions are taken into account in the same way. This is because mappings that lead to successful experiences belong to the pragmatic alignment by definition. This straightforward combination has two drawbacks. First, it still considers only the successful matches and discards all the information in the ones

⁴ It would be reasonable to use $p_{alg}(v_1 \simeq_q v_2)$ in the exploration of exp instead of choosing randomly, we do not add it for clarity. The same holds for the next criterion.

that failed. Second, the dilemma of when to choose randomly instead of following the alignment that we explained in the Alignment Criterion alg is not solved.

3.1 Learning from Unsuccessful Experiences

We now present a more elaborate method that is able to repair misleading mappings and to find missing ones more efficiently. This is achieved by updating the original confidences in the alignment with the experience of unsuccessful interactions, combining the following two ideas to deal with low quality alignments. First, to mitigate low precision, mappings involved in unsuccessful interactions are punished. Second, to mitigate low recall, the confidence in a mapping is updated taking into account the quality of the alignment possibilities that were found subsequently. The intuition behind this second idea is that, if good mappings were found after a particular choice of interpretation, it is likely that it was correct. Consider a simple analogy with human conversations: if someone is not sure of having understood a message, but the dialogue continues as expected, she will assume her understanding was correct, whereas if stranger messages arrive, her confidence will decrease.

Our method, again, divides the learning in two phases. The difference with the already presented criteria is that, in the first phase, agents also compute a distribution over the possible interpretations, representing their confidence in a mapping belonging to the pragmatic alignment. This value depends on the original confidence (given by the external alignment) and on the observations of what happened when the mapping was chosen.

In this first phase, we use a method that resembles classical temporal difference reinforcement learning techniques, but instead of computing an expected reward, we will update a value representing the confidence in that a mapping belongs to the pragmatic alignment. This confidence starts being the value given in the alignment, and it evolves with the experience. Agents will try to minimize the punishment, which is assigned to the last mapping when an interaction fails; in this way they explore different possibilities.

Let $\alpha \in (0, 1]$ be a *forgetting parameter*, and $C \in (0, 1]$ a *punishment*. As initial values, we use the confidences in the alignment:

$$\mathcal{V}_{ev}(v_1 \simeq_q v_2) = conf(v_1 \equiv v_2)$$

When an interaction finishes in failure, agents will have a sequence of the states in which they made mapping decisions like in the exp method. For each of these states, each agent will update $\mathcal{V}_{ev}(v_1 \simeq_q v_2)$ as follows:

- If $i = n$, a punishment of $-C$ is assigned for having failed:

$$\mathcal{V}_{ev}(v_{1n} \simeq_{q_n} v_{2n}) = (1 - \alpha)\mathcal{V}_{ev}(v_{1n} \simeq_{q_n} v_{2n}) + \alpha(-C)$$

- For $i < n$, each agent takes into account the mapping possibilities in future states:

$$\mathcal{V}_{ev}(v_{1i} \simeq_{q_i} v_{2i}) = (1 - \alpha)\mathcal{V}_{ev}(v_{1i} \simeq_{q_i} v_{2i}) + \alpha \max_v \mathcal{V}_{ev}(v_{1i+1} \simeq_{q_{i+1}} v_{2i+1})$$

where $v \in U(q_{i+1})$.

We do not need to allow explicitly for exploration, since the back-propagation of the punishment already has that effect.

Evolving Alignment and Experience Criterion (**ev-alg-exp**).

Let $max = \operatorname{argmax}_{v_1 \in U(q)} (\mathcal{V}_{ev}(v_1 \simeq_q v_2))$. Choose $v_1 \in U(q)$ with probability:

$$p_{ev}(v_1 \simeq_q v_2) = \begin{cases} p_{exp}(v_1 \simeq_q v_2) & \text{if } \#exp(q, v_2) > 0 \\ \frac{1}{|max|} & \text{if } \#exp(q, v_2) = 0, v_1 \in max \\ 0 & \text{if } n = 0, v_1 \notin max \end{cases}$$

Note that, since values are updated when the interaction is over, the new maximum value for future states can be used. This will back-propagate the punishment to all mappings already in the first unsuccessful interaction, repairing misleading mappings in less interactions, although it is less stable. This is the approach we use in the experimentation.

Analysis of ev-*alg-exp*: repairing misleading mappings. As we will show, the *ev-*alg-exp** technique succeeds to find misleading mappings for most configurations; however, there is one particular case in which it does not work well. The pathological case arises when, for an interaction model *IM* and an alignment *A*, the following conditions hold: 1. Two strings *s* and *s'* accepted by *IM* have the same word *v* in the *j*-th position, 2. *A* has a misleading mapping for a word in *s* before *j*, and 3. If *q* is the state for *v* in *s'*, *A* has a correct mapping $v \simeq_q w$, but it also has a misleading mapping $v' \simeq_q w$ ⁵. In this case, the technique the $v \simeq_q w$ mapping the first time *s'* is successful, thus always choosing it from there on, and not being able to decrease the value of $v' \simeq_q w$. There are possible fixes to this problem, but since it is a rare case, we choose to resort in the exploration from *exp* to repair it.

If the case above does not hold, and all messages in the protocol have some probability of being uttered, *ev-*alg-exp** always repairs misleading mappings. This is simple to see if we consider a misleading mapping $v_1 \equiv v_2$ in *q* and all mappings made after that one in an interaction. If there are no positive mappings, the value of $\mathcal{V}_{ev}(v_1 \simeq_q v_2)$ will decrease. This may not be enough to make it lower than other options, but since the values of subsequent mappings will never increase, $\mathcal{V}_{ev}(v_1 \simeq_q v_2)$ will continue to decrease, and by a greater factor in future interactions. If, on the other hand, there are positive mappings, they need to be misleading, so they will also be repaired eventually, getting to the first situation. It can be the case that this mappings are correct for other strings, but since correct mappings do not modify the values they will not damage the process, unless the case above occurs, preventing one mapping of being chosen. Since this is true for any experience including $v_1 \simeq_q v_2$, it will eventually be repaired.

3.2 The Combination Methods in Action

Let us analyse the performance of the two criteria presented in this section in the travel agency scenario. As we already mentioned, there is a misleading mapping between *Single* and *RoundTrip*; as consequence, the *exp-*alg** method will fail many times, until the agent chooses to explore. This is solved in few interactions when using the *ev-*alg-exp** criterion. Since the interaction fails right after *Single*

is mapped with *RoundTrip*, the fourth criterion will find this error in just one unsuccessful interaction. Also the first time after choosing it, the agent can confirm the $\langle \textit{Flight}, \textit{Flight} \rangle$ mapping, since its confidence will be increased with the $\langle \textit{leavingDate}, \textit{departing} \rangle$ and $\langle \textit{from}, \textit{origin} \rangle$ mappings.

To evaluate our predictions, we studied experimentally the performance of the four criteria in the travel agency scenario, letting agents interact for 60 times. We measured the proportion of successful experiences, as well as after how many interactions they converged, i.e., always understood each other. The results, which are as expected, are shown in Table 2.

Criterion	alg	exp	alg-exp	ev- <i>alg-exp</i>
successes (%)	30	92	81	96
convergence	-	7.1	20.1	3.9

Table 2: Results for the travel agency scenario

4 EXPERIMENTAL EVALUATION

To evaluate the methods we propose, we studied how they perform experimentally when used by agents with different vocabularies. In this section we present the results, after discussing the generation of data for experimentation.

4.1 Data Generation

Designing experimentation with interaction protocols raises the immediate question of how to obtain test cases. While it is simple to build random finite state automata, it is not clear that all possible protocols model a realistic interaction, and the literature does not offer a useful characterisation of interaction or conversation protocols. We chose to generate deterministic automata parametrised by the size (given by the number of states) and to only restrict their shape by using a reasonably uniform distribution of the outgoing arrows among the states.

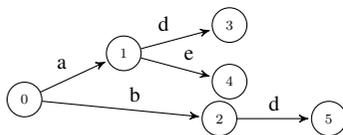
On the other side, we created vocabularies V_1 and V_2 randomly and defined a translation *A* between them. Since these are simple sets of words, the trivial similarity measure was used (1 for the same word, 0 for different ones). We labeled an interaction protocol *IM*₁ with words in V_1 and a structurally equivalent one *IM*₂ with its translations to V_2 . Finally, we explored alignments between V_1 and V_2 of different quality with parametrised values of precision and recall with respect to *A*. We used confidences of 1 for all the relations in the alignment.

4.2 Experiments

The performance of the methods we propose can be analysed in at least three different dimensions:

1. **The complexity of the interaction models:** we decided not to focus on this dimension for two reasons. First, it is already investigated in [3], and second, preliminary experiments did not show interesting variations of the performance. We used protocols with a fixed size of 90 states for all experiments.
2. **The parameters used:** For *ev-*alg-exp**, we experimented with different values of α and *C*, concluding that low (between 0.2 and 0.4) values of α gave the best results. The results for the punishment were less clear, but values between 0.7 and 0.9 seemed to be

⁵ An example is the following *IM* if the sender is always the same agent and $b \simeq_0 b'$, $a \simeq_0 a'$, but $\equiv a'$ (it is misleading), $d \simeq_1 d'$, $d \equiv d'$ and $e \equiv d'$



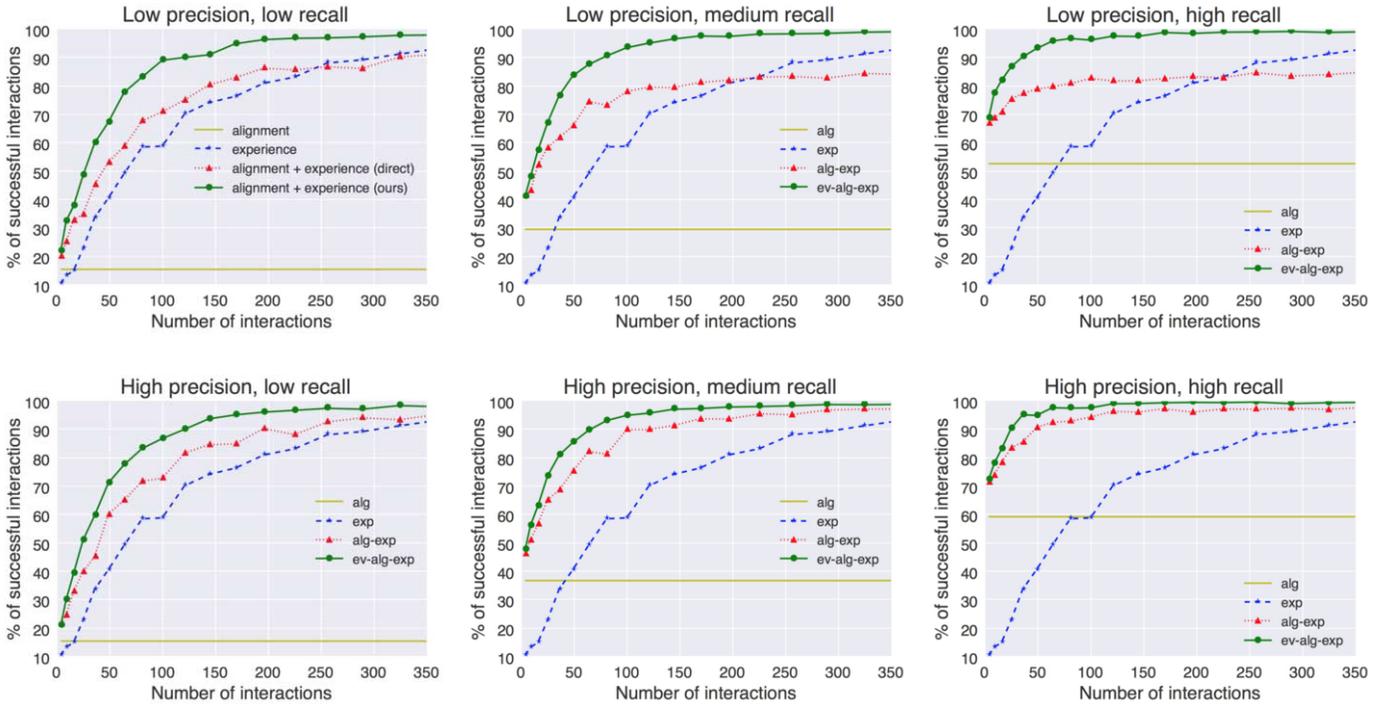


Figure 3: Results for Experiment 1, with size=90 and different alignment qualities

better. An hypothesis that should be confirmed is that this depends on the average of the mapping confidences in the alignment. We used $\alpha = 0.3$, $C = 1$, and $\xi_{1,2} = 0.1$ for the exploration parameters in the alignment and experience criteria.

3. **The quality of the alignments:** this dimension turned out to be the most interesting one, and we develop it in detail in this section.

4.2.1 Experiment 1: General Performance

The first experiment we performed provides a general comparison of the four methods. One test in this experiment is composed of two aligner agents that use the same matching criterion, one following IM_1 and the other IM_2 , each of them with an alignment with given values of precision and recall. For each learning criterion, we let agents go through a learning phase in which they interacted n times, performing the experiment for $n = i^2$ and $i \in [2, 20]$. After this training phase, we let agents interact again 100 times, without knowledge update, and measured the proportion of successful interactions. We performed 50 repetitions of each experiment, each time with a different alignment, but maintaining the same values of precision and recall.

We considered three quality classes for the precision and recall values: low: 0.2, medium: 0.5, high: 0.8, and evaluated the criteria that uses alignment with the nine combinations. Figure 3 presents the obtained results, showing the proportion of successful interactions for different lengths of the training phase. For space reasons, we only show six of the nine cases, but the remaining ones follow the same trend. Repeated interactions have no effect on the `alg` alignment; we plot the result of one experiment as a constant. The same happens for `exp` with different alignment qualities.

The two methods that combine the alignment and the learning from the experience perform better in the general case. Between them, `ev-alg-exp` is always the best one, performing better than

all other methods. Results are in general very good, achieving 90% of correct matches after only ~ 60 interactions. The method that only uses the learning also reaches values of success close to 1, but more slowly. Using only the alignment is the worst option, except when very short training periods are allowed. A more detailed analysis provides interesting observations about precision and recall:

- Recall affects performance more drastically than precision. This becomes clear when comparing the success rate for `alg`; while it increases significantly with higher values of recall, there is much less variation with different precision values. The two combining methods are also much better with high recall. This shows that errors in a contextualised environment are less dramatic, because it is more rare to find one in the expected messages.
- With low levels of precision, `ev-alg-exp` is significantly better than `alg-exp` after longer learning phases. This can be seen in the plots for low precision, particularly for high or medium recall, where the evolutionary technique reaches values close to 1 while `alg-exp` does not, being even worse than the technique without the alignment. This is explained because low precision implies higher possibility of misleading matches, which are only solved by making the alignment evolve.
- With low levels of recall, `ev-alg-exp` grows faster after short training periods. This is because it takes into account future good mappings, using the available information more efficiently.

4.2.2 Experiment 2: Focus on Precision and Recall

In Experiment 1, the effects of using alignments of different qualities are only suggested. To analyse in depth how the performance of our techniques changes with different values of precision and recall, we developed a second experiment. In Experiment 2, we let agents

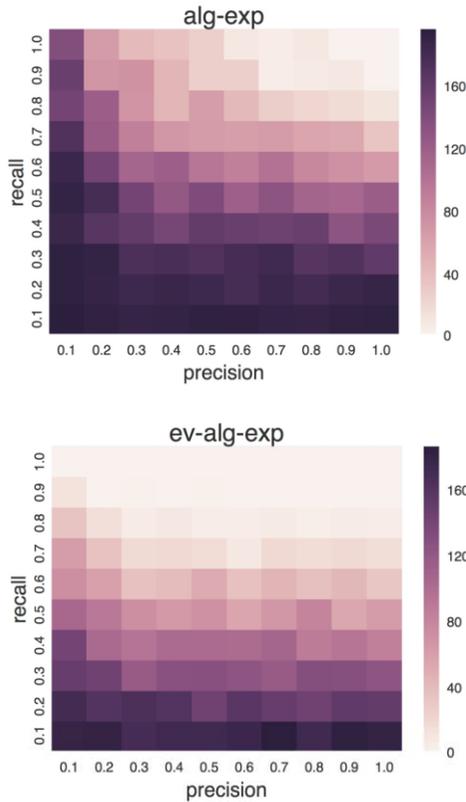


Figure 4: Results for Experiment 2

interact a large number of times (fixed in 200) and measured after how many interactions they converged when using the techniques `alg-exp` and `ev-alg-exp`. In this case, we considered convergence to be having 90% of successful interactions. The results are shown in Figure 4. The colour degrade represents the number of interactions before convergence, which increases with darkness. For the `alg-exp` technique, both low precision and low recall affect the performance, only converging fast when both values are high. As we already pointed out, low recall is more harmful than low precision. In the results for `ev-alg-exp` it can be seen that the precision has less influence; with high levels of recall, low values of convergence are achieved even with very low precision. This shows again how making the alignment evolve repairs misleading mappings, solving low quality in this dimension.

5 RELATED WORK

Although it is considered one of the main applications of ontology matching, the integration of vocabulary alignments in multi-agent interactions, and particularly the problem of how to use and repair them online, has not been deeply studied yet. Most of the work on using alignments in multi-agent systems tackles the problem of how communities of agents can achieve one common alignment from a set of heterogeneous ones. This is the approach followed by Laera et al., where argumentation techniques are used to decide between different alignments [16], and by Silva et al., who propose a method for agents to negotiate semantic bridges based on their confidence on each mapping rule. To the best of our knowledge, all existent methods consider an offline negotiation, that results in a common alignment that agents can use to communicate [18].

A well known formulation of the problem of learning meaning automatically from communication is the work of Steels [19]. In addition to the work we presented in Section 2, there exist other approaches, for example the one by Goldman et al. [12], in which the authors investigate how agents can learn to communicate in a way that maximises rewards in an environment that can be modelled as a Markov Decision Process. Our approach differs from this work in the inclusion of alignments and the modelling of an interaction context. In [4], the authors study a version of the multiagent, multiarmed bandit problem in which agents can communicate between each other with a common language, but message interpretations are not known.

A related problem not yet tackled by our approach is the one of aligning structural aspects of interaction protocols. Chopra and Singh have worked extensively on developing alignment techniques for protocols based on commitments [6, 7]. A different approach consists in developing dynamic protocols, that can be modified by agents while interacting according to the situation they are in [1, 13]. Similar approaches have been developed in the Web Services community, with the objective of making dynamic discovery and coordination of services possible [21].

6 CONCLUSIONS AND FUTURE WORK

We proposed methods that combine ontology alignments and language learning techniques, showing that they improve significantly the understanding between agents that interact in a given context. Our second method, in particular, shows how with simple techniques low quality of alignments can be mitigated.

Interesting conclusions about the quality of the alignments can be drawn from the experimentation. First, the level of recall seems to have more impact than the precision when the alignment is used for agent communication. This is worth exploring further, particularly given the current trend of favouring precision over recall in ontology alignment techniques [10].

Other directions of future research are found both from the ontologies and the agents side. The ability to estimate alignment quality measures would be useful for agents, because it would provide them with resources to choose between techniques or to fix parameters. On the other hand, while the `ev-alg-exp` criterion helps finding and avoiding possible errors in the alignment, this is only useful for one particular interaction. A method to repair ontology alignments automatically from the interaction experience could be extracted from these ideas. More generally, the development of matching techniques that create alignments to be used in agent interactions is an unexplored area, and needs both theoretical and practical development. To conclude, in our approach all the reasoning about the alignment is done locally; we think the performance of our methods would improve with a framework of negotiation in which interlocutors could discuss the situation of the alignment.

ACKNOWLEDGEMENTS

This research has been funded by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 567652 /ESSENCE: Evolution of Shared Semantics in Computational Environments/.

References

- [1] Alexander Artikis, 'Dynamic protocols for open agent systems', in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems Volume 1*, AAMAS '09, pp. 97–104, Richland, SC, (2009). International Foundation for Autonomous Agents and Multiagent Systems.
- [2] Manuel Atencia, *Semantic alignment in the context of agent interaction*, Doctoral dissertation, Universitat Autònoma de Barcelona, 2010.
- [3] Manuel Atencia and Marco Schorlemmer, 'An interaction-based approach to semantic alignment', *Journal of Web Semantics*, **12-13**, 131–147, (2012).
- [4] Samuel Barrett, Noa Agmon, Noam Hazon, Sarit Kraus, and Peter Stone, 'Communicating with unknown teammates', in *ECAI 2014 — 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic — Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, eds., Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pp. 45–50. IOS Press, (2014).
- [5] Paolo Bouquet, Jérôme Euzenat, Enrico Franconi, Luciano Serafini, Giorgos Stamou, and Sergio Tessaris, 'Specification of a common framework for characterizing alignment', Deliverable D2.2.1, Knowledge Web, (2004).
- [6] Amit K. Chopra and Munindar P. Singh, 'Multiagent commitment alignment', in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, AAMAS '09, pp. 937–944, Richland, SC, (2009). International Foundation for Autonomous Agents and Multiagent Systems.
- [7] Amit K. Chopra and Munindar P. Singh, 'Generalized commitment alignment', in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, pp. 453–461, Richland, SC, (2015). International Foundation for Autonomous Agents and Multiagent Systems.
- [8] Maxim Davidovsky, Vadim Ermolayev, and Vyacheslav Tolok, 'A survey on agent-based ontology alignment', in *ICAART 2012 — Proceedings of the 4th International Conference on Agents and Artificial Intelligence, Volume 2 - Agents, Vilamoura, Algarve, Portugal, 6-8 February, 2012*, eds., Joaquim Filipe and Ana L. N. Fred, pp. 355–361. SciTePress, (2012).
- [9] Anhai Doan and Alon Y Halevy, 'Semantic-Integration Research in the Database Community - A Brief Survey', *AI Magazine*, **26**(1), 83–94, (2005).
- [10] Zlatan Dragisic et al., 'Results of the ontology alignment evaluation initiative 2014', in *Proceedings of the 9th International Workshop on Ontology Matching collocated with the 13th International Semantic Web Conference (ISWC 2014), Riva del Garda, Trentino, Italy, October 20, 2014.*, pp. 61–104, (2014).
- [11] Jérôme Euzenat and Pavel Shvaiko, *Ontology matching*, Springer, 2007.
- [12] Claudia V. Goldman, Martin Allen, and Shlomo Zilberstein, 'Learning to communicate in a decentralized environment', *Autonomous Agents and Multi-Agent Systems*, **15**(1), 47–90, (August 2007).
- [13] Akin Günay, Michael Winikoff, and Pinar Yolum, 'Dynamically generated commitment protocols in open systems', *Autonomous Agents and Multi-Agent Systems*, **29**(2), 192–229, (March 2015).
- [14] Sandra Heiler, 'Semantic interoperability', *ACM Computing Surveys*, **27**(2), 271–273, (1995).
- [15] Ningsheng Jian, Wei Hu, Gong Cheng, and Yuzhong Qu, 'Falcon-AO: Aligning ontologies with falcon', in *Integrating Ontologies '05, Proceedings of the K-CAP 2005 Workshop on Integrating Ontologies, Banff, Canada, October 2, 2005*, volume 156 of *CEUR Workshop Proceedings*, pp. 85–91, (2005).
- [16] Loredana Laera, Ian Blacoe, Valentina A. M. Tamma, Terry R. Payne, Jérôme Euzenat, and Trevor J. M. Bench-Capon, 'Argumentation over ontology correspondences in MAS', in *6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007), Honolulu, Hawaii, USA, May 14-18, 2007*, eds., Edmund H. Durfee, Makoto Yokoo, Michael N. Huhns, and Onn Shehory, p. 228, (2007).
- [17] Maxime Morge and Jean-Christophe Routier, 'Debating over heterogeneous ontologies', *Applied Ontology*, **2**, 333–349, (2007).
- [18] Nuno Silva, Gecad Isep Ipp, and Paulo Maio, 'An approach to ontology mapping negotiation', in *Proceedings of the Workshop on Integrating Ontologies*, pp. 54–60, (2005).
- [19] Luc Steels, 'The Origins of Ontologies and Communication Conventions in Multi-Agent Systems', *Autonomous Agents and MultiAgent Systems*, **1**, 169–194, (1998).
- [20] R.S. Sutton and A.G. Barto, *Reinforcement learning: An introduction*, volume 116, Cambridge Univ Press, 1998.
- [21] Katia Sycara, Massimo Paolucci, Anupriya Ankolekar, and Naveen Srinivasan, 'Automated discovery, interaction and composition of semantic web services', *Web Semantics: Science, Services and Agents on the World Wide Web*, **1**(1), 27–46, (2003).
- [22] Jurriaan van Diggelen, Robbert-Jan Beun, Frank Dignum, Rogier M. van Eijk, and John-Jules Ch. Meyer, 'Ontology negotiation: goals, requirements and implementation', *International Journal of Agent-Oriented Software Engineering*, **1**(1), 63, (2007).
- [23] Michael Wooldridge, *An Introduction to MultiAgent Systems*, Wiley Publishing, 2nd edn., 2009.
- [24] Zhibiao Wu and Martha Stone Palmer, 'Verb semantics and lexical selection', in *32nd Annual Meeting of the Association for Computational Linguistics, 27-30 June 1994, New Mexico State University, Las Cruces, New Mexico, USA, Proceedings.*, ed., James Pustejovsky, pp. 133–138, (1994).

On the Construction of High-Dimensional Simple Games

Martin Olsen¹ and Sascha Kurz² and Xavier Molinero^{3,4}

Abstract. Voting is a commonly applied method for the aggregation of the preferences of multiple agents into a joint decision. If preferences are binary, i.e., “yes” and “no”, every voting system can be described by a (monotone) Boolean function $\chi: \{0, 1\}^n \rightarrow \{0, 1\}$. However, its naive encoding needs 2^n bits. The subclass of threshold functions, which is sufficient for homogeneous agents, allows a more succinct representation using n weights and one threshold. For heterogeneous agents, one can represent χ as an intersection of k threshold functions. Taylor and Zwicker have constructed a sequence of examples requiring $k \geq 2^{\frac{n}{2}-1}$ and provided a construction guaranteeing $k \leq \binom{n}{\lfloor n/2 \rfloor} \in 2^{n-o(n)}$. The magnitude of the worst-case situation was thought to be determined by Elkind et al. in 2008, but the analysis unfortunately turned out to be wrong. Here we uncover a relation to coding theory that allows the determination of the minimum number k for a subclass of voting systems. As an application, we give a construction for $k \geq 2^{n-o(n)}$, i.e., there is no gain from a representation complexity point of view.

1 Introduction

Consider a set $N = \{1, \dots, n\}$ of agents whose binary preferences should be aggregated to a group decision. We assume that voting is used as aggregation method, i.e., each agent can say either “yes” or “no”, which we encode by 1 and 0, respectively, to a given proposal. The group decision is then an “accept” (1) or “reject” (0). Formally, the used voting system can be modeled as a Boolean function $\chi: \{0, 1\}^n \rightarrow \{0, 1\}$. By imposing some, quite natural, additional constraints, we obtain the class of so-called *simple games*, see Subsection 2.2. They are widely applied and very useful tools for understanding decision making in political and other contexts. One major drawback is that they do not admit an obvious succinct representation. The naive approach, listing the function values of χ , needs 2^n bits. Listing so-called *minimal winning coalitions*, see Subsection 2.2, also needs $2^{n-o(n)}$ items in the worst case.

However, the subclass of threshold functions of monotone Boolean functions or *weighted games* of simple games, see Subsection 2.2, can be represented by just n integer weights w_i , for $i \in \{1, \dots, n\}$, and an integer threshold/quota q . If a representation as a threshold function/weighted games is possible, at most $O(n \log n)$ bits are needed for each integer [16]. In the case of homogeneous agents or players, May’s theorem [15] states that we can choose $w_i = 1$ and $1 \leq q \leq n$, i.e., a very succinct representation is possible. In the case of heterogeneous agents or players there are unfortunately simple games which are not weighted games if

$n \geq 4$. However, Taylor and Zwicker have constructively shown that each simple game can be represented as the intersection of at most $\binom{n}{\lfloor n/2 \rfloor} \in 2^{n-o(n)}$ weighted games, where the weights are either 0 or 1 and the quotas are 1, see Subsection 2.2. The smallest number k such that a given simple game Γ can be represented as the intersection of k weighted games is called the *dimension* of Γ . From a representation complexity point of view, we have the following important questions:

How large can the dimension of a simple game be?

And how can the corresponding weighted games be constructed?

1.1 Related Work

With respect to the first question, Taylor and Zwicker provided a sequence of examples requiring at least $k \geq 2^{\frac{n}{2}-1}$ weighted games [17]. So, there is a large gap in the knowledge of the magnitude of the worst-case situation, which was thought to be closed by Elkind et al. in 2008, see [4]. Unfortunately, their analysis is flawed, which we will demonstrate in Section 3.

Taylor and Zwicker made the observation that although there are simple games with arbitrarily large dimension, they do not seem to be used in real-world voting systems. At the time of writing [17], the authors were only aware of practical voting systems with a dimension of at least 2. Classical examples of dimension 2 are given by the Amendment of the Canadian constitution [11] and the US federal legislative system [18]. The voting systems of the Legislative Council of Hong Kong and the Council of the European Union under its Treaty of Nice rules have a dimension of exactly three, which was proven in [3, 7], respectively. Quite recently, it has been shown that the voting system of the Council of the European Union under its Treaty of Lisbon rules has a dimension between 7 and 13 368 [12]. There, the authors also address the second question by providing heuristic algorithms based on integer linear programming. Besides that, the probably first published general approach for the determination of lower bounds for the dimension of a simple game is presented.

Instead of the intersection, each simple game can also be represented as a finite union of weighted games, which leads to the notion of *co-dimension*, see e.g. [8]. Allowing arbitrary combinations of unions and intersections results in the concept of a *Boolean dimension*, which is introduced and studied in [5]. We remark that the voting system of the Council of the European Union under Lisbon rules has a Boolean dimension of exactly three [12]. For the subclass of so-called complete simple or linear games, the dimension was studied in [9].

1.2 Our Contribution

We show up a link between the dimension of simple games and coding theory. More precisely, we give a construction of a simple game

¹ BTECH, Aarhus University, Denmark, email: martino@btech.au.dk

² University of Bayreuth, Germany, email: sascha.kurz@uni-bayreuth.de

³ Univers. Politècnica de Catalunya, Spain, email: xavier.molinero@upc.edu

⁴ Research partially funded by Grant MTM 2015-66818-P from the Spanish Economy and Competitiveness Ministry.

from an error-correcting code, including the determination of the corresponding exact dimension. Using results on error-correcting codes, we can conclude the existence of simple games whose dimension asymptotically matches the worst-case upper bound $2^{n-o(n)}$ of Taylor and Zwicker [17], i.e., we close the gap in the literature that was previously filled by the flawed result of Elkind et al. [4].

We thoroughly discuss the lower bound construction of Taylor and Zwicker, i.e. we determine the corresponding exact dimension. Curiously enough, just the integer weights 0, 1, and 2 are needed for the used weighted games. It turns out that Elkind et al. considered an isomorphic variant of the example of Taylor and Zwicker.

1.3 Outline

The remaining part of this paper is structured as follows:

In Section 2, we introduce some notation and formally define the considered concepts in the paper. We also state a well-known theoretical upper bound for the dimension. Section 3 shows that the example given by Elkind et al. [4, Theorem 8] is an isomorphic variant of the example given by Taylor and Zwicker [17, Theorem 1.7.5]. The games that form the basis for our results are introduced in Section 4. Finally, Section 5 contains the proofs of high dimension and a theorem that forms the main contribution of the paper.

2 Preliminaries

We will start by briefly introducing error-correcting codes for readers not familiar with coding theory, see e.g. [2] for a more comprehensive introduction. In the second part of this section, we list the basic notation and definitions of simple games and their dimension. Here we refer the interested reader to [17].

2.1 Error-Correcting Codes

The *Hamming weight* $hw(x)$ of a bit vector $x = x_1x_2\dots x_n \in \{0, 1\}^n$ is the number of 1-bits in x : $hw(x) = |\{i : x_i = 1\}|$. The *Hamming distance* $d(x, y)$ between two bit vectors x and y is the number of bit positions, where the bits in x and y are different: $d(x, y) = |\{i : x_i \neq y_i\}|$.

Imagine a situation in which a 4 bit message has to be transmitted from a sender to a receiver in a noisy environment, where bits are risking to be flipped during the transmission. By adding extra bits to the message in a clever way, we can recover the original message if a few bits are flipped. One way of doing this is by using the well-known Hamming[8,4] code, where 4 bits are added as illustrated by the following example:

Example 1. *The Hamming[8,4] code is essentially the following set \mathcal{H} of bit vectors:*

$$\begin{aligned} \mathcal{H} = \{ & 0000\ 0000, 0001\ 1110, 0010\ 0111, 0011\ 1001, \\ & 0100\ 1011, 0101\ 0101, 0110\ 1100, 0111\ 0010, \\ & 1000\ 1101, 1001\ 0011, 1010\ 1010, 1011\ 0100, \\ & 1100\ 0110, 1101\ 1000, 1110\ 0001, 1111\ 1111 \} \end{aligned}$$

The set \mathcal{H} contains 16 vectors – one vector for each possible 4 bit message, where the message is the first 4 bits of a vector. The 4 extra bits make it possible to recover a message when bits are flipped.

The Hamming distance between any two vectors in \mathcal{H} is at least 4. This means that we can recover a message if one bit is flipped by locating the only vector in \mathcal{H} with Hamming distance 1 to the received

message. If two bits are flipped, we can only detect that something bad has happened. This is a so-called single-error correcting and double-error detecting code – a SECDED code.

Let $\mathcal{C}_8 = \mathcal{H} \setminus \{0000\ 0000, 1111\ 1111\}$ denote the subset of \mathcal{H} consisting of the 14 bit vectors with Hamming weight 4. The code \mathcal{C}_8 is referred to as a constant weight code, since all the members of \mathcal{C}_8 has the same Hamming weight. We will refer to \mathcal{C}_8 several places in the paper.

2.2 Simple Games and their Dimension

A simple game $\Gamma = (N, W)$ is a pair where $N = \{1, \dots, n\}$, for some positive integer n , denotes the set of players or agents and W is a collection of subsets of N , i.e., $W \subseteq 2^N$, satisfying the following conditions:

- (1) $\emptyset \notin W$;
- (2) $N \in W$;
- (3) $S \subseteq T \subseteq N$ and $S \in W$ implies $T \in W$.

A *coalition* S is a subset of N . If $S \in W$, then it is called winning; otherwise, it is said to be losing.

The relation to a Boolean function $\chi: \{0, 1\}^n \rightarrow \{0, 1\}$ is given as follows: Let S be the set of coordinates of the input vector x that are equal to 1, i.e., all players that vote “yes”. The players in $N \setminus S$ vote “no”. If $\chi(x) = 1$, then S is winning; otherwise, it is losing.

Conditions (1) and (2) ensure that the group decision does not contradict the individual preferences in the case of unanimity. The monotonicity condition (3) models the assumption that an enlarged set of supporters should not turn the group decision from an acceptance into rejection, which is quite reasonable. So a simple game Γ corresponds to a monotone Boolean function χ with the extra conditions $\chi(\mathbf{0}) = 0$ and $\chi(\mathbf{1}) = 1$.

Clearly, a simple game Γ is uniquely characterized by either its set W of winning or its set L of losing coalitions, which may both be as large as $2^n - 1$ in general. A first reduction is possible: A coalition S is called *minimal winning* if it is winning and all of its proper subsets are losing. Similarly, a coalition T is called *maximal losing* if it is losing and all of its proper supersets are winning. The family consisting of all minimal winning coalitions is denoted by W^m and the family of all maximal losing coalitions is denoted by L^M . Since no minimal winning coalition is a proper subset of another minimal winning coalition, we can apply Sperner’s Lemma, see e.g. [13], to conclude $|W^m| \leq \binom{n}{\lfloor n/2 \rfloor}$. Similarly, we conclude $|L^M| \leq \binom{n}{\lfloor n/2 \rfloor}$.

A simple game $\Gamma = (N, W)$ is weighted if there exists a *quota* $q \in \mathbb{R}_{>0}$ and *weights* $w_1, w_2, \dots, w_n \in \mathbb{R}_{\geq 0}$ such that $S \in W$ if and only if $\sum_{i \in S} w_i \geq q$. We remark that one can require the weights and the quota to be non-negative integers [6]. The intersection $(N, W_1) \cap (N, W_2)$ of two simple games is the simple game $(N, W_1 \cap W_2)$. Taylor and Zwicker [17] have shown that any simple game can be written as the intersection of $|L^M|$ weighted games Γ_T , $T \in L^M$, where a coalition S wins in Γ_T if $S \cap (N \setminus T) \neq \emptyset$. A weighted representation using weights 0 and 1 is given as follows: A player in $N \setminus T$ has weight 1 and all other players have weight 0 in the game Γ_T that has quota 1.

The dimension d of a simple game Γ is the smallest positive integer such that $\Gamma = \bigcap_{i=1}^d \Gamma_d$, where the games Γ_i , $i \in \{1, 2, \dots, d\}$, are weighted. From the previous considerations we conclude

$$d \leq |L^M| \leq \min \left(2^n - |W|, \binom{n}{\lfloor \frac{n}{2} \rfloor} \right). \quad (1)$$

To give an intuition of how this upper bound relates to 2^n , we can use the the following double inequality that holds for all even positive integers n [14]:

$$\sqrt{\frac{2}{\pi n}} \left(1 - \frac{1}{4n}\right) 2^n \leq \binom{n}{\frac{n}{2}} \leq \sqrt{\frac{2}{\pi n}} \left(1 - \frac{2}{9n}\right) 2^n. \quad (2)$$

For all odd positive integers n , we can use the equality $\binom{n}{\lfloor \frac{n}{2} \rfloor} = \binom{n-1}{\frac{n-1}{2}} \frac{2n}{n+1}$ and obtain the following inequalities:

$$\binom{n}{\lfloor \frac{n}{2} \rfloor} \geq \frac{n}{n+1} \sqrt{\frac{2}{\pi(n-1)}} \left(1 - \frac{1}{4(n-1)}\right) 2^n, \quad (3)$$

$$\binom{n}{\lfloor \frac{n}{2} \rfloor} \leq \frac{n}{n+1} \sqrt{\frac{2}{\pi(n-1)}} \left(1 - \frac{2}{9(n-1)}\right) 2^n. \quad (4)$$

For a bit vector $x = x_1 x_2 \dots x_n \in \{0, 1\}^n$ with n bits, we let S_x be the coalition where $i \in S$ if and only $x_i = 1$. For a coalition $S \subseteq N$, we define the bit vector x_S accordingly. We use the notation \bar{x} and \bar{S} for complements for bit vectors and sets, respectively.

3 The Example of Taylor and Zwicker

Let us reconsider the construction of a simple game with large dimension from [17, Theorem 1.7.5]. To this end, let k be an odd integer, $S = \{1, \dots, k\}$, $T = \{k+1, \dots, 2k\}$, and $N = S \cup T$. A coalition $X \subseteq N$ is winning iff either $|X| \geq k+1$ or $|X| = k$ and $|X \cap T| \equiv 0 \pmod{2}$. Denote the corresponding simple game by Γ_k . The minimal winning coalitions of Γ_k are given by $W^m =$

$$\{X_1 \cup X_2 \mid X_1 \subseteq S, X_2 \subseteq T, |X_2| \equiv 0 \pmod{2}, |X_1 \cup X_2| = k\}$$

and the maximal losing coalitions of Γ_k are given by $L^M =$

$$\{X_1 \cup X_2 \mid X_1 \subseteq S, X_2 \subseteq T, |X_2| \equiv 1 \pmod{2}, |X_1 \cup X_2| = k\}.$$

Since $k \equiv 1 \pmod{2}$, we have $n \equiv 2 \pmod{4}$ for $n = 2k = |N|$ and $|W^m| = |L^M| = \frac{1}{2} \cdot \binom{n}{n/2}$, so that the dimension of Γ_k is at most $\frac{1}{2} \cdot \binom{n}{n/2}$. We remark that Γ_k is self-dual, so that its dimension equals its co-dimension.

Theorem 1. *For each odd integer k , the dimension of Γ_k is given by 2^{k-1} .*

Proof. Let $\mathcal{C} = \{x\bar{x} : x \in \{0, 1\}^k, \sum_{i=1}^k x_i \equiv 0 \pmod{2}\}$, where \bar{x} denotes the negation of a binary vector and xy denotes the concatenation of two binary vectors x and y . We have $\mathcal{C} = L^M$, $|\mathcal{C}| = 2^{k-1}$, and we remark that the minimum Hamming distance of \mathcal{C} is 4 for $k > 1$.

For the lower bound on the dimension, we refer to [17, Theorem 1.7.5].⁵

For the other direction set $\mathcal{C}^P = \{x \in \{0, 1\}^k : \sum_{i=1}^k x_i \equiv 0 \pmod{2}\}$. Since $2^{1-1} = \frac{1}{2} \cdot \binom{2}{1}$, we can assume $k \geq 3$. We set $v = \bigcap_{x \in \mathcal{C}^P} v_x$, where $v_x = [q^x; w_1^x, \dots, w_{2k}^x]$ with

- $w_i^x = \begin{cases} 0 & : x_i = 1, \\ 2 & : x_i = 0 \end{cases}$ for all $1 \leq i \leq k$, $w_j^x = 1$ for all $k+1 \leq j \leq 2k$, and $q^x = k - (hw(x) - 1)$ if $x \neq \mathbf{0}$;
- $w_i^x = 1$ for all $1 \leq i \leq k$, $w_j^x = 0$ for all $k+1 \leq j \leq 2k$, and $q^x = 1$ if $x = \mathbf{0}$.

Let $S_1 \subseteq N$ with $|S_1| \geq k+1$. For each $x \in \mathcal{C}^P \setminus \{\mathbf{0}\}$, we have $w^x(S_1) \geq hw(x) \cdot 0 + (k+1 - hw(x)) \cdot 1 = q^x$. Since $|S_1 \cap S| \geq 1$, we additionally have $w^0(S_1) \geq 1 = q^0$, so that S_1 is winning in v . Now let S_2 be a coalition with $|S_2| = k$ and $|S_2 \cap S| \equiv 0 \pmod{2}$. If x is the characteristic vector of $S_2 \cap S$, then

- $w^x(S_2) = hw(x) \cdot 0 + (k - hw(x)) \cdot 1 = k - hw(x) < q^x$ for $x \neq \mathbf{0}$;
- $w^x(S_2) = 0 < 1 = q^x$ for $x = \mathbf{0}$,

so that S_2 is a losing coalition in v . Let S_3 be a coalition with $|S_3| = k$ and $|S_3 \cap S| \equiv 1 \pmod{2}$. Since $|S_3 \cap S| \geq 1$ we have $w^0(S_3) \geq 1 = q^0$. Now let $x \in \mathcal{C}^P \setminus \{\mathbf{0}\}$ be arbitrary. If $|S_3 \cap S| < hw(x)$, then we have $w^x(S_3) \geq (hw(x) - 1) \cdot 0 + (k - hw(x) + 1) \cdot 1 = q^x$. If $|S_3 \cap S| > hw(x)$, then there exists a player $i \in S_3 \cap S$ with $w_i^x = 2$, so that $w^x(S_3) \geq hw(x) \cdot 0 + 2 + (k - hw(x) - 1) \cdot 1 = q^x$. Thus, S_3 is winning in v . Finally, let S_4 be a coalition of cardinality $k-1$. Since $k-1$ is even, we have the following two cases:

- $|S_4 \cap S| \equiv 0 \pmod{2}, |S_4 \cap T| \equiv 0 \pmod{2}$,
- $|S_4 \cap S| \equiv 1 \pmod{2}, |S_4 \cap T| \equiv 1 \pmod{2}$.

In both cases, it is possible to extend S_4 to a coalition $S_5 \in \mathcal{L}^M$ by adding a player, so that S_4 has to be losing in v . Thus, we have $v \in \Gamma_k$ and $\dim(\Gamma_k) \leq 2^{k-1}$. \square

Now let us restate the example of [4, Theorem 8]: Let k be an odd integer and $n = 2k$, $N = \{1, \dots, n\}$. Consider the simple game where all coalitions of cardinality larger than k are winning and all coalitions of cardinality smaller than k are losing. A coalition X of cardinality k is winning iff the Hamming distance between X and $\{1, \dots, k\}$ is equivalent to 2 modulo 4. In other words, this means that $|X \cap \{1, \dots, k\}|$ is even and $|X \cap \{k+1, \dots, n\}|$ is odd.

Interchanging the first k players with the last k players yields the example of Taylor and Zwicker. Since Theorem 8 in [4] claims that the dimension is at least $\binom{2k}{k}/2$, there is a contradiction to Theorem 1. The flaw⁶ of the corresponding proof happens where it says that if x is the bit vector of a losing coalition and $x_i \neq x_j$, then switching x_i and x_j results in a bit vector of a winning coalition. An explicit counter example for $n = 6$ is given by the characteristic vectors 100110 and 010110 which both represent losing coalitions.

4 From Error Correcting Codes to Simple Games

In this section, we present a generic recipe for constructing the simple games forming the basis for our results. Throughout the paper, we let $\mathcal{C} \subseteq \{0, 1\}^n$ denote a set of bit vectors of length n having positive Hamming weight satisfying this condition:

$$\forall x \neq y \in \mathcal{C} : |hw(x) - hw(y)| < d(x, y) - 2 \quad (5)$$

For $x \in \mathcal{C}$, we define the simple game Γ_x with players $N = \{1, 2, \dots, n\}$ as follows: S wins in Γ_x if and only if $S \cap S_x \neq \emptyset$. The simple game $\Gamma_{\mathcal{C}}$ is now defined by $\Gamma_{\mathcal{C}} = \bigcap_{x \in \mathcal{C}} \Gamma_x$. In other

⁵ Using the general approach and notation of [12] we can state a quick proof: For each $x, y \in \mathcal{C}$ with $x \neq y$ there exist indices $1 \leq i \leq k, k+1 \leq j \leq 2k$ with $x_i \neq y_i, x_j \neq y_j$, and $x_i \neq x_j$. Negating x_i, x_j, y_i , and y_j gives two winning vectors x', y' with $x + y = x' + y'$, i.e., we have determined a 2-trade, so that the dimension is at least 2^{k-1} .

⁶ We would like to thank Edith Elkind for directly pointing to the position where the proof breaks down in a private communication.

words, a set S is winning if and only if S is a so-called *hitting set* for the collection of sets $\{S_x\}_{x \in \mathcal{C}}$.

The error-correcting code \mathcal{C}_8 from Example 1 is a set of bit vectors satisfying (5). Another example is the following:

Example 2. Let \mathcal{C} be defined as follows for $n = 8$:

$$\mathcal{C} = \{0000\ 1111, 1100\ 0000, 0011\ 1100\}$$

The Hamming weights of the vectors 0000 1111 and 1100 0000 differ by 2 but their Hamming distance is 6. So (5) holds for these vectors. Coalition $\{1, 5\}$ is winning in $\Gamma_{\mathcal{C}}$, since it intersects the sets $\{5, 6, 7, 8\}$, $\{1, 2\}$ and $\{3, 4, 5, 6\}$. The bit vector 1000 1000 that corresponds to the set $\{1, 5\}$ shares at least one 1-bit with all members of \mathcal{C} .

4.1 A Dimension Lemma

We now prove a lemma explicitly stating the dimension of our games.

Lemma 1. The dimension of $\Gamma_{\mathcal{C}}$ is $|\mathcal{C}|$.

Proof. The game Γ_x , $x \in \mathcal{C}$, is clearly weighted, so the dimension of $\Gamma_{\mathcal{C}}$ is not higher than $|\mathcal{C}|$.

We now assume that the dimension of $\Gamma_{\mathcal{C}}$ is less than $|\mathcal{C}|$. Let $L_x = N \setminus S_x$ for $x \in \mathcal{C}$. The coalition L_x is clearly a losing coalition in $\Gamma_{\mathcal{C}}$ because $L_x \cap S_x = \emptyset$. Using the pigeonhole principle, we conclude that there are $x, y \in \mathcal{C}$ with $x \neq y$ such that L_x and L_y lose in the same weighted game Γ' , where Γ' is one of the less than $|\mathcal{C}|$ weighted games whose intersection is $\Gamma_{\mathcal{C}}$.

By considering basic properties for the Hamming distance and the Hamming weight, we observe that (5) also holds if we replace x and y with their complements \bar{x} and \bar{y} . If one of the vectors \bar{x} or \bar{y} had all 1-bits in the $d(\bar{x}, \bar{y})$ positions, where the two vectors differ, then the left-hand side of (5) would be $d(\bar{x}, \bar{y})$ and (5) would not hold. We therefore conclude that there are players $p_x \in L_x \setminus L_y$ and $p_y \in L_y \setminus L_x$. We let A and B be the coalitions obtained if L_x and L_y swap these players: $A = (L_x \setminus \{p_x\}) \cup \{p_y\}$ and $B = (L_y \setminus \{p_y\}) \cup \{p_x\}$.

We now show that A and B are winning coalitions in $\Gamma_{\mathcal{C}}$. Without loss of generality, we consider the coalition A . It is clear that x_A and x share a 1-bit so A wins in Γ_x . Now let us assume that there is a member z of $\mathcal{C} \setminus \{x\}$ such that A loses in Γ_z . In other words, x_A and z do not share a 1-bit. The vector x_A is obtained by flipping a 0-bit and a 1-bit in the vector \bar{x} :

$$d(x_A, \bar{z}) \geq d(\bar{x}, \bar{z}) - 2. \tag{6}$$

The $d(x_A, \bar{z})$ bits shared by x_A and z are all 0 in which case we have the following:

$$d(x_A, \bar{z}) + hw(x_A) + hw(z) = n. \tag{7}$$

We now use $hw(x_A) = n - hw(x)$ together with (7):

$$d(x_A, \bar{z}) = hw(x) - hw(z). \tag{8}$$

By using $d(x, z) = d(\bar{x}, \bar{z})$ and (6) and (8), we obtain the following inequality:

$$hw(x) - hw(z) \geq d(x, z) - 2. \tag{9}$$

Since (9) contradicts (5), we conclude that A wins in Γ_z for any $z \in \mathcal{C}$. Consequently, A also wins in $\Gamma_{\mathcal{C}}$.

Summing up, we now have two coalitions L_x and L_y that lose in Γ' , and we can obtain two winning coalitions in $\Gamma_{\mathcal{C}}$ if L_x and L_y swap two players. These coalitions also win in Γ' and we obtain a contradiction, since this would mean that the total weight in Γ' of the players in L_x and L_y has increased. \square

It is worth noting that the dimension of the game $\Gamma_{\mathcal{C}}$ is $|\mathcal{C}|$ since $L^M = \{L_x\}_{x \in \mathcal{C}}$.

If we can construct games with dimension m using our approach, we can also construct games with dimension m' for every $m' \leq m$ as expressed by the following corollary:

Corollary 1. Let $\Gamma_{\mathcal{C}}$ be a simple game with n players and dimension m , then there are simple games with n players and dimension m' , $1 \leq m' \leq m$.

Proof. Just delete some elements from \mathcal{C} . \square

5 Simple Games with High Dimension

The key question we will deal with in this section is the following: Can we find families \mathcal{C} of bit vectors with high cardinality satisfying (5)? According to Lemma 1, this would automatically give us games with high dimension. From the theory on error-correcting codes, we know how to construct relatively large families of bit vectors forming SECDED constant weight codes. If we pick such a code, we clearly have a family \mathcal{C} satisfying (5). This observation is the basis for the proofs in this section. As an example, the code \mathcal{C}_8 from Example 1 corresponds to a simple game with 8 players and dimension 14.

It is important to stress that constant weight SECDED codes are not the only families satisfying the generic recipe (5) as illustrated by Example 2. There are many other families that satisfy (5), but we will use constant weight SECDED codes to construct our games with high dimension. In other words, there might be families with larger cardinalities compared to constant weight SECDED codes satisfying (5).

Agrell et al. [1] present lower bounds for cardinalities of constant weight SECDED codes. These lower bounds can be directly translated to lower bounds for dimensions for simple games if we use Lemma 1. This allows us to set up Table 1 that compares the dimensions of the games produced using composition of unanimity games [9] with the dimensions of the games based on our approach and the lower bounds from [1]. The first column displays n . The second column presents the dimensions of the games from [9] and [17]. The third column contains the dimensions of the games produced using our approach and constant weight SECDED codes. Finally, the last column shows the, slightly improved, upper bound $\binom{n}{\lfloor n/2 \rfloor} - 1$.⁷ As an example, we can see that our approach leads to a simple game with dimension 14 for $n = 8$ – the game $\Gamma_{\mathcal{C}_8}$.

We are now ready to consider all other values of n . Initially, we consider the case where n is a power of 2. The following lemma generalizes the example described earlier with $|\mathcal{C}_8| = 14$ for $n = 8$ to $n = 2^m$ for $m \geq 3$.

Lemma 2. Let $n = 2^m$ where m is an integer, $m \geq 3$. There is a set of bit vectors $\mathcal{C} \subseteq \{0, 1\}^n$ satisfying (5) with

$$|\mathcal{C}| = \frac{2}{n} \left(\frac{1}{2} \binom{n}{\frac{n}{2}} + (n-1) \binom{\frac{n}{2}-1}{\frac{n}{4}} \right). \tag{10}$$

Proof. Let $t = 2^m - 1$. The enumerator polynomial for an error-correcting code is a polynomial, where the i 'th coefficient, a_i , is the number of bit vectors of Hamming weight i . According to [19], the

⁷ Sperner's Theorem also classifies the cases where his bound is tight. Since all of the corresponding simple games are indeed weighted, the previous upper bound can be reduced by 1.

Table 1: A comparison of the dimensions of the games produced using composition of unanimity games and the dimensions of the games based on our approach.

n	Unanimity games	Our approach	$\binom{n}{\lfloor n/2 \rfloor} - 1$
6	4	4	19
7	4	7	34
8	8	14	69
9	9	18	125
10	16	36	251
11	18	66	461
12	32	132	923
13	36	166	1715
14	64	325	3431
15	81	585	6434
16	128	1170	12869
17	162	1770	24309
18	256	3540	48619
19	324	6726	92377
20	512	13452	184755

enumerator polynomial for the well-known Hamming $[t, t - m]$ code that contains bit vectors of length t is:

$$A(x) = \frac{(1+x)^t + t(1-x)(1-x^2)^{(t-1)/2}}{t+1}.$$

Let $i = \frac{t-1}{2}$ ($i = 2^{m-1} - 1$ is odd and $i + 1$ is even):

$$a_i = \frac{1}{t+1} \left(\binom{t}{i} + t(-1)^{\frac{i+1}{2}} \binom{i}{\frac{i-1}{2}} \right)$$

$$a_{i+1} = \frac{1}{t+1} \left(\binom{t}{i+1} + t(-1)^{\frac{i+1}{2}} \binom{i}{\frac{i+1}{2}} \right) = a_i$$

The extended code Hamming $[t + 1, t - m]$ is a SECEDED code. We can now let \mathcal{C} be the subset of the extended code containing the bit vectors with Hamming weight $\frac{n}{2}$. This is a constant weight SECEDED code satisfying (5).

Set $n = t + 1 = 2i + 2$. The number of bit vectors in the extended code with Hamming weight $\frac{n}{2} = i + 1$ is $a_i + a_{i+1} = 2a_{i+1}$:

$$2a_{i+1} = \frac{2}{n} \left(\binom{n-1}{\frac{n}{2}} + (n-1)(-1)^{\frac{n}{4}} \binom{\frac{n}{2}-1}{\frac{n}{4}} \right)$$

For $n \geq 8$, we have:

$$2a_{i+1} = \frac{2}{n} \left(\binom{n-1}{\frac{n}{2}} + (n-1) \binom{\frac{n}{2}-1}{\frac{n}{4}} \right).$$

We now use:

$$\binom{n}{\frac{n}{2}} = \binom{n-1}{\frac{n}{2}} + \binom{n-1}{\frac{n}{2}-1} = 2 \binom{n-1}{\frac{n}{2}}$$

to obtain

$$2a_{i+1} = \frac{2}{n} \left(\frac{1}{2} \binom{n}{\frac{n}{2}} + (n-1) \binom{\frac{n}{2}-1}{\frac{n}{4}} \right).$$

□

We now state our main theorem, where we also consider values of n that are not powers of 2.

Theorem 2. For any positive integer n there is a simple game with n players and dimension d satisfying:

$$d \geq \frac{1}{n} \binom{n}{\lfloor \frac{n}{2} \rfloor}. \tag{11}$$

If $n = 2^m$ for an integer $m \geq 3$, then there is a simple game with n players and dimension d such that

$$d = \frac{1}{n} \binom{n}{\frac{n}{2}} + \frac{2(n-1)}{n} \binom{\frac{n}{2}-1}{\frac{n}{4}}. \tag{12}$$

Proof. Graham and Sloane [10] have shown that there is constant weight SECEDED code with Hamming weight w with cardinality at least $\frac{1}{n} \binom{n}{w}$ for any w . For $w = \lfloor \frac{n}{2} \rfloor$, we get (11) by using Lemma 1. Lemma 1 and Lemma 2 give us (12). □

It follows from (2) and (3) that the lower bound presented in Theorem 2 is $2^{n-o(n)}$. Our games are easily seen to be within a factor n from the upper bound from (1). Finally, we point out that the proof of the lower bound in [10] is constructive.

6 Conclusion

We have presented a link from coding theory to the dimension of simple games. We are not aware of any other connection between coding theory and simple games. While it seems a rather tough problem to determine the exact dimension of a simple game, we have provided an exact formula for those simple games arising from error correcting codes in Lemma 1. Via this connection, any improvement on lower bounds of constant weight codes improves the stated lower bounds for the worst-case dimensions of simple games. For the other direction, it would be interesting to know whether unrestricted codes satisfying inequality (5) have some application in coding theory. Till now, it is even unclear, at least for us, if those codes can be strictly larger than constant weight codes. From our point of view, this connection should be explored in more detail.

The asymptotic magnitude of the worst-case examples with respect to the dimension of simple games is determined, which closes a gap in the literature and uncovers a flaw in a previous attempt. The bad news is that from a representation complexity point of view, the usage of intersections of weighted games cannot be a solution for all cases. From a practical point of view, one may nevertheless ask whether the set of weighted games with *small* dimension are not *too far apart* from the set of simple games, so that there is no reason to use high-dimensional simple games in reality.

From a mathematical point of view, it would be interesting to determine the exact values of the worst-case examples.

The construction of the representing weighted games is still widely open and deserves further attention.

REFERENCES

- [1] E. Agrell, A. Vardy, and K. Zeger, ‘Upper bounds for constant-weight codes’, *IEEE Transactions on Information Theory*, **46**, 2373–2395, (2000).
- [2] E.R. Berlekamp, *Algebraic Coding Theory: Revised Edition*, World Scientific, 2015.
- [3] W.-S. Cheung and T.-W. Ng, ‘A three-dimensional voting system in Hong Kong’, *European Journal of Operational Research*, **236**(1), 292–297, (2014).
- [4] E. Elkind, L. A. Goldberg, P. W. Goldberg, and M. Wooldridge, ‘On the dimensionality of voting games’, in *Proc. 23rd AAAI Conference on Artificial Intelligence*, pp. 69–74. AAAI Press, (2008).

- [5] P. Faliszewski, E. Elkind, and M. Wooldridge, 'Boolean combinations of weighted voting games', in *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems*, pp. 185–192. International Foundation for Autonomous Agents and Multiagent Systems, (2009).
- [6] J. Freixas, 'Different ways to represent weighted majority games', *Top*, **5**(2), 201–211, (1997).
- [7] J. Freixas, 'The dimension for the European Union Council under the Nice rules', *European Journal of Operational Research*, **156**(2), 415–419, (2004).
- [8] J. Freixas and D. Marciniak, 'On the notion of dimension and codimension', in *Proceedings of the 3rd International Conference Game Theory and Management*, eds., Leon A. Petrosyan and Nikolay A. Zenkevich, pp. 67–81. The International Society of Dynamic Games (Russian Chapter), (2010).
- [9] J. Freixas and M.A. Puente, 'A note about games-composition dimension', *Discrete Applied Mathematics*, **113**(23), 265–273, (2001).
- [10] R.L. Graham and N.J.A. Sloane, 'Lower bounds for constant weight codes.', *IEEE Transactions on Information Theory*, **26**(1), 37–43, (1980).
- [11] D.M. Kilgour, 'A formal analysis of the amending formula of Canada's Constitution Act, 1982', *Canadian Journal of Political Science/Revue Canadienne de Science Politique*, **16**, 771–777, (1983).
- [12] S. Kurz and S. Napel, 'Dimension of the lisbon voting rules in the EU Council: a challenge and new world record', *Optimization Letters*, 1–12, (2015). to appear.
- [13] D. Lubell, 'A short proof of Sperner's lemma', *Journal of Combinatorial Theory*, **1**(2), 299–299, (1966).
- [14] Y.L. Luke, *The special functions and their approximations*, Academic Press New York, 1969.
- [15] K.O. May, 'A set of independent necessary and sufficient conditions for simple majority decision', *Econometrica*, 680–684, (1952).
- [16] S. Muroga, *Threshold logic and its applications*, New York etc.: Wiley-Interscience, a Division of John Wiley & Sons, Inc. XIV,478 p., 1971.
- [17] A.D. Taylor and W.S. Zwicker, *Simple games: desirability relations, trading, and pseudoweightings*, Princeton University Press, New Jersey, USA, 1999.
- [18] Alan Taylor and William Zwicker, 'Weighted voting, multicameral representation, and power', *Games and Economic Behavior*, **5**(1), 170–181, (1993).
- [19] S.B. Wicker, *Error Control Systems for Digital Communication and Storage*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1995.

A Dynamic Logic of Norm Change

Max Knobbout and Mehdi Dastani and John-Jules Meyer¹

Abstract. Norms are effective and flexible means to control and regulate the behaviour of autonomous systems. Adding norms to a system changes its specification which may in turn ensure desirable system properties. As of yet, there is no generally agreed formal methodology to represent and reason about the dynamics of norms and their impacts on system specifications. In this paper, we introduce various types of norms, such as state-based or action-based norms, and gradually develop a dynamic modal logic to characterize the dynamics of such norms in a formal way. The logic can be used to prove various properties of norm dynamics and their impacts on system specification. Moreover, we show that this logic is sound and complete.

1 Introduction

Norms are widely proposed as an effective and flexible means to control and regulate the behaviour of autonomous systems. Generally, norms specify the standards of behaviours such as which actions or states should be achieved or avoided. Adding norms to a system changes its specification which may in turn incentivize/inhibit specific behaviours and thereby ensure some desirable system level properties. For example, consider the norm “individuals entering a train station should have valid tickets” being introduced in a train station. The addition of this norm incentivize having a valid ticket before entering the train station and ensures that the train station is not getting unnecessarily crowded. We assume that the addition of norms incentivizes/inhibits behaviours by various enforcement means such as regimentation (e.g. by placing ports at the entrance gates of the train station) or by means of sanctions (e.g. by random inspection of individuals at the train station and issuing fines for those who has no valid ticket). In this paper, we ignore the issue of norm enforcement and focus on how norms update system specifications, i.e., which system states or actions are considered as good/bad after the system is updated with norms. A system to which a set of norms is added, i.e., a system that is governed by a set of norms, is referred to as a normative system [15, 14, 1].

A lot of research has focused on deciding (or proving) correctness of a normative system. A normative system, i.e., a system with a set of norms, is correct if the objectives of the system designer are satisfied after the norms have been added to the system [15, 2, 12]. As of yet, there is no generally agreed formal methodology to represent and reason about the norm change in normative systems. Such a methodology allows us to formally investigate the dynamics of various types of norms, such as state-based or action-based norms, in normative systems and their impact on the system specification. For example, the methodology enables us to reason about the introduction of the above-mentioned norm in a train station before and after

an individual has entered the train station, either with or without a valid ticket. It also enables us to reason about the impact of different order of norm change on system specifications, and the interaction between norm change and agents’ behaviours.

One may identify two possible methodological approaches when dealing with norm change. On the one hand we can identify the syntactic approach [7], where norm change is considered as an operation on the underlying “code” that constitutes the system. On the other hand we can identify the semantic approach [4, 13] which aims to look at norm change as an update of the model. This work falls in the second category, but is novel because (1) instead of providing just a semantic analysis we provide a new dynamic logic to represent and reason about norm change, and (2) we provide an accompanying (sound and complete) proof system for the logic.

The view we adopt is that normative systems can be modelled by pointed labelled transition systems, which show which facts become true under execution of which actions. Moreover, we assume that updating a system with a norm modifies the system specification and thereby its behaviour. The specific problem we address in this paper is how to represent and reason about these norm updates. We introduce new types of norms that are expressive enough to model existing norm types. We propose a new dynamic norm logic with norm update operations and an accompanying proof system to reason about norm updates in normative systems. Inspired by dynamic logic [11, 16], the effect of a norm update operation is an update of the normative system. The kind of updates (norms applied to normative systems) and its effects (normative systems that are aligned with the norm) are completely novel. The contribution of this work is significant because it paves the way for the development of formal tools that can be used to prove correctness of a normative system. From a practical point of view such tools are essential to investigate the interaction between norm change and the system behaviour, in particular, whether or when the addition of some norms satisfy some desired system properties.

In this paper, we first introduce a formal framework to model normative systems and various types of norms. We then explore norms of the ‘to-be’ variant, which may forbid (or permit) certain states to occur. We devise a dynamic logic with update operation for ‘to-be’ norms. We then move to norms of the ‘to-do’ variant, which may forbid (or permit) certain actions to occur. For this norm type, we devise a dynamic logic with update operation for ‘to-do’ norms. For the proposed dynamic logics we provide accompanying proof systems.

2 Framework

This section defines the models we use for normative systems, and presents the syntax and semantics of the logic we use to express properties of these systems. We consider a normative system as a labelled transition system that gives us for each possible execution of actions the facts which become true. This is a standard way to model the be-

¹ Utrecht University, The Netherlands, email: {M.Knobbout,M.M.Dastani,J.J.C.Meyer}@uu.nl

haviour of a system, except that we additionally assume the existence of a set of violation atoms in order to model what is forbidden and permitted. This idea comes from Anderson’s reduction [5], where bad state of affairs can be labelled by a violation atom. A state with a violation atom assigned to it is then interpreted as a forbidden state. We note that a violation atom can represent a fine, but it can also simply mean that something bad has happened. In this paper, we do not wish to model whether an action is produced by a single agent or a group of agents (e.g. Concurrent Game Structures). Because of this simplification, we do not have to represent the agents explicitly, and can just assume the existence of some action alphabet. In the remainder of this paper we use the term normative system to refer to such a structure.

Definition 1 A normative system N is a tuple $(Q, Act, \rightarrow, \Pi, V, \mu)$ such that:

- Q is a non-empty finite set of states from the system.
- Act is a finite set of (domain) actions.
- $\rightarrow \subseteq Q \times Act \times Q$ is a relation between states with actions, such that for all $q \in Q$ and $\alpha \in Act$ there exists exactly one $q' \in Q$ such that $(q, \alpha, q') \in \rightarrow$. Whenever $(q, \alpha, q') \in \rightarrow$, we write $q(\alpha)$ to denote q'
- Π is a finite set of atomic propositions.
- $V \subseteq \Pi$ is a finite set of atomic violation propositions.
- μ is a valuation function mapping a state $q \in Q$ to an element from $\mathcal{P}(\Pi)$.

A pointed normative system is a pair (N, q) such that N is a normative system, and $q \in Q$ a state from N . Given such a structure, we say that a state q is forbidden whenever there exists a violation $v \in V$ such that $v \in \mu(q)$. Similarly, we say that an action α is forbidden in state q whenever there exists a violation $v \in V$ such that $v \in \mu(q(\alpha))$. Note that this model assumes that actions are deterministic (e.g. each action leads to a unique next state) and are always enabled. To model that an action α has no effect on a state q , we can simply model it by assuming that $q(\alpha) = q$, i.e. action α leads to the same state.

The language of propositional logic with action modality, written in this paper as \mathcal{L}_0 , consists of formulas φ built by the following grammar, where $p \in \Pi$ and $\alpha \in Act$:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid Do(\alpha)\varphi$$

Along a pointed normative system (N, q) , we can evaluate formulas of \mathcal{L}_0 in the following way:

- $N, q \models p$ iff $p \in \mu(q)$
- $N, q \models \neg\varphi$ iff $N, q \not\models \varphi$
- $N, q \models \varphi_1 \vee \varphi_2$ iff $N, q \models \varphi_1$ or $N, q \models \varphi_2$.
- $N, q \models Do(\alpha)\varphi$ iff $N, q(\alpha) \models \varphi$

Given a pointed normative system (N, q) , we say that a sequence of actions $\alpha_1 \dots \alpha_n$ brings about φ if and only if $N, q \models Do(\alpha_1) \dots Do(\alpha_n)\varphi$. As is standard, we say that $N \models \varphi$ holds whenever for all $q \in Q$ it holds that $N, q \models \varphi$, and that $\models \varphi$ holds (alternatively, “ φ is valid”) whenever for all normative systems N we have $N \models \varphi$. We have the following result from modal logic [9].

Theorem 1 (Sound and complete axiomatization of \mathcal{L}_0) The logic \mathcal{L}_0 is soundly and completely axiomatized by modal system \mathbf{K} (i.e. the system consisting of all propositional tautologies together with the necessitation rule and the distributive axiom) with the following ‘Function’ axiom scheme:

$$Do(\alpha)\neg\varphi \leftrightarrow \neg Do(\alpha)\varphi$$

We refer to system \mathbf{K} with this additional axiom scheme as system $\mathbf{L0}$, and write $\vdash_{\mathbf{L0}} \varphi$ whenever φ is provable in $\mathbf{L0}$. For all $\varphi \in \mathcal{L}_0$ we thus have:

$$\models \varphi \Leftrightarrow \vdash_{\mathbf{L0}} \varphi$$

This axiom represents that the arrows of a normative system are functional, i.e. a state and an action completely and uniquely define the next state. We will now introduce an example of a normative system. This example concerns a train station where individuals can enter and leave. Throughout this paper we use this example as a running example to demonstrate how various norms can alter the behaviour of the system.

2.1 Running Example

Consider the normative system N_{station} in Figure 1. This simple

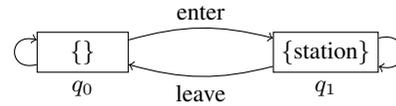


Figure 1. System N_{station} .

system models a train station where an individual/traveller can enter and leave. In particular, the system can either be in state q_0 or state q_1 , and depending on the performed action by the individual can switch between these states. Here ‘station’ denotes the fact that the individual is in the station. The reflexive arrows denote that the performance of all the remaining actions that can occur will result in the same state (e.g. action ‘enter’ in state q_1 will result in q_1). An example of a formula that holds in this system is:

$$N_{\text{station}}, q_1 \models Do(\text{leave})\neg\text{station}$$

Observe that there are no violations in this system yet. Later on we will add norms to this system and see how the system can show more interesting and complex behaviour.

2.2 Norm Types

Before we formally introduce the language of norms and norm update, it is worthwhile to reflect on the kinds of norms we want to express and model. Broadly speaking, we consider the following two classes:

1. **State-based norms** A state-based norm refers to certain states that should be achieved or avoided. These kinds of norms are of the ‘to-be’ variant. To extend the expressiveness of these kinds of norms even further, we also optionally allow the addition of a repair action. For example, the station may add a norm which states that being at the station is forbidden until a valid subscription is bought. These state-based norms are thus conditional on a repair action. This class of norms is, to our knowledge, new and allows for expressive norms that are required to model practical scenarios.
2. **Action-based norms** An action-based norm refers to certain actions that should be performed or avoided. These kinds of norms are of the ‘to-do’ variant. Again, we optionally allow the performance of a repair actions, which allows for expressive norms.

More elaborate norms pertaining to complex behaviours of the system can be acquired by combining norms in various ways. It is important to note that the addition of these norms are not ‘physical’ actions of the system. They come from outside the system (i.e. a designer) and are not triggered by the actions of the system.

3 Language for Norms and Norm Updates

In this section we begin our first step into developing our language for norms and norm updates. The kinds of norms we consider in this section are of the ‘to-be’ variant.

3.1 Language and Update

Given a normative system, we construct the norm language \mathcal{N}_1 in the following way, where $\varphi \in \mathcal{L}_0$, $v \in V$ and $Act_R \subseteq Act$:

$$n ::= (\varphi, +v, Act_R) | (\varphi, -v, Act_R)$$

Whenever we have a norm $(\varphi, \pm v, Act_R)$ (where $\pm v$ can either be $+v$ or $-v$), we refer to φ as the norm condition, $\pm v$ as the norm effect and Act_R as the set of actions from the system that will count as repair actions. These constructs relate to the kinds of norms we described earlier. Adding a norm $(\varphi, +v, Act_R)$ implies that for every φ -state the violation v will hold, until a repair action from the set Act_R occurs, at which point the behaviour of the system will revert back to what it was before the norm was added to the system. Whenever such a repair action occurs, we say that the norm effect is repaired. Note that we use the notion of repair in a rather liberal way, since a repair action may very well occur before a φ -state is ever encountered. In this case, we still say that a norm effect is repaired, even though it was never the case that we were in a state in which the norm effect $\pm v$ was realized. We note that the set Act_R can be empty, which implies that the norm is permanent. For example, the norm $(\varphi, +v, \emptyset)$ states that all φ -states are now permanently forbidden. In this way, we acquire the ‘classical’ interpretation of a state-based norm. The norm $(\varphi, -v, Act_R)$ behaves in a similar fashion, except in this case the proposition v will stop to hold until this norm effect is repaired. Both norms can be used to update pointed normative systems. That is, they transform a pointed normative system to a new pointed normative system. In this paper we want to give clear semantics to how a pointed normative system should behave when a norm from our language is added to the system. This is why we introduce the notion of *norm-aligned*. We say that an updated system is norm-aligned if the system implements the new restrictions of the added norm.

Definition 2 (Norm-Aligned (\mathcal{N}_1)) *Let (N, q) be a pointed normative system, $n = (\varphi, +v, Act_R) \in \mathcal{N}_1$ be a norm, and $(N, q)'$ be a possible update of (N, q) with n . We say that $(N, q)'$ is norm-aligned with n if for every proposition $p \in \Pi$ and every (possibly empty) sequence of actions $\alpha_1 \dots \alpha_n \in Act^*$, we have that $(N, q)' \models Do(\alpha_1) \dots Do(\alpha_n)p$ if and only if*

1. $N, q \models Do(\alpha_1) \dots Do(\alpha_n)p$, **or**
2. $p = v$, $N, q \models Do(\alpha_1) \dots Do(\alpha_n)\varphi$, and, $\alpha_1, \dots, \alpha_n \notin Act_R$.

In words, a system updated with a norm $(\varphi, +v, Act_R)$ is norm-aligned with the norm if (1) all propositions that were true before remain true, plus if (2) the norm condition φ is true and no repair action from Act_R has been performed, then v should be true as well. This notion captures both a property of success and a property of minimal change. On the one hand, it states that the norm effect should hold

under the right conditions (property of success; the norm condition is true and no repair action has been performed yet), and on the other hand it states that everything else should remain unchanged (property of minimal change). We can in a similar manner define the notion of norm-aligned for an update of the form $(\varphi, -v, Act_R)$, which should reflect that such an update removes v . This leads us to the following postulate.

Postulate 1 (Norm-Aligned (\mathcal{N}_1)) *Any normative system updated with a norm $n \in \mathcal{N}_1$ should be norm-aligned.*

This is inspired by the kind of postulates we can find in the AGM framework, which state how updates should behave [3]. A natural question that arises is how we can define updates that are norm-aligned. That is, given a pointed system (N, q) and norm $n \in \mathcal{N}_1$, how can we define $(N, q)'$ such that it is norm-aligned? The next section aims to answer this question.

3.1.1 Defining Norm-Aligned Updates

We will now show how we can update a pointed normative system to a new pointed normative system which is norm-aligned. Given a pointed normative system (N, q) and a norm n , we write $(N, q)[n]$ to denote the updated system, and we define this update as follows.

Definition 3 *Given $N = (Q, Act, \rightarrow, \Pi, V, \mu)$, $q \in Q$ and $n = (\varphi, +v, Act_R) \in \mathcal{N}_1$, we let $(N, q)[n] = (N[n], q[n])$ such that:*

- System $N[n] = (Q', Act, \rightarrow', \Pi, V, \mu')$, where:
 - $Q' = \{q^r, q^a \mid q \in Q\}$
 - $\rightarrow' = \{(q_i^a, \alpha, q_j^a) \mid q_i(\alpha) = q_j \text{ and } \alpha \notin Act_R\} \cup \{(q_i^a, \alpha, q_j^r) \mid q_i(\alpha) = q_j \text{ and } \alpha \in Act_R\} \cup \{(q_i^r, \alpha, q_j^r) \mid q_i(\alpha) = q_j\}$
 - For every state $q \in Q$:
 - $\mu'(q^r) = \mu(q)$ and
 - $\mu'(q^a) = \begin{cases} \mu(q) \cup \{v\} & \text{if } N, q \models \varphi \\ \mu(q) & \text{otherwise} \end{cases}$
- State $q[n] = q^a$

That is, for every state $q \in Q$ we create two copies in the updated system; one in which the norm effect is active (q^a) and one in which it is repaired (q^r). Thus, for each update the states of the system are duplicated. The transitions between active and repaired states are analogous to the transitions of the original system, except whenever we are in an active state and a repair action occurs, in which case we go to a repaired state. Note that we have written q_i and q_j in this definition to simply denote that these states might possibly be different; we attach no further meaning to this indexing. Alternatively, for an update with $-v$, the updated valuation function μ' becomes:

$$\mu'(q^r) = \mu(q) \text{ and } \mu'(q^a) = \begin{cases} \mu(q) \setminus \{v\} & \text{if } N, q \models \varphi \\ \mu(q) & \text{otherwise} \end{cases}$$

The definition of this update reflects the fact that if no repair action has been performed yet (meaning we are in an active state) and φ is true at this state, then v should also be true for $+v$ and false for $-v$. This leads us to the following result.

Proposition 1 *Given an arbitrary N , $q \in Q$ and $n \in \mathcal{N}_1$, the pointed system $(N, q)[n]$ is norm-aligned with n .*

With our running example, we can visualize how this update changes the behaviour of a normative system.

3.1.2 Running Example

We will return to the station system found in Figure 1. Assume that the station wants to impose a policy such that each traveller needs to have a valid travel subscription to be in the station, otherwise he is in violation. Formally, the system is updated with the norm $n_0 = (\text{station}, +v, \{\text{buy_sub}\})$, where buy_sub reflects the action of buying a subscription (which was not explicitly drawn in Figure 1): to be in a ‘station’-state causes violation v , unless this effect is repaired by buying a subscription. In Figure 2 we see how we update the normative system N_{station} to system $N_{\text{station}}[n_0]$, and states q_0 and q_1 to q_0^a and q_1^a respectively. It is important to note that the update

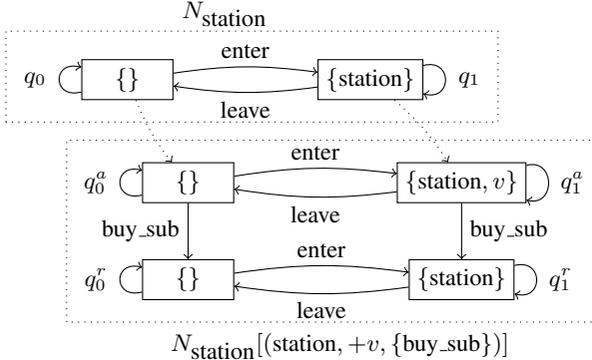


Figure 2. Above system N_{station} , and below the updated system. The dotted transitions denote the state-updates.

can affect the current state; if we would be in state q_1 in the original system, it could be that an update causes us to be in a violation-state, particularly we have:

$$(N_0, q_1)[n_0] \models v$$

If the ‘enter’ action is performed before the ‘buy_sub’ action in the updated system, we would also be in a violation-state. We thus have that:

$$(N_{\text{station}}, q_0)[n_0] \models Do(\text{enter})v$$

Otherwise this would not be the case. We have:

$$(N_{\text{station}}, q_0)[n_0] \models Do(\text{buy_sub})Do(\text{enter})\neg v$$

Thus, we see that this relatively simple norm already makes the behaviour of the system more interesting and complex. In the next section we will extend our language to reason about these dynamic updates.

3.2 Logic and Axiomatization

We can now extend modal language \mathcal{L}_0 to dynamic modal language \mathcal{L}_1 by adding norm update operation $[n]$ to \mathcal{L}_0 , where $p \in \Pi$, $\alpha \in Act$, and $n \in \mathcal{N}_1$:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid Do(\alpha)\varphi \mid [n]\varphi$$

Formulas of \mathcal{L}_1 are evaluated along pointed normative systems. This is done by adding the following rule to the satisfaction relation of \mathcal{L}_0 :

$$N, q \models [n]\varphi \text{ iff } (N, q)[n] \models \varphi$$

We reiterate that although this follows the definition of an update we can find in dynamic logic, the difference here is that we update with a norm consisting of a condition, effect and repair actions. A formula $[n]\varphi$ should be read as: “after adding norm n to the normative system it is the case that φ holds”.

3.2.1 Examples

Before we go to an axiomatization of this logic, let us look at some specific (non-)validities of this dynamic logic. First and foremost, if φ holds at a specific state, and we update with $(\varphi, +v, Act_R)$, we expect that the violation v holds. This is reflected by the following validity:

$$\models \varphi \rightarrow [(\varphi, +v, Act_R)]v$$

We expect that the other way around is not necessarily the case (the right side implying the left side), since it might be that v was already the case in a state. Again, this is reflected by the following non-validity:

$$\not\models [(\varphi, +v, Act_R)]v \rightarrow \varphi$$

However, if we know that $\neg v$ is the case and we know that after the norm update operation v holds, it can only be the case that φ holds. Thus:

$$\models \neg v \wedge [(\varphi, +v, Act_R)]v \rightarrow \varphi$$

For any action $\alpha \in Act$ such that $\alpha \in Act_R$, we have the following expected validity:

$$\models [(\varphi, \pm v, Act_R)](Do(\alpha)\psi) \leftrightarrow Do(\alpha)\psi$$

That is to say, whenever a norm effect is repaired, the truth of a formula ψ depends merely on whether it was true before the update; i.e. nothing changes. This follows from the fact that these updates are norm-aligned. However, when the norm effect is not repaired, we can infer the truth of ψ by *first* performing the action α and *then* updating the system. Or in short, it does not matter at what moment the update is performed. Thus, for any action $\alpha \in Act$ such that $\alpha \notin Act_R$ we have:

$$\models [(\varphi, \pm v, Act_R)](Do(\alpha)\psi) \leftrightarrow (Do(\alpha)[(\varphi, \pm v, Act_R)]\psi)$$

We also have the important property that order matters. First updating with norm n_0 and then with n_1 may have different results than first updating with n_1 and then with n_0 . For example, we have:

$$\begin{aligned} &\models [(\top, +v, \emptyset)]([(\top, +v, \emptyset)]v') \text{ , and,} \\ &\not\models [(\top, +v, \emptyset)]([(\top, +v, \emptyset)]v') \end{aligned}$$

Lastly, we can also have updates without any effects, since for example the update $(\perp, +v, Act_R)$ leaves the truth of any formula unchanged:

$$\models [(\perp, +v, Act_R)]\psi \leftrightarrow \psi$$

In the next section we will provide an axiomatization of our logic.

3.2.2 Axiomatization

We will now show how these properties lead to a non-trivial axiomatization of our logic. We have the following result.

Theorem 2 *The logic \mathcal{L}_1 is (soundly and completely) axiomatized by system **L1**, which consists of adding the following reduction axiom schemes to system **L0** together with the rule of replacement of equivalent formulas, where n is $(\varphi, \pm v, Act_R)$:*

1. (a) $([(\varphi, +v, Act_R)]v) \leftrightarrow (\varphi \vee v)$
 (b) $([(\varphi, -v, Act_R)]v) \leftrightarrow (\neg\varphi \wedge v)$
2. $([(\varphi, \pm v, Act_R)]p) \leftrightarrow p$ (if $v \neq p$)
3. $([n]\neg\psi) \leftrightarrow (\neg[n]\psi)$
4. $([n](\psi_1 \vee \psi_2)) \leftrightarrow (([n]\psi_1) \vee ([n]\psi_2))$
5. $([n]Do(\alpha)\psi) \leftrightarrow (Do(\alpha)\psi)$ (if $\alpha \in Act_R$)
6. $([n]Do(\alpha)\psi) \leftrightarrow (Do(\alpha)[n]\psi)$ (if $\alpha \notin Act_R$)

Proof. (Soundness) Soundness can be proven by proving soundness for every axiom independently. Soundness of each axiom can be shown by following the definitions found in the paper except for axiom 5, which requires the result that for every (N, q) we have $N[\varphi, \pm p, Act_R], q^r \models \psi$ iff $N, q \models \psi$, which can be shown by showing the existence of bi-simulation relation between the two models and states. We show soundness of axiom 6 as example, and omit the other ones due to space constraints:

$$\begin{aligned}
N, q \models [n]Do(\alpha)\psi & \Leftrightarrow \\
N[n], q^r \models Do(\alpha)\psi & \Leftrightarrow \\
N[n], q^r(\alpha) \models \psi & \Leftrightarrow \\
N[n], q(\alpha)^r \models \psi \text{ (if } \alpha \notin Act_T) & \Leftrightarrow \\
N, q(\alpha) \models [n]\psi \text{ (if } \alpha \notin Act_T) & \Leftrightarrow \\
N, q \models Do(\alpha)[n]\psi \text{ (if } \alpha \notin Act_T) & \Leftrightarrow
\end{aligned}$$

(Completeness) In order to show completeness, we define the following translation function $\tau_1 : \mathcal{L}_1 \rightarrow \mathcal{L}_0$:

$$\begin{aligned}
\tau_1(p) &= p \\
\tau_1([(\varphi, +v, Act_R)]v) &= \varphi \vee v \\
\tau_1([(\varphi, -v, Act_R)]v) &= \neg\varphi \wedge v \\
\tau_1(\neg\psi) &= \neg\tau_1(\psi) \\
\tau_1(\psi_1 \vee \psi_2) &= \tau_1(\psi_1) \vee \tau_1(\psi_2) \\
\tau_1(Do(\alpha)\psi) &= Do(\alpha)\tau_1(\psi) \\
\tau_1([n]\neg\psi) &= \tau_1(\neg[n]\psi) \\
\tau_1([n](\psi_1 \vee \psi_2)) &= \tau_1(([n]\psi_1) \vee ([n]\psi_2)) \\
\tau_1([n]Do(\alpha)\psi) &= \tau_1(Do(\alpha)\psi) \quad (\text{if } \alpha \in Act_R) \\
\tau_1([n]Do(\alpha)\psi) &= \tau_1(Do(\alpha)[n]\psi) \quad (\text{if } \alpha \notin Act_R) \\
\tau_1([n_1]([n_2]\psi)) &= \tau_1([n_1]\tau_1([n_2]\psi))
\end{aligned}$$

The first observation we can make is that this function is well-defined, i.e. for any formula $\varphi \in \mathcal{L}_1$ this function returns a formula $\tau_1(\varphi) \in \mathcal{L}_0$. The reason that this function is well-defined (i.e. always gives an answer) is that we can show under a suitable definition of the complexity of formulas (see [16]) that this translation always reduces the complexity. Using this translation, we can show that for any $\varphi \in \mathcal{L}_1$ we have $\vdash_{L_1} \varphi \leftrightarrow \tau_1(\varphi)$ (where $\vdash_{L_1} \varphi$ stands for “ φ is provably in system **L1**”), which due to space constraints we omit in this paper. By soundness of \mathcal{L}_1 , we have from $\vdash_{L_1} \varphi \leftrightarrow \tau_1(\varphi)$ that $\models \varphi \leftrightarrow \tau_1(\varphi)$. Thus, from our assumption that $\models \varphi$, we have $\models \tau_1(\varphi)$. By the completeness of system **L0** (Theorem 1), we have $\vdash_{L_0} \tau_1(\varphi)$. This implies that also $\vdash_{L_1} \tau_1(\varphi)$, since **L1** contains all the rules and axioms of **L0**. Since we have shown that $\vdash_{L_1} \varphi \leftrightarrow \tau_1(\varphi)$ and since $\vdash_{L_1} \tau_1(\varphi)$, we can apply modus ponens to acquire $\vdash_{L_1} \varphi$.

These axioms are inspired by the type of reduction axioms we may find in [16]. Axiom 1a states that if we update every φ -state with $+v$, in order for v to be true it either had to be true before the update, or φ is the case. Axiom 1b states that if we update every φ -state with $-v$, in order for v to be true it has to be both true before the update and the state itself was not updated, i.e. $\neg\varphi$ is the case. Axiom 2 states that any other proposition for which the update does not apply behave invariantly: i.e. the truth condition remains as what it was before the update. Axiom 3 and 4 both state that the dynamic

update is a *function*: a normative system N together with an update n uniquely defines an updated system $N[n]$. Axiom 5 states that when a repair action is performed, the norm is no longer in effect. Finally, axiom 6 states that if a certain action does not repair the norm effect, it does not matter if we perform the update before or after this action in order to determine the state of affairs caused by this action.

We can now show how we can derive the validity of assertions from language \mathcal{L}_1 . Previously in the paper, we already briefly mentioned without proof that the following assertion is valid:

$$\models [(\top, +v, \emptyset)]([(v, +v', \emptyset)]v')$$

Below, we prove it formally by providing a derivation from system **L1**. We use notation ‘ $RE(x,y)$ ’ to denote that we replaced a sub-formula from line x by using an acquired equivalence from line y , which is short-hand notation for applying the rule of replacement of equivalence formulas (RE) and then applying modus ponens (MP) on the result. Moreover, we use the notation ‘ $MP(x,y)$ ’ to denote that we apply modus ponens (MP) with the implication from line x and the precedent from line y . In the proof, we assume that $v \neq v'$:

$$\begin{aligned}
([(\top, +v', \emptyset)]v') &\leftrightarrow (v \vee v') && \text{Ax.1(a)} \\
([(\top, +v, \emptyset)](v \vee v')) &\leftrightarrow ([(\top, +v, \emptyset)]v \vee [(\top, +v, \emptyset)]v') && \text{Ax.4} \\
([(\top, +v, \emptyset)]v) &\leftrightarrow (\top \vee v) && \text{Ax.1(a)} \\
([(\top, +v, \emptyset)]v') &\leftrightarrow v' && \text{Ax.2} \\
([(\top, +v, \emptyset)](v \vee v')) &\leftrightarrow ((\top \vee v) \vee [(\top, +v, \emptyset)]v') && \text{RE(2,3)} \\
([(\top, +v, \emptyset)](v \vee v')) &\leftrightarrow ((\top \vee v) \vee v') && \text{RE(5,4)} \\
([(\top, +v, \emptyset)]([(v, +v', \emptyset)]v')) &\leftrightarrow ((\top \vee v) \vee v') && \text{RE(6,1)} \\
((\top \vee v) \vee v') &&& \text{Taut.} \\
([(\top, +v, \emptyset)]([(v, +v', \emptyset)]v')) &&& \text{MP(7,8)}
\end{aligned}$$

Thus, we have:

$$\vdash_{L_1} [(\top, +v, \emptyset)]([(v, +v', \emptyset)]v')$$

And by soundness of **L1** we have:

$$\models [(\top, +v, \emptyset)]([(v, +v', \emptyset)]v')$$

A natural question we may ask is whether logic \mathcal{L}_1 with corresponding system **L1** is *decidable*, i.e. whether there exists an effective procedure that tells us whether an arbitrary formula $\varphi \in \mathcal{L}_1$ is valid. We have the following result.

Theorem 3 *The logical language \mathcal{L}_1 with corresponding system **L1** is decidable.*

Proof. We use the translation function $\tau_1 : \mathcal{L}_1 \rightarrow \mathcal{L}_0$ from our completeness proof in Theorem 2 to establish the following correspondence:

$$\vdash_{L_1} \varphi \Leftrightarrow \vdash_{L_1} \tau_1(\varphi) \Leftrightarrow \vdash_{L_0} \tau_1(\varphi)$$

Thus, in order to determine whether $\vdash_{L_1} \varphi$, it is both necessary and sufficient to show that $\vdash_{L_0} \tau_1(\varphi)$. However, since logic \mathcal{L}_0 with corresponding system **L0** is decidable and since τ_1 is well defined and can be effectively computed, decidability immediately transfers to our logic and system.

This concludes our work on the logical language \mathcal{L}_1 and **L1** allowing us to reason about normative update of the ‘to-be’ variant. In the next section we take a look at normative update of the ‘to-do’ variant, leading to another dynamic logic which, as we will see later, will extend this logic.

4 Extended Norm Language and Update

In the previous section, we developed a basic logical language of normative update. In this section we consider a different kind of normative update, which corresponds to norms of the ‘to-do’ variant.

4.1 Language and Update

In the remainder of this section, we give a language to construct these kinds of norms, we show how we can update a normative system using these norms and finally we show how we can add these constructs to our logical language. We construct the norm language \mathcal{N}_2 in the following way, where $\varphi \in \mathcal{L}_0$, $v \in V$, $Act_T \subseteq Act$ and $Act_R \subseteq Act$:

$$n ::= (Act_T, \varphi, +v, Act_R) \mid (Act_T, \varphi, -v, Act_R)$$

We call the set Act_T the set of trigger actions. Adding a norm $(Act_T, \varphi, +v, Act_R)$ implies that whenever a trigger action from Act_T occurs, from that point on for every φ -state the violation v will start to hold until a repair action from the set Act_R occurs. An example norm might be that speeding will result in a violation v which can be repaired by paying a fine, which can be modelled by $(\{\text{speeding}\}, \top, +v, \{\text{pay}\})$. Again, since we want to give a clear semantic interpretation to how a pointed normative system should behave when a norm is added to the system, we extend the notion of *norm-aligned* to updates from this extended language.

Definition 4 (Norm-Aligned (\mathcal{N}_2)) Given a pointed normative system (N, q) , a norm $n = (Act_T, \varphi, +v, Act_R) \in \mathcal{N}_2$, we say that $(N, q)'$ is norm-aligned with n if for every atomic proposition p and every (possibly empty) sequence of actions $\alpha_1 \dots \alpha_n \in Act^*$, we have that $(N, q)' \models Do(\alpha_1) \dots Do(\alpha_n)p$ if and only if

1. $N, q \models Do(\alpha_1) \dots Do(\alpha_n)p$, or;
2. $p = v$, $N, q \models Do(\alpha_1) \dots Do(\alpha_n)\varphi$, and $\exists i : \alpha_i \in Act_T, \forall j, i < j \leq n : \alpha_j \notin Act_R$.

In words, a system updated with a norm $(Act_T, \varphi, +v, Act_R)$ is norm-aligned the norm if (1) all propositions that were true before remain true, plus (2) if the norm condition φ is true, a trigger action from Act_T has been performed and no repair action from Act_R has been performed yet, then v should be true as well. Again, this notion captures both a property of success and a property of minimal change. On the one hand, it states that the norm effect should hold under the right conditions (property of success; a trigger action has been performed, the norm condition is true and no repair action has been performed yet), and on the other hand it states that everything else should remain unchanged (property of minimal change). Notice that by the above interpretation of norm-aligned, whenever we have a norm $(Act_T, \varphi, \pm v, Act_R)$ and an action α such that $\alpha \in Act_T$ and $\alpha \in Act_R$, it is always the case that action α triggers and not repairs the norm effect, even though this action is both a trigger and a repair action. In other words, the trigger actions take priority over the repair actions. Again, more complex norms can be encoded by combining a multitude of these simpler norms. We again have that every update should be norm-aligned, as given by the following postulate.

Postulate 2 (Norm-Aligned (\mathcal{N}_2)) Any normative system updated with a norm $n \in \mathcal{N}_2$ should be norm-aligned.

We will now show how we can perform a norm update which results in a norm-aligned normative system. This norm update is very

similar to the previous norm update we saw in this paper, the main difference lying in the updated accessibility relation of the model. Formally:

Definition 5 Given $N = (Q, Act, \rightarrow, \Pi, V, \mu)$, $q \in Q$ and $n = (Act_T, \varphi, +v, Act_R) \in \mathcal{N}_2$, we let $(N, q)[n] = (N[n], q[n])$ such that:

- System $N[n] = (Q', Act, \rightarrow', \Pi, V, \mu')$, where:
 - $Q' = \{q^r, q^a \mid q \in Q\}$
 - $\rightarrow' = \begin{aligned} &\{(q_i^r, \alpha, q_j^r) \mid q_i(\alpha) = q_j \text{ and } \alpha \notin Act_T\} \cup \\ &\{(q_i^r, \alpha, q_j^a) \mid q_i(\alpha) = q_j \text{ and } \alpha \in Act_T\} \cup \\ &\{(q_i^a, \alpha, q_j^a) \mid q_i(\alpha) = q_j \text{ and } \alpha \notin (Act_R \setminus Act_T)\} \cup \\ &\{(q_i^a, \alpha, q_j^r) \mid q_i(\alpha) = q_j \text{ and } \alpha \in (Act_R \setminus Act_T)\} \end{aligned}$
 - For every state $q \in Q$:
 - $\mu'(q^r) = \mu(q)$ and
 - $\mu'(q^a) = \begin{cases} \mu(q) \cup \{v\} & \text{if } N, q \models \varphi \\ \mu(q) & \text{otherwise} \end{cases}$
- State $q[n] = q^r$

The norm update with $-v$ works analogously, except we remove this atomic proposition from the norm updated valuation function μ' for every active state. Again, for every state $q \in Q$ we create two copies in the updated system; one in which the norm effect is active and one in which it is repaired. There are two important differences in norm updates from \mathcal{N}_2 in comparison with norm updates from \mathcal{N}_1 . They are the following:

1. The updated accessibility relation is different. It is still the case that whenever we are in an active state and a repair action has been performed (which is not also a trigger action), we go to a repaired state. However, whenever we are in a repaired state and a trigger action has been performed, we go to an active state. In other words, with this norm update we are both able to go from active states to repaired states and vice-versa. If no trigger or repair action is performed, we remain in an active (or repaired) state.
2. With the update with norms from \mathcal{N}_2 the current state is updated to a repaired state, while with updates with norms from \mathcal{N}_1 the current state was updated to an active state.

The following proposition shows that these norm updates again result in a norm-aligned system:

Proposition 2 Given an arbitrary N , $q \in Q$ and $n \in \mathcal{N}_2$, the pointed system $(N, q)[n]$ is norm-aligned with n .

We return to our running example of the station to visualize how this norm update works.

4.1.1 Running Example

Returning to the example from Figure 1, we now want to update this system with the norm which states that if we do not check-in and enter the station, we are in violation until we leave. We assume that entering through this method is encoded by an action ‘unchecked’, which encodes the action of not checking in. We can now encode this norm by the norm $n = (\{\text{unchecked}\}, \text{station}, +v, \{\text{leave}\})$, which states that there is a violation v if a traveller is not checked in, and this violation remains as long as the traveller is in the station. By following the rules of the norm update, we see in Figure 3 how we

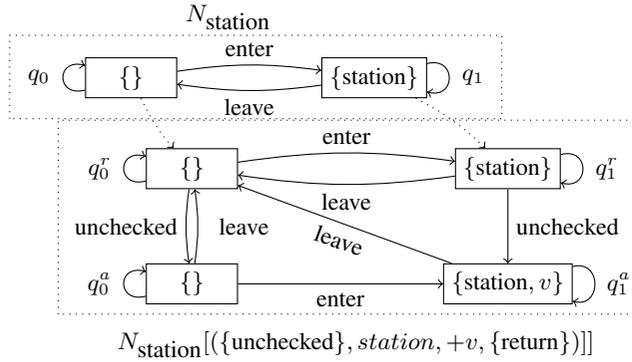


Figure 3. Above system N_{station} , and below the updated system. The dotted transitions denote the state-updates.

go from system N_{station} to $N_{\text{station}}[n]$, and states q_0 and q_1 to q_0^r and q_1^r respectively. We now have the validity that:

$$(N_{\text{station}}, q_0)[n] \models Do(\text{unchecked})Do(\text{enter})v$$

Or in words, not checking in and entering brings about a violation. Note that in principle, norms from \mathcal{N}_1 and \mathcal{N}_2 can be combined in arbitrary ways, even though in these examples we focussed on them separately.

4.2 Logic and Axiomatization

We can now add these norm update operations to our language in the same way we did to \mathcal{L}_1 to acquire \mathcal{L}_2 , which contains formulas of the form $[n]\varphi$ where $n \in \mathcal{N}_2$.

4.2.1 Examples

Before we go to an axiomatization of this logic, let us look at some specific (non-)validities of this extended dynamic logic. First, we expect that updating with $(Act_T, \varphi, +v, Act_R)$ does not immediately change the valuation of the current state. This is reflected by the following validity, which holds for any proposition p' :

$$\models ((Act_T, \varphi, +v, Act_R)[p']) \leftrightarrow p'$$

In other words, the truth of p' in the current state after an update depends merely on the truth of p' before the update. Moreover, we expect whenever a norm effect is not triggered by a certain action, it does not matter whether this update is performed before or after this action. That is, whenever $\alpha_1, \dots, \alpha_n \notin Act_T$, we have that:

$$\models ((Act_T, \varphi, +v, Act_R)(Do(\alpha_1) \dots Do(\alpha_n)\psi)) \leftrightarrow (Do(\alpha_1) \dots Do(\alpha_n)((Act_T, \varphi, +v, Act_R)\psi))$$

Of course, a more interesting scenario happens when a trigger action is performed. If we have that $\alpha \in Act_T$, we have the following validity, which elegantly showcases the connection between updates from \mathcal{N}_1 and \mathcal{N}_2 :

$$\models ((Act_T, \varphi, +v, Act_R)Do(\alpha)\psi) \leftrightarrow (Do(\alpha)[\varphi, +v, Act_T \cup Act_R]([Act_T, \varphi, +v, Act_R]\psi))$$

The truth of this validity is not at all apparent. In words, updating a system with $(Act_T, \varphi, +v, Act_R)$ and then performing a trigger

action is equivalent to first performing the trigger action, then updating the system with $(\varphi, +v, Act_T \cup Act_R)$ (note that this norm is in \mathcal{N}_1 , and that the repair actions are $Act_T \cup Act_R$ instead of just Act_R) and finally updating again with $(Act_T, \varphi, +v, Act_R)$. The reason the norm $(Act_T, \varphi, +v, Act_R)$ does not disappear on the right hand side is because it remains in effect, i.e. it can be triggered by an action at a later stage.

4.2.2 Axiomatization

In the previous section, we saw some example validities which we can use in this section to provide an axiomatization of our logic. We have the following result:

Theorem 4 *The logic \mathcal{L}_2 is axiomatized by system **L2**, which contains all the axioms and rules from **L1**, but with the following additional reduction axiom schemes together with the rule of replacement of equivalent formulas, where $n = (Act_T, \varphi, \pm v, Act_R)$:*

1. $([n]p) \leftrightarrow p$
2. $([n]\neg\psi) \leftrightarrow (\neg[n]\psi)$
3. $([n](\psi_1 \vee \psi_2)) \leftrightarrow (([n]\psi_1) \vee ([n]\psi_2))$
4. $([n]Do(\alpha)\psi) \leftrightarrow (Do(\alpha)[(\varphi, \pm v, Act_R \cup Act_T)]([n]\psi))$ (if $\alpha \in Act_T$)
5. $([n]Do(\alpha)\psi) \leftrightarrow (Do(\alpha)[n]\psi)$ (if $\alpha \notin Act_T$)

Proof. (Soundness) Soundness can again be proven by individual soundness for each axiom. Special notice should be laid on axiom 4, which contains a norm from \mathcal{N}_1 with $Act_R \cup Act_T$ as the repair actions. This is acquired from the result that for an arbitrary pointed normative system (N, q) , norm $n = (Act_T, \varphi, \pm v, Act_R) \in \mathcal{N}_2$ and formula $\psi \in \mathcal{L}_2$, we have $N[n], q^a \models \psi$ if and only if $(N[(\varphi, \pm v, Act_R \cup Act_T)], q^a)[n] \models \psi$. This result can be shown by showing the existence of bi-simulation relation between the two models and states. Intuitively, this axiom states that when a trigger action has been performed, it would be equivalent to first performing this action, then adding the norm in which the norm effect is immediately active, and finally adding the norm again. The intuitive reason for using $Act_R \cup Act_T$ instead of Act_R as the repair actions is to avoid interference with n when at a later moment a trigger action occurs again, i.e. if we would use Act_R the axiom would not be sound.

(Completeness) In order to show completeness, we define the following translation function $\tau_2 : \mathcal{L}_2 \rightarrow \mathcal{L}_0$, where below the function $\tau_1 : \mathcal{L}_1 \rightarrow \mathcal{L}_0$ is the function as defined in the proof of Theorem 2, which we use to reduce formulas of the form $[n]p$ (if $n \in \mathcal{N}_1$). Moreover, we define $f : \mathcal{N}_2 \rightarrow \mathcal{N}_1$ as:

$$f((Act_T, \varphi, \pm v, Act_R)) := (\varphi, \pm v, Act_R \cup Act_T)$$

The reduction is defined as follows:

$$\begin{aligned} \tau_2(p) &= p \\ \tau_2([n]p) &= \tau_1([n]p) & (n \in \mathcal{N}_1) \\ \tau_2([n]p) &= p & (n \in \mathcal{N}_2) \\ \tau_2(\neg\psi) &= \neg\tau_2(\psi) \\ \tau_2(\psi_1 \vee \psi_2) &= \tau_2(\psi_1) \vee \tau_2(\psi_2) \\ \tau_2(Do(\alpha)\psi) &= Do(\alpha)\tau_2(\psi) \\ \tau_2([n]\neg\psi) &= \tau_2(\neg[n]\psi) \\ \tau_2([n](\psi_1 \vee \psi_2)) &= \tau_2([n]\psi_1) \vee ([n]\psi_2) \\ \tau_2([n]Do(\alpha)\psi) &= \tau_2(Do(\alpha)\psi) & (n \in \mathcal{N}_1 \ \& \ \alpha \in Act_R) \\ \tau_2([n]Do(\alpha)\psi) &= \tau_2(Do(\alpha)[n]\psi) & (n \in \mathcal{N}_1 \ \& \ \alpha \notin Act_R) \\ \tau_2([n]Do(\alpha)\psi) &= \tau_2(Do(\alpha)[f(n)]([n]\psi)) & (n \in \mathcal{N}_2 \ \& \ \alpha \in Act_T) \\ \tau_2([n]Do(\alpha)\psi) &= \tau_2(Do(\alpha)[n]\psi) & (n \in \mathcal{N}_2 \ \& \ \alpha \notin Act_T) \\ \tau_2([n_1]([n_2]\psi)) &= \tau_2([n_1]\tau_2([n_2]\psi)) \end{aligned}$$

This time around, it is slightly harder to show that τ_2 is well-defined. This is because τ_2 on input $[n]Do(\alpha)\psi$ for which $n \in \mathcal{N}_2$ and $\alpha \in Act_T$ ‘generates’ an extra norm $f(n)$, i.e., it can occur that on a certain input, the number of dynamic operators increases. However, since the number of norms from the set \mathcal{N}_2 in the scope of a norm from \mathcal{N}_1 never increases and always eventually decreases, we know that the translation will eventually return a well-formed formula from \mathcal{L}_0 . It is possible to show that for any $\varphi \in \mathcal{L}_2$ we have that $\vdash_{\mathcal{L}_2} \varphi \leftrightarrow \tau_2(\varphi)$ (where again $\vdash_{\mathcal{L}_2} \varphi$ stands for “ φ is provably in **L2**”), which we omit due to space constraints. We can now show that from the assumption that $\models \varphi$ we have $\vdash_{\mathcal{L}_2} \varphi$. By soundness of \mathcal{L}_2 , we have from $\vdash_{\mathcal{L}_2} \varphi \leftrightarrow \tau_2(\varphi)$ that $\models \varphi \leftrightarrow \tau_2(\varphi)$. Thus, from our assumption that $\models \varphi$, we have $\models \tau_2(\varphi)$. By the completeness of system **L0**, we have $\vdash_{\mathcal{L}_0} \tau_2(\varphi)$. This implies that also $\vdash_{\mathcal{L}_2} \tau_2(\varphi)$, since **L2** contains all the rules and axioms of **L0**. Since we have shown that $\vdash_{\mathcal{L}_2} \varphi \leftrightarrow \tau_2(\varphi)$ and since $\vdash_{\mathcal{L}_2} \tau_2(\varphi)$, we can apply rule modus ponens to acquire $\vdash_{\mathcal{L}_2} \varphi$.

In words, axiom 1 states that an update does not modify the state of affairs of the current state. Axiom 2 and 3 both state that the dynamic update is a *function*: given a normative system and an update, a new normative system is uniquely determined. Axiom 4 is especially noteworthy. This axiom states that when a trigger action has been performed, it would be equivalent to first performing this action, then adding the norm in which the norm effect is immediately active, and finally adding the norm again. This axiom elegantly showcases the connection between our two types of norms, namely it both contains constructs from language \mathcal{N}_1 and \mathcal{N}_2 . Axiom 5 states the property that if the norm is not triggered by a certain action, it does not matter if we perform the update before or after this action. All these axioms are only concerned with the trigger actions, while the axioms of \mathcal{L}_1 are only concerned with the repair actions. Together they form a complete axiomatization which considers both the trigger and repair actions.

We again have a similar result about decidability.

Theorem 5 *The logical language \mathcal{L}_2 with corresponding system **L2** is decidable.*

Proof. We use the translation function $\tau_2 : \mathcal{L}_2 \rightarrow \mathcal{L}_0$ from our completeness proof in Theorem 4 to establish the following correspondence:

$$\vdash_{\mathcal{L}_2} \varphi \Leftrightarrow \vdash_{\mathcal{L}_2} \tau_2(\varphi) \Leftrightarrow \vdash_{\mathcal{L}_0} \tau_2(\varphi)$$

Thus, in order to determine whether $\vdash_{\mathcal{L}_2} \varphi$, it is both necessary and sufficient to show that $\vdash_{\mathcal{L}_0} \tau_2(\varphi)$. However, since logic \mathcal{L}_0 with corresponding system **L0** is decidable and since τ_2 is well defined and can be effectively computed, decidability again transfers to our logic and system.

Note that this proof highlights an easier way to establish the validity of a sentence. To show for example that $\vdash_{\mathcal{L}_2} [n](Do(\alpha_2)Do(\alpha_1)v)$, where $n = (\{1\}, \top, +v, \{2\})$, we can perform the following translation:

$$\begin{aligned} \tau_2([n](Do(\alpha_2)Do(\alpha_1)v)) &= \\ \tau_2(Do(\alpha_2)[n](Do(\alpha_1)v)) &= \\ Do(\alpha_2)\tau_2([n](Do(\alpha_1)v)) &= \\ Do(\alpha_2)\tau_2(Do(\alpha_1)[f(n)]([n]v)) &= \\ Do(\alpha_2)Do(\alpha_1)\tau_2([f(n)]([n]v)) &= \\ Do(\alpha_2)Do(\alpha_1)\tau_2([f(n)]\tau_2([n]v)) &= \\ Do(\alpha_2)Do(\alpha_1)\tau_2([f(n)]v) &= \\ Do(\alpha_2)Do(\alpha_1)(\top \vee v) & \end{aligned}$$

To easily establish that:

$$\vdash_{\mathcal{L}_0} Do(\alpha_2)Do(\alpha_1)(\top \vee v)$$

This concludes our work on the logical language \mathcal{L}_2 and **L2**, extending \mathcal{L}_1 and **L1**. We saw that even though our second logic extends the first, decidability is still guaranteed.

5 Related work and Conclusions

In this paper we introduced various expressive types of norms and devised new dynamic modal logics that are able to characterize the dynamics of such norms. We have shown that these logics are sound, complete and decidable, which means that we can use automated theorem provers to prove properties of norm dynamics and their effects on normative systems. The new forms of norms may have various effects and non-effects on the behaviour of a system which might initially be overlooked. With a formal and rigid framework like the one provided in this paper, we can automatically verify these effects.

The updates we consider in this paper are inspired by the kind of updates we may see in dynamic epistemic logic such as Public Announcement Logic [16], but note that the argument (a norm, which is not just a logical formula but a complex structure consisting of a condition, effect and repair actions) and result (a norm-aligned system) of our norm update is something completely different from what we may find there. Another important difference is the fact that these updates are performed on *pointed* Kripke structures, while updates from [16] are performed on just Kripke structures. Lastly, as an important note, we want to express that although these norm update operations may at first glance be related to Propositional Dynamic Logic (PDL) [11], the updates we consider are performed on Kripke structures with duplicated states and not just on valuations.

This research is related to a multitude of different papers concerned with norm change. The work presented in [7] uses the syntactic approach, where norm change is considered as an operation on the underlying “code” that constitutes the normative system. The work in [10] shows a way to program norm change, but not how to reason about this. In [4], the authors show, given a conditional norm, how a transition system can be unfolded to a tree in which the norm is into effect. The work in [13] considers norm-updates performed on normative systems, but does not consider an axiomatized logic to reason about these updates. In [6], Aucher et al. give a logical account of ought-to-be norm change via the notions of context expansion and contraction. In our work, we perform norm updates on pointed labelled transition systems to give a logical account of a much more general form of norm change via updates of the whole system at hand. To capture the dynamics of norms we have employed ideas from Dynamic Epistemic Logic, [16], where they use dynamic modal logic to characterise the dynamics of Kripke structures, albeit in a totally different (viz. epistemic) setting.

For future research there are various directions. Firstly, we can apply this framework to other well known frameworks of normative systems, such as the coloured systems considered in [14]. Secondly, we can look at other types of norms, such as norms with deadlines, and see how we can relate these to the current model. Lastly, the relation between the current framework and other frameworks of (dynamic) normative theory (e.g., [4, 8, 2, 7]) needs to be explored further.

REFERENCES

- [1] Thomas Ågotnes, Wiebe van der Hoek, Juan A. Rodríguez-Aguilar, Carlos Sierra, and Michael Wooldridge, 'On the logic of normative systems', in *Proceedings of the twentieth International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pp. 1175–1180, (2007).
- [2] Thomas Ågotnes, Wiebe van der Hoek, and Michael Wooldridge, 'Normative system games', in *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007)*, pp. 881–888, (2007).
- [3] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson, 'On the logic of theory change: Partial meet contraction and revision functions', *Journal of Symbolic Logic*, **50**(2), 510–530, (1985).
- [4] Natasha Alechina, Mehdi Dastani, and Brian Logan, 'Reasoning about normative update', in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI 2013)*, pp. 20–26, (2013).
- [5] Alan Ross Anderson, 'A reduction of deontic logic to alethic modal logic', *Mind*, **67**(265), 100–103, (1958).
- [6] Guillaume Aucher, Davide Grossi, Andreas Herzig, and Emiliano Lorini, 'Dynamic context logic', in *Proceedings of Logic, Rationality and Interaction (LORI 2009)*, pp. 15–26, (2009).
- [7] Guido Boella, Gabriella Pigozzi, and Leendert van der Torre, 'Normative framework for normative system change', in *Proceedings of the 8th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2009)*, pp. 169–176, (2009).
- [8] Nils Bulling and Mehdi Dastani, 'Verifying normative behaviour via normative mechanism design', in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pp. 103–108, (2011).
- [9] Brian F. Chellas, *Modal Logic: An Introduction*, Cambridge University Press, 1980.
- [10] Mehdi Dastani, John-Jules Ch. Meyer, and Nick A. M. Tinnemeier, 'Programming norm change', *Journal of Applied Non-Classical Logics*, 151–180, (2012).
- [11] Michael J. Fischer and Richard E. Ladner, 'Propositional modal logic of programs', in *Proceedings of the Ninth Annual ACM Symposium on Theory of Computing*, pp. 286–294, (1977).
- [12] Max Knobbout and Mehdi Dastani, 'Reasoning under compliance assumptions in normative multiagent systems', in *Proceedings of the 11th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, pp. 331–340, (2012).
- [13] Max Knobbout, Mehdi Dastani, and John-Jules Ch. Meyer, 'Reasoning about dynamic normative systems', in *Logics in Artificial Intelligence - 14th European Conference (JELIA 2014)*, pp. 628–636, (2014).
- [14] Marek J. Sergot, 'Action and agency in norm-governed multi-agent systems', in *Proceedings of the Engineering Societies in the Agents World VIII (ESAW 2007)*, pp. 1–54, (2007).
- [15] Yoav Shoham and Moshe Tennenholtz, 'On the synthesis of useful social laws for artificial agent societies', in *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI 1992)*, pp. 276–281, (1992).
- [16] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi, *Dynamic Epistemic Logic, volume 337 of Synthese Library Series*, Springer, 2007.

h-Index Manipulation by Undoing Merges

René van Bevern^{1,2} and Christian Komusiewicz³
 Hendrik Molter⁴ and Rolf Niedermeier⁴ and Manuel Sorge⁴
 Toby Walsh⁵

Abstract. The h-index is an important bibliographic measure used to assess the performance of researchers. Van Bevern et al. [*Artif. Intel.*, to appear] showed that, despite computational worst-case hardness results, substantial manipulation of the h-index of Google Scholar author profiles is possible by merging articles. Complementing this work, we study the opposite operation, the splitting of articles, which is arguably the more natural operation for manipulation and which is also allowed within Google Scholar. We present numerous results on computational complexity (from linear-time algorithms to parameterized computational hardness results) and empirically indicate that at least small improvements of the h-index by splitting merged articles are easily achievable.

1 INTRODUCTION

Lesk [14] pointed out that the h-index⁶ is the modern equivalent of the old saying “Deans can’t read, they can only count.” He also remarked that the idea of “least publishable units” by dividing one’s reports into multiple (short) papers has been around since the 1970s. A modern version of this scenario is the manipulation of an author’s h-index by splitting a publication in an author’s profile into different versions of an article (which may result in an increased h-index). Google Scholar permits such splitting. We study such manipulation in this work, introducing and discussing several models, performing a thorough analysis of the computational complexity, and providing experimental results.

Our main points of reference are three recent publications dealing with the manipulation of the h-index, particularly motivated by Google Scholar author profile manipulation [2, 13, 16]. Indeed, we will closely follow the notation and concepts introduced by van Bevern et al. [2] and we refer to this work for discussion of related work concerning strategic self-citations to manipulate the h-index [1, 4, 17], other citation indices [6, 16, 18], and manipulation in general [7, 8]. The main difference to these previous publications is that they focus on *merging* articles for increasing the h-index [2, 13, 16] or other indices like the g-index and the i10-index [16], while we focus on *splitting*.

In the manipulation scenario for merging the assumption is that an author has a publication profile, for example in Google Scholar, that

consists of single articles and aims to increase his or her h-index by merging articles. This will result in a new article with a potentially higher number of citations. The merging option is provided by Google Scholar to identify different versions of the same article, for example a journal version and a conference version.

In the case of splitting, we assume that, most of the time, an author will maintain a correct profile in which all necessary merges are performed. Some of these merges may decrease the h-index. For instance, this can be the case when the two most cited papers are the conference and journal version of the same article. A very realistic scenario is that at certain times, for example when being evaluated by their dean, an author may temporarily undo some of these merges to increase artificially his or her h-index. A further point which distinguishes manipulation by splitting from manipulation by merging is that for merging it is easier to detect whether someone cheats too much. This can be done by looking at the titles of merged articles [2]. In contrast, it is much harder to prove that someone is manipulating by splitting; the manipulator can always claim to be too busy or that he or she does not know how to operate the profile.

The main theoretical conclusion from our work is that h-index manipulation by splitting merged articles⁷ is typically computationally easier than manipulation by merging. Hence, undoing all merges and then merging from scratch might be intractable in cases while, on the contrary, computing an optimal splitting is computationally feasible. The only good news (and, in a way, a recommendation) in this sense is that if one would use the citation measure “fusionCite” as defined by van Bevern et al. [2], then manipulation is computationally much harder than for the “unionCite” measure used by Google Scholar. We also experimented with data from Google Scholar profiles [2].

Models for Splitting Articles. We consider the publication profile of an author and denote the articles in this profile by $W \subseteq V$, where V is the set of all articles. Following previous work [2], we call these articles *atomic*. Merging articles yields a partition \mathcal{P} of W in which each part $P \in \mathcal{P}$ with $|P| \geq 2$ is a *merged article*.

Given a partition \mathcal{P} of W , the aim of splitting merged articles is to find a refined partition \mathcal{R} of \mathcal{P} with a large h-index, where the *h-index of a partition* \mathcal{P} is the largest number h such that there are at least h parts $P \in \mathcal{P}$ whose number $\mu(P)$ of citations is at least h . Herein, we have multiple possibilities of defining the number $\mu(P)$ of citations of an article in \mathcal{P} [2]. The first one, $\text{sumCite}(P)$, was introduced by de Keijzer and Apt [13], and is simply the sum of the citations of each atomic article in P . Subsequently, van Bevern et al. [2] introduced the more realistic citation measures unionCite

¹ Novosibirsk State University, Novosibirsk, Russian Federation, rvb@nsu.ru

² Sobolev Institute of Mathematics, Siberian Branch of the Russian Academy of Sciences, Novosibirsk, Russian Federation

³ Institut für Informatik, Friedrich-Schiller-Universität Jena, Germany, christian.komusiewicz@uni-jena.de

⁴ Institut für Softwaretechnik und Theoretische Informatik, TU Berlin, Germany, {h.molter,rolf.niedermeier,manuel.sorge}@tu-berlin.de

⁵ TU Berlin, Germany, University of New South Wales and Data61, Sydney, Australia, toby.walsh@nicta.com.au

⁶ The h-index of a researcher is the maximum number h such that he or she has at least h articles each cited at least h times [10].

⁷ Google Scholar allows authors to group different versions of an article. We call the resulting grouping a *merged article*. Google Scholar author profiles typically contain many merged articles, e.g. an arXiv version with a conference version and with a journal version.

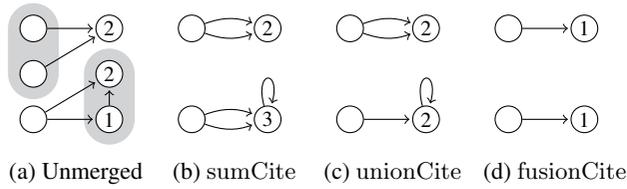


Figure 1: Vertices represent articles, arrows represent citations, numbers are citation counts. The articles on a gray background in (a) have been merged in (b)–(d), and citation counts are given according to the measures sumCite, unionCite, and fusionCite, respectively. The arrows represent the citations counted by the corresponding measure.

(used by Google Scholar), where we take the cardinality of the union of the citations, and fusionCite, where we additionally remove self-citations of merged articles as well as duplicate citations between merged articles. In generic definitions, we denote these measures by μ , see Figure 1 for an illustration and Section 2 for the formal definitions. Note that, to compute these citation measures, we need a *citation graph*, a directed graph whose vertices represent articles and in which an arc from a vertex u to a vertex v means that article u cites article v .

In this work, we introduce three different operations that may be used for undoing merges in a merged article a :

- Atomizing*: splitting a into all its atomic articles,
- Extracting*: splitting off a single atomic article from a , and
- Dividing*: splitting a into two parts arbitrarily.

See Figure 2 for an illustration of the three splitting operations. Note that the atomizing, extracting, and dividing operations are successively more powerful in the sense that successively larger h-indices can be achieved. Google Scholar offers the extraction operation.

The three splitting operations lead to three problem variants, each taking as input a citation graph $D = (V, A)$, a set $W \subseteq V$ of articles belonging to the author, a partition \mathcal{P} of W that defines already-merged articles, and a non-negative integer h denoting the h-index to achieve. For $\mu \in \{\text{sumCite}, \text{unionCite}, \text{fusionCite}\}$, we define the following problems.

ATOMIZING(μ)

Question: Is there a partition \mathcal{R} of W such that

- i) for each $R \in \mathcal{R}$ either $|R| = 1$ or there is a $P \in \mathcal{P}$ such that $R = P$,
- ii) the h-index of \mathcal{R} is at least h with respect to μ ?

EXTRACTING(μ)

Question: Is there a partition \mathcal{R} of W such that

- i) for each $R \in \mathcal{R}$ there is a $P \in \mathcal{P}$ such that $R \subseteq P$,
- ii) for each $P \in \mathcal{P}$ we have $|\{R \in \mathcal{R} \mid R \subset P \text{ and } |R| > 1\}| \leq 1$,
- iii) the h-index of \mathcal{R} is at least h with respect to μ ?

DIVIDING(μ)

Question: Is there a partition \mathcal{R} of W such that

- i) for each $R \in \mathcal{R}$ there is a $P \in \mathcal{P}$ such that $R \subseteq P$,
- ii) the h-index of \mathcal{R} is at least h with respect to μ ?

Conservative Splitting. We study for each of the problem variants an additional upper bound on the number of merged articles that are split. We call these variants *conservative*: if an insincere author would like to manipulate his or her profile temporarily, then he or she would prefer a manipulation that can be easily undone. To formally define CONSERVATIVE ATOMIZING, CONSERVATIVE EXTRACTING, and CONSERVATIVE DIVIDING, we add the following restriction to the partition \mathcal{R} : “the number $|\mathcal{P} \setminus \mathcal{R}|$ of changed articles is at most k ”.

A further motivation for the conservative variants is that, in a Google Scholar profile, an author can click on a merged article and

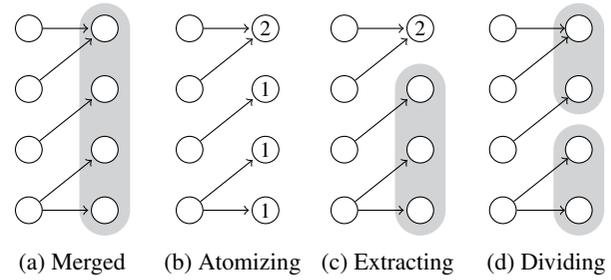


Figure 2: Vertices represent articles, arrows represent citations, numbers are citation counts. The articles on a gray background have been merged in the initial profile (a) and correspond to remaining merged articles after applying one operation in (c) and (d). Each (merged) article has the same citation count, regardless of the used measure sumCite, unionCite, and fusionCite.

tick a box for each atomic article that he or she wants to extract. Since Google Scholar uses the unionCite measure [2], CONSERVATIVE EXTRACTING(unionCite) thus corresponds closely to manipulating the Google Scholar h-index via few of the splitting operations available to the user.

Cautious Splitting. For each splitting operation, we also study an upper bound k on the number of operations. Following our previous work [2], we call this variant *cautious*. In the case of atomizing, conservativity and caution coincide since exactly one operation is performed per changed article. Thus, we obtain two cautious problem variants: CAUTIOUS EXTRACTING and CAUTIOUS DIVIDING. For both we add the following restriction to the partition \mathcal{R} : “the number $|\mathcal{R}| - |\mathcal{P}|$ of extractions (or divisions, respectively) is at most k ”. In both variants we consider k to be part of the input.

Our results. Our theoretical (complexity classification) results are summarized in Table 1. The measures sumCite and unionCite behave basically the same. In particular, in case of atomizing and extracting, manipulation is doable in linear time, while fusionCite mostly leads to (parameterized) intractability, that is, to high worst-case computational complexity. Moreover, the dividing operation (the most general one) seems to lead to computationally much harder problems than atomizing and extracting. As indicated in Table 1, the computational complexity of two specific problems remains open.

We performed experiments with real-world data [2] and the mentioned linear-time algorithms, in particular for the case directly relevant to Google Scholar, that is, using the extraction operation and the unionCite measure. Our general findings are that increases of the h-index by one or two typically are easily achievable with few operations. The good news is that dramatic manipulation opportunities due to splitting are rare. They cannot be excluded, however, and they could be easily executed when relying on standard operations and measures (as used in Google Scholar). Working with fusionCite instead of the other two could substantially hamper manipulation.

2 PRELIMINARIES

Throughout this work, we use $n := |V|$ for the number of input articles and $m := |A|$ for the overall number of arcs in the input citation graph $D = (V, E)$. Let $\text{deg}^{\text{in}}(v)$ denote the indegree of an article v in a citation graph $D = (V, A)$, that is, v ’s number of citations. Furthermore, let $N_D^{\text{in}}(v) := \{u \mid (u, v) \in A\}$ denote the set of articles that cite v and $N_{D-W}^{\text{in}}(v) := \{u \mid (u, v) \in A \wedge u \notin W\}$ be the set of articles outside W that cite v . For each part $P \in \mathcal{P}$, the

Table 1: Computational complexity of the various variants of manipulating the h-index by splitting operations. For all FPT and W[1]-hardness results we also show NP-hardness.

†: wrt. parameter h , the h-index to achieve.
 ◊: wrt. parameter k , the number of operations.
 *: wrt. parameter $h + k + s$, where s is the largest number of articles merged into one.

Problem	sumCite / unionCite	fusionCite
Atomizing	Linear (Theorem 1)	FPT† (Theorems 5, 6)
Conservative A.	Linear (Theorem 1)	W[1]-h* (Theorem 7)
Extracting	Linear (Theorem 2)	?
Conservative E.	Linear (Theorem 2)	W[1]-h* (Corollary 1)
Cautious E.	Linear (Theorem 2)	W[1]-h* (Corollary 1)
Dividing	FPT† (Theorem 3)	NP-h (Proposition 1)
Conservative D.	FPT† (Theorem 3)	W[1]-h* (Corollary 1)
Cautious D.	W[1]-h◊ (Theorem 4)	W[1]-h* (Corollary 1)

following three measures for the number $\mu(P)$ of citations of P have been introduced [2]. They are illustrated in Figure 1. The measure

$$\text{sumCite}(P) := \sum_{v \in P} \text{deg}^{\text{in}}(v)$$

defines the number of citations of a merged article P as the sum of the citations of the atomic articles it contains. This measure was proposed by de Keijzer and Apt [13]. In contrast, the measure

$$\text{unionCite}(P) := \left| \bigcup_{v \in P} N_D^{\text{in}}(v) \right|$$

defines the number of citations of a merged article P as the number of distinct atomic articles citing at least one atomic article in P . Google Scholar uses the unionCite measure [2]. The measure

$$\text{fusionCite}(P) := \left| \bigcup_{v \in P} N_{D-W}^{\text{in}}(v) \right| + \sum_{P' \in \mathcal{P} \setminus \{P\}} \begin{cases} 1 & \text{if } \exists v \in P' \exists w \in P : (v, w) \in A, \\ 0 & \text{otherwise} \end{cases}$$

is perhaps the most natural one: at most one citation of a part $P' \in \mathcal{P}$ to a part $P \in \mathcal{P}$ is counted, that is, we additionally remove duplicate citations between merged articles and self-citations of merged articles.

Our theoretical analysis is in the framework of parameterized complexity [3, 5, 9, 15]. That is, for those problems that are NP-hard, we study the influence of a *parameter*, an integer associated with the input, on the computational complexity. For a problem P , we seek to decide P using a *fixed-parameter algorithm*, an algorithm with running time $f(p) \cdot |q|^{O(1)}$, where q is the input and $f(p)$ a computable function depending only on the parameter p . If such an algorithm exists, then P is *fixed-parameter tractable* (FPT) with respect to p . W[1]-hard parameterized problems presumably do not admit FPT algorithms. For instance, to find an order- k clique in an undirected graph is known to be W[1]-hard for the parameter k . W[1]-hardness of a problem P parameterized by p can be shown via a *parameterized reduction* from a known W[1]-hard problem Q parameterized by q . That is, a reduction that runs in $f(q) \cdot n^{O(1)}$ time on input of size n with parameter q and produces instances that satisfy $p \leq f(q)$ for some function f .

3 SUM CITE AND UNION CITE

In this section, we study the sumCite and unionCite measures. We provide linear-time algorithms for atomizing and extracting and analyze the parameterized complexity of dividing with respect to the

Algorithm 1: Atomizing

Input: A citation graph $D = (V, A)$, a set $W \subseteq V$ of articles, a partition \mathcal{P} of W , a nonnegative integer h and a measure μ .

Output: A partition \mathcal{R} of W .

```

1  $\mathcal{R} \leftarrow \emptyset$ 
2 foreach  $P \in \mathcal{P}$  do
3    $\mathcal{A} \leftarrow \text{Atomize}(P)$ 
4   if  $\exists A \in \mathcal{A} : \mu(A) \geq h$  then  $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{A}$ 
5   else  $\mathcal{R} \leftarrow \mathcal{R} \cup \{P\}$ 
6 return  $\mathcal{R}$ 

```

Algorithm 2: Conservative Atomizing

Input: A citation graph $D = (V, A)$, a set $W \subseteq V$ of articles, a partition \mathcal{P} of W , nonnegative integers h and k , and a measure μ .

Output: A partition \mathcal{R} of W .

```

1  $\mathcal{R} \leftarrow \mathcal{P}$ 
2 foreach  $P \in \mathcal{P}$  do
3    $\ell_P \leftarrow 0$ 
4    $\mathcal{A} \leftarrow \text{Atomize}(P)$ 
5    $\ell_P \leftarrow \ell_P + |\{A \in \mathcal{A} \mid \mu(A) \geq h\}|$ 
6   if  $\mu(P) \geq h$  then  $\ell_P \leftarrow \ell_P - 1$ 
7 for  $i \leftarrow 1$  to  $k$  do
8    $P^* \leftarrow \arg \max_{P \in \mathcal{P}} \{\ell_P\}$ 
9   if  $\ell_{P^*} > 0$  then
10     $\mathcal{A} \leftarrow \text{Atomize}(P^*)$ 
11     $\mathcal{R} \leftarrow (\mathcal{R} \setminus \{P^*\}) \cup \mathcal{A}$ 
12     $\ell_{P^*} \leftarrow -1$ 
13 return  $\mathcal{R}$ 

```

number k of splits and the h-index h to achieve. In our results for sumCite and unionCite, we often tacitly use the observation that local changes to the merged articles do not influence the citations of other merged articles.

Manipulation by Atomizing. Recall that the atomizing operation splits a merged article into singletons and that, for the atomizing operation, the notions of *conservative* (touching few articles) and *cautious* (making few operations) manipulation coincide and are thus both captured by CONSERVATIVE ATOMIZING. Both ATOMIZING and CONSERVATIVE ATOMIZING are solvable in linear time. Intuitively, it suffices to find the merged articles which, when atomized, increase the number of articles with at least h citations the most. This leads to Algorithms 1 and 2 for ATOMIZING and CONSERVATIVE ATOMIZING. Herein, the `Atomize()` operation takes a set S as input and returns $\{\{s\} \mid s \in S\}$. The algorithms yield the following theorem.

Theorem 1. ATOMIZING(μ) and CONSERVATIVE ATOMIZING(μ) are solvable in linear time for $\mu \in \{\text{sumCite}, \text{unionCite}\}$.

Proof. We first consider ATOMIZING(μ). Let \mathcal{R} be a partition created from a partition \mathcal{P} by atomizing a part $P^* \in \mathcal{P}$. Observe that for all $P \in \mathcal{P}$ and $R \in \mathcal{R}$ we have that $P = R$ implies $\mu(P) = \mu(R)$, for $\mu \in \{\text{sumCite}, \text{unionCite}\}$. Intuitively this means that atomizing a single part $P^* \in \mathcal{P}$ does not alter the μ -value of any other part of the partition.

Algorithm 1 computes a partition \mathcal{R} that has a maximal number of parts R with $\mu(R) \geq h$ that can be created by applying atom-

Algorithm 3: Extracting

Input: A citation graph $D = (V, A)$, a set $W \subseteq V$ of articles, a partition \mathcal{P} of W , a nonnegative integer h and a measure μ .

Output: A partition \mathcal{R} of W .

```

1  $\mathcal{R} \leftarrow \emptyset$ 
2 foreach  $P \in \mathcal{P}$  do
3   foreach  $v \in P$  do
4     if  $\mu(\{v\}) \geq h$  then
5        $\mathcal{R} \leftarrow \mathcal{R} \cup \{\{v\}\}$ 
6        $P \leftarrow P \setminus \{v\}$ 
7   if  $P \neq \emptyset$  then  $\mathcal{R} \leftarrow \mathcal{R} \cup \{P\}$ 
8 return  $\mathcal{R}$ 

```

izing operations to \mathcal{P} : It applies the atomizing operation to each part $P \in \mathcal{P}$ if there is at least one singleton A in the atomization of P with $\mu(A) \geq h$. By the above observation, this cannot decrease the total number of parts in the partition that have a μ -value of at least h . Furthermore, we have that for all $R \in \mathcal{R}$, we cannot potentially increase the number of parts with μ -value at least h by atomizing R . Thus, we get the maximal number of parts R with $\mu(R) \geq h$ that can be created by applying atomizing operations to \mathcal{P} .

Obviously, if \mathcal{R} has at least h parts R with $\mu(R) \geq h$, we face a yes-instance. Conversely, if the input is a yes-instance, then there is a number of atomizing operations that can be applied to \mathcal{P} such that the resulting partition \mathcal{R} has at least h parts R with $\mu(R) \geq h$.

It is easy to see that the algorithm runs in linear time and finds a yes-instance if it exists. If the output partition \mathcal{R} does not have at least h parts R with $\mu(R) \geq h$, then the input is a no-instance.

The pseudocode for solving CONSERVATIVE ATOMIZING(μ) is given in Algorithm 2. First, in Lines 2–6, for each part P , Algorithm 2 records how many singletons A with $\mu(A) \geq h$ are created when atomizing P . Then, in Lines 7–12, it repeatedly atomizes the part yielding the most such singletons. This procedure creates the maximum number of parts that have a μ -value of at least h , since the μ -value cannot be increased by exchanging one of these atomizing operations by another.

Obviously, if \mathcal{R} has at least h parts R with $\mu(R) \geq h$, we face a yes-instance. Conversely, if the input is a yes-instance, then there are k atomizing operations that can be applied to \mathcal{P} to yield an h-index of at least h . Since Algorithm 2 takes successively those operations that yield the most new parts with h citations, the resulting partition \mathcal{R} has at least h parts R with $\mu(R) \geq h$. It is not hard to verify that the algorithm has linear running time. \square

Manipulation by Extracting. Recall that the extracting operation removes a single article from a merged article. All variants of the extraction problem are solvable in linear time. Intuitively, in the cautious case, it suffices to find k extracting operations that each increase the number of articles with h citations. In the conservative case, we determine for each merged article a set of extraction operations that increases the number of articles with h citations the most. Then we use the extraction operations for those k merged articles that yield the k largest increases in the number of articles with h citations. This leads to Algorithms 3, 4, and 5 for EXTRACTING, CAUTIOUS EXTRACTING, and CONSERVATIVE EXTRACTING, respectively, which yield the following theorem.

Theorem 2. EXTRACTING(μ), CONSERVATIVE EXTRACTING(μ)

Algorithm 4: Cautious Extracting

Input: A citation graph $D = (V, A)$, a set $W \subseteq V$ of articles, a partition \mathcal{P} of W , nonnegative integers h and k , and a measure μ .

Output: A partition \mathcal{R} of W .

```

1  $\mathcal{R} \leftarrow \emptyset$ 
2 foreach  $P \in \mathcal{P}$  do
3   foreach  $v \in P$  do
4     if  $k > 0$  and  $\mu(\{v\}) \geq h$  and  $\mu(P \setminus \{v\}) \geq h$  then
5        $\mathcal{R} \leftarrow \mathcal{R} \cup \{\{v\}\}$ 
6        $P \leftarrow P \setminus \{v\}$ 
7        $k \leftarrow k - 1$ 
8   if  $P \neq \emptyset$  then  $\mathcal{R} \leftarrow \mathcal{R} \cup \{P\}$ 
9 return  $\mathcal{R}$ 

```

Algorithm 5: Conservative Extracting

Input: A citation graph $D = (V, A)$, a set $W \subseteq V$ of articles, a partition \mathcal{P} of W , nonnegative integers h and k , and a measure μ .

Output: A partition \mathcal{R} of W .

```

1 foreach  $P \in \mathcal{P}$  do
2    $\ell_P \leftarrow 0$ 
3    $\mathcal{R}_P \leftarrow \emptyset$ 
4   foreach  $v \in P$  do
5     if  $\mu(\{v\}) \geq h$  and  $\mu(P \setminus \{v\}) \geq h$  then
6        $\mathcal{R}_P \leftarrow \mathcal{R}_P \cup \{\{v\}\}$ 
7        $P \leftarrow P \setminus \{v\}$ 
8        $\ell_P \leftarrow \ell_P + 1$ 
9   if  $P \neq \emptyset$  then  $\mathcal{R}_P \leftarrow \mathcal{R}_P \cup \{P\}$ 
10  $\mathcal{P}^* \leftarrow$  the  $k$  elements of  $P \in \mathcal{P}$  with largest  $\ell_P$ -values
11  $\mathcal{R} \leftarrow \bigcup_{P \in \mathcal{P}^*} \mathcal{R}_P \cup (\mathcal{P} \setminus \mathcal{P}^*)$ 
12 return  $\mathcal{R}$ 

```

and CAUTIOUS EXTRACTING(μ) are solvable in linear time for $\mu \in \{\text{sumCite}, \text{unionCite}\}$.

Proof. We first consider EXTRACTING(μ). Let \mathcal{R} be a partition produced from \mathcal{P} by extracting an article from a part $P^* \in \mathcal{P}$. Recall that this does not alter the μ -value of any other part, i.e., for all $P \in \mathcal{P}$ and $R \in \mathcal{R}$, we have that $P = R$ implies $\mu(P) = \mu(R)$ for $\mu \in \{\text{sumCite}, \text{unionCite}\}$.

Consider Algorithm 3. It is easy to see that the algorithm only performs extracting operations and that the running time is polynomial. So we have to argue that whenever there is a partition \mathcal{R} that can be produced by extracting operations from \mathcal{P} such that the h-index is at least h , then the algorithm finds a solution.

We show this by arguing that the algorithm produces the maximum number of articles with at least h citations possible. Extracting an article that has strictly less than h citations cannot produce an h-index of at least h unless we already have an h-index of at least h , because the number of articles with h or more citations does not increase. Extracting an article with h or more citations cannot decrease the number of articles with h or more citations. Hence, if there are no articles with at least h citations that we can extract, we cannot create more articles with h or more citations. Therefore, we have produced the maximum number of articles with h or more citations when the algorithm stops.

The pseudocode for solving `CAUTIOUS EXTRACTING(μ)` is given in Algorithm 4. We perform up to k extracting operations (Line 6). Each of them increases the number of articles that have h or more citations by one. As Algorithm 4 checks each atomic article in each merged article, it finds k extraction operations that increase the number of articles with h or more citations if they exist. Thus, it produces the maximum-possible number of articles that have h or more citations and that can be created by k extracting operations.

To achieve linear running time, we need to efficiently compute $\mu(P \setminus \{v\})$ in Line 4. This can be done by representing articles as integers and using an n -element array A which stores throughout the loop in Line 3, for each article $v \in N_D^{\text{in}}[P]$, the number $A[w]$ of articles in P that are cited by w . Using this array, one can compute $\mu(P \setminus \{v\})$ in $O(\deg^{\text{in}}(v))$ time in Line 4, amounting to overall linear time. The time needed to maintain array A is also linear: We initialize it once in the beginning with all zeros. Then, before entering the loop in Line 3, we can in $O(|N_D^{\text{in}}(P)|)$ total time store for each article $v \in N_D^{\text{in}}[P]$, the number $A[w]$ of articles in P that are cited by w . To update the array within the loop in Line 3, we need $O(\deg^{\text{in}}(v))$ time if Line 6 applies. In total, this is linear time.

Finally, the pseudocode for solving `CONSERVATIVE EXTRACTING(μ)` is given in Algorithm 5. For each merged article $P \in \mathcal{P}$, Algorithm 5 computes a set \mathcal{R}_P and the number ℓ_P of additional articles v with $\mu(v) \geq h$ that can be created by extracting. Then it chooses a set \mathcal{P}^* of k merged articles $P \in \mathcal{P}$ with maximum ℓ_P and, from each $P \in \mathcal{P}^*$, extracts the articles in \mathcal{R}_P .

This procedure creates the maximum number of articles that have a μ -value of at least h while only performing extraction operations on at most k merges.

Obviously, if the solution \mathcal{R} has at least h parts R with $\mu(R) \geq h$, then we face a yes-instance. Conversely, if the input is a yes-instance, then there are k merged articles that we can apply extraction operations to, such that the resulting partition \mathcal{R} has at least h parts R with $\mu(R) \geq h$. Since the algorithm produces the maximal number of parts R with $\mu(R) \geq h$, it achieves an h -index of at least h .

The linear running time follows by implementing the check in line 5 in $O(\deg^{\text{in}}(v))$ time as described for Algorithm 4 and by using counting sort to find the k parts to extract from in line 10. \square

Manipulation by Dividing. Recall that the dividing operation splits a merged article into two arbitrary parts. First we consider the basic and the conservative case and show that they are FPT when parameterized by the h -index. Then we show that the cautious variant is W[1]-hard when parameterized by k . `DIVIDING(μ)` is closely related to `H-INDEX MANIPULATION(μ)` [2, 13] which is, given a citation graph $D = (V, A)$, a subset of articles $W \subseteq V$, and a non-negative integer h , to decide whether there is a partition \mathcal{P} of W such that \mathcal{P} has h -index h with respect to μ . De Keijzer and Apt [13] showed that `H-INDEX MANIPULATION(sumCite)` is NP-hard, even if merges are unconstrained. The hardness of `H-INDEX MANIPULATION` for $\mu \in \{\text{unionCite}, \text{fusionCite}\}$ follows. We can reduce `H-INDEX MANIPULATION` to `CONSERVATIVE DIVIDING` by defining the partition $\mathcal{P} = \{W\}$, hence we get the following.

Proposition 1. `DIVIDING` and `CONSERVATIVE DIVIDING` are NP-hard for $\mu \in \{\text{sumCite}, \text{unionCite}, \text{fusionCite}\}$.

As to computational tractability, `DIVIDING` and `CONSERVATIVE DIVIDING` are FPT when parameterized by h —the h -index to achieve.

Theorem 3. `DIVIDING` and `CONSERVATIVE DIVIDING(μ)` can be solved in $2^{O(h^4 \log h)} \cdot n^{O(1)}$ time, where h is the h -index to achieve and $\mu \in \{\text{sumCite}, \text{unionCite}\}$.

Algorithm 6: Conservative Dividing

Input: A citation graph $D = (V, A)$, a set $W \subseteq V$ of articles, a partition \mathcal{P} of W , nonnegative integers h and k , and a measure μ .

Output: `true` if k dividing operations can be applied to \mathcal{P} to yield h -index h and `false` otherwise.

```

1 foreach  $P \in \mathcal{P}$  do
2    $D' \leftarrow$  The graph obtained from  $D$  by removing all
   citations  $(u, v)$  such that  $v \notin P$  and adding
    $h + 1$  articles  $r_1, \dots, r_{h+1}$ 
3    $W' \leftarrow P, \ell_P \leftarrow 0$ 
4   for  $i \leftarrow 0$  to  $h$  do
5     if Merge ( $D', W', h, \mu$ ) then
6        $\ell_P \leftarrow h - i$ 
7       Break
8     Add  $r_i$  to  $W'$  and add each citation  $(r_i, r_j)$ ,
        $j \in \{1, \dots, h + 1\} \setminus \{i\}$  to  $D'$ 
9 return  $\exists \mathcal{P}' \subseteq \mathcal{P}$  s.t.  $|\mathcal{P}'| \leq k$  and  $\sum_{P \in \mathcal{P}'} \ell_P \geq h$ 

```

Proof. The pseudocode is given in Algorithm 6. Herein, `Merge` (D, W, h, μ) decides `H-INDEX MANIPULATION(μ)`, that is, it returns `true` if there is a partition \mathcal{Q} of W such that \mathcal{Q} has h -index h and `false` otherwise. It follows from van Bevern et al. [2, Theorem 7] that `Merge` can be carried out in $2^{O(h^4 \log h)} \cdot n^{O(1)}$ time.

Algorithm 6 first finds, using `Merge`, the maximum number ℓ_P of (merged) articles with at least h citations that we can create in each part $P \in \mathcal{P}$. For this, we first prepare an instance (D', W', h, μ) of `H-INDEX MANIPULATION(μ)` in Lines 2 and 3. In the resulting instance, we ask whether there is a partition of P with h -index h . If this is the case, then we set ℓ_P to h and, otherwise, we add one artificial article with h citations to W' in Line 8. Then we use `Merge` again and we iterate this process until `Merge` returns `true`, or we find that there is not even one merged article contained in P with h citations. Clearly, this process correctly computes ℓ_P . Thus, the algorithm is correct. The running time is clearly dominated by the calls to `Merge`. Since `Merge` runs in $2^{O(h^4 \log h)} \cdot n^{O(1)}$ time [2, Theorem 7], the running time bound follows. \square

We note that `Merge` can be modified so that it outputs the desired partition. Hence, we can modify Algorithm 6 to output the actual solution. Furthermore, for $k = n$, Algorithm 6 solves the non-conservative variant, which is therefore also fixed-parameter tractable parameterized by h .

In contrast, for the cautious variant we show W[1]-hardness when parameterized by k , the number of allowed operations.

Theorem 4. `CAUTIOUS DIVIDING(μ)` is NP-hard and W[1]-hard when parameterized by k for $\mu \in \{\text{sumCite}, \text{unionCite}, \text{fusionCite}\}$, even if the citation graph is acyclic.

Proof. We reduce from the `UNARY BIN PACKING` problem: given a set S of n items with integer sizes $s_i, i \in \{1, \dots, n\}$, ℓ bins and a maximum bin capacity B , can we distribute all items into the ℓ bins? Herein, all sizes are encoded in unary. `UNARY BIN PACKING` parameterized by ℓ is W[1]-hard [11].

Given an instance (S, ℓ, B) of `UNARY BIN PACKING`, we produce an instance $(D, W, \mathcal{P}, h, \ell - 1)$ of `CAUTIOUS DIVIDING(sumCite)`. Let $s^* = \sum_i s_i$ be the sum of all item sizes. We assume that $B < s^*$ and $\ell \cdot B \geq s^*$ as, otherwise, the problem is trivial, since all items fit into one bin or they collectively cannot fit into all bins, respectively.

Furthermore, we assume that $\ell < B$ since, otherwise, the instance size is upper bounded by a function of ℓ and, hence, is trivially FPT with respect to ℓ . We construct the instance of CAUTIOUS DIVIDING(sumCite) in polynomial time as follows.

- Add s^* articles x_1, \dots, x_{s^*} to D . These are only used to increase the citation count of other articles.
- Add one article a_i to D and W for each s_i .
- For each article a_i , add citations (x_j, a_i) for all $1 \leq j \leq s_i$ to G . Note that, after adding these citations, each article a_i has citation count s_i .
- Add $\Delta := \ell \cdot B - s^*$ articles u_1, \dots, u_Δ to D and W .
- For each article u_i with $i \in \{1, \dots, \Delta\}$, add a citation (x_1, u_i) to D . Note that each article u_i has citation count 1.
- Add $B - \ell$ articles $h_1, \dots, h_{B-\ell}$ to D and W .
- For each article h_i with $i \in \{1, \dots, B - \ell\}$, add citations (x_j, h_i) for all $1 \leq j \leq B$ to D . Note that each article h_i has citation count B .
- Add $P^* = \{a_1, \dots, a_n, u_1, \dots, u_\Delta\}$ to \mathcal{P} , for each article h_i with $i \in \{1, \dots, B - \ell\}$, add $\{h_i\}$ to \mathcal{P} , and set $h = B$.

Now we show that (S, ℓ, B) is a yes-instance if and only if $(D, W, \mathcal{P}, h, \ell - 1)$ is a yes-instance.

(\Rightarrow) Assume that (S, ℓ, B) is a yes-instance and let S_1, \dots, S_ℓ be a partition of S such that items in S_i are placed in bin i . Now we split P^* into ℓ parts R_1, \dots, R_ℓ in the following way. Note that for each S_i , we have that $\sum_{s_j \in S_i} s_j = B - \delta_i$ for some $\delta_i \geq 0$. Furthermore, $\sum_i \delta_i = \Delta$. Recall that there are Δ articles u_1, \dots, u_Δ in P^* . Let $\delta_{<i} = \sum_{j < i} \delta_j$ and $U_i = \{u_{\delta_{<i}+1}, \dots, u_{\delta_{<i}+\delta_i}\}$, with $\delta_0 = 0$ and if $\delta_i > 0$, let $U_i = \emptyset$ for $\delta_i = 0$. We set $R_i = \{a_j \mid s_j \in S_i\} \cup U_i$. Then for each R_i , we have that $\text{sumCite}(R_i) = \text{sumCite}(\{a_j \mid s_j \in S_i\}) + \text{sumCite}(U_i)$, which simplifies to $\text{sumCite}(R_i) = \sum_{s_j \in S_i} s_j + \delta_i = B$. For each i , $1 \leq i \leq n$, we have $\text{sumCite}(\{h_i\}) = B$. Hence, $\mathcal{R} = \{R_1, \dots, R_\ell, \{h_1\}, \dots, \{h_{B-\ell}\}\}$ has h-index B .

(\Leftarrow) Assume that $(D, W, \mathcal{P}, h, \ell - 1)$ is a yes-instance and let \mathcal{R} be a partition with h-index h . Recall that \mathcal{P} consists of P^* and $B - \ell$ singletons $\{h_1\}, \dots, \{h_{B-\ell}\}$, which are hence also contained in \mathcal{R} . Furthermore, $\text{sumCite}(\{h_i\}) = B$ for each h_i and, by the definition of the h-index, there are ℓ parts R_1, \dots, R_ℓ with $R_i \subset P^*$ and $\text{sumCite}(R_i) \geq B$ for each i . Since, by definition, $\text{sumCite}(P^*) = \ell \cdot B$ and $\text{sumCite}(P^*) = \sum_{1 \leq i \leq \ell} \text{sumCite}(R_i)$ we have that $\text{sumCite}(R_i) = B$ for all i . It follows that $\text{sumCite}(R_i \setminus \{u_1, \dots, u_\Delta\}) \leq B$ for all i . This implies that packing into bin i each item in $\{s_j \mid a_j \in R_i\}$ solves the instance (S, ℓ, B) .

Note that this proof can be modified to cover also the unionCite and the fusionCite case by adding $\ell \cdot s^*$ extra x -articles and ensuring that no two articles in W are cited by the same x -article. \square

4 FUSION CITE

We now consider the fusionCite measure, which makes manipulation considerably harder than the other measures. In particular, we obtain that even in the most basic case, the manipulation problem is NP-hard.

Theorem 5. *ATOMIZING(fusionCite) is NP-hard, even if the citation graph is acyclic.*

Proof. We reduce from the NP-hard 3-SAT problem: given a 3-CNF formula F with n variables and m clauses, decide whether F allows for a satisfying truth assignment to its variables. Without loss

of generality, we assume $n + m > 3$. Given a formula F with variables x_1, \dots, x_n and clauses c_1, \dots, c_m such that $n + m > 3$, we produce an instance $(D, W, \mathcal{P}, m + n)$ of ATOMIZING(fusionCite) in polynomial time as follows.

For each variable x_i of F , add to D and W sets $\mathcal{X}_i^F := \{X_{i,1}^F, \dots, X_{i,2(n+m)}^F\}$ and $\mathcal{X}_i^T := \{X_{i,1}^T, \dots, X_{i,2(n+m)}^T\}$ of *variable articles*. Add \mathcal{X}_i^F and \mathcal{X}_i^T to \mathcal{P} and, for $1 \leq \ell \leq n + m$, add citations $(X_{i,\ell}^F, X_{i,2\ell}^T)$ and $(X_{i,\ell}^T, X_{i,2\ell}^F)$ to D . Next, for each clause c_i of F , add a *clause article* C_i to D , to W , and add $\{C_i\}$ to \mathcal{P} . Finally, if a positive literal x_i occurs in a clause c_j , then add citations $(X_{i,\ell}^T, C_j)$ to D for $1 \leq \ell \leq n + m$. If a negative literal $\neg x_i$ occurs in a clause c_j , then add citations $(X_{i,\ell}^F, C_j)$ to D for $1 \leq \ell \leq n + m$. This concludes the construction. Observe that D is acyclic since all citations go from variable articles to clause articles or to variable articles with a higher index. It remains to show that F is satisfiable if and only if $(D, W, \mathcal{P}, m + n)$ is a yes-instance.

(\Rightarrow) If F is satisfiable, then a solution \mathcal{R} for $(D, W, \mathcal{P}, m + n)$ looks as follows: for each $i \in \{1, \dots, n\}$, if x_i is true, then we put $\mathcal{X}_i^F \in \mathcal{R}$ and we put $\mathcal{X}_i^T \notin \mathcal{R}$ otherwise. All other articles of D are added to \mathcal{R} as singletons. We count the citations that every part of \mathcal{R} gets from other parts of \mathcal{R} . If x_i is true, then \mathcal{X}_i^F gets $m + n$ citations from $\{X_{i,\ell}^T\}$ for $1 \leq \ell \leq n + m$. Moreover, for the clause c_j containing the literal x_i , $\{C_j\}$ gets $n + m$ citations from $\{X_{i,\ell}^T\}$ for $1 \leq \ell \leq n + m$. Similarly, if x_i is false, then $\{X_{i,\ell}^T\}$ gets $m + n$ citations and so does every $\{C_j\}$ for each clause c_j containing the literal $\neg x_i$. Since every clause is satisfied and every variable is either true or false, it follows that each of the m clause articles gets $m + n$ citations and that, for each of the n variables x_i , either \mathcal{X}_i^F or \mathcal{X}_i^T gets $m + n$ citations. It follows that $m + n$ parts of \mathcal{R} get at least $m + n$ citations and thus, that \mathcal{R} has h-index at least $m + n$.

(\Leftarrow) Let \mathcal{R} be a solution for $(D, W, \mathcal{P}, m + n)$. We first show that, for each variable x_i , we have either $\mathcal{X}_i^T \in \mathcal{R}$ or $\mathcal{X}_i^F \in \mathcal{R}$. To this end, it is important to note two facts:

1. For each variable x_i , every variable article in $\mathcal{X}_i^T \cup \mathcal{X}_i^F$ has at most one incoming arc in D . Thus, no (singleton) variable article in \mathcal{R} can get $m + n$ citations.
2. If, for some variable x_i , the part $\mathcal{X}_i^T \in \mathcal{R}$ gets $m + n$ citations, then $\mathcal{X}_i^F \notin \mathcal{R}$ and vice versa.

Thus, since there are at most m clause articles and \mathcal{R} contains $m + n$ parts with $m + n$ citations, \mathcal{R} contains exactly one of the parts $\mathcal{X}_i^T, \mathcal{X}_i^F$ of each variable x_i . It follows that, in \mathcal{R} , all singleton clause articles have to receive $m + n$ citations. Thus, for each clause c_j , there is a literal x_i in c_j or a literal $\neg x_i$ in c_j such that $\mathcal{X}_i^T \in \mathcal{R}$ or $\mathcal{X}_i^F \in \mathcal{R}$, respectively. It follows that setting each x_i to true if and only if $\mathcal{X}_i^T \in \mathcal{R}$ gives a satisfying truth assignment to the variables of F . \square

This NP-hardness result motivates the search for fixed-parameter tractability.

Theorem 6. *ATOMIZING(fusionCite) can be solved in $O(4^{h^2}(n + m))$ time, where h is the h-index to achieve.*

Proof. We use the following procedure to solve an instance (D, W, \mathcal{P}, h) of ATOMIZING(fusionCite).

Let $\mathcal{P}_{\geq h}$ be the set of merged articles $P \in \mathcal{P}$ with $\text{fusionCite}(P) \geq h$. If $|\mathcal{P}_{\geq h}| \geq h$, then we face a yes-instance and output “yes”. To see that we can do this in linear time, note that, given \mathcal{P} , we can compute $\text{fusionCite}(P)$ in linear time for each $P \in \mathcal{P}$. Below we assume that $|\mathcal{P}_{\geq h}| < h$.

First, we atomize all $P \in \mathcal{P}$ that cannot have h or more citations, that is, for which, even if we atomize all merged articles except

for P , we have $\text{fusionCite}(P) < h$. Formally, we atomize P if $\sum_{v \in P} |N_{D-P}^{\text{in}}(v)| < h$. Let \mathcal{P}' be the partition obtained from \mathcal{P} after these atomizing operations; note that \mathcal{P}' can be computed in linear time.

The basic idea is now to look at all remaining merged articles that receive at least h citations from atomic articles; they form the set $\mathcal{P}_{<h}$ below. They are cited by at most $h - 1$ other merged articles. Hence, if the size of $\mathcal{P}_{<h}$ exceeds some function $f(h)$, then, among the contained merged articles, we find a large number of merged articles that do not cite each other. If we have such a set, then we can atomize all other articles, obtaining h-index h . If the size of $\mathcal{P}_{<h}$ is smaller than $f(h)$, then we can determine by brute force whether there is a solution.

Consider all merged articles $P \in \mathcal{P}'$ that have less than h citations but can obtain h or more citations by applying atomizing operations to merged articles in \mathcal{P}' . Let us call the set of these merged articles $\mathcal{P}_{<h}$. Formally, $P \in \mathcal{P}_{<h}$ if $\sum_{v \in P} |N_{D-P}^{\text{in}}(v)| \geq h$ and $\text{fusionCite}(P) < h$. Again, $\mathcal{P}_{<h}$ can be computed in linear time. Note that $\mathcal{P}' \setminus (\mathcal{P}_{\geq h} \cup \mathcal{P}_{<h})$ consists only of singletons.

Now, we observe the following. If there is a set $\mathcal{P}^* \subseteq \mathcal{P}_{<h}$ of at least h merged articles such that, for all $P_i, P_j \in \mathcal{P}^*$, neither P_i cites P_j nor P_j cites P_i , then we can atomize all merged articles in $\mathcal{P}' \setminus \mathcal{P}^*$ to reach an h-index of at least h . We finish the proof by showing that we can conclude the existence of the set \mathcal{P}^* if $\mathcal{P}_{<h}$ is sufficiently large and solve the problem using brute force otherwise.

Consider the undirected graph G that has a vertex v_P for each $P \in \mathcal{P}_{<h}$ and an edge between v_{P_i} and v_{P_j} if P_i cites P_j or P_j cites P_i . Note that $\{v_P \mid P \in \mathcal{P}^*\}$ forms an independent set in G . Furthermore, let I be an independent set in G that has size at least h . Let $\mathcal{P}^{**} = \{P \in \mathcal{P}_{<h} \mid v_P \in I\}$. Then, we can atomize all merged articles in $\mathcal{P}' \setminus \mathcal{P}^{**}$ to reach an h-index of at least h .

We claim that the number of edges in G is at most $(h - 1) \cdot |\mathcal{P}_{<h}|$. This is because the edge set of G can be enumerated by enumerating for every vertex v_P the edges incident with v_P that result from a citation of P from another $P' \in \mathcal{P}_{<h}$. The citations for each P are less than h as, otherwise, we would have that $P \in \mathcal{P}_{\geq h}$. Now, we can make use of Turán's Theorem, which can be stated as follows: If a graph with ℓ vertices has at most $\ell k/2$ edges, then it admits an independent set of size at least $\ell/(k + 1)$ [12, Exercise 4.8]. Hence, if $|\mathcal{P}_{<h}| \geq 2h^2 - h$, then we face a yes-instance and we can find a solution by taking an arbitrary subset $\mathcal{P}'_{<h}$ of $\mathcal{P}_{<h}$ with $|\mathcal{P}'_{<h}| = 2h^2 - h$, by atomizing every merged article outside of $\mathcal{P}'_{<h}$, and by guessing which merged articles we need to atomize inside of $\mathcal{P}'_{<h}$. If $|\mathcal{P}_{<h}| < 2h^2 - h$, then we guess which merged articles in $\mathcal{P}_{<h} \cup \mathcal{P}_{\geq h}$ we need to atomize to obtain a solution if it exists. In both cases, for each guess we need linear time to determine whether we have found a solution, giving the overall running time of $O(4^{h^2} \cdot (m + n))$. \square

For the conservative variant, however, we cannot achieve FPT, even if we add the number of atomization operations and the maximum size of a merged article to the parameter.

Theorem 7. *CONSERVATIVE ATOMIZING(fusionCite) is NP-hard and W[1]-hard when parameterized by $h + k + s$, where $s := \max_{P \in \mathcal{P}} |P|$, even if the citation graph is acyclic.*

Proof. We reduce from the CLIQUE problem: given a graph G and an integer k , decide whether G contains a clique on at least k vertices. CLIQUE parameterized by k is known to be W[1]-hard.

Given an instance (G, k) of CLIQUE, we produce an instance $(D, W, \mathcal{P}, h, k)$ of CONSERVATIVE ATOMIZING(fusionCite) in polynomial time as follows. Without loss of generality, we assume $k \geq 4$ so that $\binom{k}{2} \geq 4$. For each vertex v of G , introduce a

set R_v of $\lceil \binom{k}{2} / 2 \rceil$ vertices to D and W and add R_v as a part to \mathcal{P} . For an edge $\{v, w\}$ of G , add to D and W a vertex $e_{\{v, w\}}$ and add $\{e_{\{v, w\}}\}$ to \mathcal{P} . Moreover, add a citation from each vertex in $R_v \cup R_w$ to $e_{\{v, w\}}$. Finally, set $h := \binom{k}{2}$. Each of h, k and s in our constructed instance of CONSERVATIVE ATOMIZING(fusionCite) depends only on k in the input CLIQUE instance. It remains to show that (G, k) is a yes-instance for CLIQUE if and only if $(D, W, \mathcal{P}, h, k)$ is.

(\Rightarrow) Assume that (G, k) is a yes-instance and let S be a clique in G . Then, atomizing R_v for each $v \in S$ yields $\binom{k}{2}$ articles with at least $\binom{k}{2}$ citations in D : for each of the $\binom{k}{2}$ pairs of vertices $v, w \in S$, the vertex $e_{\{v, w\}}$ gets $\lceil \binom{k}{2} / 2 \rceil$ citations from the vertices in R_v and the same number of citations from the vertices in R_w and, thus, at least $\binom{k}{2}$ citations in total.

(\Leftarrow) Assume that $(D, W, \mathcal{P}, h, k)$ is a yes-instance and let \mathcal{R} be a solution. We construct a subgraph $S = (V_S, E_S)$ of G that is a clique of size k . Let $V_S := \{v \in V(G) \mid R_v \in \mathcal{P} \setminus \mathcal{R}\}$ and $E_S := \{\{v, w\} \in E(G) \mid \{v, w\} \subseteq V_S\}$, that is, $S = G[V_S]$. Obviously, $|V_S| \leq k$. It remains to show $|E_S| \geq \binom{k}{2}$, which implies both that $|V_S| = k$ and that S is a clique. To this end, observe that the only vertices with incoming citations in D are the vertices $e_{\{v, w\}}$ for the edges $\{v, w\}$ of G . The only citations of a vertex $e_{\{v, w\}}$ are from the parts R_v and R_w in \mathcal{P} . That is, with respect to the partition \mathcal{P} , each vertex $e_{\{v, w\}}$ has two citations. Since the h-index h to reach is $\binom{k}{2}$, at least $\binom{k}{2}$ vertices $e_{\{v, w\}}$ have to receive $\binom{k}{2} \geq 4$ citations, which is only possible by atomizing both R_v and R_w . That is, for at least $\binom{k}{2}$ vertices $e_{\{v, w\}}$, we have $\{R_v, R_w\} \subseteq \mathcal{P} \setminus \mathcal{R}$ and, thus, $v, w \subseteq V_S$ and $\{v, w\} \in E_S$. It follows that $|E_S| \geq \binom{k}{2}$. \square

The reduction given above easily yields the same hardness result for most other problem variants: a vertex $e_{\{v, w\}}$ receives a sufficient number of citations only if R_v and R_w are atomized. Hence, even if we allow extractions or divisions on R_v , it helps only if we extract or split off all articles in R_v . The only difference is that the number of allowed operations is set to $k \cdot (\lceil \binom{k}{2} / 2 \rceil - 1)$ for these two problem variants. By the same argument, we obtain hardness for the conservative variants.

Corollary 1. For $\mu = \text{fusionCite}$, CONSERVATIVE EXTRACTING(μ), CAUTIOUS EXTRACTING(μ), CONSERVATIVE DIVIDING(μ), and CAUTIOUS DIVIDING(μ) are NP-hard and W[1]-hard when parameterized by $h + k + s$, where $s := \max_{P \in \mathcal{P}} |P|$, even if the citation graph is acyclic.

5 COMPUTATIONAL EXPERIMENTS

To assess how much the h-index of a researcher can be manipulated by splitting articles, we performed computational experiments with data extracted from Google Scholar.

Description of the Data. We use three data sets provided by van Bevern et al. [2]. One data set consists of 22 selected authors of IJCAI'13. The selection of these authors was biased to obtain profiles of authors in their early career. More precisely, the selected authors have a Google Scholar profile, an h-index between 8 and 20, between 100 and 1000 citations, and have been active between 5 and 10 years. The other two data sets contain Google Scholar data of 'AI's 10 to Watch', a list of young accomplished researchers in AI compiled by *IEEE Intelligent Systems*. One data set contains five profiles from the 2011 edition, the other eight profiles from the 2013 edition of the list.

Generation of Profiles with Merged Articles. In our setting, the input consists of a profile which already contains some merged articles.

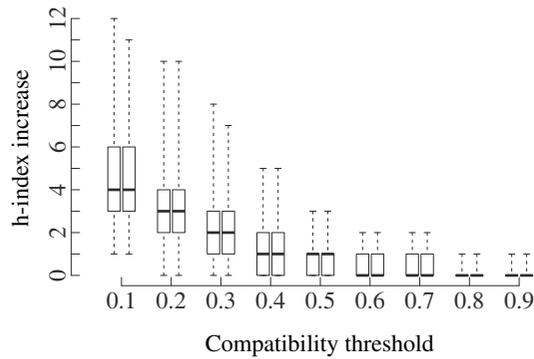


Figure 3: For each compatibility threshold, the left box shows maximum h-index increases for sumCite, the right box for unionCite.

To obtain such merged profiles, we used the compatibility graphs for each profile provided by van Bevern et al. [2], which they generated as follows. For each article u let $T(u)$ denote the set of words in its title. There is an edge between articles u and v if $|T(u) \cap T(v)| \geq t \cdot |T(u) \cup T(v)|$, where $t \in [0, 1]$ is the *compatibility threshold*. For $t = 0$, the compatibility graph is a clique; for $t = 1$ only articles with the same words in the title are adjacent. For $t \leq 0.3$, very dissimilar articles are still considered compatible [2]. Hence, we focus on $t \geq 0.4$ below.

For each profile and corresponding compatibility graph G , we obtained a profile with merged articles as follows. While the compatibility graph G contains an edge, compute a maximal clique C by a greedy algorithm, add C as a merged article to the profile, remove C from G , and continue. If the compatibility graph has no edge, then add all remaining articles as atomic articles of the profile.

Experimental Results. We implemented Algorithms 2, 4, and 5—the exact, linear-time algorithms from Section 3 for CONSERVATIVE ATOMIZING, CONSERVATIVE EXTRACTING, and CAUTIOUS EXTRACTING, respectively, each for both the sumCite and unionCite measures. Using them, we computed the maximum-possible h-index increases under the respective restrictions. The implementation is in Python 2.7.10 under Ubuntu Linux 15.10. Using an Intel Xeon E3-1231 CPU with 3.4 GHz and 32 GB RAM, the instances could be solved within three minutes altogether.

Figure 3 shows h-index increases for the IJCAI’13 authors for extracting articles: the lower edge of a box is the first quartile, the upper edge the third quartile, and the thick bar is the median; whiskers extend to the maximum and minimum values. Note that the h-index increase achievable by extracting articles is always at least as large as the one for atomizing articles. For the IJCAI’13 authors, atomizing articles yields essentially the same curve/same results as in Figure 3. Qualitatively, the results for AI’s 10 to Watch 2013 are the same, whereas AI’s 10 to Watch 2011 can achieve larger h-index increases for compatibility threshold 0.1. Hence, supposing that compatibility thresholds of at least 0.4 yield realistic profiles, we can conclude that 25% of the authors could improve their h-index by unmerging articles by at least two and some outliers by five.

The results concerning the influence of restrictions on the number of operations and number of touched merged articles are as follows. For atomizing articles, most authors can increase their h-indices in increments of one for each atomizing operation up to their individual maximum. There is, however, one IJCAI’13 author who can achieve an increment of five with one atomizing operation ($t = 0.4$, sumCite). For extracting articles, clearly, each operation can increase the h-index by at most one. The results for CAUTIOUS EXTRACTING

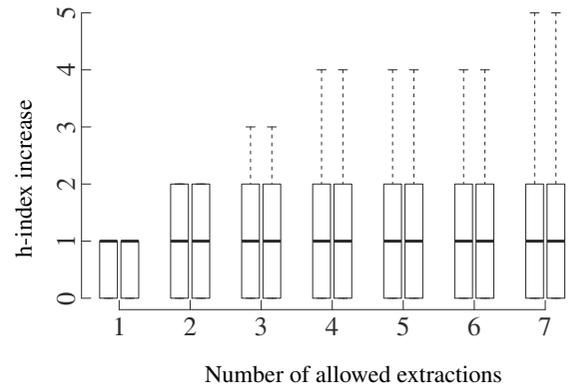


Figure 4: For each upper bound k on the number of allowed article extractions, the left box shows the maximum h-index increases for sumCite, the right box for unionCite.

over the IJCAI’13 authors are shown in Figure 4 for compatibility threshold 0.4. Interestingly, in the experiments for CONSERVATIVE EXTRACTING and $t = 0.4$, all selected IJCAI’13 authors can achieve their maximum h-index increases by extracting articles out of at most two merged articles. In general, for threshold at least 0.4, they need to touch at most three merged articles to achieve the maximum h-index increase. This is also true for AI’s 10 to Watch 2013, whereas AI’s 10 to Watch 2011 can improve further by manipulating four merged articles (for $t = 0.5$).

Summarizing, our findings indicate that realistic profiles can be manipulated by splitting articles to yield h-index increases of at most two for the majority of authors. This can mean saving at least a year of work, since the average increase of the h-indices per year is 1.22 in the considered IJCAI data set. Furthermore, our findings indicate that the increase can be obtained by tampering with a small number of merged articles.

6 CONCLUSION

Regarding theory, we leave three main open questions concerning the computational complexity of EXTRACTING(fusionCite), the parameterized complexity of DIVIDING(fusionCite), as well as the parameterized complexity of CAUTIOUS DIVIDING(sumCite / unionCite) with respect to h (see Table 1), as the most immediate challenges for future work. Also, finding hardness reductions that produce more realistic instances would be desirable. From the experimental side, evaluating the potentially possible h-index increase by splitting on real merged profiles would be interesting as well as experiments using fusionCite as a measure. Moreover, it makes sense to consider the manipulation of the h-index also in context with the simultaneous manipulation of other indices (e.g., Google’s i10-index, see also Pavlou and Elkind [16]) and to look for Pareto-optimal solutions. We suspect that our algorithms easily adapt to other indices. In addition, it is natural to consider combining merging and splitting in manipulation of author profiles. Finally, from a practical point of view, our experimental results indicate that author profiles with surprisingly large h-index may be worth inspecting concerning potential manipulation.

ACKNOWLEDGEMENTS

René van Bevern is supported by the Russian Science Foundation, grant 16-11-10041, under support of which in particular the linear running times and Theorems 5 and 7 were proven. Hendrik Molter and Manuel Sorge are supported by the DFG, project DAPA (NI 369/12).

REFERENCES

- [1] Christoph Bartneck and Servaas Kokkelmans, ‘Detecting *h*-index manipulation through self-citation analysis’, *Scientometrics*, **87**(1), 85–98, (2011).
- [2] René van Bevern, Christian Komusiewicz, Rolf Niedermeier, Manuel Sorge, and Toby Walsh, ‘H-index manipulation by merging articles: Models, theory, and experiments’, *Artificial Intelligence*, (2016). Accepted for publication, available on arXiv:1412.5498.
- [3] Marek Cygan, Fedor V Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipeczuk, Michal Pilipeczuk, and Saket Saurabh, *Parameterized Algorithms*, Springer, 2015.
- [4] Emilio Delgado López-Cózar, Nicolás Robinson-García, and Daniel Torres-Salinas, ‘The Google Scholar experiment: How to index false papers and manipulate bibliometric indicators’, *Journal of the Association for Information Science and Technology*, **65**(3), 446–454, (2014).
- [5] Rod G. Downey and Michael R. Fellows, *Fundamentals of Parameterized Complexity*, Springer, 2013.
- [6] Leo Egghe, ‘Theory and practise of the *g*-index’, *Scientometrics*, **69**(1), 131–152, (2006).
- [7] Piotr Faliszewski, Edith Hemaspaandra, and Lane A. Hemaspaandra, ‘Using complexity to protect elections’, *Communications of the ACM*, **53**(11), 74–82, (2010).
- [8] Piotr Faliszewski and Ariel D. Procaccia, ‘AI’s war on manipulation: Are we winning?’, *AI Magazine*, **31**(4), 53–64, (2010).
- [9] Jörg Flum and Martin Grohe, *Parameterized Complexity Theory*, Springer, 2006.
- [10] J. E. Hirsch, ‘An index to quantify an individual’s scientific research output’, *Proceedings of the National Academy of Sciences of the United States of America*, **102**(46), 16569–16572, (2005).
- [11] Klaus Jansen, Stefan Kratsch, Dániel Marx, and Ildikó Schlotter, ‘Bin packing with fixed number of bins revisited’, *Journal of Computer and System Sciences*, **79**(1), 39–49, (2013).
- [12] Stasys Jukna, *Extremal Combinatorics - with Applications in Computer Science*, Texts in theoretical computer science, Springer, 2001.
- [13] Bart de Keijzer and Krzysztof R. Apt, ‘The H-index can be easily manipulated’, *Bulletin of the EATCS*, **110**, 79–85, (2013).
- [14] Michael Lesk, ‘How many scientific papers are not original?’, *Proceedings of the National Academy of Sciences of the United States of America*, **112**(1), 6–7, (2015).
- [15] Rolf Niedermeier, *Invitation to Fixed-Parameter Algorithms*, number 31 in Oxford Lecture Series in Mathematics and Its Applications, Oxford University Press, 2006.
- [16] Chrystalla Pavlou and Edith Elkind, ‘Manipulating citation indices in a social context’, in *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS ’16)*, pp. 32–40. IFAAMAS, (2016).
- [17] Péter Vinkler, ‘Would it be possible to increase the Hirsch-index, π -index or CDS-index by increasing the number of publications or citations only by unity?’, *Journal of Informetrics*, **7**(1), 72–83, (2013).
- [18] Gerhard J. Woeginger, ‘An axiomatic analysis of Egghe’s *g*-index’, *Journal of Informetrics*, **2**(4), 364–368, (2008).

Planning Under Uncertainty for Aggregated Electric Vehicle Charging with Renewable Energy Supply

Erwin Walraven¹ and Matthijs T. J. Spaan¹

Abstract. Renewable energy sources introduce uncertainty regarding generated power in smart grids. For instance, power that is generated by wind turbines is time-varying and dependent on the weather. Electric vehicles will become increasingly important in the development of smart grids with a high penetration of renewables, because their flexibility makes it possible to charge their batteries when renewable supply is available. Charging of electric vehicles can be challenging, however, because of uncertainty in renewable supply and the potentially large number of vehicles involved. In this paper we propose a vehicle aggregation framework which uses Markov Decision Processes to control electric vehicles and deals with uncertainty in renewable supply. We present a grouping technique to address the scalability aspects of our framework. In experiments we show that the aggregation framework maximizes the profit of the aggregator, reduces cost of customers and reduces consumption of conventionally-generated power.

1 INTRODUCTION

The emergence of renewable energy sources in electricity grids is accompanied by several challenges [29]. For instance, power produced by solar panels and wind turbines is dependent on the weather and may cause power production peaks outside the secure range of the grid. Moreover, when many consumers use cheap electricity when renewables have a high output, the grid may become significantly congested. Traditionally such problems were addressed by expensive reinforcements of the grid, but this can be very costly [34]. A recent development is intelligently controlling generation and consumption of local consumers, and thereby creating a smart distribution grid.

Smart distribution grids offer several opportunities and challenges for the field of Artificial Intelligence, such as planning and scheduling of charging of electric vehicles [24]. In order to reduce peak loads and exploit locally produced renewable energy, such as small-scale wind power, shifting flexible electric vehicle charging demand to periods with sufficient renewable supply requires planning algorithms for so-called *aggregators*. These aggregators are entities in smart distribution grids responsible for coordinating a large number of vehicles, and need to be able to deal with uncertain information regarding the availability of renewable supply.

In this paper we consider uncertain wind power production combined with the need to coordinate charging of a large number of electric vehicles (EVs), to take advantage of renewable energy and to reduce consumption of conventionally-generated power. To make sure that vehicles charge their batteries when renewable supply is available, we present an aggregation framework based on the Multiagent Markov

Decision Process (MMDP) formalism [5]. The development of such a framework poses challenges related to the number of agents involved and the uncertainty associated with renewable energy sources. The first challenge is the main topic of this paper, and for the second challenge we build upon recent work related to modeling uncertainty of renewables [35].

Our main contributions can be summarized as follows. First, we present an electric vehicle aggregation framework which coordinates charging of collection of EVs using MMDPs. Second, we describe how the computation of value functions can be combined with tree-based representations of uncertainty in renewable wind power, such that the aggregation framework naturally accounts for uncertainty in renewable supply. Third, we develop an abstraction of the original MMDP which groups vehicles based on deadlines to keep the number of joint states and actions manageable when increasing the number of vehicles. We show how the enumeration of MMDP states and actions can be limited to reduce the number of enumerated states and actions during the computation of value functions.

In experiments based on realistic data we show that our aggregation framework is able to optimize the profit of an aggregator while reducing cost of individual consumers. Moreover, we show that electric vehicles are charging when renewable supply is available, such that consumption of conventionally-generated grid power is reduced. The experiments also show that the group-based abstraction makes our framework sufficiently scalable to control vehicles in a realistically-sized street or a small neighborhood.

The structure of the paper is as follows. In Section 2 we introduce background information about aggregation in smart grids, wind forecasting and Markov Decision Processes. Section 3 formalizes the aggregated electric vehicle charging problem. We present the corresponding MMDP formulation in Section 4, and in Section 5 we discuss an abstraction of the MMDP to improve scalability. Section 6 describes our experimental results, and the remaining sections discuss related work and our conclusions.

2 BACKGROUND

In this section we provide background information about aggregation in smart grids, wind forecasting and Markov Decision Processes.

2.1 Aggregators in Smart Grids

Aggregators in electricity grids are new entities that are acting between individual customers and the utility company [13]. From the perspective of the utility company, an aggregator represents a large number of vehicles that require power to charge their batteries. EVs provide a certain amount of flexibility since typically they do not need to be charged immediately.

¹ Delft University of Technology, Delft, The Netherlands. Email: e.m.p.walraven@tudelft.nl, m.t.j.spaan@tudelft.nl.

The flexibility of EVs can be used to address grid congestion problems. For example, during the early morning and the evening the total power demand is high since many people are at home. Current distribution grids have sufficient capacity to deal with the demand of conventional devices during such periods. However, a large number of EVs require a significant amount of power for charging, for which the capacity may not be sufficient [21, 30]. Flexibility of EVs can be used to address this problem, since EV demand can be shifted to periods in which either renewable power supply or sufficient grid capacity is available [2]. Since demand shifting for a large number of EVs requires coordination, aggregators have been proposed to control flexible demand of a large number of EVs. An aggregator is responsible for the communication technology between it and the charging points, allowing for direct control and coordination of vehicles connected to the network.

Individual customers can be incentivized to participate in aggregated charging of vehicles by providing a financial compensation. For instance, customers can sell their flexibility and get a lower charging tariff in return. From an aggregator point-of-view it is important that the cost associated with the technologies and financial compensations paid to customers are less than the profits that can be made by efficiently controlling vehicles of customers.

2.2 Wind Speed Forecasting using Scenarios

Wind forecasting methods can be categorized as either physical or statistical, where the latter are suitable for short-term prediction [11]. We use a short-term forecasting method that finds analogs [31] between observed wind speed and historical wind data [35].

The average wind speed during hour t is denoted w_t , and becomes known at the start of hour $t + 1$.² At the start of hour t , wind speed forecasts $\hat{w}_t, \hat{w}_{t+1}, \dots$ can be computed as follows. Given a sequence of past observations $w_{t-b}, \dots, w_{t-2}, w_{t-1}$ of length b , we identify similar sequences in a historical dataset containing wind speed measurements based on the Euclidean distance [35]. For each identified sequence $\hat{w}_{t-b}, \dots, \hat{w}_{t-1}$, the subsequent historical wind speed measurements $\hat{w}_t, \hat{w}_{t+1}, \dots, \hat{w}_{t+y}$ provide a scenario of length y describing future wind speed.

Probabilistic wind speed forecasts can be encoded using scenario trees [7]. Scenario trees can also be combined with wind forecasting methods such as ARMA models [28], and therefore the planning methods that we present in this paper are not limited to analog-based wind forecasting. Furthermore, the size of the tree can be managed using scenario reduction techniques [10].

2.3 Markov Decision Processes

We use techniques based on the Markov Decision Process (MDP) formalism [23] and its extension to multiple agents [5]. An MDP is a tuple (S, A, P, R, T) , where S is a finite set of states and A is a finite set of actions. The function $P : S \times A \times S \rightarrow \mathbb{R}$ defines the state transition probabilities, where $P(s, a, s')$ is the probability to transition from state s to state s' after executing action a . Similarly, the function $R : S \times A \times S \rightarrow \mathbb{R}$ defines the reward function, where $R(s, a, s')$ is the immediate reward received when transitioning from state s to s' after executing action a . The feasible set of actions that can be executed in state s is denoted $A(s)$, and the MDP has a finite time horizon T . A policy is a function $\pi : S \rightarrow A$ which maps states to actions and this function can be used by a decision maker to

² Note that throughout the paper we assume hourly intervals, but our method is trivially generalized to other intervals.

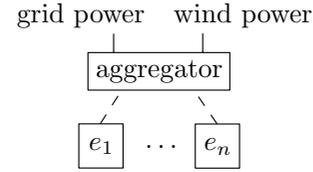


Figure 1: Vehicle aggregation with conventionally-generated grid power, wind power and n electric vehicles.

select an action for a given state. Optimal policies can be defined in terms of a value function $V^\pi : S \rightarrow \mathbb{R}$. The value of a state s under policy π , denoted by $V^\pi(s)$, is defined as the expected reward when starting from state s and following policy π thereafter. For an optimal policy π^* it holds that $V^{\pi^*}(s) \geq V^\pi(s)$ for each state $s \in S$ and for each policy π . The optimal value function of a finite-horizon MDP can be computed as follows:

$$V_t^*(s) = \max_{a \in A(s)} \sum_{s' \in S} P(s, a, s') (R(s, a, s') + V_{t+1}^*(s')), \quad (1)$$

for $t = 0, \dots, T - 1$. The corresponding time-dependent optimal policy $\pi_t^* : S \rightarrow A$ can be defined as follows:

$$\pi_t^*(s) = \arg \max_{a \in A(s)} \sum_{s' \in S} P(s, a, s') (R(s, a, s') + V_{t+1}^*(s')), \quad (2)$$

for $t = 0, \dots, T - 1$. Note that the value $V_T^*(s)$, corresponding to the final recursive step, can be defined as zero. Alternatively, it can represent a final reward corresponding to state s .

The MMDP formalism [5] generalizes MDPs to the multiagent case, in which a state $s \in S$ characterizes the joint state of the agents and actions $a \in A$ represent the joint actions that can be executed by the agents. An MMDP can still be considered as a regular MDP, and can be solved using the same algorithms (e.g., value iteration).

3 AGGREGATED EV CHARGING

We propose a vehicle aggregation framework as shown in Figure 1. The aggregator is responsible for charging n EVs and is able to use wind power generated by small-scale wind turbines in the residential area, such as wind turbines mounted on tall apartment buildings. Wind power has negligible marginal cost, and excess of wind power can be sold to the utility company. If the amount of wind power is not sufficient to charge the vehicles in time, additional conventionally-generated power can be bought from the utility company.

Now we formally introduce the optimization problem that needs to be solved by the aggregator. We consider an ordered set $E = (e_1, \dots, e_n)$ containing n electric vehicles. A vehicle e_i is connected to its charging point at the start of hour c_i , and needs to charge h_i hours before the start of hour d_i . Thus, we can define each vehicle e_i as a tuple $e_i = (c_i, d_i, h_i)$. We assume that the charging rate of each charging point is equal to z kW and that each charging point can only accommodate a single vehicle.

The aggregator is able to buy power from the utility company and pays p_t^b per kWh during hour t . If the wind turbine produces more power than needed, excess wind power can be sold to the utility company for p_t^s per kWh during hour t . The aggregator receives a fixed payment m_i from each EV $e_i \in E$ once charging has finished, which is dependent on the amount of energy used to charge the vehicle.

The power generated by the wind turbine during hour t is $g(w_t)$ kW, where w_t is the wind speed during hour t . The mapping from

wind speed to wind power can be modeled as follows [26]:

$$g(w_t) = C \cdot (1 + e^{6 - \frac{2}{3}w_t})^{-1}, \quad (3)$$

where C is the rated capacity of the wind turbine.

In order to define the objective function of the aggregator, we introduce decision variables corresponding to the charging decisions of the vehicles. Note that as the aggregator is contractually obligated to charge all vehicles by their deadline (if feasible given deadline and charge required), its payments m_i are not present in the objective function. Variable $x_{i,t}$ equals 1 if vehicle e_i charges during hour t , and is 0 otherwise. The total number of charging vehicles during hour t can be defined as $x_t = \sum_{i=1}^n x_{i,t}$. The optimization problem of the aggregator can be formulated as follows:

$$\begin{aligned} \max \quad & \sum_{t=0}^{T-1} f(x_t, w_t) \\ \text{s.t.} \quad & \sum_{t=c_i}^{d_i-1} x_{i,t} = h_i \quad i = 1, \dots, n, \end{aligned}$$

where the function f computes the benefit to be had by the aggregator when charging x_t vehicles if the wind speed is w_t during hour t . The function can be defined as follows:

$$f(x_t, w_t) = \begin{cases} p_t^s \cdot (g(w_t) - x_t \cdot z) & g(w_t) > x_t \cdot z \\ p_t^b \cdot (g(w_t) - x_t \cdot z) & \text{otherwise} \end{cases}. \quad (4)$$

Note that this function returns negative values if the amount of wind power $g(w_t)$ is not sufficient to charge x_t vehicles, because in such cases additional power needs to be bought from the utility company. The total profit of the aggregator can be defined as:

$$\sum_{i=1}^n m_i + \sum_{t=0}^{T-1} f(x_t, w_t). \quad (5)$$

If the wind speed over time and the parameters of the vehicles are known, then the optimization problem can be solved using mixed-integer programming. However, the aggregator does not know precisely how much wind power will be generated in the future, and needs to make decisions under uncertainty.

In this paper we address this problem using the MDP formalism because of two reasons. First, it allows us to conveniently separate the reasoning about exogenous wind uncertainty and the reasoning about electric vehicles, as we will show in the next section. Second, MDPs are particularly powerful in situations where the decision maker is able to control the degree of uncertainty that will be encountered in the future. For example, charging overnight before driving to work influences the uncertain demand of the vehicle at the end of the day, since the battery level upon arrival depends on the initial battery level and the distance. This paper only focuses on supply uncertainty and the problem representation, but we selected the MDP formalism based on its potential for extension to uncertainty in charging demand.

4 PLANNING FOR AGGREGATED EV CHARGING

In this section we show how the planning problem for aggregated EV charging can be formulated as a Multiagent Markov Decision Process (MMDP). First we discuss how MDP value functions can be computed in scenario trees which encode wind forecasts. Thereafter we introduce an MMDP model in which each agent represents an electric vehicle that needs to be charged.

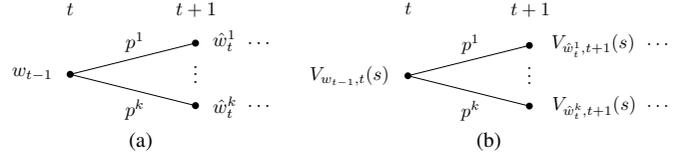


Figure 2: (a) Scenario tree representing w_{t-1} , and k branches corresponding to forecasts of w_t and their probabilities. (b) Value tree containing a value function for hour t , and k value functions for hour $t+1$.

4.1 Computing Value Functions in Scenario Trees

We use a scenario tree representation which encodes the scenarios as a tree, as illustrated in Figure 2a. The tree is constructed at the start of hour t , when w_{t-1} becomes known, and forecasted wind speed values are represented by branches j in the tree with a corresponding probability p^j . We introduce separate value functions associated with the nodes of the tree, which allows us to separate the exogenous wind uncertainty and the state transitions of the MMDP model [19]. The tree representation allows us to encode time-dependent wind forecasts, and by doing so we can avoid separate time-dependent MMDP state variables to encode wind uncertainty as part of the state transitions.

Figure 2b shows a value function $V_{w_{t-1},t}(s)$ that can be used to select an action at the start of hour t , and the corresponding tree has the same structure as the scenario tree in Figure 2a. There are k possible realizations for the wind speed during hour t , represented by $\hat{w}_t^1, \dots, \hat{w}_t^k$, and there is a probability p^j and value function $V_{\hat{w}_t^j,t+1}(s)$ corresponding to each realization. The value function $V_{w_{t-1},t}(s)$ can be computed as shown below:

$$V_{w_{t-1},t}(s) = \max_{a \in A(s)} \sum_{j=1}^k \sum_{s' \in S} (p^j \cdot P(s, a, s') \cdot (R(s, a, s', t, \hat{w}_t^j) + V_{\hat{w}_t^j,t+1}(s'))), \quad (6)$$

where the function $R(s, a, s', t, \hat{w}_t^j)$ is an augmented reward function that is also dependent on the wind speed \hat{w}_t^j during hour t . The state transitions of the MMDP model do not depend on the wind speeds, whereas the augmented reward function allows us to define a reward function that is dependent on both the state and wind speed.

The value functions for the entire scenario tree can be computed using dynamic programming, in which the value function of each node is computed using the value functions of its child nodes. In Figure 2b we show the tree for just one step ahead. However, the value functions $V_{\hat{w}_t^j,t+1}$ also need to be computed recursively based on the value functions in multiple subsequent branches. The wind forecast encoded by the scenario tree consists of a finite number of future timesteps, and therefore we have a finite planning horizon. Eventually, an optimal action can be chosen using the value function associated with the root of the tree. The tree representation of the value function corresponds to the recursive formulation in Equation 1, which we formalize below.

Proposition 1. *The value function in Equation 6 defines an optimal value function for an MDP with wind-dependent rewards, whose state transitions are independent of the wind transitions encoded by the scenario tree.*

Proof. We show that Equation 6 can be derived from Equation 1. For the purpose of the proof we make a distinction between an MMDP

state s and a global state $\langle s, t, w_{t-1} \rangle$. The MMDP state encodes the EV charging state. The global state encodes both the MMDP state as well as the wind speed during the previous time period and a time step index. Since an MMDP is an MDP, it suffices to use the equation of an optimal MDP value function in the derivation. The value function at the start of hour t maps global states to values and can be defined as follows using Equation 1:

$$V(\langle s, t, w_{t-1} \rangle) = \max_{a \in A(s)} \sum_{\langle s', t+1, \hat{w}_t^j \rangle \in Q_t} P(\langle s, t, w_{t-1} \rangle, a, \langle s', t+1, \hat{w}_t^j \rangle) \cdot \left(R(\langle s, t, w_{t-1} \rangle, a, \langle s', t+1, \hat{w}_t^j \rangle) + V(\langle s', t+1, \hat{w}_t^j \rangle) \right), \quad (7)$$

where $Q_t = \{ \langle s', t+1, \hat{w}_t^j \rangle \mid s' \in S, \hat{w}_t^j \in \{ \hat{w}_t^1, \dots, \hat{w}_t^k \} \}$ contains all possible global states at the start of hour $t+1$. The MMDP state transitions are independent of the wind transitions and the wind speed transitions are independent of the actions. Hence, it holds that $P(\langle s, t, w_{t-1} \rangle, a, \langle s', t+1, \hat{w}_t^j \rangle) = p^j \cdot P(s, a, s')$. The reward function of the MMDP model depends on the wind speed. Therefore, we define $R(\langle s, t, w_{t-1} \rangle, a, \langle s', t+1, \hat{w}_t^j \rangle) = R(s, a, s', t, \hat{w}_t^j)$ to simplify notation. The variable w_{t-1} can be left out because the reward received after hour t does not depend on the wind speed during hour $t-1$. Now the aforementioned value function can be simplified as follows:

$$V(\langle s, t, w_{t-1} \rangle) = \max_{a \in A(s)} \sum_{j=1}^k \sum_{s' \in S} p^j \cdot P(s, a, s') \cdot \left(R(s, a, s', t, \hat{w}_t^j) + V(\langle s', t+1, \hat{w}_t^j \rangle) \right). \quad (8)$$

The sum operators still define a sum over all elements in Q_t . Since the transitions of the time step counter t are assumed deterministic, the summation over all possibilities for $t+1$ can be left out. The resulting value function can be transformed to Equation 6 by defining $V(\langle s, t, w_{t-1} \rangle) = V_{w_{t-1}, t}(s)$ and $V(\langle s', t+1, \hat{w}_t^j \rangle) = V_{\hat{w}_t^j, t+1}(s')$, which is a simplification of the notation. This step completes the derivation of Equation 6 from Equation 1. An identical derivation can be used to recursively transform the value function equations in the other nodes of the value function tree. Since we consider finite-horizon forecasts and thus a value function tree with a finite number of leaves, this concludes the proof. \square

4.2 Vehicle-Based MMDP formulation

Now we describe how the aggregated EV charging problem can be formulated as MMDP, in which each agent represents an electric vehicle. At the start of hour t , we define the state h_i^t of a vehicle as the remaining number of hours during which it needs to charge (assuming a vehicle should be fully charged by the deadline). Since charging must finish before the deadline, it should hold that $h_i^{d_i} = 0$.

Each agent has two actions which it can execute: *charge* and *idle*. The *charge* action reduces the demand by one hour: $h_i^{t+1} = h_i^t - 1$, and the *idle* action does not affect its state of charge (i.e., $h_i^{t+1} = h_i^t$). We use a state-dependent action space to ensure that vehicles are guaranteed to meet their deadline. In state h_i^t the *idle* action can only be executed if $h_i^t < d_i - t$, which ensures that there is always enough time left to complete charging before the deadline. The action *charge* can be executed if $h_i^t > 0$, and must be executed if $h_i^t = d_i - t$. By using the state-dependent action space that we just described, it is guaranteed that $h_i^{d_i} = 0$. This is formalized in the following proposition.

Proposition 2. *The state-dependent action space ensures that a vehicle e_i always completes charging before its deadline d_i .*

Proof. In order to show that a vehicle always finishes charging before its deadline, we need to show that the action *idle* is never executed in situations where it would lead to a violation of the deadline. For this purpose we assume the contrary, namely that the *idle* action is executed in state h_i^t , leading to a state h_i^{t+1} in which the demand is one higher than the time left for charging: $h_i^{t+1} = (d_i - (t+1)) + 1$. Since the *idle* action was executed, it holds that $h_i^t = h_i^{t+1}$. Now we derive $h_i^t = h_i^{t+1} = (d_i - (t+1)) + 1 = d_i - t$. In state h_i^t , however, action *charge* must have been executed according to our state-dependent action space. This contradicts the assumption that *idle* was executed in state h_i^t . We can conclude that the action *idle* is never executed if it leads to a situation in which it violates a deadline, and we can conclude that our state-dependent action space ensures that vehicles meet their deadline (i.e., $h_i^{d_i} = 0$). \square

Until now we defined the states and state-dependent action space for an individual vehicle. For multiple vehicles the joint states and actions of the MMDP can be created by taking the Cartesian product of the states and actions of individual vehicles. For example, if there are two vehicles with states h_1^t and h_2^t at the start of hour t , then their joint state is (h_1^t, h_2^t) and an example of a joint action is $(charge, idle)$. The joint reward function of the agents can be computed using the function $f(x_t, w_t)$ defined in Equation 4, where x_t is the number of charging vehicles and w_t is the wind speed during hour t . For instance, if a joint action dictates that x_t vehicles need to charge during step t when the wind speed is w_t , then the MMDP reward is equal to $f(x_t, w_t)$. The state transitions of the electric vehicles are assumed deterministic and therefore we do not define a probabilistic transition function. The probabilistic transitions of wind speed are encoded separately using the scenario tree, as discussed in the previous section.

In our MMDP formulation the individual vehicles are transition-independent (i.e., P can be computed as the product of individual transition functions defined over the individual states and actions of each vehicle), as the decision whether or not to charge a particular vehicle only affects that vehicle's state of charge. However, since they are coupled through the joint reward function (only a certain number of vehicles can be charged for free using renewable energy), the value function is not factored. Specific solution algorithms have been designed for transition-independent Decentralized MDPs [4, 9], in which vehicles would take decisions in a decentralized manner. However, these solution techniques do not apply to our MMDP model in which an aggregator controls vehicles in a centralized manner. Other solution algorithms for transition-independent MMDPs exploit sparse reward structures [25], in which only a small subset of the joint actions has a non-zero reward. The latter is not the case in our model.

4.3 Reducing Enumerated States

In this section we present an optimization which reduces the number of states that need to be enumerated in each node of the value function tree when recursively computing the value functions. The number of enumerated states can be reduced by observing that some parts of the state space cannot be reached. For instance, states representing a situation in which a deadline is going to be violated will never be encountered, as stated in Proposition 2, and therefore such states do not need to be considered. When recursively computing a value function $V_{w_{t'-1}, t'}(s)$ corresponding to time $t' \geq t$, it is necessary to

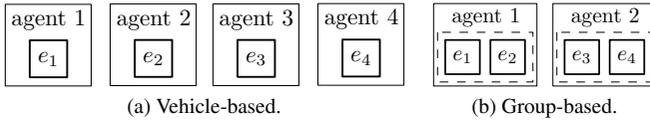


Figure 3: Vehicle-based and group-based MMDPs.

determine which states s need to be enumerated. For instance, suppose that state $s = (h_1^{t'}, h_2^{t'})$ encodes the joint state of two vehicles at time t' , then all possible combinations of $h_1^{t'}$ and $h_2^{t'}$ can be enumerated in order to enumerate all possible states s . The states $h_i^{t'}$ which need to be enumerated for vehicle $e_i \in E$ can be defined as follows:

$$\max(0, h_i^t - (t' - t)) \leq h_i^{t'} \leq \min(h_i^t, d - t'). \quad (9)$$

The lowerbound is achieved when charging as fast as possible during hours $t, \dots, t' - 1$, and the upperbound is achieved when being idle as much as possible during this period. The actions $a \in A(s)$ that need to be enumerated during the computation of $V_{w_{t'-1}, t'}(s)$ can be defined using the state-dependent action space.

5 GROUP-BASED MMDPs

In order to reduce the number of joint states and actions when increasing the number of electric vehicles, we present a group-based MMDP formulation in which each agent represents a group of vehicles. The difference between vehicle-based and group-based MMDP formulations is illustrated in Figure 3. The grouping technique is based on deadlines of vehicles, which is formalized below.

Definition 1 (Vehicle group). *A vehicle group $G_d \subseteq E$ is defined as a subset of vehicles whose deadline is equal to d . In other words, for each $e_i \in G_d$ it holds that $d_i = d$.*

The state of group G_d at the start of hour t is defined as $s_d^t = \sum_{e_i \in G_d} h_i^t$, which is simply the aggregated demand of the vehicles belonging to the group. It should hold that $s_d^d = 0$, since the deadline of the vehicles belonging to the group is identical. Our group-based planner only requires that all vehicles in a group share the same deadline, hence an aggregator could create many G_d sets. If the available renewable energy is split among them equally (for instance), each such set can be planned for separately. The action space A_d contains charging actions corresponding to group G_d . Each action $a \in A_d$ corresponds to the number of vehicles that is charging within the group. After executing action a , the demand of the entire group is reduced accordingly: $s_d^{t+1} = s_d^t - a$.

Similar to the vehicle-based formulation, for multiple groups the joint states and joint actions can be defined by taking the Cartesian product of the states and actions of the groups. For example, if there are two groups with states s_1^t and s_2^t , then the joint state of the groups is (s_1^t, s_2^t) . If there is one vehicle that is charging within both groups, then $(1, 1)$ would be a joint action. In the next section we will elaborate on the state-dependent action space which ensures that the planner does not violate the deadline of a group, similar to the state-dependent action space of the vehicle-based formulation. The joint reward can be computed using the function $f(x_t, w_t)$, similar to the vehicle-based formulation, where x_t is the number of charging vehicles and w_t is the wind speed during hour t .

Even with grouping of vehicles, obstacles to scalability might remain. In particular, it might be the case (and even likely in a typical overnight charging scenario) that many vehicles share the same deadline and hence certain G_d sets will be large, resulting in large A_d

sets. A potential solution to this problem is restricting the A_d sets, by considering charging only multiples of l vehicles, i.e.,

$$A_d = \{0, l, 2l, 3l, \dots, |G_d|\}. \quad (10)$$

The loss of fine-grained control will typically be compensated by the ability to solve for larger sets of vehicles. This aspect will also be studied in our experiments.

Example 1 (Vehicle grouping). *In our example formulation we consider six electric vehicles connected to an aggregator at time $t = 0$. The relevant properties of the individual vehicles are shown in Table 1. First we compare the number of states and actions of vehicle-based and group-based MMDP models. When formulating a vehicle-based MMDP, the total number of states is equal to $\prod_{i=1}^6 (h_i^0 + 1) = 2160$ and the number of actions is equal to $2^6 = 64$. A group-based MMDP formulation can be created by defining a group G_4 with demand 3, a group G_5 with demand 3 and a group G_6 with demand 11. The number of states in such a formulation is equal to $(3 + 1) \cdot (3 + 1) \cdot (11 + 1) = 192$ and the number of actions equals $3 \cdot 2 \cdot 4 = 24$. Clearly, the total number of states and actions decreased compared to the vehicle-based MMDP formulation. A Dynamic Bayesian network representation of the group-based MMDP is shown in Figure 4. It should be noted that the wind speed transitions in the actual implementation are encoded in a tree-based fashion, as discussed in Section 4.1.*

Table 1: Deadlines and demand of example vehicles.

i	1	2	3	4	5	6
d_i	4	4	5	6	6	6
h_i^0	2	1	3	4	5	2

5.1 Planning with Group-Based MMDPs

A group-based MMDP can directly be solved by computing value functions in the scenario tree. However, due to the aggregation of multiple vehicles into groups it becomes less straightforward which states and actions need to be enumerated in each node of the tree. In this section we first define which states need to be enumerated, and thereafter we discuss the state-dependent action space which ensures that the planner does not violate deadlines of vehicles.

We consider a group G_d , for which we can assume that s_d^t is known at the start of hour t , as well as h_i^t for each $e_i \in G_d$. This assumption can be made since the aggregator is able to observe the states of the individual vehicles before making a decision for hour t . When recursively computing the value function $V_{w_{t-1}, t}(s)$, it is necessary to know which states $s = s_d^{t'}$ need to be enumerated for timesteps $t' \geq t$. For this purpose we generalize the bounds shown in Equation 9 to bounds on the demand of a group as shown below.

$$\sum_{e_i \in G_d} \max(0, h_i^t - (t' - t)) \leq s_d^{t'} \leq \sum_{e_i \in G_d} \min(h_i^t, d - t') \quad (11)$$

The lower bound has been defined by taking the sum of the lower bounds on the demand $h_i^{t'}$ for each vehicle $e_i \in G_d$. Similarly, the upper bound has been defined by taking the sum of the upper bounds on the demand. The resulting bounds can be used to ensure that we do not enumerate unreachable states in case we use a group-based formulation.

Similar to the vehicle-based MMDP formulation, the executed actions need to ensure that the demand of an entire group is decreased

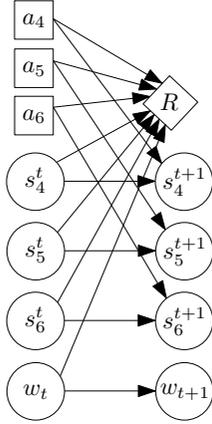


Figure 4: Dynamic Bayesian network corresponding to the group-based MMDP of the example instance.

to zero before its deadline. Therefore, we define a state-dependent action space for a group-based state $s_d^{t'}$. For convenience we let $\lfloor s_d^{t'} \rfloor$ denote the lower bound on $s_d^{t'}$ and $\lceil s_d^{t'} \rceil$ denotes the upper bound on $s_d^{t'}$. In other words, we obtain the following equations:

$$\lfloor s_d^{t'} \rfloor = \sum_{e_i \in G_d} \max(0, h_i^t - (t' - t)), \quad (12)$$

$$\lceil s_d^{t'} \rceil = \sum_{e_i \in G_d} \min(h_i^t, d - t'). \quad (13)$$

Now we can restrict the actions $a \in A(s_d^{t'})$ for a state $s_d^{t'}$ ($t \leq t' < d$) as follows:

$$\max(0, s_d^{t'} - \lceil s_d^{t'+1} \rceil) \leq a \leq \min(|G_d|, s_d^{t'} - \lfloor s_d^{t'+1} \rfloor). \quad (14)$$

In the computation of the state-dependent action space $A(s_d^{t'})$ we also use the lower- and upper bound on $s_d^{t'+1}$. These bounds have been properly defined in Equations 12 and 13. The state-dependent action space ensures that in state $s_d^{t'}$ an action is selected in such a way that $\lfloor s_d^{t'+1} \rfloor \leq s_d^{t'+1} \leq \lceil s_d^{t'+1} \rceil$. It holds that $0 = \lfloor s_d^d \rfloor \leq s_d^d \leq \lceil s_d^d \rceil = 0$, which implies that the total group demand is reduced to zero before the deadline.

Proposition 3. *The state-dependent action space for a group G_d ensures that all vehicles $e_i \in G_d$ always complete charging before their deadline d .*

Proof. If the group demand $s_d^{t'+1}$ at time $t'+1$ is higher than $\lceil s_d^{t'+1} \rceil$, then it is impossible to reduce the demand to zero before the deadline. We will show that this never occurs. If $s_d^{t'} > \lceil s_d^{t'+1} \rceil$, then the state-dependent action space defines that at least $s_d^{t'} - \lceil s_d^{t'+1} \rceil$ vehicles will be charged, such that $s_d^{t'+1} \leq s_d^{t'} - (s_d^{t'} - \lceil s_d^{t'+1} \rceil) = \lceil s_d^{t'+1} \rceil$. In other words, any action executed in state $s_d^{t'}$ guarantees that $s_d^{t'+1}$ does not exceed $\lceil s_d^{t'+1} \rceil$. Therefore, we can conclude that the state-dependent action space ensures that the planner does not violate the deadline of a group. \square

We have shown that our group-based formulation defines states for groups of vehicles, while still being able to meet the deadlines of all individual vehicles in the EV fleet. It should be noted, however, that the group-based MMDP formulation does not define a Markovian state representation for the original EV charging problem. In other

words, the state representation of the group-based formulation does not preserve sufficient information to derive the individual states of all the vehicles within the groups. Due to the aggregation of multiple vehicles into one group the upper bound on the number of vehicles that still needs to charge (i.e., the upper bound on a) may overestimate the number of vehicles that is actually available for charging. In the example below we discuss this potential overestimate in an example. An overestimate might only occur during planning when selecting the actions to compute value functions. When the resulting value function is used to select actions to control the vehicles, then such an overestimate never occurs, because the feasible actions can be determined using the actual state of the individual vehicles.

Example 2 (Overestimate of demand). *Using the previous example instance we illustrate why infeasible actions may be enumerated during the computation of value functions. We consider group G_4 containing two vehicles with demand $h_1^t = 2$ and $h_2^t = 1$ at time $t = 0$. By definition it holds that $s_4^t = 3$. We consider the group-based state $s_4^{t'}$ at time $t' = 1$, for which it holds that $1 \leq s_4^{t'} \leq 3$. In state $s_4^{t'} = 2$, the upper bound on the number of vehicles with non-zero demand is $\min(|G_4|, s_4^{t'} - \sum_{e_i \in G_4} \max(0, h_i^t - ((t'+1) - t))) = \min(2, 2 - \max(0, 2 - 2) - \max(0, 1 - 2)) = 2$, which represents that we can charge at most two vehicles simultaneously in this state. However, it may be possible that $h_1^{t'} = 2$ and $h_2^{t'} = 0$, and then only one vehicle can be charged. In this case the number of vehicles with non-zero demand is overestimated by 1.*

6 EXPERIMENTS

This section describes the results of our experiments. We use historical wind data from the Sotavento wind farm in Spain.³ We simulate the hourly average wind speed for the period from September 2, 2012 until September 26, 2012. The forecasts are based on data from the period September 1, 2009 until December 31, 2009. Unless stated otherwise, the capacity of the wind turbine involved is 50 kW. We assume that the charging rate of the vehicles is equal to 3 kW, which corresponds to a compact car. The electricity price during the simulation is time-dependent, for which we use data from a European power market, which gives us an hourly price (unit EUR/kWh). Unless stated otherwise, the feed-in tariff is 50 percent of the tariff for buying power. To define EVs we use realistic vehicle arrival and departure times from a Dutch mobility study, conducted by Statistics Netherlands [6].

6.1 Aggregator Profit and Power Consumption

First we investigate whether the aggregator is able to make profit by coordinating vehicles. We simulate 25 days, and during each day we charge 20 vehicles. For each vehicle $e_i \in E$, the payment m_i is 10 percent lower than the minimum cost the customer would pay to the utility company without participation, which provides an incentive for the customers to subscribe to the aggregator. In order to compensate for the discount given to customers, the aggregator needs to efficiently use zero-cost wind power. It is estimated that there is a 25 percent market share of EVs starting in 2020 [14], hence 20 vehicles can represent a realistically-sized street or a small neighborhood.

Figure 5 shows the cumulative daily profit of the aggregator for several different planners, which needs to be maximized. In addition to our MMDP planner with groups, we use a greedy planner which charges each vehicle during its individually cheapest hours (i.e., min cost), and another greedy planner which charges the vehicles as fast as

³ Data is available on www.sotaventogalicia.com.

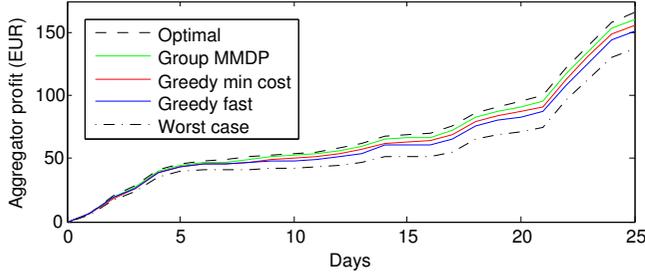


Figure 5: Cumulative profit made by the aggregator.

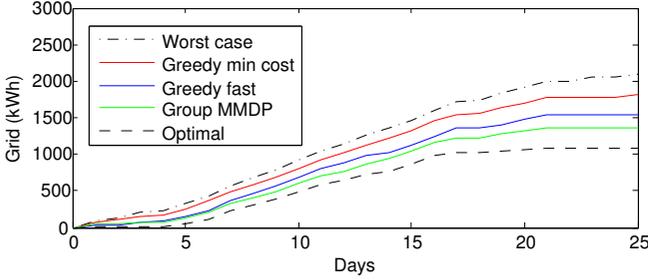


Figure 6: Cumulative power consumption of the vehicles.

possible. Lower- and upper bounds on the profit have been computed using a mixed-integer programming formulation, which computes omniscient optimal and worst-case charging schedules based on the wind speed during the day. In practice it would not be possible to find such schedules, since wind speed in the future is uncertain.

We conclude that the aggregator is able to make profit by coordinating vehicles, even if it provides a financial compensation to customers of the vehicles. Moreover, the group-based MMDP planner outperforms two greedy planners in terms of profit, and its profit is close to the profit of the omniscient optimal planner.

Although the main objective of the aggregator is optimizing its profit, it may be able to reduce power consumption from the grid, since it is able to charge vehicles during periods in which wind speed is high. Figure 6 shows the cumulative grid power consumption corresponding to the simulation of the previous experiment. We observe that the grid power consumption of the MMDP planner is lower than the power consumption of the greedy planners involved in the experiment. Therefore, we conclude that an aggregator that aims to maximize its profit also reduces grid power consumption, which can be considered as one of its side effects.

6.2 Vehicle-Based and Group-Based MMDPs

Next we study the influence of grouping on the scalability of MMDP formulations for electric vehicle charging. To study the difference between vehicle-based and group-based MMDPs, we constructed a set of EVs $E' = (e_1, \dots, e_{15})$, in which the first three vehicles do not have common deadlines. When we run vehicle-based and group-based planners on the first $1 \leq \delta \leq 15$ vehicles of E' , we expect that grouping only provides improved scalability if $\delta > 3$. In Figure 7 we show the running times of vehicle-based and group-based MMDPs for an increasing δ (i.e., number of vehicles), which confirms our expectation that group-based formulations require less computation time if groups of vehicles can be created. Note that a log scale is used for the y -axis representing the running time.

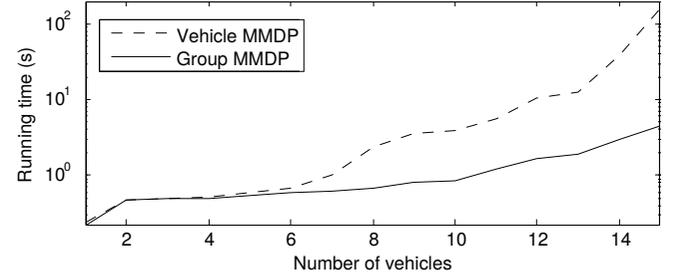


Figure 7: Running time comparison between vehicle-based and group-based MMDP formulations (log scale).

6.3 Action Space Compression

When after grouping large sets of vehicles remain, it may be desirable to perform action space compression to reduce the number of enumerated actions, as defined in Equation 10. This means that the planner only considers charging multiples of l vehicles. For a case of 15 vehicles, Figure 8a shows the effect on runtime of increasing l (i.e., the level of discretization of the action space) and Figure 8b shows the corresponding profit. We can see that as expected a small loss is incurred, but the running time required for the computation of the value functions decreases significantly. The dashed lines represent the profit of the omniscient optimal and greedy min cost planners in the simulation. Our MMDP planner still makes more profit compared to the greedy min cost planner in the simulation.

6.4 Influence of Wind Turbine Capacity

Until now we assumed a fixed turbine capacity, but it can be expected that the turbine capacity influences the profit of the aggregator. In order to study this influence, we run simulations in which we charge 15 vehicles during each day, and we assume that wind power cannot be sold to the utility company. The latter is assumed because this eliminates the influence of selling wind power in our experiment. Small-scale wind power involves turbines with a capacity of at most 50 kW, and therefore we repeat the simulation for an increasing turbine capacity up to 50kW, as shown in Figure 9a. We can derive three conclusions. First, if the turbine capacity is too low then the aggregator is not able to make profit. This is caused by the fact that the charging cost will exceed the customer payments if there is almost no wind power available. Second, a relatively small wind turbine may already be sufficient to make profit. Third, the experiment shows that it is likely that our framework can be used in the residential area where wind turbines typically have a capacity up to a few kilowatts [3].

6.5 Influence of Customer Payments

In the previous experiment we observed that the financial compensation paid to the customers influences the profit of the aggregator, and we expect that profit becomes negative if the compensations are too high compared to the usage of zero-cost wind power. In the current experiment we assume that the payments m_i are α percent lower than the minimum cost the customer would pay to the utility company without participation ($0 < \alpha \leq 100$), and we run simulations for an increasing value of α . The parameter α is called the vehicle discount. In Figure 9b we show the profit of the aggregator as a function of the vehicle discount, which confirms our expectation that it is impossible to make profit if the discount is too high. In order to provide an incentive to customers of EVs to participate, it is sufficient to have a small

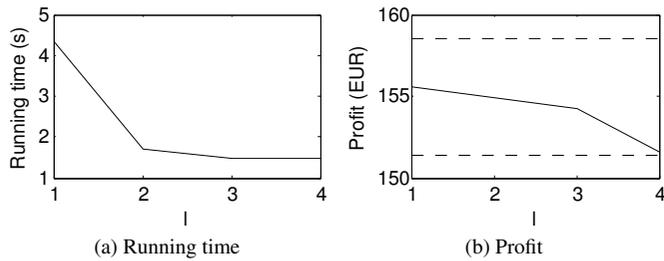


Figure 8: Effect of action space compression (10).

non-zero α , and therefore we conclude that the payments m_i of our framework provide an incentive to customers to participate.

7 RELATED WORK

Leterme et al. discuss an MDP-based approach to control EVs for wind balancing, in which wind uncertainty is encoded as a tree [19], but in contrast to our work their solution does not control individual EVs. Huang et al. [16] cluster EVs based on remaining parking time and use Monte Carlo simulations to estimate a value function. Our scenario-tree encoding of the wind uncertainty provides a more advanced representation of wind uncertainty and cannot directly be combined with their approach. Other objective functions, such as waiting time at charging stations, have also been studied in existing work [37]. Aggregators can use reinforcement learning to learn a consumption pattern of their fleet before buying energy in the day-ahead market [32]. Currently our work only focuses on uncertainty in renewable supply, and it does not model bids in a day-ahead or intraday energy market.

In the power systems community research has focused on matching demand and supply in the unit commitment problem using multi-stage stochastic programming and mixed-integer programming, where exogenous uncertainty in the supply is also characterized using scenarios [22]. Multi-stage stochastic programming methods are typically used for problems with exogenous uncertainty that cannot be controlled by the decision maker [8], whereas Markov Decision Processes are well-suited if control actions influence the uncertainty encountered in the future. For example, stochastic state transitions in our MDP models can also be used to model uncertainty in arrival time and departure time of electric vehicles, which is hard to model in a multi-stage stochastic programming formulation. Research has also focused on inclusion of network characteristics in aggregate models of multiple EVs [17]. Compared to our work, existing work in this area focuses more on modeling the electrical aspects and the impact on the power system. Congestion management schemes have been developed for electric vehicles, which typically assume a deterministic setting in which there is no uncertainty during optimization and execution [33]. Our work can be used for congestion management if renewable supply is uncertain.

Reducing computational requirements by aggregating states of MDPs has been studied in the context of stochastic bisimulation [12], which is an exact method to compute an equivalent smaller-sized MDP, and symmetry reduction [18]. Both methods can theoretically be combined with our work, but require a given MDP which needs to be minimized [20] and often require full state-space enumeration. The latter leads to problems in the multiagent setting because of the exponential growth of the number of states. Our group-based model can be created without needing an initial model, but the abstraction method is not exact. Other abstraction methods include temporal

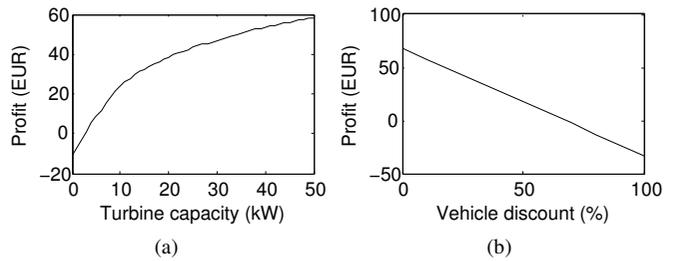


Figure 9: Profit for increasing turbine capacity (a) and discount (b).

abstractions, such as macro-actions [15] and Semi-MDPs [27], which would allow an aggregator to solve an abstract planning problem to select sub-policies rather than actions. However, these abstractions do not address scalability problems that follow from the large number of EVs, and it is hard to combine such abstraction techniques with exogenous wind uncertainty.

Constrained MDPs [1] include constraints in the dual formulation of a linear program. This framework can also be used to impose constraints to make sure that deadlines are satisfied, but it would be difficult to separate the reasoning about exogenous wind uncertainty in the corresponding linear programming formulations. Moreover, linear programs for Constrained MDPs are typically based on the assumption that the planning horizon is infinite.

8 CONCLUSIONS

In this paper we present an aggregated charging technique based on Multiagent Markov Decision Processes which accounts for the uncertainty in renewable supply and coordinates the charging process of several EVs. We use groups of vehicles to create an abstraction of the MMDP, which reduces the number of joint states and actions and it reduces the running time required to compute MMDP solutions. Our experiments show that our framework is able to charge a collection of EVs, reduces cost of the individual customers and reduces consumption of conventionally-generated power. Moreover, our work demonstrates that AI methods have the potential to support the development of smart grids. For example, an interesting application of our work can be found in parking garages with local grid capacity constraints, where charging of a large number of EVs needs to be coordinated and peak loads must be prevented.

In future work we aim to include information about uncertain demand in our MMDP formulations, which can be naturally included in stochastic state transitions. Our work can also be extended to asynchronous events and actions using Generalized Semi-MDPs [36], and it can be combined with wind scenario trees generated by ARMA models [28]. Another interesting direction is creating groups of vehicles based on additional characteristics besides their deadline, such as the charging rate and spatial location in the network. Our method can also be combined with power flow computations to derive the power flows through the network. This is useful if capacity violations must be prevented in a congested network.

ACKNOWLEDGMENTS

The work presented in this paper is funded by the Netherlands Organisation for Scientific Research (NWO), as part of the Uncertainty Reduction in Smart Energy Systems program.

REFERENCES

- [1] E. Altman, *Constrained Markov Decision Processes*, CRC Press, 1999.
- [2] P. Bach Andersen, J. Hu, and K. Heussen, 'Coordination strategies for distribution grid congestion management in a multi-actor, multi-objective setting', in *3rd IEEE PES Innovative Smart Grid Technologies Europe*, (2012).
- [3] D. Ayhan and S. Sağlam, 'A technical review of building-mounted wind power systems and a sample simulation model', *Renewable and Sustainable Energy Reviews*, **16**(1), 1040–1049, (2012).
- [4] R. Becker, S. Zilberstein, V. Lesser, and C. V. Goldman, 'Transition-Independent Decentralized Markov Decision Processes', in *Proceedings of the 2nd International Conference on Autonomous Agents and Multiagent Systems*, (2003).
- [5] C. Boutilier, 'Planning, Learning and Coordination in Multiagent Decision Processes', in *Proceedings of the 6th Conference on Theoretical Aspects of Rationality and Knowledge*, pp. 195–210, (1996).
- [6] Centraal Bureau voor de Statistiek (CBS) / Rijkswaterstaat (RWS). Onderzoek Verplaatsingen in Nederland 2013 - OViN 2013, 2014. DANS. <http://dx.doi.org/10.17026/dans-x9h-dsdg>.
- [7] A. J. Conejo, M. Carrion, and J. M. Morales, *Decision Making Under Uncertainty in Electricity Markets*, Springer, 2010.
- [8] B. Defourmy, D. Ernst, and L. Wehenkel, 'Multistage Stochastic Programming: A Scenario Tree Based Approach to Planning under Uncertainty', in *Decision Theory Models for Applications in Artificial Intelligence: Concepts and Solutions*, eds., L E Sucar, E F Morales, and J Hoey, 97–143, Information Science Publishing, (2012).
- [9] J. S. Dibangoye, C. Amato, A. Doniec, and F. Charpillat, 'Producing efficient error-bounded solutions for transition independent decentralized MDPs', in *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems*, (2013).
- [10] J. Dupačová, N. Gröwe-Kuska, and W. Römisch, 'Scenario reduction in stochastic programming: An approach using probability metrics', *Mathematical Prog. Series A*, (2003).
- [11] G. Giebel, R. Brownsword, G. Kariniotakis, M. Denhard, and C. Draxl, 'The State-Of-The-Art in Short-Term Prediction of Wind Power', Technical report, ANEMOS.plus project, (2011).
- [12] R. Givan, T. Dean, and M. Greig, 'Equivalence notions and model minimization in Markov decision processes', *Artificial Intelligence*, **147**(1-2), 163–223, (2003).
- [13] L. Gkatzikis, I. Koutsopoulos, and T. Salonidis, 'The Role of Aggregators in Smart Grid Demand Response Markets', *IEEE Journal on Selected Areas in Communications*, **31**(7), 1247–1257, (2013).
- [14] S. W. Hadley and A. A. Tsvetkova, 'Potential impacts of plug-in hybrid electric vehicles on regional power generation', *The Electricity Journal*, **22**(10), 56–68, (2009).
- [15] M. Hauskrecht, N. Meuleau, L. P. Kaelbling, and C. Boutilier, 'Hierarchical Solution of Markov Decision Processes using Macro-actions', in *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pp. 220–229, (1998).
- [16] Q. Huang, Q. Jia, Z. Qiu, X. Guan, and G. Deconinck, 'Matching EV Charging Load With Uncertain Wind: A Simulation-Based Policy Improvement Approach', *IEEE Transactions on Smart Grid*, **6**(3), 1425–1433, (2015).
- [17] S. Izadkhast, P. Garcia-Gonzalez, P. Frias, L. Ramirez-Elizondo, and P. Bauer, 'An aggregate model of plug-in electric vehicles including distribution network characteristics for primary frequency control', *IEEE Transactions on Power Systems*, **31**(4), 1–12, (2015).
- [18] M. Kwiatkowska, G. Norman, and D. Parker, 'Symmetry Reduction for Probabilistic Model Checking', in *Computer Aided Verification*, pp. 234–248, (2006).
- [19] W. Leterme, F. Ruelens, B. Claessens, and R. Belmans, 'A Flexible Stochastic Optimization Method for Wind Power Balancing With PHEVs', *IEEE Transactions on Smart Grid*, **5**(3), 1238–1245, (2014).
- [20] L. Li, T. J. Walsh, and M. L. Littman, 'Towards a Unified Theory of State Abstraction for MDPs', in *Proceedings of the 9th International Symposium on Artificial Intelligence and Mathematics*, (2006).
- [21] J. A. P. Lopes, F. J. Soares, and P. M. R. Almeida, 'Identifying management procedures to deal with connection of electric vehicles in the grid', in *IEEE PowerTech*, (2009).
- [22] G. Morales-España, *Unit Commitment: Computational Performance, System Representation and Wind Uncertainty Management*, Ph.D. dissertation, Pontifical Comillas University, KTH Royal Institute of Technology and Delft University of Technology, 2014.
- [23] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, John Wiley & Sons, Inc., 1994.
- [24] E. S. Rigas, S. D. Ramchurn, and N. Bassiliades, 'Managing electric vehicles in the smart grid using artificial intelligence: A survey', *IEEE Transactions on Intelligent Transportation Systems*, **16**(4), 1619–1635, (2015).
- [25] J. Scharpff, D. M. Roijers, F. A. Oliehoek, M. T. J. Spaan, and M. M. de Weerd, 'Solving Transition-Independent Multi-agent MDPs with Sparse Interactions', in *Proceedings of the AAAI Conference on Artificial Intelligence*, (2016).
- [26] P. Ströhle, E. H. Gerding, M. M. de Weerd, S. Stein, and V. Robu, 'Online Mechanism Design for Scheduling Non-Preemptive Jobs under Uncertain Supply and Demand', in *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems*, pp. 437–444, (2014).
- [27] R. S. Sutton, D. Precup, and S. Singh, 'Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning', *Artificial Intelligence*, **112**(1), 181–211, (1999).
- [28] J. L. Torres, A. García, M. De Blas, and A. De Francisco, 'Forecast of hourly average wind speed with ARMA models in Navarre (Spain)', *Solar Energy*, **79**(1), 65–77, (2005).
- [29] F. Ueckerdt, R. Brecha, and G. Luderer, 'Analyzing Major Challenges of Wind and Solar Variability in Power Systems', *Renewable Energy*, **81**, 1 – 10, (2015).
- [30] E. Ungar and K. Fell, 'Plug in, turn on, and load up', *IEEE Power and Energy Magazine*, **8**(3), 30–35, (2010).
- [31] H. M. van den Dool, 'A New Look at Weather Forecasting through Analogues', *Monthly Weather Review*, **117**(10), 2230–2247, (1989).
- [32] S. Vandael, B. Claessens, D. Ernst, T. Holvoet, and G. Deconinck, 'Reinforcement Learning of Heuristic EV Fleet Charging in a Day-Ahead Electricity Market', *IEEE Transactions on Smart Grid*, **6**(4), 1795–1805, (2015).
- [33] R. A. Verzijlbergh, *The Power of Electric Vehicles - Exploring the value of flexible electricity demand in a multi-actor context*, Ph.D. dissertation, Delft University of Technology, 2013.
- [34] R. A. Verzijlbergh, L. J. de Vries, and Z. Lukszo, 'Renewable Energy Sources and Responsive Demand. Do We Need Congestion Management in the Distribution Grid?', *IEEE Transactions on Power Systems*, **29**(5), 2119–2128, (2014).
- [35] E. Walraven and M. T. J. Spaan, 'Planning under Uncertainty with Weighted State Scenarios', in *Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence*, (2015).
- [36] H. L. S. Younes and R. G. Simmons, 'Solving Generalized Semi-Markov Decision Processes using Continuous Phase-Type Distributions', *Proceedings of the 19th National Conference on Artificial Intelligence*, 742–747, (2004).
- [37] T. Zhang, W. Chen, Z. Han, and Z. Cao, 'Charging Scheduling of Electric Vehicles With Local Renewable Energy Under Uncertain Electric Vehicle Arrival and Grid Power Price', *IEEE Transactions on Vehicular Technology*, **63**(6), 2600–2612, (2014).

On the Computation of Top-k Extensions in Abstract Argumentation Frameworks

Said Jabbour¹ and Badran Raddaoui² and Lakhdar Sais³ and Yakoub Salhi⁴

Abstract. Formal argumentation has received a lot of attention during the last two decades, since abstract argumentation framework provides the basis for various reasoning problems in Artificial Intelligence. Unfortunately, the exponential number of its possible semantics extensions makes some reasoning problems intractable in this framework. In this paper, we investigate the pivotal issue of efficient computation of acceptable arguments called *extensions* according to a given semantics. In particular, we address this aspect by applying a strategy of how to use preferences at the semantics level in order to determine what are “desirable” outcomes of the argumentation process. Then, we present a new approach for computing the Top-*k* extensions of an abstract argumentation framework, according to a user-specified preference relation. Indeed, an extension is a Top-*k* extension for a given semantics if it admits less than *k* extensions preferred to it with respect to a preference relation. Our experiments on various datasets demonstrate the effectiveness and scalability of our approach and the accuracy of the proposed enumeration method.

1 Introduction

Since its emergence, the formal study of *argumentation* has proven to be fruitful. It has been intensively studied in various subfields of Artificial Intelligence, namely for handling inconsistency and uncertainty [36, 22], and for representing dialogue and making decision [2]. Furthermore, argumentation finds applications in the context of multi-agent systems, mostly in negotiation context [38], where agents try to convince each other by exchanging reasons in favor or against different offers, but also in persuasion [33, 23] or for opponent modeling [37].

Argumentation is a form of reasoning that is basically concerned with the exchange of interacting arguments. This set of arguments may come either from a dialogue between several intelligent agents but also from the available (and possibly contradictory) pieces of information at the disposal of one unique agent. Usually, the interaction between arguments takes the form of a conflict, called *attack*. Such attacks are exploited to decide what can be reasonably concluded. There have been a number of proposals for capturing this cognitive process in computational models of argumentation. A particularly popular formalism are *Dung’s abstract argumentation frameworks* [17]. Dung’s theory of argumentation provides a general model, which is widely recognized as a fundamental reference in computational argumentation in virtue of its simplicity, general-

ity, and ability to model reasoning in a variety of non-classical logics, e.g. [17, 9, 16]. In this framework, the starting point is a graph, with vertices corresponding to arguments, and edges representing the notion of *attack* between arguments. Abstract argumentation frameworks are so-called because they abstract away from the origin and the internal structure of individual arguments, and focus only on the interactions between arguments. The concept of *extension* plays a pivotal role in this simple setting, where an extension is a set of arguments which can be accepted together. Meanwhile, various acceptability *semantics* for abstract argumentation frameworks have been also stated for characterizing “reasonable” sets of arguments. In addition, several inference relations can be defined within Dung’s theory of argumentation in the light of extensions. Usually, inference is defined at the argument level: an argument is accepted in a given argumentation framework when it belongs to one (*credulous acceptability*) (resp. all (*skeptical acceptability*)) extension(s) of under some semantics.

In Dung’s abstract argumentation frameworks, most of the reasoning tasks have been shown to suffer from high computational complexity [19]. Indeed, many important reasoning problems for abstract argumentation frameworks are located at the second level of the polynomial hierarchy. Consequently, most computational argumentation research is theoretical in nature. The few existing implementations either follow a straight-forward reduction-based approach or focus on certain tractable classes of argumentation systems.

In this paper, we deal with the significant problem of *enumerating* semantics extensions, i.e., constructing *all* sets of acceptable arguments prescribed for a given abstract argumentation framework: its solution provides complete information concerning the justification status of arguments and subsumes the solutions to the other problems. Our goal is to contribute to the evolution of the field from the theoretical side to the development of applicable solutions.

On one hand, the complexity of extension-related decision problems has been deeply investigated and, for most of the semantics proposed in the literature they have been proven to be intractable [18]. On the other hand, there are few domain-independent argumentation tools (e.g. ASPARTIX [20], CEGARTIX [25], dynPARTIX[15], ConArg [8], Dung-O-Matic⁵, Tweety [41], and CoQuiAAS [29]) that aim to identify extensions of abstract argumentation frameworks. Finally, one of the main bottlenecks in abstract argumentation frameworks, in general, concerns the large number of possible extensions (that can be exponential in the worst case), from which it is difficult for the user to retrieve relevant information. Then, reducing the huge size of the output is clearly an important research issue.

In this work, we address the following research question:

Given an abstract argumentation framework, how to enumerate

¹ CRIL CNRS UMR 8188, France, email: jabbour@cril.fr

² LIAS/ENSMA - University of Poitiers, France, email: badran.raddaoui@ensma.fr

³ CRIL CNRS UMR 8188, France, email: sais@cril.fr

⁴ CRIL CNRS UMR 8188, France, email: salhi@cril.fr

⁵ http://www.arg.dundee.ac.uk/?page_id=279

all extensions prescribed by a given semantics according to some ranking expressed by a preference relation, with the aim of reducing the computational effort?

In other words, we consider forms of selecting the *best* extensions based on some user-given preference criterion while retrieving only the Top- k extensions.

The layout of this paper follows the research question and is organized as follows. Section 2 presents the existing Dung's well-studied abstract argumentation framework. Next, we review some preferences relations over arguments in abstract argumentation frameworks in Section 3. Section 4 studies the way to select the desirable outcomes of a given argumentation process by using preferences. To do this, we impose various partial orderings on the set of extensions based on arguments comparison while taking into account various influences such as relevance of the arguments, number of arguments attacking it, and successive attack-defense sequences. Section 5 provides a classical propositional satisfiability (SAT) based encoding, which allows us to benefit from cutting-edge SAT solvers in the design of argumentation procedures. In Section 6, we conduct a set of empirical studies to examine the performance of the proposed algorithm on some datasets. Avenues for further research are discussed in Section 7.

2 Background on Abstract Argumentation

In this section, we briefly outline the basic concepts of Dung's abstract argumentation framework [17].

An *abstract argumentation framework* (AF, for short) is a pair $\langle \mathcal{A}, \mathcal{R} \rangle$, where \mathcal{A} is a finite set, whose elements are called *arguments*, and $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$ is a binary relation over \mathcal{A} , whose elements are referred to as *attacks*. An argument is an abstract entity whose role is entirely determined by its relationships with other arguments. An AF can simply be represented as a directed graph, called *attack graph*, where nodes are the arguments and edges represent the attack relation.

Given two arguments a and b , we say that a *attacks* b iff there is $(a, b) \in \mathcal{R}$. Moreover, a set $S \subseteq \mathcal{A}$ *attacks* an argument $b \in \mathcal{A}$ iff there is $a \in S$ such that a attacks b . A set $S \subseteq \mathcal{A}$ is said to be *conflict-free* if there are no arguments $a, b \in S$ such that a attacks b . An argument $a \in \mathcal{A}$ is *defended* by a set $S \subseteq \mathcal{A}$ iff $\forall b \in \mathcal{A}$ such that b attacks a , there is $c \in S$ such that c attacks b .

Using the notions of conflict-freeness and defence, we can define a number of argumentation semantics, each embodying a particular rationality criterion, in order to identify reasonable sets of arguments, called *extensions*.

Definition 1 Given an AF $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$. A set $\mathcal{E} \subseteq \mathcal{A}$ of arguments is said to be:

- admissible iff \mathcal{E} is conflict-free and all its arguments are defended by \mathcal{E} ,
- a complete extension iff \mathcal{E} is admissible and \mathcal{E} contains all and only the arguments it defends,
- a grounded extension iff \mathcal{E} is a minimal (w.r.t. set inclusion) complete set of arguments,
- a preferred extension iff \mathcal{E} is a maximal (w.r.t. set inclusion) admissible set of arguments.

Basically, each of these semantics corresponds to some properties which certify whether a set of arguments can be profitably used to support a point of view in a dispute.

Let $\text{Ext}_x(\mathcal{F})$ denotes the set of extensions of the AF \mathcal{F} under semantics x where $x \in \{a, c, g, p\}$ and a, c, g and p stands for admissible, complete, grounded and preferred semantics respectively. When the semantics is not important, or when it is clear from the context to which semantics we refer to, we use the notation $\text{Ext}(\mathcal{F})$ for short.

Despite the descriptive power offered by AFs one significant problem is the apparent intractability of many natural questions concerning acceptability under all semantics.

The following example illustrates the case where an AF admits an exponential number of possible extensions.

Example 1 Let the AF $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ s.t. $\mathcal{A} = \{a_1, b_1, \dots, a_n, b_n\}$ and $\mathcal{R} = \{(a_1, b_1), (b_1, a_1), \dots, (a_n, b_n), (b_n, a_n)\}$. Let us determine the extensions of \mathcal{F} under some semantics. Clearly, a set $\mathcal{E} \subseteq \mathcal{A}$ is a preferred extension if and only if $|\mathcal{E}| = n$ and $\{a_i, b_i\} \not\subseteq \mathcal{E}$ for every $i = 1, \dots, n$. Thus, \mathcal{F} has 2^n preferred extensions. As a consequence, the number of complete extensions in \mathcal{F} is greater than 2^n , since every preferred extension is complete.

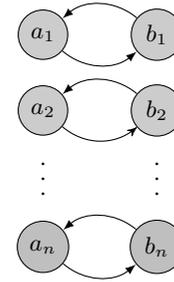


Figure 1. an AF illustrating the exponential number of extensions under preferred and complete semantics.

As shown in the previous example, the number of possible extensions is very large. Consequently, every effort to lessen the burden of computation of these extensions would be an important step forward. Namely, this can be done by taking advantage of *preference* relations within an argumentation process for selecting only *best* extensions.

3 Preferences in Abstract Argumentation

Elicitation, modeling, and reasoning with preferences in the context of argumentation has been experiencing much interest in recent years. In their seminal paper [39], Simari and Loui have emphasized the importance of considering preferences when evaluating arguments in a given framework. Indeed, numerous proposals have been made in the literature for refining argumentation frameworks using preferences. In [5], the authors stressed the importance of incorporating preferences or priorities in argumentation frameworks. In fact, the selection of a unique set of justified/acceptable arguments often requires a *preference* relation on arguments to determine *successful* attacks between arguments. These preferences may be derived, for example, from relative specificity/priority of arguments or from the relative strength of the beliefs with which an argumentation framework is built.

Various approaches (e.g. [7, 27, 3, 31]) extend the Dung abstract argumentation model by introducing a pre-order or a total order over arguments through a preference relation, which states for each couple of arguments either they are incomparable or which is the most

preferred. Likewise, other approaches formalize the role of preferences in the underlying logical formalism that instantiate a Dung-style framework [34, 32]. In these frameworks, different definitions of preference relations between arguments have been introduced and analyzed. Further, in [21] the authors develop a formal method to rank arguments according to how strongly they are justified under different graded semantics, so that extensions are graded with respect to the attacks and counter-attacks on their contained arguments. Typically, in all preference-based argumentation frameworks, preference elicitation and reasoning focus on preferences over single arguments. However, evaluating the outcomes of argumentation in the framework seems to have received relatively limited attention in the literature. To the best of our knowledge, few works have been done on instantiations of argumentation frameworks with preferences over (set of) extensions. Particularly, in order to rank extensions, [40] defined a hybrid operator to compute the relevance of each extension by aggregating the strength of its arguments. Notice that another notion of argument strength has been put in relation with the class of games of strategy [30]. Another extension of Dung's framework has been proposed in [3], to compare conflict-free sets of arguments. The extensions proposed in [6, 4, 1] assume different principles for comparing existing semantics with respect to them. In [35], the authors proposed various agent's preferences criteria in order to compare different outcomes of argumentation process where an outcome specifies acceptable, rejected and undecided arguments. Let us also mention the work of Kaci [26] that addressess the problem of rank-ordering acceptable extensions. Recently, Konieczny et al. [28] defined different criteria for pairwise comparison of extensions in AFs. Choices are based either on the attack relation between extensions or on the number of times an argument appears in the extensions.

In a number of applications the *outcomes* of interest are sets of such atomic outcomes [10, 11]. For instance, when considering an argumentation framework, we need to select a set of arguments that are acceptable together, with respect to quality, relevance, etc. However, unlike most past works which attempt to reason with preferences over arguments, we are interested in similar forms of reasoning over sets of arguments, i.e., comparing and assessing relevant extensions, appears far from being a straightforward task. The idea is that, instead of modifying the inputs of Dung's argumentation framework, we extend the outcomes with preferences. More specifically, the setting of our problem is:

Given a set of extensions $\text{Ext}(\mathcal{F})$ of an AF \mathcal{F} , elicit an (possibly partial) ordering over $\text{Ext}(\mathcal{F})$, and find at least Top- k best subset of $2^{\mathcal{A}}$ with respect to this ordering. The idea here is that outcomes promote certain extensions and that different audiences have different preferences over extensions, from which the preferences over outcomes are derived.

4 Preferences over Extensions

In this section, we introduce an approach based on applying a strategy of how to use preferences at the semantics level in order to determine what are *desirable* outcomes of the argumentation process. A most common type of a preference relation is a pre-order, a relation that is reflexive and transitive. The objective is to characterize a class of preference relations aimed to rank subsets of arguments (i.e., extensions) including interaction effects among the arguments.

In the literature, various ordering over sets have been proposed in order to compare sets [11]. These pre-orders can be naturally extended to AFs as follows:

Definition 2 Given an AF $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ where \gg is a pre-order on \mathcal{A} . Let $\mathcal{E}_1, \mathcal{E}_2 \in \text{Ext}(\mathcal{F})$. Then,

- $\mathcal{E}_1 \succeq_1 \mathcal{E}_2$ if for every argument $b \in \mathcal{E}_2$, there exists $a \in \mathcal{E}_1$ s.t. $a \gg b$
- $\mathcal{E}_1 \succeq_2 \mathcal{E}_2$ if for every argument $a \in \mathcal{E}_1$, there exists $b \in \mathcal{E}_2$ s.t. $a \gg b$
- $\mathcal{E}_1 \succeq_3 \mathcal{E}_2$ iff $\mathcal{E}_1 \succeq_1 \mathcal{E}_2$ and $\mathcal{E}_1 \succeq_2 \mathcal{E}_2$
- $\mathcal{E}_1 \succeq_4 \mathcal{E}_2$ if there is an argument $a \in \mathcal{E}_1$ s.t. for every argument $b \in \mathcal{E}_2$, $a \gg b$
- $\mathcal{E}_1 \succeq_5 \mathcal{E}_2$ if for every argument $b \in \mathcal{E}_2 \setminus \mathcal{E}_1$, there exists $a \in \mathcal{E}_1 \setminus \mathcal{E}_2$ s.t. $a \gg b$
- $\mathcal{E}_1 \succeq_6 \mathcal{E}_2$ if for every argument $a \in \mathcal{E}_1 \setminus \mathcal{E}_2$, there exists $b \in \mathcal{E}_2 \setminus \mathcal{E}_1$ s.t. $a \gg b$
- $\mathcal{E}_1 \succeq_7 \mathcal{E}_2$ if $\mathcal{E}_1 = \mathcal{E}_2$, or if $\mathcal{E}_2 \setminus \mathcal{E}_1 \neq \emptyset$ and for every argument $b \in \mathcal{E}_2 \setminus \mathcal{E}_1$, there exists $a \in \mathcal{E}_1 \setminus \mathcal{E}_2$ s.t. $a \gg b$
- $\mathcal{E}_1 \succeq_8 \mathcal{E}_2$ if $\mathcal{E}_1 = \mathcal{E}_2$, or if $\mathcal{E}_1 \setminus \mathcal{E}_2 \neq \emptyset$ and for every argument $a \in \mathcal{E}_1 \setminus \mathcal{E}_2$, there exists $b \in \mathcal{E}_2 \setminus \mathcal{E}_1$ s.t. $a \gg b$

The pre-order \succeq_1 means that an extension \mathcal{E}_1 is preferred to another extension \mathcal{E}_2 if for each argument b in \mathcal{E}_2 , there exists at least one element in \mathcal{E}_1 , which is preferred to b . Note also that $\forall \mathcal{E}_1, \mathcal{E}_2 \in \text{Ext}(\mathcal{F})$, if $\mathcal{E}_1 \succeq_4 \mathcal{E}_2$ then $\mathcal{E}_1 \succeq_1 \mathcal{E}_2$.

Contrary to preference-based argumentation frameworks, rather than modifying the original framework, all the aforementioned relations aim to select optimal extensions through a pre-order on sets of arguments. In other words, these relations aim to derive a preference relation on the power set of $\text{Ext}(\mathcal{F})$ from a preference relation over arguments in $\text{Ext}(\mathcal{F})$. Thus, none of the previous pre-order are entirely free of \gg . The above pre-order is a general definition that allows for a range of possible preference relations to be proposed. Note that instances of \succeq_i , $i = 1, \dots, 8$ depend on the choice of the basic relation \gg between arguments.

Now, let us define a basic ordering for pairwise comparison of extensions as follows.

Definition 3 Let $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ be an AF and $\mathcal{E}_1, \mathcal{E}_2 \in \text{Ext}(\mathcal{F})$. Then, $\mathcal{E}_1 \succeq_9 \mathcal{E}_2$ iff $|\mathcal{E}_1| \geq |\mathcal{E}_2|$.

That is, the relation \succeq_9 shows how to evaluate extensions by comparing them according to their size. However, in argumentation literature it has been acknowledged that arguments may not have equal strength. Of course, some arguments may be stronger than others for different reasons; for instance, because they are built from more certain information. Note that the relation $a \gg b$ can stand that an argument a is stronger (more preferred) to b . In this case, a subset of arguments $\mathcal{A}_{\mathcal{R}} \subseteq \mathcal{A}$ is specified as the most *relevant* and thus preferred extensions must express such relevance.

Clearly, the preference relation \succeq_9 does not discriminate better between extensions, i.e., many of them might have the same size.

Thus, a finer way to select extensions seems to compare them in pairwise w.r.t. $\mathcal{A}_{\mathcal{R}}$ as stated in Definition 4.

Definition 4 Let $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ be an AF and $\mathcal{E}_1, \mathcal{E}_2 \in \text{Ext}(\mathcal{F})$. Let $\mathcal{A}_{\mathcal{R}} \subseteq \mathcal{A}$ be the set of relevant arguments in \mathcal{F} . Then, $\mathcal{E}_1 \succeq_{10} \mathcal{E}_2$ iff $|\mathcal{E}_1 \cap \mathcal{A}_{\mathcal{R}}| \geq |\mathcal{E}_2 \cap \mathcal{A}_{\mathcal{R}}|$.

Clearly, in the absence of a characterization of the set of relevant arguments $\mathcal{A}_{\mathcal{R}}$ in the given AF, we need to address the general problem of elicitation of preferences among arguments from the attack

graph. To do this, one could consider the criterion of the frequency of an argument a , denoted by $\text{freq}(a, \text{Ext}(\mathcal{F}))$, i.e., the number of all extensions it appears in. Indeed, an argument can be considered as *at least as good as* another one if it is more frequent.

Definition 5 Let $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ be an AF s.t. $a, b \in \mathcal{A}$. Then, $a \gg b$ iff $\text{freq}(a, \text{Ext}(\mathcal{F})) > \text{freq}(b, \text{Ext}(\mathcal{F}))$.

The preference relation \gg is based on a notion of relevance of the arguments. It is important to note that for a given AF, this relation requires the computation of all extensions.

Another natural way to compare arguments can be done according to their corresponding attackers. Then, an argument highly attacked can be considered as the less preferred one. However, given an attack graph many arguments can be equally attacked. That is, one needs to prefer an argument with more arguments defending it and so on. For the next preference relation, we need some further notations.

Definition 6 Let $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ be an AF and $a \in \mathcal{A}$. We define the sequence of attacks of a as $\mathcal{R}_a = \langle R_1 \dots R_n \rangle$, which satisfies:

1. $\forall i, j \in \{1 \dots n\} \ i \neq j, R_i \cap R_j = \emptyset$,
2. $R_1 = \{(b, a) \in \mathcal{R}\}$ and $\forall i \in \{2 \dots n\}, R_i = \{(b, a) \in \mathcal{R} \mid (a, c) \in R_{i-1}\}$.

We are now ready to state the concept of sequence of attack-defence.

Definition 7 Let $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ be an AF, $a \in \mathcal{A}$ and $\mathcal{R}_a = \langle R_1 \dots R_n \rangle$ the sequence of attacks of a . Then, the sequence of attack-defence of a , denoted by $\mathcal{A}_a = \langle A_1 \dots A_n \rangle$, is defined as follows:

1. $\forall i \in \{1 \dots n\}, A_i \subseteq \mathcal{A}$,
2. $\forall i \in \{1 \dots n\}, A_i = \{b \mid (b, c) \in R_i\}$.

As is easily observed, A_1 contains all arguments attacking the initial argument a , A_2 contains arguments attacking A_1 (i.e., those defending a), A_3 contains arguments attacking A_2 (i.e., those attacking indirectly a), and so on.

Now, we associate each argument with a numeric vector depending on its sequence of attack-defence.

Definition 8 Let $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ be an AF, $a \in \mathcal{A}$ and $\mathcal{A}_a = \langle A_1 \dots A_n \rangle$ the sequence of attack-defence associated to a . Then, the attack-defence vector of a is defined as:

$$\mathcal{AD}_a = \langle |A_1|, -|A_2|, |A_3|, \dots, (-1)^{n-1}|A_n| \rangle$$

According to the definition of attack-defence sequence, it can be noticed that \mathcal{A}_a reflects only one direction. The next definition presents the sequence of defence-attack of an argument in the light of its attack-defence sequence.

Definition 9 Let $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ be an AF, $a \in \mathcal{A}$ and $\mathcal{A}_a = \langle A_1 \dots A_n \rangle$ the sequence of attack-defence of a . The sequence of defence-attack of a , denoted as $\mathcal{B}_a = \langle B_1 \dots B_n \rangle$, should satisfy the following conditions:

1. $\forall i \in \{1 \dots n\}, B_i \subseteq A_i$,
2. $\forall i \in \{1 \dots n\}, B_i = \{b \mid (a, b) \in \mathcal{R}, b \in A_i\}$.

Similarly, we define a numeric vector to each argument depending on its associated sequence of defence-attack as follows.

Definition 10 Let $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ be an AF, $a \in \mathcal{A}$ and $\mathcal{B}_a = \langle B_1 \dots B_n \rangle$ the sequence of defence-attack of a . Then, the defence-attack vector of a is defined as:

$$\mathcal{DA}_a = \langle -|B_1|, |B_2|, -|B_3|, \dots, (-1)^n |B_n| \rangle$$

Notice that the attack-defence and defence-attack vectors are both important for a fine elicitation of preferences so as to be able to choose the best argument.

In the following, we provide a new numeric vector that will be used to measure the strength of a given argument.

Definition 11 Let $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ be an AF and $a \in \mathcal{A}$. Let \mathcal{AD}_a and \mathcal{DA}_a be the attack-defence and defence-attack vectors of a , respectively. Then, the conflict vector of a , denoted by \mathcal{ADDA}_a , is defined as:

$$\mathcal{ADDA}_a = \langle |A_1|, -|B_1|, -|A_2|, |B_2|, \dots, (-1)^{n-1}|A_n|, (-1)^n |B_n| \rangle$$

We now present the new preference relation among two arguments based on an ordering of criteria, that is basically lexicographic, over their associated conflict vectors.

Let us remark that the two corresponding numeric vectors might have different lengths ⁶.

Definition 12 Let $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ be an AF and $a, b \in \mathcal{A}$. Let \mathcal{ADDA}_a and \mathcal{ADDA}_b be two conflict vectors of a and b , respectively. Then, $a \gg_{cv} b$ iff $\mathcal{ADDA}_a \preceq_{lex} \mathcal{ADDA}_b$.

Let us illustrate the above definitions through the following example:

Example 2 Let the AF $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ depicted in Figure 2. Then, the attack-defence sequence associated to the argument x_1 is $\mathcal{A}_{x_1} = \langle \{y_2\}, \{x_2\}, \{y_1, y_2\}, \{x_1, z_1\} \rangle$, while the defence-attack sequence of x_1 is $\mathcal{B}_{x_1} = \langle \{\}, \{\}, \{y_1, y_2\}, \{z_1\} \rangle$. For the argument y_2 , we have $\mathcal{A}_{y_2} = \langle \{x_2\}, \{y_1, y_2\}, \{x_1, z_1\}, \{y_2, x_1\} \rangle$ and $\mathcal{B}_{y_2} = \langle \{x_2\}, \{\}, \{x_1\}, \{x_1\} \rangle$. Then, we build the conflict vectors $\mathcal{ADDA}_{x_1} = \langle 1, 0, -1, 0, 2, -1, -2, 1 \rangle$ and $\mathcal{ADDA}_{y_2} = \langle 1, -1, -2, 0, 2, -1, -2, 1 \rangle$. As $\mathcal{ADDA}_{y_2} \preceq_{lex} \mathcal{ADDA}_{x_1}$, then $y_2 \gg_{cv} x_1$. This conclusion is rational. Indeed, both x_1 and y_2 are attacked by only one argument (e.g. y_2 and x_2 , respectively). However, y_2 attacks x_2 while y_2 is not attacked by x_1 .

Intuitively, it is important to note that the lexicographic preference relation implies that arguments against a directly (i.e., A_1) or indirectly (i.e., A_3) are considered as *enemies* of a , while those attacking the enemies are considered as *friends* of a (i.e., A_2). As $B_i \subseteq A_i$, B_i is considered as an enemy (respectively friend) iff A_i so is. So, the idea is to reduce the number of enemies, while maximizing the number of friends.

In the following, we are interested in how preferences among extensions can be defined using attacks between arguments. Before proceeding, we begin by extending the above definitions on arguments to sets of arguments. So, the notions of attack-defence and defence-attack of an extension can be defined in a simple way, as stated by the next definition:

⁶ Let $u = \langle u_1, u_2, \dots, u_m \rangle$ and $v = \langle v_1, v_2, \dots, v_n \rangle$ be two vectors with $m \leq n$. $u \preceq_{lex} v$, if u is a prefix of v or $\exists k \leq m$ s.t. $u_k < v_k$ and $u_i = v_i$ foreach $i < k$.

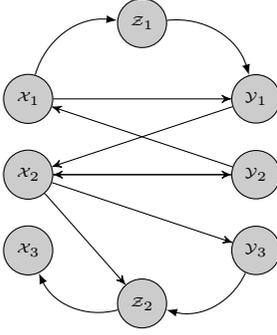


Figure 2. Attack graph of Example 2

Definition 13 Let $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ be an AF s.t. $\mathcal{E} \in \text{Ext}(\mathcal{F})$. We define $\mathcal{A}_{\mathcal{E}} = \bigcup_{a \in \mathcal{E}} \mathcal{A}_a$ and $\mathcal{B}_{\mathcal{E}} = \bigcup_{a \in \mathcal{E}} \mathcal{B}_a$ ⁷. For $\text{AD}_{\mathcal{E}}$ and $\text{DA}_{\mathcal{E}}$, they can be defined in a similar way as in Definition 8 and Definition 10, respectively.

Analogously to Definition 11, the conflict vector associated to an extension \mathcal{E} can be introduced as follows.

Definition 14 Let $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ be an AF and $\mathcal{E} \in \text{Ext}(\mathcal{F})$. Let $\text{AD}_{\mathcal{E}}$ and $\text{DA}_{\mathcal{E}}$ be the attack-defence and defence-attack vectors of \mathcal{E} , respectively. Then, we define:

$$\text{ADDA}_{\mathcal{E}} = \bigoplus_{a \in \mathcal{E}} \text{ADDA}_a$$
⁸

Let us now turn to the ordering relation between extensions.

Definition 15 Let $\mathcal{F} = \langle \mathcal{A}, \mathcal{R} \rangle$ be an AF and $\mathcal{E}_1, \mathcal{E}_2 \in \text{Ext}(\mathcal{F})$. Then, $\mathcal{E}_1 \succeq_{cv} \mathcal{E}_2$ iff

- $|\mathcal{E}_1| < |\mathcal{E}_2|$, or
- $|\mathcal{E}_1| = |\mathcal{E}_2|$ and $\text{ADDA}_{\mathcal{E}_1} \succeq_{lex} \text{ADDA}_{\mathcal{E}_2}$.

That is, the preference relation \succeq_{cv} considers the comparison between conflict vectors as a tiebreaker, i.e., when the two extensions \mathcal{E}_1 and \mathcal{E}_2 are of the same size. Notice that it is also possible to consider comparison of conflict vectors as a first criteria and tiebreak using another one such as size.

5 Top-k Extensions in Abstract Argumentation

Generally, an AF may admit an exponentially large number of extensions (cf. Example 1), from which it is difficult for the user to retrieve relevant information. In particular, if there is a lot of semantics extensions, only few (if any) arguments are in all of them. Then, using skeptical inference gives almost no information. Conversely, the credulous inference may lead to many information.

Recall that the results in the previous section show how to take advantage of the attack graph for comparing extensions using different criteria in order to derive significant inferences.

⁷ Let $U = \langle u_1 \dots u_n \rangle$ and $V = \langle v_1 \dots v_m \rangle$ be two sequences of sets. $U \uplus V = \langle u_1 \cup v_1 \dots u_n \cup v_n \rangle$. For two sequences of different size, the smallest is completed with empty sets.

⁸ Let $x = \langle x_1, x_2, \dots, x_m \rangle$ and $y = \langle y_1, y_2, \dots, y_n \rangle$ be two vectors with $m \leq n$. $x \oplus y$ is the vector $\langle x_1 + y_1, x_2 + y_2, \dots, x_m + y_m, x_{m+1}, \dots, x_n \rangle$.

In this section, we present a new approach that aims to reduce the size of the outcome of a given AF by focusing the enumeration problem only on the *best* extensions. Given a fixed bound k , the goal is to select the Top- k extensions according to a given preference relation. The problem we are interested in can be stated as follows:

Given an AF \mathcal{F} , a preference relation over extensions and a positive integer k , compute the set of Top- k extensions (under a given semantics).

To solve this problem, our approach follows those proposed by Jabbour et al. [24] in the context of Boolean satisfiability and data mining.

We describe the idea more formally below:

Definition 16 Let \mathcal{F} be an AF. Let \mathcal{E} be an extension s.t. $\mathcal{E} \in \text{Ext}(\mathcal{F})$, k a positive integer, and \succeq_* a preference relation over $\text{Ext}(\mathcal{F})$. Then, \mathcal{E} is called a Top- k extension for a given semantics if it admits less than k extensions preferred to it w.r.t. to \succeq_* .

According to Definition 16, it can be checked that the number of Top- k extensions is not necessarily equal to k . Indeed, it can be strictly greater or smaller than k . For instance, the number of Top- k extensions does not exceed k if it is assumed that the preference relation is a total order.

Notice also that setting the value of k aims to reduce the size of the output. This parameter is made available to the user to reduce the output to a reasonable size, in order to enable him to make his choice among the Top- k extensions. The usual way is that the user starts by setting k to a small value (e.g. Top-1) and then increases such bound if he is not satisfied by the first provided extensions. However, if the total number of extensions is known, in this case, the value k can be set as a function of the size of the output.

Our approach is based on a reduction of the problem of enumerating Top- k extensions of an AF to the problem of computing Top- k models of a propositional formula. Let us recall two recent contributions on which our approach is based.

On one hand, computing Top- k models of a propositional formula in *conjunctive normal form* (CNF) with an original application to data mining has been recently proposed in [24]. The authors present an efficient algorithm to enumerate Top- k models based on a preference relation over models of the considered propositional formula restricted to a subset of its variables. That is, they use some preference relations called δ -preference relations (Definition 17) allowing to add constraints to the CNF in order to avoid the enumeration of models that are less preferred to the previously found ones. A δ -preference relation is then dynamically used as a lower bound during the enumeration process.

Definition 17 ([24]) Let Φ be a propositional formula and \succeq a preference relation on the models of Φ . Then, \succeq is a δ -preference relation if there exists a polytime function f_{\succeq} from Boolean interpretations to the set of CNF formulae such that, for all model \mathcal{M} of Φ and for all Boolean interpretation \mathcal{M}' , \mathcal{M}' is a model of $\Phi \wedge f_{\succeq}(\mathcal{M})$ iff \mathcal{M}' is a model of Φ and $\mathcal{M} \not\succeq \mathcal{M}'$.

On the other hand, different algorithms have been studied for enumerating extensions prescribed by a given semantics in AFs. Among the various semantics from the literature, we are only interested in the *complete* semantics, which is one of the most important semantics in Dung's theory, since it is introduced as a way to connect preferred (maximum complete) and grounded (minimum complete) semantics. Intuitively, the notion of complete extensions captures some kind of

confidence of a rational agent who believes in every thing he can defend [17]. Another point to mention is that it has been proved that complete-based semantics (that is semantics whose set of extensions are subsets of the set of all complete extensions), when used for the purpose of logical inference, tend to produce fully instantiated argumentation formalisms that satisfy reasonable properties [12]. In [13], the authors encode the constraints corresponding to complete extensions as a propositional formula. Then, a SAT-solver is called to solve it, thus returning a complete extension. Our approach combines these two interesting techniques in a single framework.

Now, in order to compute the set of Top- k extensions of a given AF, we only need to show that the preference relations defined in Section 4 are δ -preference relations. Accordingly, such δ -preference relations are used in our implementation of the Top- k extensions enumeration problem.

Proposition 1 *The preference relations \succeq_i , $i \in \{1, \dots, 10, cv\}$ are δ -preference relations.*

Proof 1 *Let \mathcal{F} be an AF. Denote by $\Phi_{\mathcal{F}}$ the CNF formula encoding the AF \mathcal{F} where \mathcal{V}_E is the set of variables corresponding to the arguments in \mathcal{F} . Let us show that \succeq_1 , \succeq_9 , and \succeq_{10} are δ -preference relations. For the remaining preference relations, the proof is similar. Given a model \mathcal{M} of $\Phi_{\mathcal{F}}$ corresponding to an extension $\mathcal{E} \in \text{Ext}(\mathcal{F})$, we only need to define for each preference relation \succeq , its associated lower bound $f_{\succeq}(\mathcal{M})$. Notice that arguments and boolean variables representing such arguments will be denoted similarly.*

- $f_{\succeq_1}(\mathcal{M})$: let us recall that \succeq_1 is based on a pre-order over arguments. This pre-order is a δ -preference relation. One can define $f_{\succeq_1}(\mathcal{M}) = (\bigvee_{x_i \in \text{Pref}_{\mathcal{E}}} x_i) \wedge (\bigvee_{y \in \mathcal{E}} \neg y)$, where $\text{Pref}_{\mathcal{E}}$ is the set of arguments that are preferred to each argument in \mathcal{E} . Indeed, for an extension \mathcal{E} , $(\bigvee_{x_i \in \text{Pref}_{\mathcal{E}}} x_i)$ allows to look for extensions having at least one argument preferred to those in \mathcal{E} while $(\bigvee_{y \in \mathcal{E}} \neg y)$ excludes \mathcal{E} itself.
- $f_{\succeq_9}(\mathcal{M})$: let us recall that \succeq_9 associates with each extension \mathcal{E} a value corresponding to the size of \mathcal{E} . This pre-order is a δ -preference relation. Now, we can define a lower bound $f_{\succeq_9}(\mathcal{M}) = \sum_{x_i \in \mathcal{A}} x_i \geq |\mathcal{E}|$. Indeed, for an extension \mathcal{E} , it is sufficient to add such cardinality constraint to allow the solver to eliminate the set of less-preferred models to \mathcal{M} . This constraint is then combined with the clause $(\bigvee_{y \in \mathcal{E}} \neg y)$ allowing to exclude \mathcal{E} in the next run. The same constraint can be slightly modified for \succeq_{10} leading to $f_{\succeq_{10}}(\mathcal{M}) = \sum_{x_i \in \mathcal{A}_R} x_i \geq |\mathcal{E} \cap \mathcal{A}_R|$.

Computing Top- k Complete Extensions

Our algorithm for computing Top- k complete extensions is based on the Algorithm 1 proposed in [24] to enumerate the Top- k models of a propositional formula (Top- k SAT problem). Indeed, as mentioned previously, we first encode the constraints corresponding to complete extensions as a propositional formula using the approach proposed in [13], then we apply Algorithm 1. In order to make the paper self contained, we recall in details the Jabbour et al. algorithm [24] used for computing Top- k models using the δ -preference relations defined above. The basic idea is simply to use the formula $f_{\succeq}(\mathcal{M})$ associated to a model \mathcal{M} to obtain models that are at least as preferred as \mathcal{M} . This algorithm takes as input a CNF formula Φ , a preference relation \succeq , a strictly positive integer k , and a set X of propositional variables allowing to define the equivalence relation \approx_X [24]. It has as output a set \mathcal{L} of Top- k models of Φ satisfying the two properties given in the definition of the Top- k SAT problem [24].

Algorithm 1: Top- k

Input: a CNF formula Φ , a pre-order relation \succeq , an integer $k \geq 1$, and a set X of Boolean variables

Output: A set of Top- k models \mathcal{L}

```

1  $\Phi' \leftarrow \Phi$ ;
2  $\mathcal{L} \leftarrow \emptyset$ ; /* Set of all Top- $k$  models */
3 while ( $\text{solve}(\Phi')$ ) do /*  $\mathcal{M}$  is a model of  $\Phi'$  */
4   if ( $\exists M' \in \mathcal{L}. M \approx_X M' \ \& \ M \succ M'$ ) then
5      $\text{replace}(M, M', \mathcal{L})$ ;
6   else if ( $\forall M' \in \mathcal{L}. M \not\approx_X M' \ \& \ |\text{pref}(M, \mathcal{L})| < k$ ) then
7      $S \leftarrow \text{min\_top}(k, \mathcal{L})$ ;
8      $\text{add}(M, \mathcal{L})$ ;
9      $\text{remove}(k, \mathcal{L})$ ;
10     $S \leftarrow \text{min\_top}(k, \mathcal{L}) \setminus S$ ;
11     $\Phi' \leftarrow \Phi' \wedge \bigwedge_{M' \in S} f_{\succeq}(M')$ ;
12  else
13     $\Phi' \leftarrow \Phi' \wedge f_{\succeq}(M)$ ;
14   $\Phi' \leftarrow \Phi' \wedge \overline{M}$ ;
15 return  $\mathcal{L}$ ;

```

Algorithm Description In the while-loop, we use lower bounds for finding optimal models. These lower bounds are obtained by using the fact that the pre-order relation considered is a δ -preference relation. In each step, the lower bound is integrated by using the formula:

$$\bigwedge_{M' \in S} f_{\succeq}(M')$$

- **Lines 4 – 5.** Let us first mention that the procedure $\text{replace}(\mathcal{M}, M', \mathcal{L})$ replaces M' with \mathcal{M} in \mathcal{L} . We apply this replacement because there exists a model M' in \mathcal{L} which is equivalent to M' and \mathcal{M} allows to have a better bound. It is important to note that as the encoding of the constraints corresponding to complete extensions as a propositional formula requires the use of additional variables in addition to variables encoding the arguments, noted X , we consider only the projection of each model over the variables representing the arguments.
- **Lines 6 – 11.** In the case where \mathcal{M} is not equivalent to any model in \mathcal{L} and the number of models in \mathcal{L} preferred to it is strictly less than k ($|\text{pref}(\mathcal{M}, \mathcal{L})| < k$), we add \mathcal{M} to \mathcal{L} ($\text{add}(\mathcal{M}, \mathcal{L})$). Note that S contains first the models of \mathcal{L} before adding \mathcal{M} that have exactly $k - 1$ models preferred to them in this set. After adding \mathcal{M} to \mathcal{L} , we remove from \mathcal{L} the models that are not Top- k , i.e., they have more than $k - 1$ models in \mathcal{L} that are strictly preferred to them ($\text{remove}(k, \mathcal{L})$). Next, we modify the content of S . Note that the elements of S before adding \mathcal{M} are used as bounds in the previous step. Hence, in order to avoid adding the same bound several times, the new content of S corresponds to the models in \mathcal{L} that have exactly $k - 1$ models preferred to them in \mathcal{L} ($\text{min_top}(k, \mathcal{L})$) deprived of the elements of the previous content of S . In line 11, we integrate lower bounds in Φ' by using the elements of S . Indeed, for all model \mathcal{M} of a formula $\Phi' \wedge \bigwedge_{M' \in S} f_{\succeq}(M')$, $M' \not\approx \mathcal{M}$ holds, for any $M' \in S$.
- **Lines 12 – 13.** In the case where \mathcal{M} is not a Top- k model, we integrate its associated lower bound.
- **Line 14.** This instruction enables us to avoid finding the same model in two different steps of the while-loop.

6 Implementation and Experiments

In this section, we carried out an experimental evaluation of the performance of our Algorithm for computing Top- k complete extensions. The primary goal is to assess the effectiveness of our proposed approach. For this purpose, we consider the problem of enumerating Top- k complete extensions according to a preference relation denoted Top- k_{\succeq_*} , where $*$ \in $\{9, 10, 11\}$. First, notice that in order to derive top- k complete extensions we have only consider methods that are not based on a given preference on the set of arguments. The idea is that, instead of modifying the inputs of the argumentation framework, we only extend the outcomes with preference relations. These outcomes promote certain extensions and that different audiences have different preferences over extensions, from which the preferences over outcomes are derived. Let us also stress that \succeq_{11} is a special case of \succeq_{cv} where extensions are compared according to the sum of attacks over their arguments.

For our experiments, we consider the SAT-based encoding of the preferred extensions enumeration problem described in [13]. Indeed, a complete labeling corresponds to a complete extension and vice-versa. In this encoding, a complete extension can be extracted from a model of the formula by considering the true Boolean variables corresponding to arguments labeled *in*. To enumerate the Top- k complete extensions of the encoded propositional formula, we use Algorithm 1 described in Section 5. As a lower bound or δ -preference relations, we exploit those introduced in Section 4. The algorithm is implemented on the top of the state-of-the-art SAT solver MiniSAT 2.2⁹. The cardinality constraint involved in the encoding of the δ -preference relation is managed by maintaining counters during search. Indeed, we maintain dynamically the sum of the variables involved in the cardinality constraint.

In our experiments, the set of instances are taken from the web site of the argumentation project on *New Methods for Analyzing, Comparing and Solving Argumentation Problems*¹⁰. All the experiments were conducted on Intel Xeon quad-core machines with 32GB of RAM running at 2.66 Ghz. Table 1 presents the characteristics of the different AFs used in the experimental analysis. The first column gives the name of the instance. In the second and third column, we give the number of arguments (#args) and attacks relations (#attacks), respectively. In the fourth column, we report in parenthesis the time in seconds needed to compute the Top-1000 complete extensions for the set of considered preference relations \succeq_9 , \succeq_{10} , and \succeq_{11} , respectively. The last column presents the computation time needed to compute all complete extensions.

Table 1. Top-1000 vs all Extensions (time)

instance	#args	#attacks	Top-1000	Ext(F)
af_nbr10_150_10_0.65	150	697	(0.2, 0.14, 0.22)	0.7
af_nbr10_175_10_0.65	175	809	(0.09, 0.1, 0.15)	0.2
af_nbr10_200_10_0.65	200	920	(0.63, 0.88, 0.51)	12.8
af_nbr10_225_10_0.65	225	1035	(0.34, 0.53, 0.63)	4.4
af_nbr10_275_10_0.65	275	1289	(6.4, 5.4, 8.5)	> 300
af_nbr10_300_10_0.65	300	1368	(0.9, 0.4, 0.5)	21
af_nbr9_75_7_0.9	75	433	(9.1, 10.2, 11.5)	> 300

Empirical Evaluations In order to analyze the behavior of our Top- k algorithm over $\text{Ext}(\mathcal{F})$, we conducted two kinds of experiments. In the first one, the value of k is fixed to 1000. We evaluate

⁹ MiniSAT: <http://minisat.se/>

¹⁰ <http://www.dbai.tuwien.ac.at/proj/argumentation/dynpartix>

then the time needed to compute all complete extensions compared to the time needed for generating the Top-1000. We consider the preference relation \succeq_9 , \succeq_{10} , and \succeq_{11} .

Results on a representative set of instances are shown in Table 1. The other instances present similar behavior. As expected, the CPU time needed for computing the Top-1000 complete extensions is clearly less than the time needed to find all of them. Then, our experimental evaluation clearly shows that finding the Top- k complete extensions can be computed efficiently for reasonable values of k with respect to the set of all extensions.

In the second experiment, k is varied from 1 to 10000. As shown in Table 2, the number of Top- k complete extensions increases, in general, with k for all the considered preference relations. More interestingly, the number of extensions is closed to k , especially for \succeq_{10} and \succeq_{11} . This clearly shows that these two criteria are more appropriate than the basic one considering the size based preference relation. Indeed, the preference relation \succeq_9 does not discriminate better between extensions, i.e., many of them might have the same size. Consequently, the number of Top- k will be greater than k .

Table 2. af_nbr10_150_10_0.4 instance: Number of Top- k complete extensions

k	Preference: \succeq_9	Preference: \succeq_{10}	Preference: \succeq_{11}
1	96	2	2
100	878	113	109
1000	3069	1080	1134
5000	6344	5176	5098
7500	9344	7977	7312
10000	11780	10335	10278

7 Conclusion and Perspectives

During the last years, abstract argumentation frameworks provided the basis for various reasoning problems in the area of Artificial Intelligence. Unfortunately, the large number of its possible extensions makes some reasoning problems intractable in this framework, since enumerating all the extensions is computationally expensive. This paper makes a connection between orderings on a powerset and its application to ordering extensions in an abstract argumentation framework. We have described and studied various forms of selecting the best extensions using some criterion in order to get more significant inferences. To do this, we imposed several partial orderings on the set of extensions based on comparing arguments taking into account various influences such as relevance of the arguments, number of arguments attacking it, and attack-defense sequences. The selection of extensions allows us to limit the number of extensions considered to a specified limit k , this set will be referred to as the Top- k extensions. For this problem, we considered an approach proposed in the context of data mining that computes the Top- k models, on a propositional formula that represents the complete extensions enumeration problem. The implemented algorithm tested on various datasets demonstrates the effectiveness and scalability of our approach and the accuracy of the enumeration method in abstract argumentation.

For future work, the experimental results suggest to study other semantics (such as preferred, semi-stable, and stage semantics, etc) and to analyse additional experimentation by considering different benchmarks extracted from real debates. We also plan to apply more programming techniques like the parallelism [14] to improve the current proposal of the Top- k computation. We are also interested in

identifying other ordering criteria to compare outcomes of an argumentation framework. Finally, it would be interesting to apply our framework to further formalisms extending the Dung-style frameworks such as abstract dialectical frameworks, and weighted argumentation systems.

REFERENCES

- [1] Leila Amgoud and Jonathan Ben-Naim, 'Ranking-based semantics for argumentation frameworks', in *Proc. of the 7th SUM Conference*, pp. 134–147, (2013).
- [2] Leila Amgoud and Henri Prade, 'Using arguments for making and explaining decisions', *Artificial Intelligence*, **173**(3-4), 413–436, (2009).
- [3] Leila Amgoud and Srdjan Vesic, 'Repairing preference-based argumentation frameworks', in *Proc. of the 21th IJCAI Conference*, pp. 665–670, (2009).
- [4] Leila Amgoud and Srdjan Vesic, 'A new approach for preference-based argumentation frameworks', *Annals of Mathematics and Artificial Intelligence*, **63**(2), 149–183, (2011).
- [5] Leila Amgoud and Srdjan Vesic, 'Rich preference-based argumentation frameworks', *International Journal Approximate Reasoning*, **55**(2), 585–606, (2014).
- [6] Pietro Baroni and Massimiliano Giacomin, 'On principle-based evaluation of extension-based argumentation semantics', *Artificial Intelligence*, **171**(10-15), 675–700, (2007).
- [7] Trevor J. M. Bench-Capon, 'Persuasion in practical argument using value-based argumentation frameworks', *Journal of Logic and Computation*, **13**(3), 429–448, (2003).
- [8] Stefano Bistarelli and Francesco Santini, 'Conarg: A constraint-based computational framework for argumentation systems', in *Proc. of the 23rd ICTAI Conference*, pp. 605–612, (2011).
- [9] Andrei Bondarenko, Phan Minh Dung, Robert A. Kowalski, and Francesca Toni, 'An abstract, argumentation-theoretic approach to default reasoning', *Artificial Intelligence*, **93**, 63–101, (1997).
- [10] Ronen I. Brafman, Carmel Domshlak, Solomon Eyal Shimony, and Y. Silver, 'Preferences over sets', in *Proc. of the 21rd AAAI Conference*, pp. 1101–1106, (2006).
- [11] Gerhard Brewka, Mirosław Truszczynski, and Stefan Woltran, 'Representing preferences among sets', in *Proc. of the 24th AAAI Conference*, (2010).
- [12] Martin W. A. Caminada and Dov M. Gabbay, 'A logical account of formal argumentation', *Studia Logica*, **93**(2-3), 109–145, (2009).
- [13] Federico Cerutti, Paul E. Dunne, Massimiliano Giacomin, and Mauro Vallati, 'Computing preferred extensions in abstract argumentation: A sat-based approach', in *Proc. of the 2nd TAFE Workshop*, pp. 176–193, (2013).
- [14] Federico Cerutti, Ilias Tachmazidis, Mauro Vallati, Sotirios Batsakis, Massimiliano Giacomin, and Grigoris Antoniou, 'Exploiting parallelism for hard problems in abstract argumentation', in *Proc. of the 29th AAAI Conference*, pp. 1475–1481, (2015).
- [15] Günther Charwat and Wolfgang Dvorák, 'dynPARTIX 2.0 - dynamic programming argumentation reasoning tool', in *Proc. of the 3rd COMMA Conference*, pp. 507–508, (2012).
- [16] Yannis Dimopoulos, Bernhard Nebel, and Francesca Toni, 'On the computational complexity of assumption-based argumentation for default reasoning', *Artificial Intelligence*, **141**(1/2), 57–78, (2002).
- [17] Phan Minh Dung, 'On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games', *Artificial Intelligence*, **77**(2), 321–358, (1995).
- [18] Paul E. Dunne and Michael Wooldridge, 'Complexity of abstract argumentation', *Argumentation in AI*, Springer, **186**, 85–104, (2009).
- [19] Wolfgang Dvorák, Matti Järvisalo, Johannes Peter Wallner, and Stefan Woltran, 'Complexity-sensitive decision procedures for abstract argumentation', *Artificial Intelligence*, **206**, 53–78, (2014).
- [20] Uwe Egly, Sarah Alice Gaggl, and Stefan Woltran, 'ASPARTIX: implementing argumentation frameworks using answer-set programming', in *Proc. of the 24th ICLP Conference*, pp. 734–738, (2008).
- [21] Davide Grossi and Sanjay Modgil, 'On the graded acceptability of arguments', in *Proc. of the 24th IJCAI Conference*, pp. 868–874, (2015).
- [22] Anthony Hunter, 'A probabilistic approach to modelling uncertain logical arguments', *International Journal of Approximate Reasoning*, **54**(1), 47–81, (2013).
- [23] Anthony Hunter, 'Modelling the persuadee in asymmetric argumentation dialogues for persuasion', in *Proc. of the 24th IJCAI Conference*, pp. 3055–3061, (2015).
- [24] Saïd Jabbour, Lakhdar Sais, and Yakoub Salhi, 'The Top-k frequent closed itemset mining using Top-k SAT problem', in *Proc. of the 24th ECML/PKDD Conference*, pp. 403–418, (2013).
- [25] Wolfgang Dvorák Matti Järvisalo, Johannes Peter Wallner, and Stefan Woltran, 'CEGARTIX: a sat-based argumentation system', in *Proc. of the 3rd POS Workshop*, (2012).
- [26] Souhila Kaci, 'Refined preference-based argumentation frameworks', in *Proc. of the 2nd COMMA Conference*, pp. 299–310, (2010).
- [27] Souhila Kaci and Leendert W. N. van der Torre, 'Preference-based argumentation: Arguments supporting multiple values', *Int. J. Approx. Reasoning*, **48**(3), 730–751, (2008).
- [28] Sébastien Konieczny, Pierre Marquis, and Srdjan Vesic, 'On supported inference and extension selection in abstract argumentation frameworks', in *Proc. of the 13th ECSQARU Conference*, pp. 49–59, (2015).
- [29] Jean-Marie Lagniez, Emmanuel Lonca, and Jean-Guy Mailly, 'Co-quaas: A constraint-based quick abstract argumentation solver', in *Proc. of the 27th ICTAI Conference*, pp. 928–935, (2015).
- [30] Paul-Amoury Matt and Francesca Toni, 'A game-theoretic measure of argument strength for abstract argumentation', in *Proc. of the 11th JELIA Conference*, pp. 285–297, (2008).
- [31] Sanjay Modgil, 'Reasoning about preferences in argumentation frameworks', *Artificial Intelligence*, (2009).
- [32] Sanjay Modgil and Henry Prakken, 'A general account of argumentation with preferences', *Artificial Intelligence*, **195**, 361–397, (2013).
- [33] Henry Prakken, 'Formal systems for persuasion dialogue', *Knowledge Engineering Review*, **21**(2), 163–188, (2006).
- [34] Henry Prakken and Giovanni Sartor, 'Argument-based extended logic programming with defeasible priorities', *Journal of Applied Non-Classical Logics*, **7**(1), 25–75, (1997).
- [35] Iyad Rahwan and Kate Larson, 'Pareto optimality in abstract argumentation', in *Proc. of the 23rd AAAI Conference*, pp. 150–155, (2008).
- [36] Iyad Rahwan and Guillermo R. Simari, 'Argumentation in artificial intelligence', Springer, (2009).
- [37] Tjitze Rienstra, Matthias Thimm, and Nir Oren, 'Opponent models with uncertainty for strategic argumentation.', in *Proc. of the 23rd IJCAI Conference*, pp. 322–338, (2013).
- [38] Avi Rosenfeld, Inon Zuckerman, Erel S. Halevi, Osnat Drein, and Sarit Kraus, 'Negochat: A chat-based negotiation agent', in *Proc. of the 13th AAMAS Conference*, pp. 525–532, (2014).
- [39] Guillermo Ricardo Simari and Ronald Prescott Loui, 'A mathematical treatment of defeasible reasoning and its implementation', *Artificial Intelligence*, **53**(2-3), 125–157, (1992).
- [40] Nouredine Tamani and Madalina Croitoru, 'A quantitative preference-based structured argumentation system for decision support', in *Proc. of the 23rd FUZZ-IEEE Conference*, pp. 1408–1415, (2014).
- [41] Matthias Thimm, 'Tweety: A comprehensive collection of java libraries for logical aspects of artificial intelligence and knowledge representation', in *Proc. of the 16th KR Conference*, (2014).

On Revision of Partially Specified Convex Probabilistic Belief Bases¹

Gavin Rens² and Thomas Meyer³ and Giovanni Casini⁴

Abstract. We propose a method for an agent to revise its incomplete probabilistic beliefs when a new piece of propositional information is observed. In this work, an agent's beliefs are represented by a set of probabilistic formulae – a belief base. The method involves determining a *representative* set of 'boundary' probability distributions consistent with the current belief base, revising each of these probability distributions and then translating the revised information into a new belief base. We use a version of Lewis Imaging as the revision operation. The correctness of the approach is proved. An analysis of the approach is done against six rationality postulates. The expressivity of the belief bases under consideration are rather restricted, but has some applications. We also discuss methods of belief base revision employing the notion of optimum entropy, and point out some of the benefits and difficulties in those methods. Both the boundary distribution method and the optimum entropy methods are reasonable, yet yield different results.

1 INTRODUCTION

Suppose an agent represents its probabilistic knowledge with a set of statements; every statement says something about the probability of some features the agent is aware of. Ideally, the agent would want to have enough information to, at least, identify one probability distribution over all the situations (worlds) it deems possible. However, if the agent could not gather sufficient data or if it was not told or given sufficient information, it would not be able to pinpoint exactly one probability distribution. An agent with this sort of ignorance, can be thought of as having beliefs compatible with a *set of* distributions. Now, this agent might need to revise its beliefs when new (non-probabilistic) information is received, even though the agent's beliefs do not characterize a *particular* probability distribution over its current possible worlds.

Several researchers argue that using a single probability distribution requires the agent to make unrealistically precise uncertainty distinctions [9, 27, 29]. "One widely-used approach to dealing with this has been to consider sets of probability measures as a way of modeling uncertainty," [9]. However, simply applying standard probabilistic conditioning to each of the measures/distributions in the set individually and then combining the results is either not recommended

because it produces unsatisfactory results [9, 27] or because it is not computable due to the set being infinite. The framework presented in this paper proposes two ways to go from one 'probabilistically incomplete' belief base to another when new information is acquired.

Both belief revision methods presented, essentially follow this process: From the original belief base, determine a relatively small set of belief states / probability distributions which are 'compatible' with the belief base and in a sense, representative of the belief base. (We shall use the terms *belief state* and *probability distribution* interchangeably). Then revise only the belief states in this representative set. Finally, induce a new, revised belief base from the revised representative set.

We shall present two approaches to determine the representative set of belief states from the current belief base: (i) The approach we focus on involves finding belief states which, in a sense, are at the boundaries of the constraints implied by the belief base. These 'boundary belief states' can be thought of as drawing the outline of the convex space of beliefs. This outline is then revised to form a new outline shape, which can be translated into a new belief base. (ii) As a possible alternative approach, the representative set is a *single* belief state which can be imagined to be at the center of the outline of the first approach. This 'central' belief state is found by determining the one in the space of beliefs which is least biased or most entropic in terms of information theory [11, 4].

For approach (i) – where the canonical set is the set of boundary belief states – we shall prove that the revised canonical set characterizes the set of all belief states which would have resulted from revising all (including interior) belief states compatible with the original belief base.

The next section provides the relevant background theory and notation. Section 3 presents a generalized imaging method for revising probabilistic belief states. Then we describe the application of generalized imaging in our main contribution; revising boundary belief states instead of all belief states. The subsequent section explain two other approaches of revising our belief bases, based on optimum entropy. The first method finds a single representative belief state through maximum entropy inference and the second method revises boundary belief states using minimum cross-entropy inference. All three methods can be considered motivated methods, yet yield different results. Then, in Section 6, we shall list six traditional rationality postulates, and check how well the main approach fares against them. The related work is discussed in Section 7. We end with a section on future possible directions of research and some concluding remarks.

2 PRELIMINARIES

We shall work with classical propositional logic. Let \mathcal{P} be the finite set of n atomic propositional variables (*atoms*, for short). Formally,

¹ This is a modified and extended version of a paper that was presented at the Sixteenth International Workshop on Non-Monotonic Reasoning, 22–24 April 2016, Cape Town, South Africa.

² Centre for AI Research, and University of KwaZulu-Natal, School of Mathematics, Statistics and Computer Science, and CSIR Meraka, South Africa, email: gavinrens@gmail.com

³ Centre for AI Research, and University of Cape Town, Department of Computer Science, and CSIR Meraka, South Africa, email: tmeyer@cs.uct.ac.za

⁴ University of Luxembourg, Computer Science and Communication Research Unit, Luxembourg, email: giovanni.casini@uni.lu

a *world* is a unique assignment of truth values to all the atoms in \mathcal{P} . There are thus 2^n conceivable worlds. An agent may consider some non-empty subset W of the conceivable worlds; W is called the possible worlds. Often, in the exposition of this paper, a world will be referred to by its truth vector. For instance, if the vocabulary is placed in order $\langle q, r \rangle$ and $w_3 \models \neg q \wedge r$, then w_3 may be referred to as 01.⁵ Let L be all propositional formulae which can be formed from \mathcal{P} and the logical connectives \wedge and \neg , with \top abbreviating tautology and \perp abbreviating contradiction.

Let β be a sentence in L . $[\beta]$ denotes the set of β -worlds, that is, the elements of W satisfying β . The worlds satisfying all sentences in a set of sentences K are denoted by $[K]$.

We define the probabilistic language $L^{prob} = \{(\alpha) \bowtie x \mid \alpha \in L, \bowtie \in \{\leq, =, \geq\}, x \in [0, 1]\}$. We propose a belief base (BB) to be a consistent (logically satisfiable) subset of L^{prob} . A BB specifies an agent's knowledge.

The basic semantic element of an agent's beliefs is a probability distribution or a *belief state*

$$b = \{(w_1, p_1), (w_2, p_2), \dots, (w_n, p_n)\},$$

where p_i is the probability that w_i is the actual world in which the agent is. $\sum_{(w,p) \in b} p = 1$. We may also use c to refer to a belief state. For parsimony, let $b = \langle p_1, \dots, p_n \rangle$ be the probabilities that belief state b assigns to w_1, \dots, w_n where $\langle w_1, w_2, w_3, w_4 \rangle = \langle 11, 10, 01, 00 \rangle$, and $\langle w_1, w_2, \dots, w_8 \rangle = \langle 111, 110, \dots, 000 \rangle$. Let Π be the set of all belief states over W .

$b(\alpha)$ abbreviates $\sum_{w \in W, w \models \alpha} b(w)$. b satisfies formula $(\alpha) \bowtie x$ (denoted $b \models (\alpha) \bowtie x$) iff $b(\alpha) \bowtie x$. If B is a set of formulae, then b satisfies B (denoted $b \models B$) iff $\forall \gamma \in B, b \models \gamma$. If B and B' are sets of formulae, then B entails B' (denoted $B \models B'$) iff for all $b \in \Pi$, $b \models B'$ whenever $b \models B$. If $B \models \{\gamma\}$ then we simply write $B \models \gamma$. B is logically equivalent to B' (denoted $B \equiv B'$) iff $B \models B'$ and $B' \models B$.

Instead of an agent's beliefs being represented by a single belief state, a BB B represents a *set* of belief-states: Let $\Pi^B := \{b \in \Pi \mid b \models B\}$. A BB B is *satisfiable* (*consistent*) iff $\Pi^B \neq \emptyset$. We can now also define entailment as $B \models B'$ iff $\Pi^B \subseteq \Pi^{B'}$.

The technique of *Lewis imaging* for the revision of belief states, requires a notion of distance between worlds to be defined. Various notions of distance are possible, however, a study of their influence on the imaging technique is beyond the scope of this paper. We use a pseudo-distance measure between worlds, as defined by Lehmann et al. [18] and adopted by Chhogyal et al. [3]. We add a 'faithfulness' condition, which we feel is lacking from the definition of Lehmann et al. [18]: without this condition, a pseudo-distance measure would allow all worlds to have zero distance between them. Boutilier [2] mentions this condition, and we use his terminology: "faithfulness".

Definition 1 A pseudo-distance function $d : W \times W \rightarrow \mathbb{Z}$ satisfies the following five conditions: for all worlds $w, w', w'' \in W$,

1. $d(w, w') \geq 0$ (Non-negativity)
2. $d(w, w) = 0$ (Identity)
3. $d(w, w') = d(w', w)$ (Symmetry)
4. $d(w, w') + d(w', w'') \geq d(w, w'')$ (Triangular Inequality)
5. if $w \neq w'$, then $d(w, w') > 0$ (Faithfulness)

Presently, the foundation theory, or paradigm, for studying belief change operations is commonly known as AGM theory [1, 7]. Typically, belief change (in a static world) can be categorized as expansion, revision or contraction, and is performed on a belief set, the set

of sentences K closed under logical consequence. Expansion (denoted $+$) is the logical consequences of $K \cup \{\alpha\}$, where α is new information and K is the current belief set. Contraction of α is the removal of some sentences until α cannot be inferred from K . Revision is when α is (possibly) inconsistent with K and K is (minimally) modified so that the new K remains consistent and entails α . In this view, when the new information is consistent with the original beliefs, expansion and revision are equivalent.

3 GENERALIZED IMAGING

It is not yet universally agreed what revision means in a probabilistic setting. One school of thought says that probabilistic expansion is equivalent to Bayesian conditioning. This is evidenced by Bayesian conditioning (BC) being defined only when $b(\alpha) \neq 0$, thus making BC expansion equivalent to BC revision. In other words, one could define expansion (restricted revision) to be

$$b \text{ BC } \alpha = \{(w, p) \mid w \in W, p = b(w \mid \alpha), b(\alpha) \neq 0\},$$

where $b(w \mid \alpha)$ can be defined as $b(\phi_w \wedge \alpha) / b(\alpha)$ and ϕ_w is a sentence identifying w (i.e., a complete theory for w).

To accommodate cases where $b(\alpha) = 0$, that is, where α contradicts the agent's current beliefs and its beliefs need to be revised in the stronger sense, we shall make use of *imaging*. Imaging was introduced by Lewis [20] as a means of revising a probability distribution, and has been discussed in other work too [7, 6, 3, 25]. Informally, Lewis's original solution for accommodating contradicting evidence α is to move the probability of each world to its closest, α -world. Lewis made the strong assumption that every world has a *unique* closest α -world. More general versions of imaging allows worlds to have *several*, equally proximate, closest worlds.

Gärdenfors [7] calls one generalization of Lewis's imaging, *general imaging*. Our method is also a generalization of Lewis's imaging. We thus refer to his as *Gärdenfors's general imaging* and to our method as *generalized imaging* to distinguish them. It should be noted that all three these imaging methods are general revision methods and can be used in place of Bayesian conditioning for expansion. "Thus imaging is a more general method of describing belief changes than conditionalization," [7, p. 112] in the sense that Bayesian conditioning cannot deal with contradicting evidence but imaging can.

Let $Min(\alpha, w, d)$ be the set of α -worlds closest to w with respect to pseudo-distance d . Formally,

$$Min(\alpha, w, d) := \{w' \in [\alpha] \mid \forall w'' \in [\alpha], d(w', w) \leq d(w'', w)\},$$

where $d(\cdot)$ is some pseudo-distance measure between worlds (e.g., Hamming or Dalal distance).

Example 1 Let the vocabulary be $\{q, r, s\}$. Let α be $(q \wedge r) \vee (q \wedge \neg r \wedge s)$. Suppose d is Hamming distance. Then

$$\begin{aligned} Min((q \wedge r) \vee (q \wedge \neg r \wedge s), 111, d) &= \{111\} \\ Min((q \wedge r) \vee (q \wedge \neg r \wedge s), 110, d) &= \{110\} \\ Min((q \wedge r) \vee (q \wedge \neg r \wedge s), 101, d) &= \{101\} \\ Min((q \wedge r) \vee (q \wedge \neg r \wedge s), 100, d) &= \{110, 101\} \\ Min((q \wedge r) \vee (q \wedge \neg r \wedge s), 011, d) &= \{111\} \\ Min((q \wedge r) \vee (q \wedge \neg r \wedge s), 010, d) &= \{110\} \\ Min((q \wedge r) \vee (q \wedge \neg r \wedge s), 001, d) &= \{101\} \\ Min((q \wedge r) \vee (q \wedge \neg r \wedge s), 000, d) &= \{110, 101\} \end{aligned}$$

⁵ $w \models \alpha$ is read 'w is a model for/satisfies α '.

Definition 2 (GI) Then generalized imaging (denoted GI) is defined as

$$b \text{ GI } \alpha := \{(w, p) \mid w \in W, p = 0 \text{ if } w \notin [\alpha], \\ \text{else } p = \sum_{\substack{w' \in W \\ w \in \text{Min}(\alpha, w', d)}} b(w') / |\text{Min}(\alpha, w', d)|\}.$$

In words, $b \text{ GI } \alpha$ is the new belief state produced by taking the generalized image of b with respect to α . Notice how the probability mass of non- α -worlds is shifted to their closest α -worlds. If a non- α -world w^\times with probability p has n closest α -worlds (equally distant), then each of these closest α -worlds gets p/n mass from w^\times .

We define $b_\alpha^\circ := b \circ \alpha$ so that we can write $b_\alpha^\circ(w)$, where \circ is a revision operator.

Example 2 Continuing on Example 1: Let $b = (0, 0.1, 0, 0.2, 0, 0.3, 0, 0.4)$.

$(q \wedge r) \vee (q \wedge \neg r \wedge s)$ is abbreviated as α .

$$b_\alpha^{\text{GI}}(111) = \sum_{\substack{w' \in W \\ 111 \in \text{Min}(\alpha, w', d)}} b(w') / |\text{Min}(\alpha, w', d)| \\ = b(111) / |\text{Min}(\alpha, 111, d)| + b(011) / |\text{Min}(\alpha, 011, d)| \\ = 0/1 + 0/1 = 0.$$

$$b_\alpha^{\text{GI}}(110) = \sum_{\substack{w' \in W \\ 110 \in \text{Min}(\alpha, w', d)}} b(w') / |\text{Min}(\alpha, w', d)| \\ = b(110) / |\text{Min}(\alpha, 110, d)| + b(100) / |\text{Min}(\alpha, 100, d)| \\ + b(010) / |\text{Min}(\alpha, 010, d)| + b(000) / |\text{Min}(\alpha, 000, d)| \\ = 0.1/1 + 0.2/2 + 0.3/1 + 0.4/2 = 0.7.$$

$$b_\alpha^{\text{GI}}(101) = \sum_{\substack{w' \in W \\ 101 \in \text{Min}(\alpha, w', d)}} b(w') / |\text{Min}(\alpha, w', d)| \\ = b(101) / |\text{Min}(\alpha, 101, d)| + b(100) / |\text{Min}(\alpha, 100, d)| \\ + b(001) / |\text{Min}(\alpha, 001, d)| + b(000) / |\text{Min}(\alpha, 000, d)| \\ = 0/1 + 0.2/2 + 0/1 + 0.4/2 = 0.3.$$

And $b_\alpha^{\text{GI}}(100) = b_\alpha^{\text{GI}}(011) = b_\alpha^{\text{GI}}(010) = b_\alpha^{\text{GI}}(001) = b_\alpha^{\text{GI}}(000) = 0$. \square

4 REVISION VIA GI AND BOUNDARY BELIEF STATES

The most obvious way to revise a given belief base (BB) B is to revise every individual belief state in Π^B and then induce a new BB from the set of revised belief states. Formally, given observation α , first determine a new belief state b^α for every $b \in \Pi^B$ via the defined revision operation:

$$\Pi^{B^\alpha} = \{b^\alpha \in \Pi \mid b^\alpha = b \text{ GI } \alpha, b \in \Pi^B\}.$$

If there is more than only a single belief state in Π^B , then Π^B contains an infinite number of belief states. Then how can one compute Π^{B^α} ? And how would one subsequently determine B^α from Π^{B^α} ?

In the rest of this section we shall present a finite method of determining Π^{B^α} . What makes this method possible is the insight that Π^B can be represented by a finite set of ‘boundary’ belief states – those belief states which, in a sense, represent the limits or the convex hull of Π^B . We shall prove that the set of revised boundary belief states defines Π^{B^α} . Inducing B^α from Π^{B^α} is then relatively easy, as will be seen.

Let W^{perm} be every permutation on the ordering of worlds in W . For instance, if $W = \{w_1, w_2, w_3, w_4\}$, then $W^{\text{perm}} = \{\langle w_1, w_2, w_3, w_4 \rangle, \langle w_1, w_2, w_4, w_3 \rangle, \langle w_1, w_3, w_2, w_4 \rangle, \dots, \langle w_4, w_3, w_2, w_1 \rangle\}$. Given an ordering $W^\# \in W^{\text{perm}}$, let $W^\#(i)$ be the i -th element of $W^\#$; for instance, $\langle w_4, w_3, w_2, w_1 \rangle(2) = w_3$. Suppose we are given a BB B . We now define a function which, given a permutation of worlds, returns a belief state where worlds earlier in the ordering are assigned maximal probabilities according to the boundary values enforced by B .

Definition 3 *MaxASAP*($B, W^\#$) is the $b \in \Pi^B$ such that for $i = 1, \dots, |W|$, $\forall b' \in \Pi^B$, if $b' \neq b$, then $\sum_{j=1}^i b(W^\#(j)) \geq \sum_{k=1}^i b'(W^\#(k))$.

Example 3 Suppose the vocabulary is $\{q, r\}$ and $B_1 = \{(q) \geq 0.6\}$. Then, for instance, *MaxASAP*($B_1, (01, 00, 11, 10)$) = $\{(01, 0.4), (00, 0), (11, 0.6), (10, 0)\} = \{(11, 0.6), (10, 0), (01, 0.4), (00, 0)\}$. \square

Definition 4 We define the boundary belief states of BB B as the set $\Pi_{\text{bnd}}^B := \{b \in \Pi^B \mid W^\# \in W^{\text{perm}}, b = \text{MaxASAP}(B, W^\#)\}$.

Example 4 Suppose the vocabulary is $\{q, r\}$ and $B_1 = \{(q) \geq 0.6\}$. Then

$$\Pi_{\text{bnd}}^{B_1} = \{\{(11, 1.0), (10, 0.0), (01, 0.0), (00, 0.0)\}, \\ \{(11, 0.0), (10, 1.0), (01, 0.0), (00, 0.0)\}, \\ \{(11, 0.6), (10, 0.0), (01, 0.4), (00, 0.0)\}, \\ \{(11, 0.6), (10, 0.0), (01, 0.0), (00, 0.4)\}, \\ \{(11, 0.0), (10, 0.6), (01, 0.4), (00, 0.0)\}, \\ \{(11, 0.0), (10, 0.6), (01, 0.0), (00, 0.4)\}\}.$$

\square

Next, the revision operation is applied to every belief state in Π_{bnd}^B . Let $(\Pi_{\text{bnd}}^B)_\alpha^{\text{GI}} := \{b' \in \Pi \mid b' = b_\alpha^{\text{GI}}, b \in \Pi_{\text{bnd}}^B\}$.

Example 5 Suppose the vocabulary is $\{q, r\}$ and $B_1 = \{(q) \geq 0.6\}$. Let α be $(q \wedge \neg r) \vee (\neg q \wedge r)$. Then

$$(\Pi_{\text{bnd}}^{B_1})_\alpha^{\text{GI}} = \{\{(11, 0.0), (10, 0.5), (01, 0.5), (00, 0.0)\}, \\ \{(11, 0.0), (10, 1.0), (01, 0.0), (00, 0.0)\}, \\ \{(11, 0.0), (10, 0.3), (01, 0.7), (00, 0.0)\}, \\ \{(11, 0.0), (10, 0.6), (01, 0.4), (00, 0.0)\}, \\ \{(11, 0.0), (10, 0.8), (01, 0.2), (00, 0.0)\}\}.$$

(Two revision operations produce $\{(11, 0), (10, 0.5), (01, 0.5), (00, 0)\}$.) \square

To induce the new BB B_{bnd}^α from $(\Pi_{\text{bnd}}^B)_\alpha^{\text{GI}}$, the following procedure is executed. For every possible world, the procedure adds a sentence enforcing the upper (resp., lower) probability limit of the world, with respect to all the revised boundary belief states. Trivial limits are excepted.

For every $w \in W$, $(\phi_w) \leq \bar{y} \in B^\alpha$, where $\bar{y} = \max_{b \in (\Pi_{bnd}^\alpha)^{GI}} b(w)$, except when $\bar{y} = 1$, and $(\phi_w) \geq \underline{y} \in B^\alpha$, where $\underline{y} = \min_{b \in (\Pi_{bnd}^\alpha)^{GI}} b(w)$, except when $\underline{y} = 0$.

The intention is that the procedure specifies B^α to represent the upper and lower probability envelopes of the set of revised boundary belief states. And thus, by Theorem 1, B^α defines the entire revised belief state space.

Example 6 Continuing Example 5, using the translation procedure just above, we see that $B_{1bnd}^\alpha = \{(\phi_{11}) \leq 0, (\phi_{10}) \geq 0.3, (\phi_{01}) \leq 0.7, (\phi_{00}) \leq 0.0\}$.

Note that if we let $B' = \{((q \wedge \neg r) \vee (\neg q \wedge r)) = 1, (q \wedge \neg r) \geq 0.3\}$, then $\Pi^{B'} = \Pi^{B_{1bnd}^\alpha}$. \square

Example 7 Suppose the vocabulary is $\{q, r\}$ and $B_2 = \{(\neg q \wedge \neg r) = 0.1\}$. Let α be $\neg q$. Then

$$\begin{aligned} \Pi_{bnd}^{B_2} &= \{(11, 0.9), (10, 0), (01, 0), (00, 0.1)\}, \\ &\quad \{(11, 0), (10, 0.9), (01, 0), (00, 0.1)\}, \\ &\quad \{(11, 0), (10, 0), (01, 0.9), (00, 0.1)\}, \end{aligned}$$

$$\begin{aligned} (\Pi_{bnd}^{B_2})_\alpha^{GI} &= \{(11, 0), (10, 0), (01, 0.9), (00, 0.1)\}, \\ &\quad \{(11, 0), (10, 0), (01, 0), (00, 1)\} \text{ and} \end{aligned}$$

$$B_{2bnd}^\alpha = \{(\phi_{11}) \leq 0, (\phi_{10}) \leq 0, (\phi_{01}) \leq 0.9, (\phi_{00}) \geq 0.1\}.$$

Note that if we let $B' = \{(\neg q) = 1, (\neg q \wedge r) \leq 0.9\}$, then $\Pi^{B'} = \Pi^{B_{2bnd}^\alpha}$. \square

Note that every world in W can be associated with the size of $Min(\alpha, w, d)$ for some α and d . Denote this size as $\#(w)$. Let $W^{Min(\alpha, d)}$ be a partition of W such that every block (equivalence class) blk of the partition is defined as follows. $blk = \{w_1, \dots, w_k\}$ iff $\#(w_1) = \dots = \#(w_k)$. Let $[w]$ denote block blk iff $w \in blk$. Finally, let w^i indicate that $i = \#(w)$, in other words, $[w^i]$ is the block containing all worlds such that $i = |Min(\alpha, w, d)|$. Let $m := \max_{w \in W} \{|Min(\alpha, w, d)|\}$.

Observation 1 Let $\delta_1, \delta_2, \dots, \delta_m$ be positive integers such that $i < j$ iff $\delta_i < \delta_j$. Let $\nu_1, \nu_2, \dots, \nu_m$ be values in $[0, 1]$ such that $\sum_{k=1}^m \nu_k = 1$. Associate with every ν_i a maximum value it is allowed to take: $most(\nu_i)$. For every ν_i , we define the assignment value

$$av(\nu_i) := \begin{cases} most(\nu_i) & \text{if } \sum_{k=1}^i \nu_k \leq 1 \\ 1 - \sum_{k=1}^{i-1} \nu_k & \text{otherwise} \end{cases}$$

Determine first $av(\nu_1)$, then $av(\nu_2)$ and so on. Then

$$\frac{av(\nu_1)}{\delta_1} + \dots + \frac{av(\nu_m)}{\delta_m} > \frac{\nu'_1}{\delta_1} + \dots + \frac{\nu'_m}{\delta_m}$$

whenever $\nu'_i \neq av(\nu_i)$ for some i . \square

For instance, let $\delta_1 = 1, \delta_2 = 2, \delta_3 = 3, \delta_4 = 4$. Let $most(\nu_1) = 0.5, most(\nu_2) = 0.3, most(\nu_3) = 0.2, most(\nu_4) = 0.3$. Then $av(\nu_1) = 0.5, av(\nu_2) = 0.3, av(\nu_3) = 0.2, av(\nu_4) = 0$ and

$$\frac{0.5}{1} + \frac{0.3}{2} + \frac{0.2}{3} + \frac{0}{4} = 0.716.$$

But

$$\frac{0.49}{1} + \frac{0.3}{2} + \frac{0.2}{3} + \frac{0.01}{4} = 0.709.$$

And

$$\frac{0.5}{1} + \frac{0.29}{2} + \frac{0.2}{3} + \frac{0.01}{4} = 0.714.$$

Lemma 1 essentially says that the belief state in Π^B which causes a revised belief state to have a maximal value at world w (w.r.t. all belief states in Π^B), will be in Π_{bnd}^B .

Lemma 1 For all $w \in W$, $\arg \max_{b_X \in \Pi^B} \sum_{w' \in W} b_X(w') / |Min(\alpha, w', d)|$ is in Π_{bnd}^B .

Proof:

Note that $\sum_{w' \in W} b(w') / |Min(\alpha, w', d)|$ can be written in the form

$$\frac{\sum_{w' \in [w^1]} b(w')}{1} + \dots + \frac{\sum_{w' \in [w^m]} b(w')}{m}.$$

Observe that there must be a $W^\# \in W^{perm}$ such that $W^\# = \langle w_1^1, \dots, w_{n_1}^1, \dots, w_1^m, \dots, w_{n_m}^m \rangle$. Then by the definition of the set of boundary belief states (Def. 4), $MaxASAP(B, W^\#)$ will assign maximal probability mass to $[w^1] = \{w_1^1, \dots, w_{n_1}^1\}$, then to $[w^2] = \{w_1^2, \dots, w_{n_2}^2\}$ and so on.

That is, by Observation 1, for some $b_X \in \Pi_{bnd}^B$, $b_X(w) = \max_{b_X \in \Pi^B} \sum_{w' \in W} b_X(w') / |Min(\alpha, w', d)|$ for all $w \in W$. Therefore, $\arg \max_{b_X \in \Pi^B} \sum_{w' \in W} b_X(w') / |Min(\alpha, w', d)|$ is in Π_{bnd}^B . \square

Let

$$\bar{x}^w := \max_{b \in \Pi_{bnd}^B} b(w) \quad \bar{X}^w := \max_{b \in \Pi^B} b(w)$$

$$\bar{y}^w := \max_{b \in (\Pi_{bnd}^B)_\alpha^{GI}} b(w) \quad \bar{Y}^w := \max_{b \in (\Pi^B)_\alpha^{GI}} b(w)$$

$$\underline{x}^w := \min_{b \in \Pi_{bnd}^B} b(w) \quad \underline{X}^w := \min_{b \in \Pi^B} b(w)$$

$$\underline{y}^w := \min_{b \in (\Pi_{bnd}^B)_\alpha^{GI}} b(w) \quad \underline{Y}^w := \min_{b \in (\Pi^B)_\alpha^{GI}} b(w)$$

Lemma 2 states that for every world, the upper/lower probability of the world with respect to Π_{bnd}^B is equal to the upper/lower probability of the world with respect to Π^B . The proof requires Observation 1 and Lemma 1.

Lemma 2 For all $w \in W$, $\bar{y}^w = \bar{Y}^w$ and $\underline{y}^w = \underline{Y}^w$.

Proof:

Note that if $w \notin [\alpha]$, then $\bar{y}^w = \bar{Y}^w = 0$ and $\underline{y}^w = \underline{Y}^w = 0$.

We now consider the cases where $w \in [\alpha]$.

$$\bar{y}^w = \bar{Y}^w$$

iff

$$\max_{b \in (\Pi_{bnd}^B)} b(w) = \max_{b \in (\Pi^B)} b(w)$$

iff

$$\begin{aligned} &\max_{b_X \in \Pi_{bnd}^B} \sum_{w' \in W} b_X(w') / |Min(\alpha, w', d)| \\ &= \max_{b_X \in \Pi^B} \sum_{w' \in W} b_X(w') / |Min(\alpha, w', d)| \end{aligned}$$

if $\bar{b}_x(w) = \bar{b}_X(w)$, where

$$\bar{b}_x(w) := \max_{b_x \in \Pi_{bnd}^B} \sum_{\substack{w' \in W \\ w \in \text{Min}(\alpha, w', d)}} b_x(w') / |\text{Min}(\alpha, w', d)|$$

and

$$\bar{b}_X(w) := \max_{b_X \in \Pi^B} \sum_{\substack{w' \in W \\ w \in \text{Min}(\alpha, w', d)}} b_X(w') / |\text{Min}(\alpha, w', d)|.$$

Note that

$$\sum_{\substack{w' \in W \\ w \in \text{Min}(\alpha, w', d)}} b(w') / |\text{Min}(\alpha, w', d)|$$

can be written in the form

$$\frac{\sum_{\substack{w' \in [w^1] \\ w \in \text{Min}(\alpha, w', d)}} b(w')}{1} + \dots + \frac{\sum_{\substack{w' \in [w^m] \\ w \in \text{Min}(\alpha, w', d)}} b(w')}{m}.$$

Then by Observation 1, $\bar{b}_X(w)$ is in Π_{bnd}^B . And also by Lemma 1, the belief state in Π_{bnd}^B identified by $\bar{b}_X(w)$ must be the one which maximizes

$$\sum_{\substack{w' \in W \\ w \in \text{Min}(\alpha, w', d)}} b_x(w') / |\text{Min}(\alpha, w', d)|,$$

where $b_x \in \Pi_{bnd}^B$. That is, $\bar{b}_x = \bar{b}_X$.

With a symmetrical argument, it can be shown that $\underline{y}^w = \underline{Y}^w$. \square

Informally, the following theorem says that the BB determined through the method of revising boundary belief states captures exactly the same beliefs and ignorance as the belief states in Π^B which have been revised. This correspondence relies on the fact that the upper and lower probability envelopes of Π^B can be induced from Π_{bnd}^B , which is what Lemma 2 states.

Theorem 1 Let $(\Pi_{\alpha}^B)^{\text{Gl}} := \{b_{\alpha}^{\text{Gl}} \in \Pi \mid b \in \Pi^B\}$. Let B_{bnd}^{α} be the BB induced from $(\Pi_{bnd}^B)^{\text{Gl}}_{\alpha}$. Then $\Pi_{bnd}^B = (\Pi_{\alpha}^B)^{\text{Gl}}$.

Proof:

We show that $\forall b' \in \Pi, b' \in \Pi_{bnd}^B \iff b' \in (\Pi_{\alpha}^B)^{\text{Gl}}$.

(\Rightarrow) $b' \in \Pi_{bnd}^B$ implies $\forall w \in W, \underline{y}^w \leq b'(w) \leq \bar{y}^w$ (by definition of B_{bnd}^{α}). Lemma 2 states that for all $w \in W, \bar{y}^w = \bar{Y}^w$ and $\underline{y}^w = \underline{Y}^w$. Hence, $\forall w \in W, \underline{Y}^w \leq b'(w) \leq \bar{Y}^w$. Therefore, $b'(w) \in (\Pi_{\alpha}^B)^{\text{Gl}}$.

(\Leftarrow) $b'(w) \in (\Pi_{\alpha}^B)^{\text{Gl}}$ implies $\forall w \in W, \underline{Y}^w \leq b'(w) \leq \bar{Y}^w$. Hence, by Lemma 2, $\forall w \in W, \underline{y}^w \leq b'(w) \leq \bar{y}^w$. Therefore, by definition of B_{bnd}^{α} , $b' \in \Pi_{bnd}^B$. \square

5 REVISION VIA OPTIMUM ENTROPY INFERENCE

Another approach to the revision of a belief base (BB) is to determine a representative of Π^B (call it b_{rep}), change the representative belief state via the defined revision operation and then induce a new BB from the revised representative belief state. Selecting a representative probability distribution from a family of such functions is not new [8, 22, e.g.]. More formally, given observation α , first determine $b_{rep} \in \Pi^B$, then compute its revision b_{rep}^{α} , and finally induce B^{α} from b_{rep}^{α} .

We shall represent Π^B (and thus B) by the single ‘least biased’ belief state, that is, the belief state in Π^B with *highest entropy*:

Definition 5 (Shannon Entropy)

$$H(b) := - \sum_{w \in W} b(w) \ln b(w),$$

where b is a belief state.

Definition 6 (Maximum Entropy) Traditionally, given some set of distributions Π , the most entropic distribution in Π is defined as

$$b^H := \arg \max_{b \in \Pi} H(b).$$

Suppose $B_2 = \{(-q \wedge \neg r) = 0.1\}$. Then the belief state $b \in \Pi^{B_2}$ satisfying the constraints posed by B_2 for which $H(b)$ is maximized is $b_{rep} = b^H = \langle 0.3, 0.3, 0.3, 0.1 \rangle$.

The above distribution can be found directly by applying the principle of maximum entropy: The true belief state is estimated to be the one consistent with known constraints, but is otherwise as unbiased as possible, or ‘‘Given no other knowledge, assume that everything is as random as possible. That is, the probabilities are distributed as uniformly as possible consistent with the available information,’’ [24]. Obviously world 00 must be assigned probability 0.1. And the remaining 0.9 probability mass should be uniformly spread across the other three worlds.

Applying Gl to b_{rep} on evidence $\neg q$ results in $b_{rep}^{\neg q} = \langle 0, 0, 0.6, 0.4 \rangle$.

Example 8 Suppose the vocabulary is $\{q, r\}$, $B_1 = \{(q) \geq 0.6\}$ and α is $(q \wedge \neg r) \vee (\neg q \wedge r)$. Then $b_{rep} = \arg \max_{b \in \Pi^{B_1}} H(b) = \langle 0.3, 0.3, 0.2, 0.2 \rangle$. Applying Gl to b_{rep} on α results in $b_{rep}^{\alpha} = \langle 0, 0.61, 0.39, 0 \rangle$. b_{rep}^{α} can be translated into B_{1rep}^{α} as $\{(q \wedge \neg r) = 0.61, (\neg q \wedge r) = 0.39\}$. \square

Still using $\alpha = (q \wedge \neg r) \vee (\neg q \wedge r)$, notice that $\Pi^{B_{1rep}^{\alpha}} \neq \Pi^{B_{1bnd}^{\alpha}}$. But how different are $B_{1rep}^{\alpha} = \{(q \wedge \neg r) = 0.61, (\neg q \wedge r) = 0.39\}$ and $B_{1bnd}^{\alpha} = \{(q \wedge r) \leq 0, (q \wedge \neg r) \geq 0.3, (\neg q \wedge r) \leq 0.7, (\neg q \wedge \neg r) \leq 0.0\}$? Perhaps one should ask, how different B_{1rep}^{α} is from the representative of B_{1bnd}^{α} : The least biased belief state satisfying B_{1bnd}^{α} is $\langle 0, 0.5, 0.5, 0 \rangle$. That is, How different are $\langle 0, 0.61, 0.39, 0 \rangle$ and $\langle 0, 0.5, 0.5, 0 \rangle$?

In the case of B_2 , we could compare $B_{2bnd}^{\neg q} = \{(\phi_{11}) \leq 0, (\phi_{10}) \leq 0, (\phi_{01}) \leq 0.9, (\phi_{00}) \geq 0.1\}$ with $b_{rep}^{\neg q} = \langle 0, 0, 0.6, 0.4 \rangle$. Or if we take the least biased belief state satisfying $B_{2bnd}^{\neg q}$, we can compare $\langle 0, 0, 0.5, 0.5 \rangle$ with $\langle 0, 0, 0.6, 0.4 \rangle$.

It has been extensively argued [11, 26, 23] that maximum entropy is a reasonable inference mechanism, if not the most reasonable one (w.r.t. probability constraints). On the other hand, the boundary belief states method also seems like a very reasonable inference mechanism for revising BBs as defined here, in the sense that it causes information loss. Resolving this misalignment in the results of the two methods is an obvious task for future research.

An extended version of maximum entropy is *minimum cross-entropy* (MCE) [17, 5]:

Definition 7 (Minimum Cross-Entropy) The ‘directed divergence’ of distribution c from distribution b is defined as

$$R(c, b) := \sum_{w \in W} c(w) \ln \frac{c(w)}{b(w)}.$$

$R(c, b)$ is undefined when $b(w) = 0$ while $c(w) > 0$; when $c(w) = 0$, $R(c, b) = 0$, because $\lim_{x \rightarrow 0} \ln(x) = 0$. Given new evidence

$\phi \in L^{prob}$, the distribution c satisfying ϕ diverging least from current belief state b is

$$\arg \min_{c \in \Pi, c \Vdash \phi} R(c, b).$$

Definition 8 (MCI) Then MCE inference (denoted (MCI)) is defined as

$$b \text{ MCI } \alpha := \arg \min_{b' \in \Pi, b' \Vdash (\alpha)=1} R(b', b).$$

In the following example, we interpret revision as MCE inference.

Example 9 Suppose the vocabulary is $\{q, r\}$ and $B_1 = \{(q) \geq 0.6\}$. Let α be $(q \wedge \neg r) \vee (\neg q \wedge r)$. Then

$$\begin{aligned} \Pi_{bnd}^{B_1} = & \{ \{(11, 1.0), (10, 0.0), (01, 0.0), (00, 0.0)\}, \\ & \{(11, 0.0), (10, 1.0), (01, 0.0), (00, 0.0)\}, \\ & \{(11, 0.6), (10, 0.0), (01, 0.4), (00, 0.0)\}, \\ & \{(11, 0.6), (10, 0.0), (01, 0.0), (00, 0.4)\}, \\ & \{(11, 0.0), (10, 0.6), (01, 0.4), (00, 0.0)\}, \\ & \{(11, 0.0), (10, 0.6), (01, 0.0), (00, 0.4)\} \}, \end{aligned}$$

$$\begin{aligned} (\Pi_{bnd}^{B_1})_{\alpha}^{\text{MCI}} = & \{ \{(11, 0), (10, 0), (01, 1), (00, 0)\}, \\ & \{(11, 0), (10, 1), (01, 0), (00, 0)\}, \\ & \{(11, 0), (10, 0.6), (01, 0.4), (00, 0)\} \} \text{ and} \end{aligned}$$

$$B_{1bnd}^{\alpha} = \{(\phi_{11}) \leq 0, (\phi_{00}) \leq 0\}.$$

Note that if we let $B' = \{(q \wedge \neg r) \vee (\neg q \wedge r) = 1\}$, then $\Pi^{B'} = \Pi^{B_{1bnd}^{\alpha}}$. \square

Recall from Example 6 that B' included $(q \wedge \neg r) \geq 0.3$. Hence, in this particular case, combining the boundary belief states approach with MCI results in a less informative revised belief base than when GI is used. The reason for the loss of information might be due to $R(\cdot, \{(11, 1.0), (10, 0.0), (01, 0.0), (00, 0.0)\})$ and $R(\cdot, \{(11, 0.6), (10, 0.0), (01, 0.0), (00, 0.4)\})$ being undefined: Recall that $R(c, b)$ is undefined when $b(w) = 0$ while $c(w) > 0$. But then there is no belief state c for which $c \Vdash \alpha$ and $R(\cdot)$ is defined (with these two belief states as arguments). Hence, there are no revised counterparts of these two belief states in $(\Pi_{bnd}^{B_1})_{\alpha}^{\text{MCI}}$. We would like to analyse MCI more within this framework. In particular, in the future, we would like to determine whether a statement like Theorem 1 holds for MCI too.

In MCE inference, b -consistency of evidence ϕ is defined as: There exists a belief state c such that $c \Vdash \phi$ and c is *totally continuous* with respect to b (i.e., $b(w) = 0$ implies $c(w) = 0$). MCE is undefined when the evidence is not b -consistent. This is analogous to Bayesian conditioning being undefined for $b(\alpha) = 0$. Obviously, this is a limitation of MCE because some belief states may not be considered as candidate revised belief states.

6 RATIONALITY POSTULATES

In this section, we assess the operation of revising a belief base B by α via GI (denoted B^{α}) with respect to several rationality postulates. Katsuno and Mendelzon [12] modified the eight AGM belief revision postulates [1] to the following six ((R1)-(R6)). The postulates are intended to be an ideal standard of rationality with respect to the behavior of any revision operator. We shall simply translate (\implies) each postulate into one appropriate for belief bases (in the notation of this paper) in order to get a sense of which ones are satisfied

and which not. We then intend to use this information to guide us in drawing up appropriate generalised postulates in the future. ⁶ In their notation [12], given a knowledge base represented by a propositional sentence ψ and an observation represented by a propositional sentence μ , $\psi \circ \mu$ denotes the *revision* of ψ by μ .⁷

$$\begin{aligned} \text{(R1)} \quad & \psi \circ \mu \text{ implies } \mu \implies \\ \text{(RB1)} \quad & B^{\alpha} \models (\alpha) = 1. \end{aligned}$$

Proposition 1 Postulate (RB1) holds.

Proof:

By definition of GI, all probability mass is shifted to closest α -worlds. For every revised boundary belief state b_{bnd} , it is thus the case that $b_{bnd}(\alpha) = 1$. By the process of inducing B^{α} , it must be that B^{α} entails $(\alpha) = 1$. \square

$$\text{(R2)} \quad \text{If } \psi \wedge \mu \text{ is satisfiable, then } \psi \circ \mu \equiv \psi \wedge \mu \implies$$

$$\text{(RB2)} \quad \text{If } B \cup \{(\alpha) = 1\} \text{ is satisfiable, then } B^{\alpha} \equiv B \cup \{(\alpha) = 1\}.$$

Proposition 2 Postulate (RB2) does not hold.

Proof:

Recall that $B_1 = \{(q) \geq 0.6\}$. Example 6 shows that $B_1^{(q \wedge \neg r) \vee (\neg q \wedge r)} \equiv B' = \{((q \wedge \neg r) \vee (\neg q \wedge r)) = 1, (q \wedge \neg r) \geq 0.3\}$. Therefore, $B' \models (q \wedge \neg r) \geq 0.3$. But $B_1 \cup \{((q \wedge \neg r) \vee (\neg q \wedge r)) = 1\}$, although satisfiable, does not entail $(q \wedge \neg r) \geq 0.3$. Hence, $B' \not\equiv B_1 \cup \{((q \wedge \neg r) \vee (\neg q \wedge r)) = 1\}$. \square

Let $B_4 = \{(q \wedge r) = 0.2, (q \wedge \neg r) = 0.8\}$. Then $B_4^{\alpha} \equiv B_4 \equiv B_4 \cup \{(q) = 1\}$. Notice that B_4 specifies a particular belief state. One can see that Postulate (RB2) will hold whenever the BB to be revised specifies a particular belief state (i.e., whenever $|\Pi^B| = 1$).

$$\text{(R3)} \quad \text{If } \mu \text{ is satisfiable, then } \psi \circ \mu \text{ is also satisfiable} \implies$$

$$\text{(RB3)} \quad \text{If } (\alpha) = 1 \text{ is satisfiable, then } B^{\alpha} \text{ is also satisfiable.}$$

Proposition 3 Postulate (RB3) holds.

Proof:

$(\alpha) = 1$ is satisfiable iff α is. And if α is satisfiable, then every boundary belief state of B revised by α is defined. Hence, B^{α} must be defined, i.e., satisfiable. \square

$$\text{(R4)} \quad \text{If } \psi \equiv \psi' \text{ and } \mu \equiv \mu', \text{ then } \psi \circ \mu \equiv \psi' \circ \mu' \implies$$

$$\text{(RB4)} \quad \text{If } B \equiv C \text{ and } \alpha \equiv \beta, \text{ then } B^{\alpha} \equiv C^{\beta}.$$

Proposition 4 Postulate (RB4) holds.

Proof:

By definition, if $B \equiv C$, then $\Pi^B = \Pi^C$. Hence, the boundary belief states for B and C are the same. And given $\alpha \equiv \beta$, it must be the case that $B^{\alpha} \equiv C^{\beta}$. \square

$$\text{(R5)} \quad (\psi \circ \mu) \wedge \phi \text{ implies } \psi \circ (\mu \wedge \phi) \implies$$

$$\text{(RB5)} \quad B^{\alpha} \cup \{(\beta) = 1\} \models B^{\alpha \wedge \beta}.$$

Proposition 5 Postulate (RB5) does not hold.

Proof:

Let α be \top . Then $B^{\alpha} \equiv B$ and $B^{\alpha \wedge \beta} \equiv B^{\beta}$. Now let β be $(q \wedge \neg r) \vee (\neg q \wedge r)$. We can thus ask whether $B \cup \{(\beta) = 1\} \models B^{\beta}$.

Consider Example 6. Recall that $B_1^{(q \wedge \neg r) \vee (\neg q \wedge r)} \models (q \wedge \neg r) \geq 0.3$. That is, $\forall b \in \Pi_{bnd}^{B_1^{(q \wedge \neg r) \vee (\neg q \wedge r)}}$, $b \models (q \wedge \neg r) \geq 0.3$. And recall

⁶ In these postulates, it is sometimes necessary to write an observation α as a BB, i.e., as $\{(\alpha) = 1\}$ – in the present framework, observations are regarded as certain.

⁷ \circ is some revision operator.

that $B_1 \cup \{((q \wedge \neg r) \vee (\neg q \wedge r)) = 1\} \not\models (q \wedge \neg r) \geq 0.3$. That is, $\exists b \in \Pi^{B_1 \cup \{((q \wedge \neg r) \vee (\neg q \wedge r)) = 1\}}$ s.t. $b \not\models (q \wedge \neg r) \geq 0.3$. Hence, $\Pi^{B_1 \cup \{((q \wedge \neg r) \vee (\neg q \wedge r)) = 1\}} \not\subseteq \Pi^{B_1^{(q \wedge \neg r) \vee (\neg q \wedge r)}}$ and by the definition of \models , $B_1 \cup \{((q \wedge \neg r) \vee (\neg q \wedge r)) = 1\} \not\models B_1^{(q \wedge \neg r) \vee (\neg q \wedge r)}$. \square

If $B^\alpha \cup \{(\beta) = 1\}$ is not satisfiable, then (RB5) holds trivially. If $B^\alpha \cup \{(\beta) = 1\}$ is satisfiable, then α logically entails β , implying that $\alpha \wedge \beta \equiv \alpha$. Then we can ask whether $B^\alpha \cup \{(\beta) = 1\} \models B^\alpha$? Assuming B specifies a particular belief state, clearly $\Pi^{B^\alpha \cup \{(\beta) = 1\}} \subseteq \Pi^{B^\alpha}$ and (RB5) holds. So the difficulties come in when $|\Pi^B| > 1$.

(R6) If $(\psi \circ \mu) \wedge \phi$ is satisfiable, then $\psi \circ (\mu \wedge \phi)$ implies $(\psi \circ \mu) \wedge \phi$
 \implies

(RB6) If $B^\alpha \cup \{(\beta) = 1\}$ is satisfiable, then $B^{\alpha \wedge \beta} \models B^\alpha \cup \{(\beta) = 1\}$.

Proposition 6 *Postulate (RB6) does not hold.*

Proof:

Let α be \top . Then (RB6) becomes: If $B \cup \{(\beta) = 1\}$ is satisfiable, then $B^\beta \models B \cup \{(\beta) = 1\}$.

Consider Example 6. Let β be $(q \wedge \neg r) \vee (\neg q \wedge r)$. Note that $B_1 \cup \{((q \wedge \neg r) \vee (\neg q \wedge r)) = 1\}$ is satisfiable. Then by (RB6), $B_1^{(q \wedge \neg r) \vee (\neg q \wedge r)} \models B_1 \cup \{((q \wedge \neg r) \vee (\neg q \wedge r)) = 1\}$. But this is false:

Let $b = \langle 0, 0.5, 0.5, 0 \rangle$. Then

$$b \in \Pi^{B_1^{(q \wedge \neg r) \vee (\neg q \wedge r)}} = \Pi^{\{((q \wedge \neg r) \vee (\neg q \wedge r)) = 1, (q \wedge \neg r) \geq 0.3\}}, \text{ but}$$

$$b \notin \Pi^{B_1 \cup \{((q \wedge \neg r) \vee (\neg q \wedge r)) = 1\}} = \Pi^{\{(q \geq 0.6, ((q \wedge \neg r) \vee (\neg q \wedge r)) = 1\}}.$$

Hence, $\Pi^{B_1^{(q \wedge \neg r) \vee (\neg q \wedge r)}} \not\subseteq \Pi^{B_1 \cup \{((q \wedge \neg r) \vee (\neg q \wedge r)) = 1\}}$. Therefore, $B_1^{(q \wedge \neg r) \vee (\neg q \wedge r)} \not\models B_1 \cup \{((q \wedge \neg r) \vee (\neg q \wedge r)) = 1\}$. \square

From the discussion of Postulate (RB5), we notice that given the antecedent of (RB6), $B^\alpha \cup \{(\beta) = 1\} \equiv B^\alpha$. Thus, the consequent of (RB6) becomes $B^\alpha \models B^\alpha$. This means that (RB6) holds when $|\Pi^B| = 1$. Again, the difficulties come in when $|\Pi^B| > 1$.

7 RELATED WORK

Voorbraak [27] proposed the partial probability theory (PTT), which allows probability assignments to be partially determined, and where there is a distinction between probabilistic information based on (i) hard background evidence and (ii) some assumptions. He does not explicitly define the ‘‘constraint language’’, however, from his examples and discussions, one can infer that he has something like the language L^{PTT} in mind: it contains all formulae which can be formed with sentences in our L^{prob} in combination with connectives \neg, \wedge and \vee . A ‘‘belief state’’ in PTT is defined as the quadruple $\langle \Omega, \mathcal{B}, \mathcal{A}, \mathcal{C} \rangle$, where Ω is a sample space, $\mathcal{B} \subset L^{PTT}$ is a sets of probability constraints, $\mathcal{A} \subset L^{PTT}$ is a sets of assumptions and $\mathcal{C} \subseteq W$ ‘‘represents specific information concerning the case at hand’’ (an observation or evidence).⁸ Our epistemic state can be expressed as a restricted PTT ‘‘belief state’’ by letting $\Omega = W$, $\mathcal{B} = B$, $\mathcal{A} = \emptyset$ and $\mathcal{C} = \{w \in W \mid w \models \alpha\}$, where B is a belief base and α is an observation in our notation.

Voorbraak [27] mentions that he will only consider conditioning where the evidence does not contradict the current beliefs. He defines the set of belief states corresponding to the conditionalized PPT

⁸ Voorbraak’s ‘‘belief state’’ would rather be called an *epistemic state* or *knowledge structure* in our language.

‘‘belief state’’ as $\{b(\cdot \mid C) \in \Pi \mid b \in \Pi^{B \cup \mathcal{A}}, b(C) > 0\}$. In our notation, this corresponds to $\{(b \text{BC} \alpha) \in \Pi \mid b \in \Pi^B, b(\alpha) > 0\}$, where α corresponds to C . Voorbraak [27] proposes *constraining* as an alternative to conditioning: Let $\phi \in L^{prob}$ be a probability constraint. In our notation, constraining Π^B on ϕ produces $\Pi^{B \cup \{\phi\}}$.

Note that expanding a belief set reduces the number of models (worlds) and expanding a PPT ‘‘belief state’’ with extra constraints also reduces the number of models (belief states / probability distributions).

In the context of belief sets, it is possible to obtain any belief state from the ignorant belief state by a series of expansions.

In PPT, constraining, but not conditioning, has the analogous property. This is one of the main reasons we prefer to constraining and not conditioning to be the probabilistic version of expansion. [27, p. 4]

But Voorbraak does not address the issue that C and ϕ are different kinds of observations, so constraining, as defined here, cannot be an alternative to conditioning. C cannot be used directly for constraining and ϕ cannot be used directly for conditioning.

W.l.o.g., we can assume C is represented by α . If we take $b \text{GI} \alpha$ to be an expansion operation whenever $b(\alpha) > 0$, then one might ask, Is it possible to obtain any belief base B' from the ignorant belief base $B = \emptyset$ by a series of expansions, using our approach? The answer is, No. For instance, there is no observation or series of observations which can change $B = \{\}$ into $B' = \{(q) \geq 0.6\}$. But if we were to allow sentences (constraints) in L^{prob} to be observations, then we could obtain any B' from the ignorant B .

Grove and Halpern [9] investigate what ‘‘update’’ (incorporation of an observation with current beliefs, such that the observation does not contradict the beliefs) means in a framework where beliefs are represented by a set of belief states. They state that the main purpose of their paper is to illustrate how different the set-of-distributions framework can be, ‘‘technically’’, from the standard single-distribution framework. They propose six postulates characterizing what properties an update function should have. They say that some of the postulates are obvious, some arguable and one probably too strong. Out of seven (families of) update functions only the one based on conditioning ($Upd_{cond}(\cdot)$) and the one based on constraining ($Upd_{constrain}(\cdot)$) satisfy all six postulates, where $Upd_{cond}(\Pi^B, \alpha) := \{(b \text{BC} \alpha) \in \Pi \mid b \in \Pi^B, b(\alpha) > 0\}$ and where they interpret Voorbraak’s constraining [27] as $Upd_{constrain}(\Pi^B, \alpha) := \{b \in \Pi^B \mid b(\alpha) = 1\}$. Grove and Halpern [9] do not investigate the case when an observation must be incorporated while it is (possibly) inconsistent with the old beliefs (i.e., revision). It would be interesting to analyse the present work against their six postulates.

Kern-Isberner [14] develops a new perspective of probabilistic belief change. Based on the ideas of Alchourr3n et al. [1] and Katsuno and Mendelzon [12] (KM), the operations of revision and update, respectively, are investigated within a probabilistic framework. She employs as basic knowledge structure as belief base (b, \mathcal{R}) , where b is a probability distribution (belief state) of background knowledge and \mathcal{R} is a set of probabilistic conditionals of the form $A \rightsquigarrow B[x]$ meaning ‘‘The probability of B , given A , is x ’’. A universal inference operation – based on the techniques of optimum entropy – is introduced as an ‘‘adequate and powerful method to realize probabilistic belief change’’.

By having a belief state available in the belief base, minimum cross-entropy can be used. The intention is then that an agent with belief base (b, T) should always reason w.r.t. belief state $b^T := \arg \min_{c \in \Pi, c \models T} R(c, b)$. Kern-Isberner [14] then defines the prob-

abilistic belief revision of (b, \mathcal{R}) by evidence \mathcal{S} as $(b, \mathcal{R} \cup \mathcal{S})$. And the probabilistic belief update of (b, \mathcal{R}) by evidence \mathcal{S} is defined as $(b^{\mathcal{R}}, \mathcal{S})$.⁹ She distinguishes between revision as a knowledge adding process, and updating as a change-recording process. Kern-Isberner [14] sets up comparisons of maximum cross-entropy belief change with AGM revision and KM update. Cases where, for update, new information \mathcal{R} is inconsistent with the prior distribution b , or, for revision, is inconsistent with b or the context \mathcal{R} , are not dealt with [14, p. 399, 400].

Having a belief state available for modification when new evidence is to be adopted is quite convenient. As Voorbraak [27] argues, however, an agent's ignorance can hardly be represented in an epistemic state where a single belief state must always be chosen.

We would like to investigate the representation of conditional probabilistic information such as is done in the work of Kern-Isberner [14, 15] and Yue and Liu [29], for instance.

Yue and Liu [29] propose a probabilistic revision operation for imprecise probabilistic beliefs in the framework of Probabilistic Logic Programming (PLP). New evidence may be a probabilistic (conditional) formula and needs not be consistent with the original beliefs. Revision via imaging (e.g., GI) also overcomes this consistency issue. Essentially, their *probabilistic epistemic states* Ψ are induced from a PLP program which is a set of formulae, each formula having the form $(\psi \mid \phi)[l, u]$, meaning that the probability of the conditional $(\psi \mid \phi)$ lies in the interval $[l, u]$. The operator they propose has the characteristic that if an epistemic state Ψ represents a single probability distribution, revising collapses to Jeffrey's rule and Bayesian conditioning. They mention that it is required that the models (distributions) of Ψ is a convex set. There might thus be an opportunity to employ their revision operation on a representative set of boundary distributions as proposed in this paper.

Another PLP, proposed by Michels et al. [21], also allow for incomplete specification of probabilities. The language of their logic is however more expressive than ours, and they focus on inference (probabilistic query answering). They mention that they want to allow for learning in their system – one could possibly interpret some kinds of learning in this setting as belief revision.

8 CONCLUSION AND FUTURE DIRECTIONS

In this paper, we propose an approach how to generate a new probabilistic belief base from an old one, given a new piece of non-probabilistic information, where a belief base is a finite set of sentences, each sentence stating the likelihood of a proposition about the world. In this framework, an agent's belief base represents the set of belief states compatible with the sentences in it. In this sense, the agent is able to represent its knowledge *and* ignorance about the true state of the world.

We used a version of the so-called *imaging* approach to implement the revision operation.

Three methods were proposed: revising a finite set of 'boundary belief states' via generalized imaging, revising a finite set of 'boundary belief states' via minimum cross-entropy and revising a least biased belief state. We focussed on the first method and showed that the latter two give different results.

There were two main contribution of this paper. The first was to prove that the set of belief states satisfying B_{new} is exactly those belief states satisfying the original belief base, revised. The second was to uncover an interesting conflict in the results of the three belief

base revision methods. It is worth further understanding the reasons behind such a difference, as such an investigation could give more insight about the mechanisms behind the methods and indicate possible pros and cons of each. Importantly, further analysis with respect to rationality postulated is necessary, as mentioned in § 8. Such an analysis may also bring insights into the differing results.

The computational complexity of $Min(\cdot)$ is in $O(|W|^2)$ and the complexity of GI is thus in $O(|W|^2|W|^2) = O(|W|^4)$ in the worst case. However, this complexity is highly dependent on the observation and the distance function. Note that $|\Pi_{bnd}^B| \leq |W^{perm}| = |W|!$. GI is applied to every belief state in Π_{bnd}^B . Hence, the complexity of the method, in the worst case, is in $O(|W|!|W|^4)$.

As far as we know, there is no analytic solution to determine the distribution in Π^B with maximum entropy / minimum cross-entropy. To narrow in on the computational complexity of these methods, we would have to know what class of optimization problem they are (convex?), and thus what techniques are used to solve them. Our knowledge in this area is lacking and it would require some time for investigation.

The proposal or design, and justification of rationality postulates similar to (RB1)-(RB6) in the section above is called for. An analysis of the postulates with respect to the revision operation must then be carried out. An attempt may be made to design the revision operation so as to make more of the postulates hold, or we may attempt to justify why our revision process does not / need not satisfy postulates it fails at.

Given that we have found that the belief base resulting from revising via the boundary-belief-states approach differs from the belief base resulting from revising via the representative-belief-state approach, the question arises, When is it appropriate to use a representative belief state defined as the most entropic belief state of a given set Π^B ? This is an important question, especially due to the popularity of employing the Maximum Entropy principle in cases of underspecified probabilistic knowledge [11, 8, 10, 27, 14, 16] and the principle's well-behavedness [26, 22, 13].

As far as we know, imaging for belief change has never been applied to (conditional) probabilistic evidence. Due to issues with many revision methods required to be consistent with prior beliefs, and imaging not having this limitation, it might be worthwhile investigating.

The translation from the set of belief states back to a belief base is a mapping from every belief state to a probability formula. The size of the belief base is thus in the order of $|W^{perm}|$, where $|W|$ is already exponential in the size of \mathcal{P} , the set of atoms. As we saw in several examples in this paper, the new belief base often has a more concise equivalent counterpart. It would be useful to find a way to consistently determine more concise belief bases than our present approach does.

Does a similar result as Theorem 1 holds for Bayesian conditioning? This is an important question we would like to answer and which credal set theory [19, 28] may answer.

ACKNOWLEDGEMENTS

The work of Giovanni Casini has been supported by the Fonds National de la Recherche, Luxembourg, and cofunded by the Marie Curie Actions of the European Commission (FP7-COFUND) (AFR/9181001). Thomas Meyer has received funding from the European Unions Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 690974. We thank the reviewers for their valuable comments and suggestions.

⁹ This is a very simplified version of what she presents. Please refer to the paper for details.

REFERENCES

- [1] C. Alchourrón, P. Gärdenfors, and D. Makinson, 'On the logic of theory change: Partial meet contraction and revision functions', *Journal of Symbolic Logic*, **50**(2), 510–530, (1985).
- [2] C. Boutilier, 'A unified model of qualitative belief change: A dynamical systems perspective', *Artif. Intell.*, **98**(1–2), 281–316, (1998).
- [3] K. Chhogyal, A. Nayak, R. Schwitter, and A. Sattar, 'Probabilistic belief revision via imaging', in *PRICAI 2014: Trends in Artif. Intell.: Thirteenth Pacific Rim Intl. Conf. on Artif. Intell.*, eds., D.-N. Pham and S.-B. Park, volume 8862 of *LNAI*, 694–707, Springer Intl. Publishing, Switzerland, (2014).
- [4] T. Cover and J. Thomas, *Elements of Information Theory*, Wiley, New York, 1991.
- [5] I. Csiszár, 'I-divergence geometry of probability distributions and minimization problems', *Annals of Probability*, **3**, 146–158, (1975).
- [6] D. Dubois and H. Prade, 'Belief revision and updates in numerical formalisms: An overview, with new results for the possibilistic framework', in *Proceedings of the Thirteenth Intl. Joint Conf. on Artif. Intell. (IJCAI93)*, pp. 620–625, San Francisco, CA, USA, (1993). Morgan Kaufmann Publishers Inc.
- [7] P. Gärdenfors, *Knowledge in Flux: Modeling the Dynamics of Epistemic States*, MIT Press, Massachusetts/England, 1988.
- [8] M. Goldszmidt, P. Morris, and J. Pearl, 'A maximum entropy approach to nonmonotonic reasoning', in *Proceedings of the Eighth Natl. Conf. on Artif. Intell. (AAAI-90)*, pp. 646–652. AAAI Press, (1990).
- [9] A. Grove and J. Halpern, 'Updating sets of probabilities', in *Proceedings of the Fourteenth Conf. on Uncertainty in Artif. Intell., UAI'98*, pp. 173–182, San Francisco, CA, USA, (1998). Morgan Kaufmann.
- [10] D. Hunter, 'Maximum entropy updating and conditionalization', in *Existence and Explanation*, eds., W. Spohn, B. Van Fraassen, and B. Skyrms, volume 49 of *The University of Western Ontario Series in Philosophy of Science*, 45–57, Springer Netherlands, (1991).
- [11] E. Jaynes, 'Where do we stand on maximum entropy?', in *The Maximum Entropy Formalism*, 15–118, MIT Press, (1978).
- [12] H. Katsuno and A. Mendelzon, 'On the difference between updating a knowledge base and revising it', in *Proceedings of the Second Intl. Conf. on Principles of Knowledge Representation and Reasoning*, pp. 387–394, (1991).
- [13] G. Kern-Isberner, 'Characterizing the principle of minimum cross-entropy within a conditional-logical framework', *Artif. Intell.*, **98**(12), 169 – 208, (1998).
- [14] G. Kern-Isberner, 'Revising and updating probabilistic beliefs', in *Frontiers in Belief Revision*, eds., M.-A. Williams and H. Rott, volume 22 of *Applied Logic Series*, 393–408, Kluwer Academic Publishers, Springer Netherlands, (2001).
- [15] G. Kern-Isberner, 'Linking iterated belief change operations to non-monotonic reasoning', in *Proceedings of the Eleventh Intl. Conf. on Principles of Knowledge Representation and Reasoning*, pp. 166–176, Menlo Park, CA, (2008). AAAI Press.
- [16] G. Kern-Isberner and W. Rödder, 'Belief revision and information fusion on optimum entropy', *Intl. Journal of Intelligent Systems*, **19**(9), 837–857, (2004).
- [17] S. Kullback, *Information theory and statistics*, volume 1, Dover, New York, 2nd edn., 1968.
- [18] D. Lehmann, M. Magidor, and K. Schlechta, 'Distance semantics for belief revision', *Journal of Symbolic Logic*, **66**(1), 295–317, (2001).
- [19] I. Levi, *The enterprise of knowledge*, The MIT press, 1983.
- [20] D. Lewis, 'Probabilities of conditionals and conditional probabilities', *Philosophical Review*, **85**(3), 297–315, (1976).
- [21] S. Michels, A. Hommersom, P. Lucas, M. Velikova, and P. Koopman, 'Inference for a new probabilistic constraint logic', in *Proceedings of the Twenty-Third Intl. Joint Conf. on Artif. Intell. (IJCAI-13)*, ed., F. Rossi, pp. 2540–2543, Menlo Park, CA, (August 2013). AAAI Press.
- [22] J. Paris, *The Uncertain Reasoner's Companion: A Mathematical Perspective*, Cambridge University Press, Cambridge, 1994.
- [23] J. Paris and A. Vencovsk, 'In defense of the maximum entropy inference process', *Intl. Journal of Approximate Reasoning*, **17**(1), 77–103, (1997).
- [24] D. Poole and A. Mackworth, *Artif. Intell.: Foundations of Computational Agents*, Cambridge University Press, New York, USA, 2010.
- [25] G. Rens and T. Meyer, 'A new approach to probabilistic belief change', in *Proceedings of the Intl. Florida AI Research Society Conf. (FLAIRS)*, eds., I. Russell and W. Eberle, pp. 582–587. AAAI Press, (2015).
- [26] J. Shore and R. Johnson, 'Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy', *Information Theory, IEEE Transactions on*, **26**(1), 26–37, (Jan 1980).
- [27] F. Voorbraak, 'Partial Probability: Theory and Applications', in *Proceedings of the First Intl. Symposium on Imprecise Probabilities and Their Applications*, pp. 360–368, (1999). url: decsai.ugr.es/ smc/isipta99/proc/073.html.
- [28] P. Walley, *Statistical Reasoning with Imprecise Probabilities*, Chapman and Hall, 1991.
- [29] A. Yue and W. Liu, 'Revising imprecise probabilistic beliefs in the framework of probabilistic logic programming', in *Proceedings of the Twenty-third AAAI Conf. on Artif. Intell. (AAAI-08)*, pp. 590–596, (2008).

Solving Dynamic Controllability Problem of Multi-Agent Plans with Uncertainty Using Mixed Integer Linear Programming

Guillaume Casanova¹ and Cédric Pralet¹ and Charles Lesire¹ and Thierry Vidal²

Abstract.

Executing multi-agent missions requires managing the uncertainty about uncontrollable events. When communications are intermittent, it additionally requires for each agent to act only based on its local view of the problem, that is independently of events which are controlled or observed by the other agents. In this paper, we propose a new framework for dealing with such contexts, with a focus on mission plans involving temporal constraints. This framework, called Multi-agent Simple Temporal Network with Uncertainty (MaSTNU), is a combination between Multi-agent Simple Temporal Network (MaSTN) and Simple Temporal Network with Uncertainty (STNU). We define the dynamic controllability property for MaSTNU, and a method for computing offline valid execution strategies which are then dispatched between agents. This method is based on a mixed-integer linear programming formulation and can also be used to optimize criteria such as the temporal flexibility of multi-agent plans.

1 Introduction

In robotic applications such as the autonomous exploration of large and hazardous areas, better performances can be obtained by using multiple robots. This can indeed lead to a faster achievement of the mission due to parallel realizations of tasks, and bring redundancy for continuing the mission in case of robot failures. One difficulty to overcome in this context is that the tasks allocated to robots must be coordinated, since there may exist precedence or synchronization constraints between tasks, or more generally constraints on the minimum/maximum temporal distances between tasks. To handle these multi-agent temporal constraints, *Multi-agent Simple Temporal Networks* (MaSTNs [1]) were recently introduced, with techniques for computing, in a distributed way, allowed distances between time-points involved in plans [1], or earliest/latest occurrence times of time-points [2].

However, one issue when using MaSTN for robotic missions is that MaSTN are not designed for obtaining decision strategies which are robust to the uncertainty about the occurrence time of uncontrollable time-points. For instance, they are not adapted to obtain plans which are feasible whatever the exact duration of tasks turn out to be. Along this line, they are not as expressive as the framework of Simple Temporal Network with Uncertainty (STNU [9]),

which makes an explicit distinction between *executable* time-points, which can be directly controlled, and *contingent* time-points, which cannot. In STNU, robust execution strategies describe a way to set the occurrence time of executable time-points depending on time-point occurrences observed so far, and these strategies are built under the strong assumption that the realization of every time-point in the temporal network is instantaneously observed. Such an assumption is often violated for multi-robot systems, since each event might be observable only from a restricted set of geographical positions.

This is why we propose a new framework for managing temporal constraints over multi-agent systems. This framework, called *Multi-agent Simple Temporal Network with Uncertainty* (MaSTNU), can be seen as an attempt to combine MaSTN and STNU. It is equipped with algorithms to compute robust execution strategies which respect every temporal constraint of the multi-agent plan despite the uncertainty about the occurrence time of contingent time-points, and which are applicable even in constrained environments featuring intermittent communications between agents. Such distributed execution strategies are obtained using a centralized offline procedure based on a Mixed Integer Programming (MIP) formulation of what we call the *multi-agent dynamic controllability problem*. This procedure is run at the mission center before triggering the coordinated deployment of the agents on the field.

The paper is organized as follows. Sect. 2 introduces some background on STN, MaSTN, STNU, and dynamic controllability checking. Sect. 3 presents the MaSTNU framework. Sect. 4 details our MIP approach for dealing with multi-agent dynamic controllability. Sect. 5 provides experimental results and discusses several ways to optimize execution strategies.

2 Background

2.1 Simple Temporal Network

A standard framework for reasoning about temporal constraints is the framework of *Simple Temporal Problems* (STPs [4]). Basically, an STP is a pair $S = (V, E)$ defined by a set $V = \{v_1, \dots, v_n\}$ of time-points representing event occurrence times, and a set E of temporal constraints between these time-points. Each constraint $e \in E$ takes the form $v_j - v_i \in [L_{ij}, U_{ij}]$, where $L_{ij} \in \mathbb{R} \cup \{-\infty\}$ and $U_{ij} \in \mathbb{R} \cup \{+\infty\}$ respectively specify a minimum and a maximum temporal distance between v_i and v_j . A specific time-point v_0 called the *reference-point* is usually added to V for representing a reference temporal position, and unary temporal constraints such as $v_i \in [L_{0i}, U_{0i}]$ can then be easily expressed as distance constraints with regards to this reference-point (constraints $v_i - v_0 \in [L_{0i}, U_{0i}]$).

¹ Onera – The French Aerospace Lab, F-31055, Toulouse, France, email: name.surname@onera.fr

² Ecole Nationale d'Ingenieurs de Tarbes, Tarbes, France, email: name.surname@enit.fr

STPs have a natural graphical representation called *Simple Temporal Networks* (STNs), which contain one vertex per time-point in V and one edge $v_i \rightarrow v_j$ labeled by $[L_{ij}, U_{ij}]$ per temporal constraint in E . STNs are appealing in practice to deal with temporal aspects because several problems that can be formulated on STNs are solvable in polytime [4], such as determining whether there exists an assignment of time-points satisfying all temporal constraints.

STNs were extended to deal with multi-agent problems on one hand and with uncertain temporal durations on the other hand. Thereafter, we give some background on these two distinct extensions.

2.2 Multi-agent Simple Temporal Network

STNs were extended to a multi-agent context, where time-points are not controlled by a single agent but are instead partitioned among a set of agents \mathcal{A} . This extension is called MaSTN for *Multiagent Simple Temporal Network* [1]. Formally, an MaSTN is defined by:

- a set of *local STNs* (one per agent $a \in \mathcal{A}$); the *local STN* associated with agent a , denoted by S_L^a , is defined by V^a , the set of *local time-points* owned by a , and E_L^a , the set of *local edges* which connect only time-points in V^a ;
- a set of *external edges* E_X , each of which constrains the temporal distance between two time-points belonging to distinct agents; apart from its local edges in E_L^a , each agent a is aware of the subset of external constraints which hold on one of its local vertices.

Fig. 1 gives an example of an MaSTN involving three agents A , B , C . Agent A (resp. B and C) owns variables v_1^A to v_6^A (resp. v_1^B to v_8^B and v_1^C to v_8^C). Edge (v_1^A, v_2^A) is an example of a local edge for agent A . Edge (v_3^A, v_3^B) is an example of an external edge in E_X . It enforces some synchronization between agents A and B .

MaSTN algorithms were defined to compute, in a distributed way, possible temporal distance between pairs of time-points (distributed partial path-consistency algorithms [1]), as well as earliest/latest dates associated with time-points (distributed arc-consistency algorithms [2]).

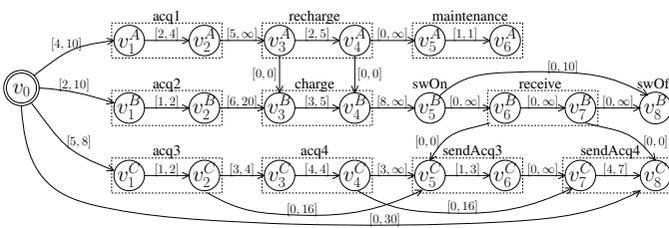


Figure 1. Example of a Multi-agent STN involving 3 agents A , B , C (one line per agent)

2.3 Simple Temporal Network with Uncertainty

In another direction, STNs were extended to *Simple Temporal Networks with Uncertainty* (STNUs [9]) in order to represent *uncertain durations*, that is durations whose value is fixed by an external process rather than by the planning agent itself.

Formally, an STNU is a triple (V, E, C) , where V is a set of time-points, E is a set of *requirement links*, and C is a set of *contingent links*. Each requirement link is defined as in standard STN. Each contingent link is defined by a pair of time-points (v_i, v_j) and by a temporal interval $[L_{ij}, U_{ij}]$ with $0 < L_{ij} < U_{ij} < \infty$. The duration

of such a contingent link, that is distance $v_j - v_i$, is known to be between L_{ij} and U_{ij} , but its precise value is not controlled. In this case, v_i and v_j are respectively called the *activation* time-point and the *contingent* time-point. Last, a time-point cannot be the contingent time-point of two distinct contingent links. Any time-point which is not a contingent time-point for some contingent link is called an *executable* time-point. In the following, we denote by V_E the set of executable time-points and by V_C the set of contingent time-points.

Fig. 2 gives an example of an STNU involving six time-points plus the reference time-point v_0 . Requirement links such as $v_2^B - v_1^B \in [4, 6]$ are depicted using continuous lines, while contingent links such as $v_1^B - v_1^A \in [1, 4]$ are depicted using dashed lines. In this STNU, the set of executable time-points is $V_E = \{v_1^A, v_2^A, v_2^B, v_3^B\}$ and the set of contingent time-points is $V_C = \{v_1^B, v_3^A\}$.

The fundamental problem associated with an STNU is to determine whether it is *dynamically controllable*, which informally means that there exists a way to dynamically assign values to executable time-points depending on observations collected, so that all requirement links are satisfied whatever the precise values of contingent links turn out to be at execution time.

More formally, dynamic controllability over STNU can be defined as follows. First, a *projection* of an STNU is an STN obtained by replacing each contingent link $v_j - v_i \in [L_{ij}, U_{ij}]$ by a deterministic link $v_j - v_i \in [d, d]$ with $d \in [L_{ij}, U_{ij}]$. A *schedule* is an assignment of values to all time-points. An execution strategy R can then be defined as a mapping from projections to schedules. An execution strategy R is said to be *valid* when for every projection p , schedule $R(p)$ satisfies all requirement links. An execution strategy R is said to be *dynamic* iff for every executable time-point v and every projections p_1, p_2 of the STNU, if the assignment of all time-points scheduled before v are the same in $R(p_1)$ and $R(p_2)$, then the values assigned to v in $R(p_1)$ and $R(p_2)$ are the same. In other words, the execution time of v can only depend on the information gathered before executing v . Last, an STNU is *Dynamically Controllable (DC)* iff there exists an execution strategy with is both valid and dynamic.

The STNU in Fig. 2 is dynamically controllable, and a valid dynamic execution strategy can be: (a) execute v_1^A at time 0, (b) wait for v_1^B to happen, (c) execute v_2^B at time $v_1^B + 4$, (d) execute v_2^A at time $v_2^B + 6$, (e) execute v_3^B at time $v_2^B + 6$, (f) wait for v_3^A to happen.

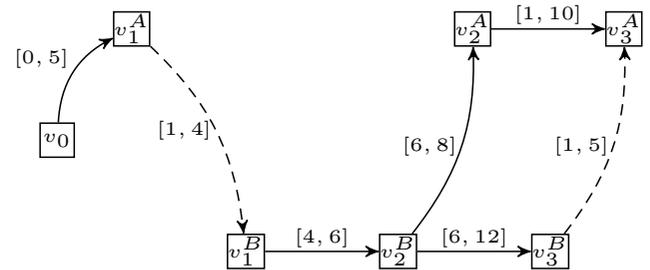


Figure 2. Example of dynamically controllable STNU

2.4 Checking Dynamic Controllability using MIP

Several algorithms do exist for checking dynamic controllability [9, 7, 5]. One of them consists in using graph-based algorithms for com-

puting mandatory *wait* constraints on requirement links. A wait constraint (v_k, w_{ijk}) on a requirement link $(v_i, v_j) \in E$ means that v_j can only be executed either after v_k is executed, or after w_{ijk} time units since the execution of v_i .

In a completely different direction, DC checking on STNUs can also be formulated as a Mixed-Integer linear Program (MIP) [3]. One advantage of such a MIP formulation is that it can be adapted for answering more general queries, such as minimally updating time bounds on contingent links such that a non-DC STNU becomes DC.

Fig.3 gives the disjunctive linear model introduced in [3], from which a MIP model can be obtained using some linearization steps. Roughly speaking, the model contains two continuous decision variables l_{ij} and u_{ij} for each pair of time-points (v_i, v_j) . Variables l_{ij} and u_{ij} respectively represent the lower and upper bounds imposed on the distance $v_j - v_i$ between time-points v_i and v_j . The model also contains a set of continuous wait variables w_{ijk} (one variable per triple of time-points (v_i, v_j, v_k) such that v_k is a contingent time-point). These variables have the same meaning as seen previously. Discrete decision variables are present in the MIP model after the linearization process. If a solution is found to the problem, then the STNU is DC. See [3] for details concerning the correctness of the modeling and the linearization process.

An optimization function f_{opt} can easily be added to the model, for instance to maximize the flexibility of solutions by using $f_{opt} = \sum_{i < j} (u_{ij} - l_{ij})$.

$$\begin{aligned} \forall (v_i, v_j) \in E, L_{ij} \leq l_{ij} \leq u_{ij} \leq U_{ij} & \quad (1) \\ \forall (v_i, v_j) \in C, (l_{ij} = L_{ij}) \wedge (u_{ij} = U_{ij}) & \quad (2) \\ \forall v_i, v_j, v_k \in V, \begin{cases} l_{ik} \leq u_{ij} + l_{jk} \leq u_{ik} \\ l_{ik} \leq l_{ij} + u_{jk} \leq u_{ik} \\ u_{ik} \leq u_{ij} + u_{jk} \\ l_{ij} + l_{jk} \leq l_{ik} \end{cases} & \quad (3) \\ \forall (v_i, v_k) \in C, \forall v_j \in V_E, (l_{jk} < 0) \vee \begin{pmatrix} u_{ij} \leq l_{ik} - l_{jk} \\ l_{ij} \geq u_{ik} - u_{jk} \end{pmatrix} & \quad (4) \\ \forall (v_i, v_k) \in C, \forall v_j \in V_E, u_{ik} - u_{jk} \leq w_{ijk} & \quad (5) \\ \forall (v_i, v_j) \in E, \forall v_k \in V_C, \min(l_{ik}, w_{ijk}) \leq l_{ij} & \quad (6) \\ \forall (v_i, v_k), (v_m, v_j) \in C^2, & \\ (w_{ijk} < 0) \vee (w_{ijk} - l_{mj} \leq w_{imk}) & \quad (7) \\ \forall (v_i, v_k) \in C, \forall v_m, v_j \in V, w_{ijk} - u_{mj} \leq w_{imk} & \quad (8) \end{aligned}$$

Figure 3. Disjunctive linear model for encoding DC on STNU [3]

3 Multi-agent Simple Temporal Network with Uncertainty (MaSTNU)

3.1 Framework Definition

As explained in the introduction, STNU cannot be directly reused in a multi-agent setting, where each agent only controls a subset of the executable time-points and only observes the occurrence of a subset of the contingent time-points. This is why we introduce Multi-agent STNU (MaSTNU).

Formally, an MaSTNU is a quadruplet (\mathcal{A}, V, E, C) , with \mathcal{A} a set of agents and (V, E, C) an STNU (V denotes the set of executable and contingent time-points, E the set of requirement links, and C the set of contingent links). Additionally, as in MaSTN, time-points in V are partitioned among \mathcal{A} , that is for every time-point $v \in V$ there exists a unique agent $a \in \mathcal{A}$ which *owns* v , denoted by $owner(v) = a$. Semantically speaking, if v is an executable time-point, then the owner of v is the agent which controls the execution of the event associated with v . If v is a contingent time-point, the owner of v is the unique agent which is assumed to instantaneously observe the realization of v . Time-points owned by other agents are not supposed to be directly observed, however information about their realization can be obtained thanks to external contingent links. Reference-point v_0 represents a clock synchronized between agent and is considered to be simultaneously owned by all agents.

In the following, for each agent $a \in \mathcal{A}$, V^a denotes the set of time-points owned by a , called the *local time-points* of a . We denote by E_L^a (resp. C_L^a) the set of *local* requirement links (resp. contingent links), which hold only on time-points owned by a . Analogously to MaSTN, we also define E_X (resp. C_X) as the set of *external* requirement links (resp. contingent links), which connect two time-points owned by different agents.

Fig. 4 gives an example of an MaSTNU involving two agents A and B , which respectively own time-points $V^A = \{v_1^A, v_2^A, v_3^A\}$ and $V^B = \{v_1^B, v_2^B, v_3^B\}$. The link from v_1^A to v_1^B is an external contingent link, the link from v_2^B to v_2^A is an external requirement link, the link from v_1^B to v_2^B is a local requirement link, and there is no local contingent link. Semantically speaking, external contingent links model observations received by an agent, the source of these observations being owned by other agents. For instance, the source of a contingent link might be the start of a data transmission process triggered by one agent, and the target of this link might be the end of this data transmission process, observed by the receiving agent. As in MaSTN, external requirement links correspond to synchronization constraints between agents. For instance, they can serve to express that there must not be more than 10 time units between successive surveillances of a given area by two distinct agents.

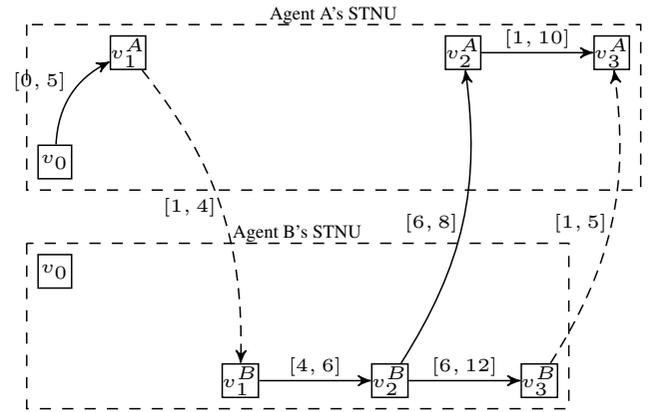


Figure 4. Example of an MaSTNU

3.2 Dynamic Controllability Revisited

The multi-agent nature of MaSTNU requires an adaptation of the dynamic controllability property. Indeed, given an MaSTNU (\mathcal{A}, V, E, C) , computing a valid dynamic execution strategy for STNU (V, E, C) does not necessarily give an applicable multi-agent strategy. As an illustration, consider the MaSTNU provided in Fig. 4. The STNU associated with it is the STNU previously shown in Fig. 2. By considering the execution strategy seen for this STNU and by partitioning it between agents, we obtain the following strategy:

- for agent A : (a) execute v_1^A at time 0, (b) execute v_2^A at time $v_2^B + 6$, (c) wait for v_3^A to happen;
- for agent B : (a) wait for v_1^B to happen, (b) execute v_2^B at time $v_1^B + 4$, (c) execute v_3^B at time $v_2^B + 6$.

The issue with such a strategy is that agent A has no guarantee to be able to execute it, because it might not observe external time-point v_2^B owned by agent B .

This is why we introduce a new definition of dynamic controllability which is adapted to MaSTNU. Let (\mathcal{A}, V, E, C) be an MaSTNU and let R be an execution strategy for the associated STNU (V, E, C) . Execution strategy R is said to be *distributed* iff for every projections p, p' (that is for every two possible assignments of the duration of contingent links) and for every agent $a \in \mathcal{A}$, if schedule $R(p)$ and schedule $R(p')$ assign the same value to all contingent time-points owned by a , then they also assign the same value to all executable time-points owned by a . In other words, each agent only acts based on its own immediate observations, which means that the execution strategy is robust to the missing observations of external time-points.

An MaSTNU (\mathcal{A}, V, E, C) is then said to be *dynamically controllable* iff STNU (V, E, C) admits an execution strategy which is *valid* (it induces schedules which satisfy all requirement links), *dynamic* (decisions are made only based on past information), and *distributed* (previous definition).

For the MaSTNU given in Fig. 4, an example of a valid, dynamic and distributed execution strategy is:

- for agent A : (a) execute v_1^A at time 4, (b) execute v_2^A at time 19, (c) wait for v_3^A to happen;
- for agent B : (a) wait for v_1^B to happen, (b) execute v_2^B at time $v_1^B + 6$ if $v_1^B \leq 7$ and at time $v_1^B + 4$ otherwise, (c) execute v_3^B at time $v_2^B + 8$.

In the following, we introduce techniques for checking DC for MaSTNU and for automatically computing distributed strategies.

4 Dynamic Controllability Check and Computation of Execution Strategies

To check DC for an MaSTNU (\mathcal{A}, V, E, C) , we first check DC for STNU (V, E, C) . If this STNU is not DC, then the MaSTNU is not DC either, because acceptable execution strategies for MaSTNU are more restricted than acceptable execution strategies for STNU. Otherwise, if STNU (V, E, C) is DC, we perform additional operations to determine whether the original MaSTNU is DC.

4.1 From one MaSTNU to a set of local STNUs

The key idea in our method is to transform the original MaSTNU S into a *distributed* MaSTNU, which contains no external link between agents, and then to partition this distributed MaSTNU into a

set of local STNUs $\{S^a \mid a \in \mathcal{A}\}$. Fig. 5 shows an example of such a process. The reason why we consider distributed MaSTNUs as the transformation target is that if each agent $a \in \mathcal{A}$ uses a valid dynamic execution strategy R^a for its own local STNU S^a , then the global strategy obtained by joining strategies R^a is valid and dynamic, and it is also distributed because we are sure that in R^a , each agent acts only based on the observations it is supposed to get at execution (no possible occurrence of external time-points in the execution strategy thanks to the partitioning). In other words, the set of local execution strategies $\{R^a \mid a \in \mathcal{A}\}$ allows to dynamically control the MaSTNU.

Globally, to transform the original MaSTNU into a set of local STNUs, we need to perform two kinds of operations:

1. to replace external requirement links of the original MaSTNU by requirement links which are local to agents, as done in Fig. 5 for external constraint $v_2^A - v_2^B \in [6, 8]$ which will necessarily be satisfied thanks to the two internal requirements $v_2^A - v_0 \in [19, 19]$ and $v_2^B - v_0 \in [11, 13]$ which are present in the partitioned MaSTNU; the introduced local requirement links can be stronger than in the initial MaSTNU, and they are implicitly used to coordinate agent actions;
2. to remove external contingent links and to replace them by local contingent links, as done in Fig. 5 for (v_1^A, v_1^B) which is replaced by (v_0, v_1^B) ; more generally, the external source v of a contingent link (v, w) must be replaced by a local source u contained in the agent which owns w .

The way these two operations are realized is presented in the two following sections. Compared to DC reasoning on STNU, it is worth mentioning that the transformation of the original MaSTNU into several local STNUs is a *combinatorial decision problem*, because for instance there is not necessarily a unique way of distributing/sharing the satisfaction of external requirement links among agents, or a unique way of reassigning the contingency source of a contingent time-point. The associated decision problem is formalized using a MIP model, which allows us to reuse elements from the existing MIP model given in Sect. 2.4 for standard STNUs. In the MIP model built, we capture several constraints guaranteeing the satisfaction of the external requirement links of the original MaSTNU, and several constraints guaranteeing that the local contingency assumptions made in the distributed MaSTNU are not restrictive with regards to the set of possible scenarios covered by the external contingent links of the original MaSTNU. By adding a linear optimization function, MIP solvers can then be used to find an optimal distribution of temporal constraints such that all local STNUs are DC.

In the following, as in the MIP model of DC for STNU, we use, for every $i < j$, variables l_{ij} and u_{ij} to represent the lower and upper bounds imposed on the distance $v_j - v_i$ between v_i and v_j . Moreover, for $i < j$, we also use u_{ji} as a substitute for $-l_{ij}$.

4.2 Internalization of external requirement links

Let us consider an external requirement link $e = (v_i, v_j)$, with $owner(v_i) = a, owner(v_j) = b, a \neq b$. Initially, e is labeled by $[L_{ij}, U_{ij}]$. The main issue is that a and b might not have enough information during execution to ensure that e is respected. For example, if b waits to observe v_i before executing v_j then it might fail at respecting e if the delay for observing v_i is greater than U_{ij} . Similarly, a has no information about when it should be executing v_i in a way such that b can execute v_j and respect e .

To make sure that the upper bound of e is respected during execution without using any communication between a and b , it suffices to

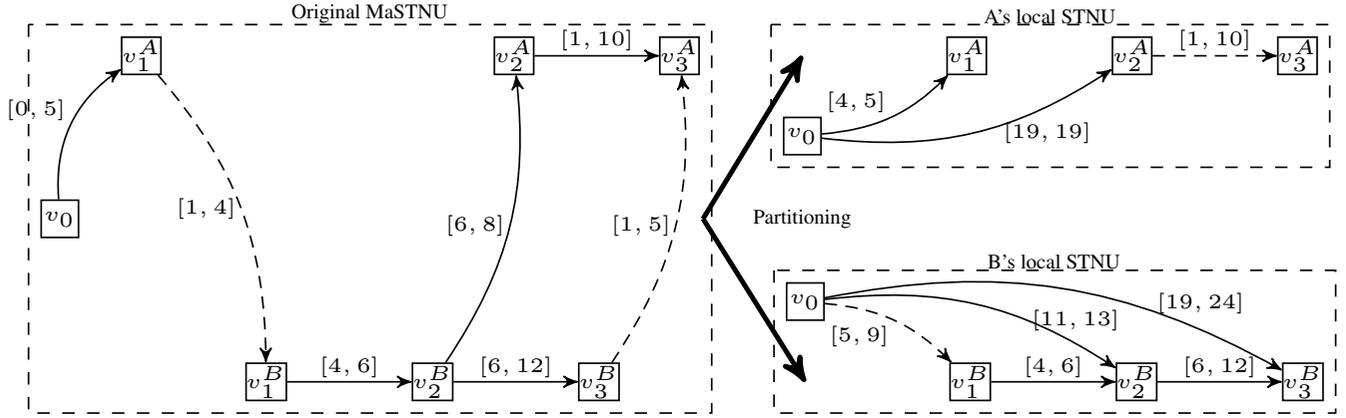


Figure 5. Original MaSTNU and its partitioning

find a path $p = [p_1, \dots, p_k]$ composed of time-points, such that:

1. each link (p_i, p_{i+1}) involved in p is either an internal link ($owner(p_i) = owner(p_{i+1})$), or an external contingent link ($(p_i, p_{i+1}) \in C_X$ or $(p_{i+1}, p_i) \in C_X$);
2. path p defines a path from v_i to v_j which is shorter than U_{ij} , so that if all constraints between time-points in p are satisfied, then e is also satisfied.

Such paths are called *distributed paths*. A contingent link in a distributed path is necessarily satisfied at execution by definition. An internal requirement link in a distributed path is satisfiable by the agent holding it at execution as long as its local STNU is DC. Therefore, if all local STNUs are DC, then all constraints in path p are satisfiable at execution, and therefore the original external requirement link is satisfiable as well. Similar distributed paths must be found to justify that the lower bound of e is satisfied.

In order to find such paths, it actually suffices to decide on the sequence of contingent links to use for justifying the satisfaction of bound u_{ij} , because successive links which are internal to a single agent can be harmlessly collapsed into a single internal link thanks to the path consistency property over STNU. See Figure 6 for an illustration of a path $p = [v_i, v_{k_1}, v_{l_1}, v_{k_2}, v_{l_2}, \dots, v_{k_n}, v_{l_n}, v_j]$ covering the satisfaction of requirement link (v_i, v_j) . On this example, the satisfaction of requirement link (v_i, v_j) is covered by the satisfaction of some requirement over (v_i, v_{k_1}) and (v_{l_1}, v_j) ; the satisfaction of (v_{l_1}, v_j) is itself covered by the satisfaction of some requirement over (v_{l_1}, v_{k_2}) and (v_{l_2}, v_j) ... and so on until there is no more external link to satisfy. Informally speaking, the path built use a sequence of quadrilaterals, and it introduces some new temporal distance constraints with regards to time-points which are correlated through contingent links. Exploiting these correlations is the only way to be robust to the absence of communication. Also, thanks to the path consistency property again, by imposing that local STNUs must be DC, it suffices to search for distributed paths which cross each agent at most once.

In order to formalize such a process, we define two sets:

- set $\overline{E_X}$ which contains all possible external links which are not contingent links, and therefore might require justification; this set is given by $\overline{E_X} = \{(v_i, v_j) \in V^2 \mid (owner(v_i) \neq owner(v_j)) \wedge ((v_i, v_j) \notin C_X) \wedge ((v_i, v_j) \notin C_X)\}$;

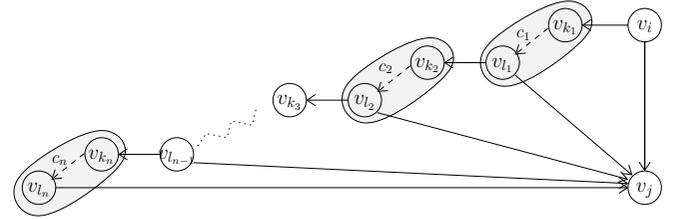


Figure 6. Justification path for an external requirement link between v_i and v_j

- set Q which contains all quadruplets (v_i, v_j, v_k, v_l) such that
 - $(v_i, v_j) \in \overline{E_X}$ is an external requirement link whose satisfaction which might have to be justified when building justification paths;
 - (v_k, v_l) is associated with a contingent link which can be used in the justification for (v_i, v_j) , which means that (1) either $(v_k, v_l) = (v_0, v_0)$ (case in which the upper bound u_{ij} over $v_j - v_i$ is justified by a path through the reference time-point), (2) or $(owner(v_k) = owner(v_i)) \wedge ((v_k, v_l) \in C_X \vee (v_l, v_k) \in C_X)$.

Note that the size of Q is at most cubic in the number of time-points, since v_k and v_l are related by a contingent link and because a contingent time-point can only have a unique contingent link pointing to it.

To model the requirement to cover external requirement links by distributed paths, we introduce a MIP modeling which uses the following variables:

- $\forall (v_i, v_j) \in \overline{E_X}, b_{ij} \in \{0, 1\}$ is a boolean decision variable encoding that we need to justify external requirement link $v_j - v_i \leq u_{ij}$;
- $\forall (v_i, v_j, v_k, v_l) \in Q, z_{ijkl} \in \{0, 1\}$ is a boolean decision variable encoding that contingent link between v_k and v_l (either link (v_k, v_l) or link (v_l, v_k)) is used to justify the satisfaction of $v_j - v_i \leq u_{ij}$;
- $\forall (v_i, v_j) \in \overline{E_X}, h_{ij} \in [0, H]$ are decision variables encoding the *height* of the justification of the satisfaction of $v_j - v_i \leq u_{ij}$, with H a constant equal to $\max(|A| - 2, |C_X|)$; these height variables

are used to avoid cycles, that is to avoid cases in which the satisfaction of the upper bound on an external link e is justified by the upper bound associated with an external link e' , and in which the upper bound of e' is justified by the upper bound of e ; see Fig. 7 for an illustration of what could happen without preventing cycles in justifications; it also helps bounding the search process since it suffices to consider distributed paths which cross each agent at most once and each contingent link at most once, which explains the value chosen for the upper bound of h_{ij} .

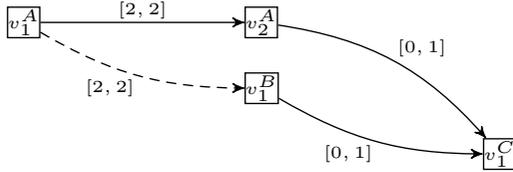


Figure 7. Cycles in justifications without the use of heights variables: the satisfaction of external requirement $v_1^C - v_2^A \leq 1$ can be justified by the satisfaction of external requirement $v_1^C - v_1^B \leq 1$, and reciprocally

We impose several linear constraints for representing the satisfaction of the external requirements by distributed paths. First, the lower and upper bounds associated with contingent links of the original MaSTNU cannot be shrunk:

$$\forall (v_i, v_j) \in C_X, (l_{ij} \leq L_{ij}) \wedge (u_{ij} \geq U_{ij}) \quad (9)$$

Next, every external requirement link in the initial MaSTNU must be justified:

$$\forall (v_i, v_j) \in E_X \text{ s.t. } U_{ij} \neq +\infty, b_{ij} = 1 \quad (10)$$

$$\forall (v_i, v_j) \in E_X \text{ s.t. } L_{ij} \neq -\infty, b_{ji} = 1 \quad (11)$$

If an external requirement link (original or intermediate) must be justified, then there exists a unique contingent link justifying it:

$$\forall (i, j) \in \overline{E}_X, b_{ij} = \sum_{(v_i, v_j, v_k, v_l) \in Q} z_{ijkl} \quad (12)$$

An external requirement link is justified if there exists a shorter distributed path:

$$\forall (i, j, k, l) \in Q, u_{ij} \geq u_{ik} + u_{kl} + u_{lj} + (z_{ijkl} - 1)M \quad (13)$$

In the previous equation, M is a large constant equal to $L_{ij} - U_{ik} - U_{kl} - U_{lj}$, so that the constraint is always satisfied when $z_{ijkl} = 0$.

Then, every external requirement link used in a justification must also be justified (again, see Fig. 6 for an illustration):

$$\forall (v_i, v_j, v_k, v_l) \in Q \text{ s.t. } (v_l, v_j) \in \overline{E}_X, z_{ijkl} \leq b_{lj} \quad (14)$$

Finally, we are preventing cycles in justifications thanks to the following set of constraints (in the following equation, H is the maximum value of h_{ij} variables):

$$\forall (v_i, v_j, v_k, v_l) \in Q \text{ s.t. } (v_l, v_j) \in \overline{E}_X \\ h_{ij} + (1 - z_{ijkl})(H + 1) \geq h_{lj} + 1 \quad (15)$$

Fig. 8 shows a representation of the MaSTNU obtained after the internalization process of external requirement links. In this example, A and B tighten internal constraints (v_0, v_1^A) , (v_0, v_2^A) , (v_0, v_2^B) and (v_0, v_3^B) so that external requirement link (v_2^B, v_3^A) is satisfied at execution. Contrarily to the mono-agent STNU solution seen in Fig. 2, v_2^A has no temporal flexibility anymore.

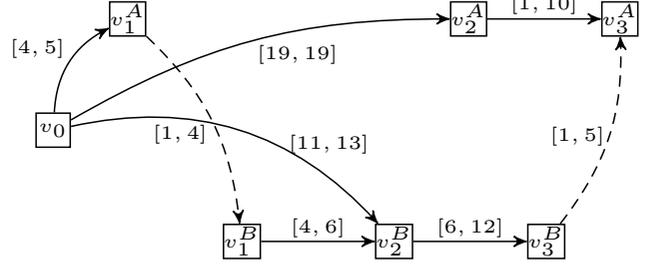


Figure 8. MaSTNU after internalization of external requirement links

4.3 Internalization of external contingent links

External contingent links also have to be internalized, otherwise external time-points might appear in execution strategies, which would invalidate the distributivity of these strategies. Globally, the idea in the internalization of links in C_X is that every potential situation which may be encountered owing to the original MaSTNU must be covered by scenarios considered at the level of local STNUs.

To illustrate the transformation proposed, let us consider an external contingent link $c = (v_i, v_j)$, with $owner(v_i) = a$, $owner(v_j) = b$, $a \neq b$. c is labeled by $[L_{ij}, U_{ij}]$, with $L_{ij} > 0$. Any execution strategy directly using the fact that “ v_j occurs necessary between L_{ij} and U_{ij} units of time after v_i ” cannot be sound as b does not directly observe v_i . This is why we need to explicitly erase c from the MaSTNU representation while keeping the uncontrollable status of v_j . The only solution to do this is to replace link (v_i, v_j) by an internal contingent link (v_k, v_j) in the set of local contingent constraints of agent b . In this case, we say that we use substitution triangle (v_i, v_j, v_k) . In the following, we define the set of candidate substitution triangles by $T = \{(v_i, v_j, v_k) \mid (v_i, v_j) \in C_X, v_k \in V^{owner(v_j)} \setminus \{v_j\}\}$. For every external contingent link (v_i, v_j) , as there is a freedom in the local time-point v_k chosen for activating v_j , we add in the MIP model the following set of decision variables:

- $\forall (v_i, v_j, v_k) \in T, c_{kj} \in \{0, 1\}$ is a boolean decision variable encoding that we substitute external contingent link (v_i, v_j) by a new internal contingent link (v_k, v_j) .

Several constraints are imposed over these variables. First, every external contingent link must be substituted by exactly one internal contingent link:

$$\forall (v_i, v_j) \in C_X, \sum_{v_k \mid (v_i, v_j, v_k) \in Q} c_{kj} = 1 \quad (16)$$

If an external contingent link (v_i, v_j) is substituted by an internal contingent link (v_k, v_j) , then the bounds specified by (v_i, v_j) must not be less restrictive than the bounds given by path $v_i \rightarrow v_k \rightarrow v_j$:

$$\forall (v_i, v_j, v_k) \in T, \begin{cases} u_{kj} \geq u_{ki} + u_{ij} + (c_{kj} - 1)M' \\ 0 < l_{kj} \leq l_{ki} + l_{ij} + (1 - c_{kj})M' \end{cases} \quad (17)$$

with M' a large constant.

If external contingent link (v_i, v_j) is substituted by internal contingent link (v_k, v_j) , then the associated requirement link over (v_i, v_k) must be justified:

$$\forall (v_i, v_j, v_k) \in T, \begin{cases} (v_i, v_k) \in \overline{E}_X : c_{kj} \leq b_{ik} \\ (v_k, v_i) \in \overline{E}_X : c_{kj} \leq b_{ki} \end{cases} \quad (18)$$

Fig. 9 shows the distributed MaSTN obtained after internalizing both external requirement links and external contingent links. Compared to the previous example, (v_0, v_3^B) had to be constrained further. Moreover, (v_0, v_1^B) and (v_2^A, v_3^A) are now contingent links.

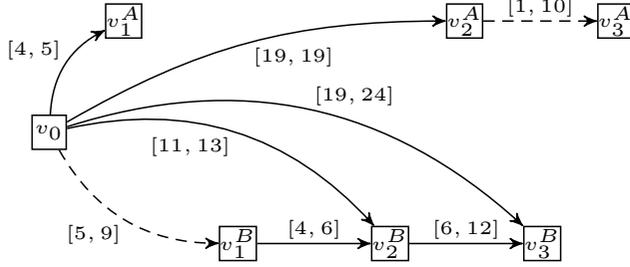


Figure 9. MaSTN after internalization of external requirement and contingent links

4.4 Dynamic Controllability of Local STNUs

Finally, we have to express that the local STNU associated with each agent must be dynamically controllable. This is expressed by adapting the model provided in Section 2.4. An adaptation is required because due to the choice in the internalization of external contingent links, the set of local contingent link is not fixed. This implies for instance that the constraint given in Eq. 8 must be replaced by:

$$\forall v_k \in C_X, \forall v_i \in V^{owner(v_k)} \setminus \{v_k\}, \forall v_m, v_j \in V, \\ w_{ijk} - u_{mj} \leq w_{imk} + (1 - c_{ik})M'' \quad (19)$$

with M'' a large constant. Similar transformations must be applied for Eq. 4, 5, and 7.

4.5 Distributing local STNUs

If a solution to the global MIP problem exists, this solution describes a distributed MaSTNU. The latter can be partitioned into a set of N local STNUs, one for each agent, while ignoring external constraints. Local STNUs can be dispatched between agents, and the mission can start.

4.6 Discussion

Completeness The techniques defined for checking DC are sound but not complete, essentially because of the internalization of external contingent links, which can make lose some information on the correlation between time-points. More precisely, by transforming an MaSTNU into a set of local STNUs, we do not represent some correlations between contingent time-points. For instance, consider an MaSTNU involving one contingent time-point x belonging to agent A , two contingent time-points y, z belonging to agent B , and two contingent links $y - x \in [2, 3]$ and $z - x \in [2, 3]$. In this case, the temporal constraint $z - y \in [-1, 1]$ necessarily holds (y and x are correlated). With our approach, we cannot represent it by a contingent link between y and z (negative lower bound), and we cannot add a new time-point t in B pointing to both y and z since t would have to be observable by B . We believe that all these points are more STNU related issues (representing non-causal uncertainty).

MIP versus propagation techniques for STNU As mentioned previously, searching for robust execution strategies is a combinatorial task and we cannot directly reuse polynomial DC checking techniques available in the literature [9, 7, 5]. Such a combinatorial aspect is also present in [3] for building an STNU which is DC from an initial STNU which is not.

Case without contingent links The techniques defined can also be used in the particular case where there is no contingent link, that is where the MaSTNU is actually an MaSTN. With regards to existing work on MaSTN, one contribution is that the model introduced allows to compute robust execution strategies which can be executed independently by the agents. We are not aware of previous works on this point for MaSTN. Also, when there is no contingent link, it can be shown that there is actually no boolean decision variable in the model and the MIP becomes a linear program solvable in polytime.

Objective functions Our method can optimize the set of local STNUs by maximizing objective function $f_{opt} = \sum_{i < j} (u_{ij} - l_{ij})$. Many other metrics can be used. For example, Fig 10 shows the local STNUs obtained when minimizing the latest execution time of the last time-point, to finish the mission as soon as possible (minimization of $f_{opt} = \max_i u_{0i}$). With this new objective function, the maximum mission completion time is reduced from 29 time units to 24 time units.

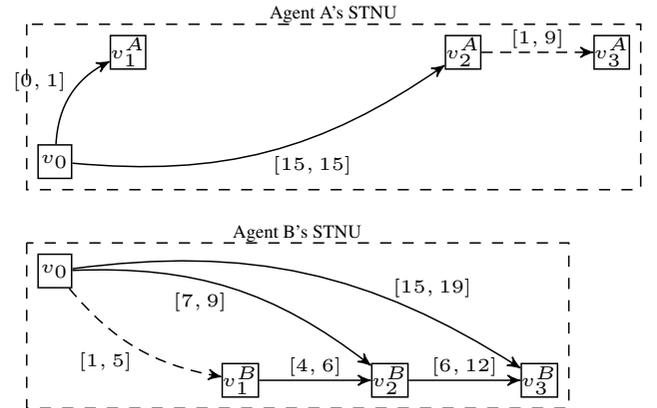


Figure 10. Mission duration optimization

Objective functions can also be used to balance the flexibility of solutions between agents, in order to avoid overly constrained agents. In this case, we define the *normalized flexibility* metrics f : $\forall a \in \mathcal{A}, f(a) = \frac{1}{|VA|} \sum_{v_i, v_j \in VA} (u_{ij} - l_{ij})$. The corresponding objective function to maximize is then $f_{opt} = \min_{a \in \mathcal{A}} f(a)$. This is particularly useful when the number of agents is high, since in this case the maximum global flexibility can often be reached by constraining as much as possible a unique agent. Another option can be to keep the initial objective function and constrain the problem such that each agent achieves a minimal threshold t of normalized flexibility: $\forall a \in \mathcal{A}, f(a) \geq t$.

5 Experiments

Running Times We tested our MIP approach using the CPLEX solver on 500 instances of randomly-generated MaSTNUs, ranging homogeneously from 10 to 40 nodes. All experiments were run on 3.0GHz Intel cores and 4GB memory. Finding the optimal solution to the MIP problem typically takes between 0.2 seconds for the 6-nodes example used in this paper, and 1200 seconds for a 4-agents and 40-nodes example containing 4 external contingent links and 10 external requirement links. However, in this last case a first solution, that strictly improves the solution found by removing external contingent links, is found in 80 seconds, at the expense of a drop in the flexibility of 75% compared to the optimal solution.

It must be noted that our current implementation of DC for local STNU (section 2.4) is based on the $O(N^5)$ -time DC-checking techniques from [9], and is the primary cause of the scalability performances. A more efficient algorithm in $O(N^4)$ -time can be found in [6], however this algorithm is more complex to translate into a MIP formalism and is beyond the scope of this paper.

Impact of Observations on Temporal Flexibility For the sake of consistency with real life applications vocabulary, in the remainder of this paper we will refer to external contingent links as *observations*.

We want to measure the improvements made by our method on the "quality" of resulting execution strategies. To this end, we define the *Relative temporal flexibility* of a solution as being the ratio of the value of the objective function of solution to the value of the objective function of the corresponding STNU:

$$\text{Relative Temporal Flexibility} = \frac{f_{opt}(MaSTNU)}{f_{opt}(STNU)}$$

This metric gives us a strong indicator of the "quality" of a solution compared to best and worst cases. At 100% it means the solution is as much flexible as the corresponding STNU, at 0% it means that the solution found is totally rigid with no tolerance for execution error.

We also assume that external contingent links take values in $[0, x]$, i.e observations of events by other agents are made within x unit of times. We compare x to the temporal flexibility of external requirement links in order to obtain the *relative delay of observations*:

$$\text{Relative delay of observations} = \frac{|E^X|}{|C^X|} \cdot \frac{\sum_{(v_i, v_j) \in C^X} (u_{ij})}{\sum_{(v_i, v_j) \in E^X} (u_{ij} - l_{ij})}$$

This ratio measure the "quality" of observations (lower ratio means better quality), which may be translated as cost in real applications.

Fig 11 shows the impact of the number of observations (i.e number of external contingent links) and their quality (i.e their immediacy) on the relative flexibility of the solution found by our method. The parameters of the generated MaSTNUs are as follow: 4 agents of 5 nodes each, 4 external requirement links (connecting two randomly-chosen nodes from distinct agents). The number of internal requirement (resp. contingent) links for each agent were randomly drawn from 4 to 6 (resp. from 0 to 2). Additional nodes and external contingent links, representing observations, were then added depending on the experiments.

Each point on a curve represents the mean of the relative flexibility of the solution found over 10 MaSTNUs randomly-generated with the corresponding set of parameters. We display the results for two limit cases, the "Full observations" and the "No observations" cases, and an intermediary one, the "Half observations" case.

The "No observation" curve corresponds to MaSTNUs without any external contingent links: $C_X = \emptyset$. In this case, agents cannot receive any informations during execution, so the flexibility of the solution is minimal: agents must agree on a rigid schedule before

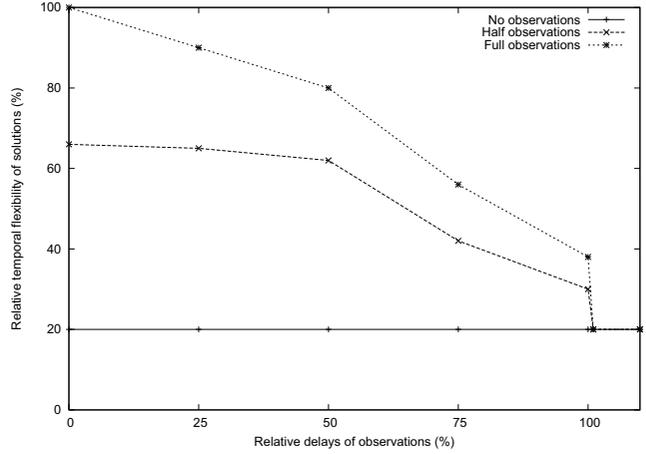


Figure 11. Influence of numbers of observations and delays

execution start. This threshold depends on the MaSTNUs considered, other MaSTNUs may have lower or higher relative flexibility than the 20% reported in our experiments in case of absence of observations.

The "Full observations" curve corresponds to MaSTNUs wherein each event is observed by each other agent: $\forall v \in V, \forall a \in A, a \neq \text{owner}(v), \exists w \in V^a$ s.t. $(v, w) \in C_X$. In the extreme case with no delays of observation, the resulting solution has the same quality than if the MaSTNU were considered as a STNU. In the opposite extreme case with high delays of observation, no useful informations can be received from the observations, and the agents must act on their own as in the no observation case.

The "Half observations" curve corresponds to MaSTNUs identical to the "Full observations" set, except half of vertices in V are not observed by any other agent. The quality of the solution actually depends on which vertices are observed: for instance observations of external vertices are more likely to be useful than observations of internal vertices.

Higher numbers of observations lead to higher flexibility of solutions, at the expense of increased computational costs and potentially increased workload during the mission if the observations were not initially scheduled.

6 Conclusion

Dynamic controllability is an important property for temporal plans with uncertainty as it improves the odds of success of a mission. In this paper we showed how to handle uncertain temporal constraints in multi-agent temporal plans thanks to Multi-agent Simple Temporal Network with Uncertainty, and how to use a MIP model to get executable plans which are dispatched between the agents. There are several future work directions for improving the management of MaSTNU, such as solving the MIP incrementally to repair infeasible MaSTNUs by adding observations one by one while optimizing computation times, or distributing the MIP solving process in order to reoptimize temporal plans during the mission, or taking into account the existence of communication rendez-vous [8].

REFERENCES

- [1] James C. Boerkoel, Léon Planken, Ronald Wilcox, and Julie A. Shah, 'Distributed Algorithms for Incrementally Maintaining Multiagent Simple Temporal Networks', in *Int. Conf. on Automated Planning and Scheduling (ICAPS)*, Rome, Italy, (2013).
- [2] Guillaume Casanova, Cédric Pralet, and Charles Lesire, 'Managing dynamic multi-agent simple temporal network', in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 1171–1179, Istanbul, Turkey, (2015).
- [3] Jing Cui, Peng Yu, Cheng Fang, Patrik Haslum, and Brian C. Williams, 'Optimising bounds in simple temporal networks with uncertainty under dynamic controllability constraints', in *Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 52–60, Jerusalem, Israel, (2015).
- [4] Rina Dechter, Itay Meiri, and Judea Pearl, 'Temporal Constraint Networks', *Artificial Intelligence*, **49**(1-3), 61–95, (1991).
- [5] Luke Hunsberger, 'Efficient execution of dynamically controllable simple temporal networks with uncertainty', *Acta Informatica*, **52**(8), 1–59, (2015).
- [6] Paul Morris, 'A structural characterization of temporal dynamic controllability', in *Principles and Practice of Constraint Programming (CP)*, pp. 375–389, Nantes, France, (2006).
- [7] Paul Morris, 'Dynamic controllability and dispatchability relationships', in *Integration of AI and OR Techniques in Constraint Programming - 11th International Conference (CPAIOR)*, pp. 464–479, Cork, Ireland, (2014).
- [8] Paul H. Morris and Nicola Muscettola, 'Managing temporal uncertainty through waypoint controllability', in *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1253–1258, Stockholm, Sweden, (1999).
- [9] Paul H. Morris, Nicola Muscettola, and Thierry Vidal, 'Dynamic Control Of Plans With Temporal Uncertainty', in *Int. Joint Conference on Artificial Intelligence (IJCAI)*, Seattle, WA, USA, (2001).

Combining Efficient Preprocessing and Incremental MaxSAT Reasoning for MaxClique in Large Graphs

Hua Jiang¹ and Chu-Min Li² and Felip Manyà³

Abstract. We describe a new exact algorithm for MaxClique, called LMC (short for Large MaxClique), that is especially suited for large sparse graphs. LMC is competitive because it combines an efficient preprocessing procedure and incremental MaxSAT reasoning in a branch-and-bound scheme. The empirical results show that LMC outperforms existing exact MaxClique algorithms on large sparse graphs from real-world applications.

1 INTRODUCTION

In an undirected graph $G=(V, E)$, where V is the set of vertices and E is the set of edges, a *clique* C is a subset of V such that all its vertices are adjacent to each other. The size of C is its cardinality. The density of G is computed as $2 \times |E| / (|V| \times (|V|-1))$. The *Maximum Clique Problem* (MaxClique) is to find a clique of maximum size in G , denoted by $\omega(G)$.

MaxClique is *NP-Hard* [12] and has many practical applications such as fault diagnosis [6], bioinformatics and chemoinformatics [9], coding theory [10], economics [7], and social network analysis [2]. The most deeply studied MaxClique algorithms are exact algorithms based on the branch-and-bound (BnB) scheme [1, 8, 11, 16, 17, 20, 22, 29, 32]. There are also efficient heuristic algorithms such as [5, 13, 23, 24] that find approximate solutions.

In recent years, special attention has been paid to large graphs from real-world applications such as graphs compiled from Internet, social networks, biological networks, collaboration networks and interaction networks. They usually have very low density, contain a high amount of vertices, and have common statistical properties such as small-world property, power-law degree distributions, and clustering [21]. Certainly, cliques are also a valuable property for analyzing such graphs. For example, in biological networks, a clique might be a functional group; in social networks of acquaintance, a clique might identify an organization or a community; and in web networks, a clique might help to find a certain topic.

State-of-the-art exact MaxClique algorithms are effective in solving DIMACS [14] and randomly generated graphs, but unfortunately very few of them are able to solve large graphs from real-world applications. PMC [25] and BBMCS [27] are exceptions. They were designed to solve large sparse graphs. PMC implements a BnB scheme, and uses approximate graph coloring bounds to prune search and parallelization to speed up the algorithm. BBMCS is a very recent al-

gorithm for large sparse graphs, which derives from an efficient bit-string encoding and bit-parallel algorithm called BBMCI [28].

Standard MaxSAT reasoning was proposed for MaxClique in [20] and was combined with an incremental upper bound in [16]. Incremental MaxSAT reasoning was proposed in [17] and has shown a superiority over standard MaxSAT reasoning for MaxClique. However, it is very hard to make sophisticated techniques such as incremental MaxSAT reasoning or standard MaxSAT reasoning effective for large graphs, so that it is noted in [27] that sophisticated techniques such as MaxSAT reasoning are not useful for large graphs. In this paper, we present a new exact MaxClique algorithm for large graphs, called LMC (short for Large MaxClique), that combines a novel efficient preprocessing and incremental MaxSAT reasoning. The empirical results on a representative sample of large sparse graphs from real-world applications show that LMC is a fast algorithm for graphs with millions of vertices, and substantially outperforms PMC and BBMCS, proving that MaxSAT reasoning can be very effective for large graphs.

The paper is organized as follows. Section 2 describes the new algorithm and the techniques it implements. Section 3 reports and analyzes the empirical results. Finally, Section 4 concludes.

2 LMC: A NEW ALGORITHM FOR LARGE SPARSE GRAPHS

We describe algorithm LMC, designed for large sparse graphs, which is composed of two main components: an efficient preprocessing procedure *Initialize* and a main search procedure *SearchMaxClique* employing incremental MaxSAT Reasoning.

2.1 The preprocessing procedure

Preprocessing in MaxClique BnB algorithms is decisive for efficiency, especially for solving large sparse graphs. Preprocessing generally performs the following three tasks:

- Derive a vertex ordering for search;
- Find an initial clique;
- Reduce the input graph; i.e., remove as many vertices that do not belong to any maximum clique as possible.

We define a novel preprocessing procedure called *Initialize*, which performs efficiently all these three tasks at the same time. The starting point is the notion of *core*, which was first used in social network analysis [30]. Let $deg_G(v)$ denote the number of vertices that are adjacent to v in graph $G = (V, E)$, or degree of v in G (G is omitted when it is clear from the context). A subgraph G' induced by $V' \subseteq V$, written as $G'=G(V')$, is a core of order k or a k -core

¹ Huazhong University of Sciences and Technology (HUST) & JiangHan University, China, email: jh_hgt@163.com

² Corresponding author, MIS, Université de Picardie Jules Verne, France, email: chu-min.li@u-picardie.fr

³ Artificial Intelligence Research Institute (IIIA, CSIC), Spain, email: felip@iiia.csic.es

iff $\deg_{G'}(v) \geq k$ for each $v \in V'$, and G' is the subgraph with the largest number of vertices with this property. The core number of a vertex v , denoted by $k(v)$, is the highest order of a core that contains v . The core number of a graph $G = (V, E)$, denoted by $k(G)$, is the maximum core number among the vertices of G . Refer to the graph in Figure 1. The set $\{v_1, v_2, v_3, v_4, v_5\}$ induces a 2-core, and the set $\{v_1, v_2, v_3, v_4, v_5, v_6\}$ induces a 1-core. The core number of the graph is 2, because the core number of v_6 is 1 and the core number of the other vertices is 2.

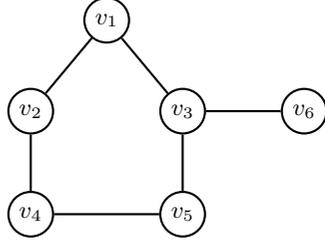


Figure 1. A graph with $k(G)=2$ and $\omega(G)=2$.

The core number of a graph $G = (V, E)$, as well as the core number of each vertex in V , can be efficiently computed in time $O(|E|)$ with an algorithm based on the following property [3, 4]: If we recursively delete all vertices of degree less than k and all edges incident with them in a graph G , the remaining subgraph is the k -core of G .

If a graph G has a clique C of size r , then G must have a core of order greater than or equal to $r - 1$, because $\deg_{G(C)}(v) = r - 1$ for each $v \in C$. So, $k(G) \geq \omega(G) - 1$; i.e., $\omega(G) \leq k(G) + 1$. Similarly, if a vertex v is in a clique C of size r , then $k(v) \geq r - 1$. In other words, if we want to find a clique of size greater than r , we just need to consider the vertices v such that $k(v) \geq r$. The vertices whose core number is less than r can be discarded, because they cannot be in any clique of size larger than r .

Procedure *Initialize*, showed in Algorithm 1, performs the reduction of the input graph G based on the core number of each vertex of G . It removes all the vertices whose core number is less than a lower bound lb of $\omega(G)$. The procedure should be called by a BnB algorithm searching for a clique of size larger than lb . At the same time, it also derives an initial clique C_0 and determines an initial vertex ordering O_0 for the subsequent search. In the pseudo-code, cur_core denotes the order of the current core, max_core denotes the core number of graph G , and $core_number[v_i]$ denotes the core number of vertex v_i .

To compute the core number of each vertex of G , procedure *Initialize* first sorts all vertices of V in increasing degree ordering (i.e., $\deg(v_1) \leq \deg(v_2) \leq \dots \leq \deg(v_n)$), and assigns $\deg(v_1)$ to cur_core . G is a core of order cur_core , because $\deg(v_i) \geq cur_core$ for each $v_i \in V$ and G is the largest subgraph with this property. Moreover, cur_core is the core number of the smallest vertex v_1 , because it is the greatest order of a core that contains v_1 . Afterwards, the procedure considers the set $V \setminus \{v_1\}$: updating the degree of the vertices adjacent to v_1 (line 12), moving them for keeping the increasing degree ordering (line 13). If the degree of v_2 is greater than cur_core , then the subgraph induced by $V \setminus \{v_1\}$ is a new core of order greater than cur_core . In this case, cur_core is updated with the new order. Otherwise, v_2 still belongs to a core of order cur_core . In this way, the core number of all the vertices of G can be computed successively (line 6).

Note that if the set of vertices $\{v_i, v_{i+1}, \dots, v_{|V|}\}$ is in the increasing ordering of their degree in the subgraph induced by $\{v_i, v_{i+1}, \dots, v_{|V|}\}$ and the smallest v_i is adjacent to the other vertices

(i.e., $\deg(v_i) = |V| - i$), then all pairs of vertices in $\{v_i, v_{i+1}, \dots, v_{|V|}\}$ must be adjacent. So, $\{v_i, v_{i+1}, \dots, v_{|V|}\}$ forms a clique in this case. As soon as a vertex v_i with such a property is found, the algorithm ends the loop and the core number of all the vertices greater than or equal to v_i is set to cur_core (line 7-10), because they cannot belong to a core of order greater than cur_core . As a result, $\{v_i, v_{i+1}, \dots, v_{|V|}\}$ is the initial clique C_0 of G (line 14). If $|C_0|$ is greater than lb , lb is set to $|C_0|$. Finally, the vertices with core number less than lb are removed from G , because they cannot be in any clique larger than lb .

Algorithm 1: Initialize(G, lb), a preprocessing for large sparse graphs

Input: $G=(V, E)$, a lower bound lb of $\omega(G)$

Output: an initial clique C_0 , the core number of G , a reduced graph G' of G , and an initial vertex ordering O_0

```

1 begin
2   Sort  $V$  in increasing degree ordering;
3    $cur\_core \leftarrow \deg(v_1)$ ;
4   for  $i := 1$  to  $|V|$  do
5     if  $\deg(v_i) > cur\_core$  then  $cur\_core \leftarrow \deg(v_i)$ ;
6      $core\_number[v_i] \leftarrow cur\_core$ ;
7     if  $\deg(v_i) = |V| - i$  then
8       for  $j := i + 1$  to  $|V|$  do
9          $core\_number[v_j] \leftarrow cur\_core$ ;
10      break;
11     for each neighbor  $v$  of  $v_i$  in  $\{v_{i+1}, v_{i+2}, \dots, v_{|V|}\}$  do
12        $\deg(v) \leftarrow \deg(v) - 1$ ;
13       Move  $v$  and re-index vertices  $\{v_{i+1}, v_{i+2}, \dots, v_{|V|}\}$ 
        for keeping the increasing degree ordering;
14    $C_0 \leftarrow \{v_i, v_{i+1}, \dots, v_{|V|}\}$ ;
15    $max\_core \leftarrow$  the maximum core number in vertices of  $V$ ;
16   if  $|C_0| > lb$  then  $lb \leftarrow |C_0|$ ;
17    $G' \leftarrow$  reduced  $G$  by removing all vertices with core number
        less than  $lb$ ;
18    $O_0 \leftarrow$  the ordering in which the core number of each vertex
        is computed;
19   return  $(C_0, max\_core, G', O_0)$ ;

```

PMC and BBMCSP also use core numbers in their preprocessing to reduce the input graph. Algorithm 1 differs from the preprocessing of PMC and BBMCSP in that Algorithm 1 performs the following three tasks at the same time: derive the vertex ordering for the subsequent search, find an initial clique, and reduce the graph G by computing the core number of each vertex. However, PMC and BBMCSP perform the three tasks separately. In fact, PMC and BBMCSP use the original algorithm proposed in [4] to compute the core number of each vertex. After computing the core number of a vertex v , it only updates the degree of the vertices adjacent to v and with degree greater than $\deg(v)$. In Algorithm 1, the degrees of all uncomputed vertices adjacent to v are updated. So, the vertex ordering that Algorithm 1 uses to compute the core numbers (i.e., v_1 is the vertex with the smallest degree in G , v_2 is the vertex with the smallest degree in G after removing v_1 , and so on) is exactly the *degeneracy ordering*, which is a typical initial vertex ordering proposed in [8] and used in many BnB MaxClique algorithms. Algorithm 1 returns this ordering as the initial ordering O_0 for the subsequent search. Furthermore, Algorithm 1 naturally derives the initial clique C_0 in line 14, while PMC uses a heuristic [25] to derive C_0 in a separate sub-procedure

that can roughly be stated as follows: Let C_0 be the largest clique found so far. For each vertex v in decreasing core number order, if the core number of v is greater than or equal to $|C_0|$, greedily form a clique with v and the neighbors of v . That heuristic has time complexity $O(|E| \cdot \Delta(G))$, where $\Delta(G)$ is the maximum degree in graph G . BBMCSP uses a similar heuristic to derive C_0 [27].

The complexity of Algorithm 1 is dominated by the computation of the core number of vertices. Observe that the cost to compute C_0 and O_0 is negligible. So, although *Initialize* performs more tasks than the original algorithm in [4], the complexity of Algorithm 1 remains $O(|E|)$. Note that the complexity of the preprocessing in PMC and BBMCSP is dominated by the computation of C_0 , which is in $O(|E| \cdot \Delta(G))$.

2.2 The BnB algorithm for MaxClique with incremental MaxSAT reasoning

The graph G preprocessed by procedure *Initialize* will be solved by a BnB MaxClique algorithm called *SearchMaxClique*, which is presented in this section. In the first subsection, we describe the branch and bound scheme in *SearchMaxClique*. In the second subsection, we present incremental MaxSAT reasoning for *SearchMaxClique* to reduce the search space.

2.2.1 The BnB scheme in SearchMaxClique

SearchMaxClique is an improved variant of algorithm DoMC [17]. Given a graph G , whose vertices are ordered following a given ordering O , *SearchMaxClique* searches recursively for a clique larger than the current maximum clique C_{max} , combined with the growing clique C .

Algorithm 2: *SearchMaxClique*(G, C_{max}, C, O), an algorithm for finding a clique of size larger than $|C_{max}|$

Input: $G=(V, E)$, the largest clique C_{max} found so far, the current growing clique C , a vertex ordering O
Output: a clique C , if $|C| > |C_{max}|$, otherwise, C_{max}

```

1 begin
2   if  $|V|=0$  then return  $C$ ;
3    $B \leftarrow \text{GetBranches}(G, |C_{max}|-|C|, O)$ ;
4   if  $B=\emptyset$  then return  $C_{max}$ ;
5    $A \leftarrow V \setminus B$ ;
6   Let  $B=\{b_1, b_2, \dots, b_{|B|}\}$  in the increasing ordering w.r.t.  $O$ ;
7   for  $i := |B|$  to 1 do
8      $P \leftarrow \Gamma(b_i) \cap (\{b_{i+1}, b_{i+2}, \dots, b_{|B|}\} \cup A)$ ;
9      $C' \leftarrow \text{SearchMaxClique}(G(P), C_{max}, C \cup \{b_i\}, O)$ ;
10    if  $|C'| > |C_{max}|$  then  $C_{max} \leftarrow C'$ ;
11  return  $C_{max}$ ;

```

Algorithm 2 shows the pseudo-code of *SearchMaxClique*. If the set of vertices V is non-empty, it calls function *GetBranches* to partition V into two sets A and B , respecting the vertex ordering O , in such a way that the size of a maximum clique in A is not greater than $|C_{max}| - |C|$, and $B=V \setminus A=\{b_1, b_2, \dots, b_{|B|}\}$ is called the set of branching vertices. If B is empty, the search is pruned and it returns the current largest clique C_{max} . Otherwise, the algorithm searches recursively for a maximum clique, containing $b_i \in B$ and of size greater than $|C_{max}|$, in the subgraphs induced by $\Gamma(b_i) \cap (\{b_{i+1}, b_{i+2}, \dots, b_{|B|}\} \cup A)$ for $i = |B|, \dots, 1$, where $\Gamma(b_i)$ denotes the set of vertices adjacent to b_i in G and $b_1 < b_2 < \dots < b_{|B|}$ w.r.t. O .

Algorithm 3: *GetBranches*(G, r, O), for an algorithm searching for a maximum clique with more than r vertices in G .

Input: $G=(V, E)$, an integer r and a vertex ordering O over V
Output: a set B of branching vertices

```

1 begin
2    $B \leftarrow \emptyset$ ;  $\Pi \leftarrow \emptyset$ ; /*  $\Pi$  is a set of Independent Sets (IS) */
3   while  $V$  is not empty do
4      $v \leftarrow$  the biggest vertex of  $V$  w.r.t.  $O$ ;
5     remove  $v$  from  $V$ ;
6     if there is an IS  $D$  in  $\Pi$  in which  $v$  is not adjacent to any vertex then
7        $\text{insert } v$  into  $D$ ;
8     else if  $|\Pi| < r$  then
9       create a new IS  $D = \{v\}$ ;  $\Pi \leftarrow \Pi \cup \{D\}$ ;
10      else
11        if There is an IS  $D_1$  in which  $v$  has only one adjacent vertex  $u$ , and  $u$  can be inserted into another IS  $D_2$  then
12           $\text{insert } u$  into  $D_2$ ;  $\text{insert } v$  into  $D_1$ ;
13          else  $B \leftarrow B \cup \{v\}$ ;
14       $B \leftarrow \text{IncMaxSAT}(G, O, V \setminus B, B)$ ;
15  return  $B$ ;

```

Function *GetBranches*(G, r, O) is described in Algorithm 3, where r is an integer and O is a vertex ordering. *GetBranches* returns a set B of branching vertices by showing that $A=V \setminus B$ does not contain any clique of size greater than r . The set B is derived by using a greedy sequential coloring process that successively assigns the smallest possible color to each vertex in V w.r.t. ordering O . Note that each color is represented by a natural integer from 1, and that all the vertices with the same color form an independent set (IS) in which no vertex is adjacent to another. The vertices that cannot be assigned a color smaller than or equal to r form the set B . Since vertices in $A=V \setminus B$ can be colored using r colors, they cannot form a clique of size larger than r .

Since the greater the cardinality of B , the greater the remaining search space to be explored, *GetBranches* reduces B using procedure *Re-NUMBER* [32]. It works as follows: when the coloring process fails to color a vertex v with a color less than or equal to r , it checks whether there exists a vertex u and two ISs D_1 and D_2 , such that u is the only vertex adjacent to v in D_1 but there is no vertex adjacent to u in D_2 . In this case, u is moved to D_2 and v is inserted into D_1 , obtaining this way one additional vertex in A and one less in B .

Finally, *GetBranches* further reduces B by applying incremental MaxSAT reasoning [17], which is described in the next subsection.

2.2.2 Incremental MaxSAT reasoning

We first explain the rationale to use MaxSAT reasoning in BnB MaxClique algorithms. Then, we give basic notions of MaxSAT and describe standard MaxSAT reasoning for MaxClique. Finally, we describe incremental MaxSAT reasoning.

BnB MaxClique algorithms compute upper bounds (UBs) of $\omega(G)$ to prune search [22, 31, 32, 20, 19, 29], using approximate algorithms such as greedy sequential coloring to derive good quality UBs with low overhead. Given a graph $G = (V, E)$, greedy sequential coloring successively assigns the smallest possible color (from 1) to each vertex v in a predefined ordering, and the computed UB of $\omega(G)$ is

the greatest color r needed to color G . Let $D_i = \{v \mid v \in V \text{ and } v \text{ is assigned color } i\}$ for $i=1, \dots, r$. D_i is an IS, and the vertex coloring process partitions V into r ISs. In Figure 1, assuming that a coloring process colors the vertices in V in the ordering $v_1, v_2, v_3, v_4, v_5, v_6$, then three ISs $D_1 = \{v_1, v_4, v_6\}$, $D_2 = \{v_2, v_3\}$ and $D_3 = \{v_5\}$ are obtained.

The coloring process has a low time complexity $O(n^2)$. Nevertheless, the derived UB may not be tight enough; e.g., in Figure 1, $UB=3$ and $\omega(G)=2$. A subset of q ISs is said to be *conflicting* if the q ISs cannot form a clique of size q . Recent approaches [20, 19] use *standard MaxSAT reasoning* to improve the coloring-based UB by detecting disjoint conflicting subsets of ISs, after encoding MaxClique to MaxSAT. They apply the following proposition [20]: Let G be a graph that can be partitioned into r ISs. If the r ISs can be partitioned into c disjoint conflicting subsets of ISs, then $\omega(G) \leq r - c$.

Recall that a literal is a propositional variable x or its negation \bar{x} , and a clause is a disjunction of literals. A clause is unit if it contains exactly one literal. A partial MaxSAT (PMaxSAT) instance is a multiset of clauses in which some clauses are declared to be hard and the others are declared to be soft. Given a PMaxSAT instance, the PMaxSAT problem is to find an assignment that satisfies all hard clauses and the maximum number of soft clauses [18].

After G is partitioned into ISs, MaxClique can be reduced to PMaxSAT as follows [20]: Given a graph $G = (V, E)$, we define a propositional variable x_i for each $v_i \in V$ with the intended meaning that x_i is true iff v_i belongs to the maximum clique C_{max} , and derive the PMaxSAT instance ϕ that contains (i) a hard clause $\bar{x}_i \vee \bar{x}_j$ for each pair of non-adjacent vertices v_i and v_j , stating that v_i and v_j cannot be both in C_{max} , and (ii) a soft clause for each IS in the partition, which is the disjunction of the variables associated to the vertices in the IS. Notice that at most one vertex in each IS can be in C_{max} . So, at most one variable can be assigned true in each soft clause. An assignment that satisfies all hard clauses and the maximum number of soft clauses in ϕ identifies a maximum clique in G (containing one vertex per satisfied soft clause).

Nevertheless, the purpose of encoding G as a PMaxSAT instance ϕ in [20] is not to solve ϕ with a MaxSAT solver, but to detect conflicts in ϕ with an approximate PMaxSAT algorithm to improve the coloring-based UB of $\omega(G)$.

Example 1 (from [20]) The vertices of the graph G in Figure 1 can be partitioned into three ISs: $\{v_1, v_4, v_6\}$, $\{v_2, v_3\}$, $\{v_5\}$. The PMaxSAT encoding ϕ is formed by the hard clauses: $\{\bar{x}_1 \vee \bar{x}_4, \bar{x}_1 \vee \bar{x}_5, \bar{x}_1 \vee \bar{x}_6, \bar{x}_2 \vee \bar{x}_3, \bar{x}_2 \vee \bar{x}_5, \bar{x}_2 \vee \bar{x}_6, \bar{x}_3 \vee \bar{x}_4, \bar{x}_4 \vee \bar{x}_6, \bar{x}_5 \vee \bar{x}_6\}$, and the soft clauses: $\{x_1 \vee x_4 \vee x_6, x_2 \vee x_3, x_5\}$. Standard MaxSAT Reasoning detects a conflict as follows. Assume that x_5 is true (i.e., v_5 is in C_{max}). Literal \bar{x}_5 must be removed from the hard clauses $\bar{x}_1 \vee \bar{x}_5, \bar{x}_2 \vee \bar{x}_5$, and $\bar{x}_5 \vee \bar{x}_6$, resulting in three hard unit clauses: \bar{x}_1, \bar{x}_2 , and \bar{x}_6 . The satisfaction of these three unit clauses removes x_1, x_2 and x_6 from the soft clauses $x_1 \vee x_4 \vee x_6$ and $x_2 \vee x_3$, and results in two soft unit clauses: x_4 and x_3 . The satisfaction of these two new soft unit clauses makes the hard clause $\bar{x}_3 \vee \bar{x}_4$ falsified. Hence, the three soft clauses $x_1 \vee x_4 \vee x_6, x_2 \vee x_3$ and x_5 cannot be satisfied simultaneously, meaning that the three corresponding ISs cannot form a clique of size 3. Consequently, the coloring-based UB of $\omega(G)$ is improved from 3 to 2.

The improvement of UB depends on the number of disjoint conflicts detected. Standard MaxSAT reasoning has no impact if the improved UB is still greater than the size of the largest clique found so far. It is easy to see that the greater the average cardinality of ISs,

the harder to detect conflicts. This fact might explain why standard MaxSAT reasoning in sparse graphs is not as useful as in medium and dense graphs, because the ISs in sparse graphs are generally very large.

Algorithm 4: IncMaxSAT(G, O, A, B), incremental MaxSAT reasoning to reduce the set of branching vertices B

Input: $G=(V, E)$, V is ordered w.r.t. O and is partitioned into two subsets: A and B , and A is partitioned into ISs

Output: a set of branching vertices B

```

1 begin
2    $\phi \leftarrow$  the PMaxSAT instance: hard clauses encoded from  $G$ 
   and soft clause according to the IS partition of  $A$ ;
3   while  $B$  is not empty do
4      $v \leftarrow$  the biggest vertex in  $B$  w.r.t.  $O$ ;
5     Add the soft unit clause  $\{v\}$  in  $\phi$ ;
6     if a conflict can be detected in  $\phi$  then
7       Let  $c_1, c_2, \dots, c_p$  be the conflicting soft clauses;
8       Remove  $c_1, c_2, \dots, c_p$  from  $\phi$ ;
9       Add the soft clauses  $c_1 \vee z_1, c_2 \vee z_2, \dots, c_p \vee z_p$  to  $\phi$ ;
10      Add the constraint  $z_1 + z_2 + \dots + z_p = 1$  to  $\phi$ ;
11      Remove  $v$  from  $B$ ;
12    else break;
13  return  $B$ ;
```

To remedy that drawback of standard MaxSAT reasoning, *incremental MaxSAT reasoning* was proposed in [17], resulting in two efficient algorithms: DoMC and SoMC. These algorithms partition the set of vertices V into two sets, A and B , in such a way that the vertices in A are colored with $|C_{max}|$ colors, and $B = V \setminus A = \{b_1, b_2, \dots, b_{|B|}\}$ is the set of branching vertices. If B is empty, the search is pruned. Otherwise, the algorithms get the PMaxSAT instance ϕ : the hard clauses encode the non-adjacent vertices of G , and there is a soft clause for each IS of the partition of A . The MaxSAT encoding is implicit. The propositional variables are directly represented by the vertices, the soft clauses are represented by the corresponding ISs, and the hard clauses are represented using the adjacency matrix. In addition, each vertex is associated with the list of the non-adjacent vertices. In this way, no extra space is needed, and the cost of the encoding is negligible once A and B are obtained.

Then, incremental MaxSAT reasoning successively adds to ϕ the highest vertex b_i of B as a soft unit clause $\{b_i\}$ for $i=|B|$ to 1, respecting the ordering O . If a new conflict is detected in ϕ after adding $\{b_i\}$, then b_i is removed from B and added to A . Let c_1, c_2, \dots, c_p be the soft clauses involved in the detected conflict. Note that at least one of these soft clauses is falsified by each truth assignment that satisfies all the hard clauses. These soft clauses are weakened before detecting the next conflict: A fresh propositional variable z_i is added to soft clause c_i for $i=1$ to p , and a hard constraint $z_1 + z_2 + \dots + z_p = 1$ is added to require that exactly one of these variables is assigned true. The weakened soft clauses can then be used to detect further conflicts. The constraint $z_1 + z_2 + \dots + z_p = 1$ is treated as follows: if one variable z_i is assigned true, the other variables are assigned false. Adding the fresh variables and the hard constraint allows one to satisfy exactly one soft clause that was previously falsified by each assignment.

If $b_{|B|}, b_{|B|-1}, \dots$, and b_i are removed from B and added to A , and a conflict is detected for each one of these vertices, then the set $A \cup \{b_{|B|}, b_{|B|-1}, \dots, b_i\}$ cannot form a clique of size greater than

$|C_{max}|$. To see this, note that the PMaxSAT instance ϕ contains now $|C_{max}|+|B|-i+1$ soft clauses and, for each truth assignment satisfying all the hard clauses, $|B|-i+1$ soft clauses have to be satisfied because of the fresh variables. So, at most $|C_{max}|$ soft clauses can be satisfied by the vertices.

If a conflict cannot be detected when adding a soft unit clause $\{b_i\}$ to ϕ , the reduced set $B=\{b_1, b_2, \dots, b_i\}$ is returned. Algorithm 4 shows the pseudo-code of incremental MaxSAT reasoning.

Example 2 (adapted from [17]). Refer to the graph G in Figure 1. Assume that $|C_{max}|=2$, and a coloring process partitions the graph into $A=\{D_1, D_2\}$ and $B=\{v_5, v_6\}$, where $D_1=\{v_1, v_4\}$ and $D_2=\{v_2, v_3\}$. *IncMaxSAT* encodes the graph into a PMaxSAT instance ϕ by directly treating the vertices as Boolean variables, the ISs D_1 and D_2 as soft clauses, and the non-adjacency relations between vertices as hard clauses. Then, it adds a new soft unit clause $\{v_5\}$ to ϕ , and proves that $\{D_1, D_2, \{v_5\}\}$ cannot form a clique of size 3 as follows. If v_5 is in the clique (i.e., v_5 is assigned true), then v_1 and v_2 cannot be in the clique (i.e., v_1 and v_2 should be assigned false), because they are non-adjacent to v_5 . So, the only remaining vertices v_4 in D_1 and v_3 in D_2 should be in the clique, which is impossible because v_4 and v_3 are non-adjacent (if so, the hard clause $\bar{v}_3 \vee \bar{v}_4$ would be falsified). So the three soft clauses D_1, D_2 and $\{v_5\}$ are conflicting.

Then, *IncMaxSAT* adds a fresh variable z_1 (z_2, z_3) to D_1 ($D_2, \{v_5\}$) together with the hard constraint $z_1+z_2+z_3=1$, before adding the soft unit clause $\{v_6\}$ to ϕ . It proves that the three soft clauses $D_1=\{v_1, v_4, z_1\}, \{v_5, z_3\}$ and $\{v_6\}$ are conflicting as follows. Assume that v_6 is true. Then, v_1, v_4 and v_5 should be false, because they are not adjacent to v_6 . However, z_1 and z_3 cannot both be true because of the hard constraint $z_1+z_2+z_3=1$. A fresh variable z_4 (z_5, z_6) is then added to D_1 ($\{v_5, z_3\}$ and $\{v_6\}$) together with the hard constraint $z_4+z_5+z_6=1$.

The two conflicts are clearly disjoint. Recall that ϕ contains now four soft clauses. Given a truth assignment satisfying all the hard clauses, at most two soft clauses can be satisfied due to the original variables v_i ($1 \leq i \leq 6$), and the other two soft clauses are satisfied due to the fresh variables z_i ($1 \leq i \leq 6$). This shows that $A \cup \{v_5, v_6\}$ cannot form a clique of size greater than 2.

The advantage of incremental MaxSAT reasoning over standard MaxSAT reasoning for MaxClique is that if it eliminates all the branching vertices, then the search is pruned; otherwise, the set of branching vertices is generally significantly reduced. We note that although large real-world graphs are usually sparse, they might contain cliques with hundreds of vertices. So, BnB MaxClique algorithms will develop wider and deeper search trees for such graphs. The reduction of the set of branching vertices B by incremental MaxSAT reasoning has a dramatic impact on performance when hard large sparse graphs are solved, as we will see in Section 3.

2.3 Algorithm LMC for large sparse graphs

Algorithm 5 describes LMC, which is especially suited for large sparse graphs. Roughly speaking, given a graph G , LMC calls procedure *Initialize* to preprocess both G and the first level subgraphs in the search tree, and then calls the search procedure *SearchMaxClique* to recursively search for a maximum clique in the reduced subgraphs.

LMC first calls *Initialize*($G, 0$) (the initial *lb* of $\omega(G)$ is 0) to derive an initial clique C_0 , the core number of G and of each vertex, a reduced subgraph G' and an initial ordering O_0 . If the size of C_0 is $k(G) + 1$, then C_0 is a maximum clique of G and is returned (line

Algorithm 5: LMC(G), a BnB algorithm for MaxClique in large sparse graphs

Input: $G=(V, E)$

Output: a maximum clique C_{max} of G

```

1 begin
2    $(C_0, k(G), G', O_0) \leftarrow Initialize(G, 0)$ ;
3   if  $|C_0| = k(G) + 1$  then return  $C_0$ ;
4    $C_{max} \leftarrow C_0$ ;
5    $V' \leftarrow$  the vertex set of  $G'$ ;
6   Order  $V'$  w.r.t the initial ordering  $O_0$ ;
7   for  $i := |V'|$  to 1 do
8      $P \leftarrow \Gamma(v_i) \cap \{v_{i+1}, v_{i+2}, \dots, v_{|V'|}\}$ ;
9      $(C'_0, k(G(P)), G'', O'_0) \leftarrow$ 
10       $Initialize(G(P), |C_{max}| - 1)$ ;
11     if  $|C'_0| \geq |C_{max}|$  then  $C_{max} \leftarrow C'_0 \cup \{v_i\}$ ;
12     if  $k(G(P)) + 1 \geq |C_{max}|$  then
13       Construct the adjacency matrix for  $G''$ ;
14        $C' \leftarrow SearchMaxClique(G'', C_{max}, \{v_i\}, O'_0)$ ;
15       if  $|C'| > |C_{max}|$  then  $C_{max} \leftarrow C'$ ;
16   return  $C_{max}$ 

```

3), because $k(G) + 1$ is an UB of $\omega(G)$. Otherwise, LMC unrolls the first level subgraphs induced by the set of candidates $\Gamma(v_i) \cap \{v_{i+1}, \dots, v_{|V'|}\}$, denoted by P , for $i = |V'|$ to 1, where vertices follow the initial ordering O_0 . For each first level subgraph $G(P)$ of the search tree, LMC calls *Initialize*($G(P), |C_{max}| - 1$) to compute an initial clique C'_0 of $G(P)$, the core number of $G(P)$ and of each vertex in $G(P)$, a subgraph G'' of $G(P)$ obtained by removing all the vertices whose core number is less than $|C_{max}| - 1$, and a vertex ordering O'_0 . Finally, the search procedure *SearchMaxClique* is called to recursively search for a clique containing v_i , of size greater than $|C_{max}|$, in the subgraph G'' . Observe that the size of a maximum clique in $G(P)$ is at most $k(G(P)) + 1$. When $k(G(P)) + 1 < |C_{max}|$, a clique containing v_i of size greater than $|C_{max}|$ cannot be found from G'' and the search in G'' is pruned.

LMC also calls *Initialize* for the first level subgraphs. The rationale is: (i) the vertex ordering computed by *Initialize* is *degeneracy ordering*; re-ordering the vertices in the subgraphs near the root of the search tree was proven to be beneficial for BnB MaxClique algorithms [15]. (ii) Since G is large, the first level subgraphs may still contain a lot of vertices. With a growing lower bound $|C_{max}|$ of $\omega(G)$, the first level subgraphs can be further reduced, which is beneficial for speeding up the search in *SearchMaxClique*. We note that the first level subgraphs are also preprocessed in BBMCS.

Maintaining an adjacency matrix for large sparse graphs has a costly space complexity. LMC does not construct a global adjacency matrix for the input graph G . When the search in the first level subgraphs is necessary, the adjacency matrix for them is constructed dynamically to serve *SearchMaxClique* (line 13). Since G is sparse, the number of vertices in the first level subgraphs should be substantially reduced by the preprocessing in line 9 and line 10. So, constructing an adjacency matrix for the reduced first level subgraphs is feasible and beneficial. In the implementation, we use bit-sets to store the adjacency matrix.

3 EMPIRICAL INVESTIGATION

We empirically evaluated LMC, and compared it with PMC and BBMCS, which are, to our best knowledge, the two most efficient

Table 1. The graphs tested in the experiments, excluding those graphs whose *init* and *search* times are below 10s for all solvers.

Graph	$ V $	$ E $	Graph	$ V $	$ E $	Graph	$ V $	$ E $
adaptive	6815744	13624320	inf-great-britain.osm	7733822	8156517	soc-ljournal-2008	5363186	49514271
aff-digg	872622	22501700	inf-road-usa	23947347	28854312	soc-orkut	2997166	106349209
aff-flickr-user-groups	395979	8537703	packing-500x100	2145852	17488243	soc-orkut-dir	3072441	117185083
aff-orkut-user2groups	8730857	327036486	x100-b050	2146057	5743132	soc-pokec	1632803	22301964
bio-human-gene1	22283	12323680	rec-amazon-ratings	168792	17351416	soc-sinaweibo	58655849	261321033
bio-human-gene2	14340	9027024	rec-dating	74424	2811458	soc-twitter-higgs	456631	12508442
bio-mouse-gene	45101	14461095	rec-eachmovie	755761	13396042	soc-wiki-conflict	118100	2027871
bn-human-BNU_1.0	1827218	143158339	rec-epinions	220970	17233144	soc-youtube-growth	3223589	9376594
025864_session_1-bg	1827241	133727516	rec-libimseti-dir	71567	9991339	socfb-A-anon	3097165	23667394
bn-human-BNU_1.0	1398408	42296922	rec-movieLens	8388608	63501393	socfb-B-anon	2937612	20959854
025864_session_2-bg	1717207	22855526	rgg_n_2_23_s0	16777216	132557200	socfb-konect	59216214	92522012
025865_session_1-bg	4802000	42681372	rgg_n_2_24_s0	38120	16115324	socfb-uci-uni	58790782	92208195
bn-human-BNU_1.0	11621692	78621046	sc-TSOPF-RS	952203	20770807	tech-as-skitter	1694616	11094209
025865_session_2-bg	4194304	12582869	-b2383-c1	5921786	23667162	tech-ip	2250498	21643497
channel-500x100	8388608	25165784	sc-ldoor	88784	2093195	tech-p2p	5792297	147829887
x100-b050	16777216	50331601	sc-reJ9	639014	3214986	twitter_mpi	9862152	99940317
dbpedia-link	8658744	45671471	soc-BlogCatalog	4847571	42851237	web-ClueWeb09-50m	428136613	446534058
delaunay_n22	21198119	28857767	soc-FourSquare	101163	2763066	web-baidu-baike	2141300	17014946
delaunay_n23	4588484	6879133	soc-LiveJournal1	149700	5448197	web-indochina-2004-all	7414865	150984819
delaunay_n24	12057441	18082179	soc-buznet	770799	5907132	web-it-2004-all	41291318	1027474947
friendster	16002413	23998813	soc-catster	426820	8543549	web-uk-2002-all	18520343	261787258
hugebubbles-00020	87273	297456	soc-digg	1715255	15555041	web-wiki-ch-internal	1930275	8956902
hugebubbles-00000	2394385	4659565	soc-dogster	2523386	7918801	web-wiki-en-growth	1870709	36532531
hugebubbles-00010	2104544	5572584	soc-flickr-und	65608366	1806067135	web-wikipedia-link.en	27154756	31024475
hugebubbles-00020	50912018	54054660	soc-flixster	7489073	112305407	web-wikipedia-link.it	2936413	86754664
ia-enron-email-dynamic	11548845	12369181	soc-friendster	5204176	48709773			
ia-wiki-Talk-dir			soc-livejournal					
ia-wiki-user-edits-page			-user-groups					
inf-europe.osm			soc-livejournal07					
inf-germany.osm								

exact MaxClique algorithms for large sparse graphs. LMC was implemented in C and compiled using GNU gcc -O3. The algorithms are also called solvers when they are used to solve MaxClique instances. The experiments were performed on an Intel Xeon CPU X5460@3.16GHz under Linux with 32GB of memory.

We next describe the compared solvers and the tested graphs, and then discuss and analyze the experimental results.

3.1 The compared solvers and the tested graphs

The source code of PMC [25] is publicly available at <https://www.cs.purdue.edu/homes/dgleich/codes/maxcliques/>. We compiled it using the provided Makefile and ran it with `./pmc -f G -a 0` to solve G . We used the Linux binary executable of BBMCSP [27] available at http://venus.elai.upm.es/logs/results_sparse/bin/.

The measures considered for each tested graph are the following:

The size of initial clique C_0 (ω_0). PMC and BBMCSP use a dedicated heuristic to compute C_0 , while LMC derives C_0 in procedure *Initialize* as a by-product of computing core numbers.

The time for preprocessing (*init*). For PMC and BBMCSP, it includes the time of the k -core analysis, finding C_0 , and reducing the graph. For LMC, it is the time needed by procedure *Initialize* to preprocess the input graph at the root of the search tree. It does not include the preprocessing of the first level subgraphs.

The time for search (*search*). The runtime after finishing the preprocessing. For LMC, it includes both the time of search and preprocessing of first level subgraphs. The cut-off time was set to 5 hours.

We considered 170 real-world graphs from the Network Data Repository [26] available at <http://networkrepository.com>, including the 90 graphs reported to evaluate BBMCSP in [27]. The number of vertices ranges from 4K to 400M. We exclude the graphs whose *init* and *search* times are below 10s for all the solvers and report results for the remaining 77 graphs, providing a clearer comparison. Table 1

shows the number of vertices ($|V|$) and number of edges ($|E|$) of these 77 graphs.

3.2 Comparison of LMC with PMC and BBMCSP

Table 2 shows the experimental results, where $k(G)+1$ is the UB of $\omega(G)$ derived by the k -core analysis. When the size of an initial clique (ω_0) is $k(G)+1$, the search time is 0 in LMC and BBMCSP because no search is performed. PMC does not return *search* = 0 in some cases for some unknown reason. The best times are in bold.

LMC solves all graphs in at most 1120s, while PMC (BBMCSP) cannot solve 12 (6) instances in 5h. In addition, BBMCSP runs out of memory on two graphs. Furthermore, LMC is almost always faster than BBMCSP and PMC. For example, for *bio-mouse-gene*, the total time (*init+search*) of LMC is 147s, which is 4.4 and 35.5 times faster than PMC (643s) and BBMCSP (5220s); for *soc-sinaweibo*, the total time of LMC is 80s, which is 59 and 40 times faster than PMC (4742s) and BBMCSP (3206s), respectively.

Observe that the *init* times of LMC are significantly below the ones of PMC and BBMCSP due to the overhead of the dedicated heuristic for finding the initial clique in PMC and BBMCSP. Surprisingly, although the method for finding an initial clique in LMC is very simple, it finds larger initial cliques than the ones found by PMC and BBMC for 14 and 18 instances, respectively. For example, the initial clique found by LMC for *tech-p2p* is of size 172, while the initial clique found by PMC and BBMCSP is of size 155 and 153, respectively.

Although the initial clique found by LMC is smaller than the one found by PMC and BBMCSP in many cases, the search after the preprocessing of LMC is almost always faster than PMC and BBMCSP. For example, the search time of LMC for *rec-epinions* is 123 and 20 times faster than PMC and BBMCSP, respectively; and LMC finishes the search in 150s for *twitter_mpi*, whereas PMC and BBMCSP cannot terminate in 5h. For these two graphs, the initial cliques found by LMC (resp. 2 and 79) are smaller than the ones found by PMC (resp. 7 and 113) and BBMCSP (resp. 7 and 121). This also happens on

Table 2. Runtimes in seconds of LMC, PMC and BBMCSP to solve the large graphs in Table 1. For each graph, the cpu time limit is 5h, ω_0 is the size of the initial clique found by the solvers, *init* denotes the time for preprocessing, and *search* denotes the time for search after the preprocessing.

Graph	$k(G)$	ω	LMC			PMC			BBMCSP		
			ω_0	init	search	ω_0	init	search	ω_0	init	search
adaptive	3	2	2	1.74	0.01	2	4.25	6.09	2	13.42	1.44
aff-digg	646	32	29	1.98	175.6	24	62.38	>5h	25	49.01	1397
aff-flickr-user-groups	187	14	12	0.85	3.62	10	9.43	53.06	10	15.79	22.42
aff-orkut-user2groups	472	6	2	83.06	436.5	6	932.4	>5h	5	2410	3727
bio-human-gene1	2048	1335	1328	0.65	1047	1272	90.48	>5h	1268	16.55	>5h
bio-human-gene2	1903	1300	1290	0.48	178.4	1241	66.54	>5h	1229	7.77	>5h
bio-mouse-gene	1046	561	435	1.29	145.8	525	33.06	610.4	520	24.35	5196
bn-human-BNU_1_0025864_session_1-bg	1210	294	222	14.72	642.8	271	177.8	>5h	276	277.3	>5h
bn-human-BNU_1_0025864_session_2-bg	1088	271	199	13.91	544.1	271	154.9	1595	271	252.4	912.8
bn-human-BNU_1_0025865_session_1-bg	912	196	159	3.90	192.9	172	56.16	12611	186	84.43	>5h
bn-human-BNU_1_0025865_session_2-bg	582	201	83	2.18	40.91	201	20.73	120.9	201	33.76	12.64
channel-500x100x100-b050	10	4	4	10.19	2.10	4	18.96	32.37	4	33.44	11.41
dbpedia-link	140	33	10	17.62	16.07	30	4050	3774	32	243.4	118.1
delaunay_n22	5	4	3	2.69	0.12	4	6.21	13.54	4	15.75	2.51
delaunay_n23	5	4	3	5.60	0.25	4	12.46	25.70	4	32.26	5.13
delaunay_n24	5	4	3	11.55	0.50	4	24.76	32.82	4	67.06	10.71
friendster	52	37	17	8.26	1.42	37	18.71	24.36	37	67.91	5.56
hugebubbles-00020	3	2	2	15.78	0.01	2	23.47	41.27	2	62.42	6.78
hugetrace-00000	3	2	2	2.24	0.01	2	3.52	4.91	2	10.04	1.06
hugetrace-00010	3	2	2	6.35	0.01	2	9.60	16.51	2	30.21	2.99
hugetrace-00020	3	2	2	10.14	0.01	2	16.19	27.23	2	47.16	4.30
ia-enron-email-dynamic	54	33	24	0.02	0.02	28	0.48	16.03	30	0.15	0.02
ia-wiki-Talk-dir	132	26	25	0.35	0.37	22	4.18	8.01	16	12.56	0.89
ia-wiki-user-edits-page	67	15	13	0.27	0.10	14	689.9	>5h	11	46.60	0.34
inf-europe_osm	4	4	3	9.34	0.01	4	7.93	0.00	4	12.83	0.00
inf-germany_osm	4	3	3	2.13	0.01	3	2.06	2.57	3	16.43	0.01
inf-great-britain_osm	4	3	3	1.32	0.01	3	1.47	1.78	3	11.10	0.01
inf-road-usa	4	4	3	7.93	0.01	3	0.01	5.96	4	10.13	0.00
packing-500x100x100-b050	10	4	4	3.17	0.94	4	8.16	11.96	4	15.21	5.75
rec-amazon-ratings	30	5	4	1.15	0.18	4	3.39	5.32	4	10.40	2.24
rec-dating	261	13	8	1.76	12.16	11	11.89	161.4	9	35.32	55.74
rec-eachmovie	221	12	11	0.24	0.81	11	9.68	49.98	10	2.87	1.51
rec-epinions	149	8	2	1.69	2.50	7	169.0	308.8	7	44.00	51.87
rec-libimseti-dir	274	14	13	1.79	10.47	12	14.97	159.3	11	38.46	45.53
rec-movieLens	532	29	24	0.96	17.72	23	18.08	541.8	20	14.05	127.0
rgg_n_2_23_s0	21	21	21	9.16	0.00	21	11.48	0.00	21	15.92	0.00
rgg_n_2_24_s0	21	21	21	19.43	0.00	21	25.13	0.00	21	33.80	0.00
sc-TSOPF-RS-b2383-c1	656	7	3	0.84	6.82	7	3.28	162.7	6	3.25	2.65
sc-ldoor	35	21	21	1.50	2.46	21	10.56	9.85	21	5.83	1.84
sc-rel9	5	4	3	2.03	0.19	4	6.53	21.14	4	28.76	4.56
soc-BlogCatalog	222	45	41	0.16	1.81	39	2.67	14.36	37	2.12	4.02
soc-FourSquare	64	30	25	0.19	0.22	29	36.82	36.53	27	7.09	0.39
soc-LiveJournal1	373	321	320	5.30	0.10	314	8.26	5.71	316	53.89	0.03
soc-buzznet	154	31	28	0.22	1.15	25	4.41	17.56	23	3.52	2.37
soc-catster	420	81	56	0.32	4.78	80	8.08	16631	58	6.20	1.35
soc-digg	237	50	18	0.57	2.56	46	2.37	13.32	42	9.35	2.43
soc-dogster	249	44	39	0.61	2.65	40	10.95	18.42	33	13.94	4.50
soc-flickr-und	569	98	74	1.49	24.32	77	11.50	1027	68	24.24	155.3
soc-flixster	69	31	30	0.60	0.18	29	1.46	1.84	29	10.28	0.91
soc-friendster	305	129	14	726.7	393.6	129	1253	7458	out of memory		
soc-livejournal-user-groups	117	9	4	19.83	42.48	8	7360	>5h	8	1351	2064
soc-livejournal07	375	358	358	6.32	0.05	358	7.21	1.35	356	62.03	0.01
soc-ljournal-2008	426	400	389	5.50	0.04	400	9.16	2.50	400	47.18	0.01
soc-orkut	231	47	17	23.39	29.25	43	72.84	268.5	46	190.8	85.72
soc-orkut-dir	254	51	14	26.01	34.08	48	79.94	291.3	50	209.2	96.67
soc-pokec	48	29	15	4.28	1.90	29	8.30	8.36	29	34.53	6.68
soc-sinaweibo	194	44	8	53.14	26.83	37	3493	1249	41	3012	194.8
soc-twitter-higgs	126	71	21	1.35	1.71	71	15.13	28.28	70	29.11	4.43
soc-wiki-conflict	146	25	22	0.17	0.46	21	7.48	62.82	22	1.75	0.77
soc-youtube-growth	89	20	18	1.44	0.39	17	51.63	33.97	18	32.41	2.01
socfb-A-anon	75	25	23	3.75	5.02	23	12.43	22.21	24	40.41	15.48
socfb-B-anon	64	24	11	3.64	4.55	24	10.44	19.41	24	35.46	14.10
socfb-konect	17	6	6	17.94	0.15	6	26.20	20.26	6	131.8	1.72
socfb-uci-uni	17	6	6	20.01	0.16	6	30.28	20.69	6	164.5	1.75
tech-as-skitter	112	67	57	0.87	0.10	66	1.55	0.87	50	11.09	0.14
tech-ip	254	4	3	2.22	7.47	3	67.79	>5h	4	103.9	24.76
tech-p2p	854	178	172	26.21	208.8	155	229.8	>5h	153	564.2	>5h
twitter_mpi	678	131	79	15.68	149.5	113	1456	>5h	121	537.8	>5h
web-ClueWeb09-50m	189	56	41	130.9	2.40	55	411.8	258.2	out of memory		
web-baidu-baike	79	31	12	2.84	1.11	31	38.13	30.62	31	46.77	8.46
web-indochina-2004-all	6870	6848	6848	5.61	82.38	6848	3621	162.5	6848	86.62	0.85
web-it-2004-all	3225	3222	3222	47.20	2.99	3222	682.1	>5h	3222	451.8	0.45
web-uk-2002-all	944	944	944	16.40	0.00	944	35.30	9.71	944	21.29	0.00
web-wiki-ch-internal	121	33	13	1.17	0.52	32	13.67	10.53	33	14.90	2.17
web-wikipedia-growth	207	31	15	6.26	6.41	31	354.9	273.5	31	131.9	52.22
web-wikipedia.link_en	378	44	4	6.99	2.09	43	24.79	93.24	44	57.70	13.80
web-wikipedia.link_it	895	870	869	5.67	0.25	870	152.1	>5h	869	85.28	0.06

other graphs such as *dbpedia-link*, *friendster*, *soc-livejournal-user-groups* and *soc-orkut-dir*. This fact provides evidence that incremental MaxSAT reasoning is effective in pruning search in large sparse graphs, even with a smaller initial clique.

Overall, LMC is clearly superior to PMC and BBMCSP. These results show that the combination of the novel preprocessing and incremental MaxSAT reasoning in the proposed BnB scheme is competitive to find maximum cliques in large sparse graphs, which is

analyzed in details in the next subsection.

3.3 The impact of preprocessing and incremental MaxSAT reasoning in LMC

To analyze the impact of preprocessing in LMC, we report in Table 3 the density of the graphs whose search time is more than 5s in Table 2, before and after the preprocessing at the root of the search tree

(see line 2 in Algorithm 5), and the ratio of the number of vertices of the reduced graph to the number of vertices of the original graph, as well as the mean density of a subgraph $G(P)$ and of its reduced graph G'' (see line 8 and line 9 in Algorithm 5), and the mean ratio of the number of vertices of G'' to the number of vertices of $G(P)$. We can see that most graphs are already significantly reduced by preprocessing at the root of the search tree, which considerably increases their density. The preprocessing in the first level of the search tree further reduces the graphs and substantially increases the density of G'' , which is really solved by the BnB algorithm *SearchMaxClique* (line 14 of Algorithm 5). While an IS in the original graph G with very low density is usually huge, an IS in G'' is much smaller, which explains the effectiveness of incremental MaxSAT reasoning when solving G'' in LMC, because conflicts among small ISs can be easily detected to efficiently reduce the set B of branching vertices.

Table 3. Impact of the preprocessing in LMC. The columns d and d' are respectively the density of the original graph G and of its reduced graph G' in the root of the search tree, rt the ratio of the number of vertices of G' to the number of vertices of G ; \bar{d}_P and \bar{d}'' the mean density of the subgraph $G(P)$ and of the reduced subgraph G'' in the first level of the tree, and rt'' the ratio of the number of vertices of G'' to the number of vertices of $G(P)$. The entries marked with '-' mean that all vertices of the first level subgraphs have been removed by the preprocessing.

Graph	Original Graph			First Level Subgraphs		
	d	rt	d'	\bar{d}_P	rt''	\bar{d}''
aff-digg	0.000059	0.16	0.002066	0.302	0.32	0.35
aff-orkut-user2groups	0.000009	0.78	0.000014	0.005	0.01	0.18
bio-human-gene1	0.049641	0.20	0.636241	0.968	0.79	0.98
bio-human-gene2	0.087802	0.27	0.693864	0.973	0.80	0.98
bio-mouse-gene	0.014219	0.38	0.080813	0.780	0.28	0.93
bn-human-BNU_1_0	0.000086	0.13	0.003767	0.591	0.15	0.70
025864_session_1-bg						
bn-human-BNU_1_0	0.000080	0.14	0.003416	0.587	0.17	0.69
025864_session_2-bg						
bn-human-BNU_1_0	0.000043	0.06	0.009783	0.524	0.29	0.66
025865_session_1-bg						
bn-human-BNU_1_0	0.000016	0.05	0.005656	0.461	0.22	0.65
025865_session_2-bg						
dbpedia-link	0.000001	0.18	0.000028	0.147	0.01	0.52
rec-dating	0.001218	0.81	0.001869	0.056	0.10	0.28
rec-libimseti-dir	0.000706	0.79	0.001128	0.044	0.04	0.28
rec-movieleens	0.003902	0.91	0.004667	0.255	0.27	0.35
sc-TSOPF-RS-b2383-c1	0.022181	1.00	0.022181	0.004	0.00	-
soc-flickr-und	0.000011	0.03	0.007760	0.349	0.18	0.62
soc-friendster	0.000001	0.42	0.000004	0.041	0.01	0.33
soc-livejournal-user-groups	0.000004	0.42	0.000022	0.044	0.01	0.37
soc-orkut-dir	0.000025	0.82	0.000036	0.185	0.03	0.40
soc-orkut	0.000024	0.75	0.000039	0.191	0.02	0.44
soc-sinaweibo	0.000001	0.12	0.000006	0.055	0.01	0.61
socfb-A-anon	0.000005	0.13	0.000191	0.163	0.01	0.77
tech-ip	0.000009	0.19	0.000206	0.002	0.01	0.15
tech-p2p	0.000009	0.03	0.003944	0.236	0.09	0.67
twitter_mpi	0.000002	0.03	0.001355	0.241	0.10	0.64
web-indochina-2004-all	0.000005	0.01	0.999713	0.999	0.89	0.99
web-wikipedia-growth	0.000021	0.49	0.000071	0.172	0.02	0.43

We now show the individual impact of preprocessing the first level subgraphs and of incremental MaxSAT reasoning in LMC by comparing it with the following solvers:

LMC\prep1. It is LMC without preprocessing the first level subgraphs; i.e., line 9 and line 10 that call procedure *Initialize* are removed in Algorithm 5.

LMC\MaxSAT. It is LMC without incremental MaxSAT reasoning; i.e., line 14 in the *GetBranches* function (Algorithm 3) is removed.

Table 4 shows the search tree size and the search time of LMC, LMC\prep1 and LMC\MaxSAT on the graphs of Table 3. With preprocessing and incremental MaxSAT reasoning, the search tree size of LMC is always the smallest. The search time of LMC is comparable with that of LMC\prep1 and LMC\MaxSAT on easy

Table 4. Search tree sizes in thousands and search times in seconds (s) of LMC, LMC\prep1 and LMC\MaxSAT. The cpu time limit is 5h.

Graph	LMC		LMC\prep1		LMC\MaxSAT	
	tree	search	tree	search	tree	search
aff-digg	5602	175.6	7055	181.5	13553	202.9
aff-orkut-user2groups	4173	436.5	6831	316.6	4173	457.3
bio-human-gene1	11.19	1047	12.53	1081	59.46	1461.8
bio-human-gene2	14.16	178.4	15.45	122.8	35.85	392.4
bio-mouse-gene	76.92	145.8	78.85	153.2	611.1	703.1
bn-human-BNU_1_0	210.1	642.8	-	>5h	-	>5h
025864_session_1-bg						
bn-human-BNU_1_0	215.4	544.1	-	>5h	-	>5h
025864_session_2-bg						
bn-human-BNU_1_0	104.3	192.9	-	>5h	7490	1184
025865_session_1-bg						
bn-human-BNU_1_0	53.70	40.91	105.7	60.11	96.31	40.17
025865_session_2-bg						
dbpedia-link	684.6	16.07	686.7	13.30	685.0	16.12
rec-dating	138.3	12.16	142.8	9.87	138.3	12.21
rec-libimseti-dir	173.3	10.47	174.7	9.19	173.3	10.43
rec-movieleens	120.8	17.72	144.3	10.51	160.1	17.81
sc-TSOPF-RS-b2383-c1	34.25	6.82	34.26	11.87	34.25	6.83
soc-flickr-und	81.01	24.32	307.0	61.49	639.7	39.22
soc-friendster	2308	393.6	2318	412.3	2308	388.8
soc-livejournal-user-groups	2403	42.48	2403	31.32	2403	42.88
soc-orkut-dir	714.0	34.08	731.5	27.81	724.3	34.14
soc-orkut	745.5	29.25	751.6	23.87	754.5	29.51
soc-sinaweibo	712.7	26.83	714.4	30.50	713.5	26.93
socfb-A-anon	355.2	5.02	355.2	3.76	355.2	4.99
tech-ip	79.29	7.47	79.29	9.70	79.29	7.48
tech-p2p	235.1	208.8	1116	746.5	2922	831.0
twitter_mpi	323.1	149.6	-	>5h	2911	249.7
web-indochina-2004-all	0.14	82.38	0.14	24.36	0.14	82.45
web-wikipedia-growth	358.4	6.41	359.6	5.18	360.7	6.35

graphs. However, LMC is substantially faster than LMC\prep1 and LMC\MaxSAT on hard graphs. In particular, the search time of LMC is smaller than 1047s for all the graphs, while LMC\prep1 fails to solve 4 graphs and LMC\MaxSAT fails to solve 2 graphs within 5h. In addition, LMC is 14 and 4 times faster than LMC\MaxSAT for *bio-human-gene1* and *tech-p2p*, respectively.

4 CONCLUSIONS

Clique is a valuable property for analyzing real-world large sparse graphs. We have proposed a new exact algorithm, called LMC, to find maximum cliques in large sparse graphs that combines a novel preprocessing and incremental MaxSAT reasoning in a BnB scheme. The preprocessing procedure *Initialize* performs effectively three tasks at the same time with a very low overhead: derive a vertex ordering, reduce the graph, and compute an initial clique. LMC also applies preprocessing to the subgraphs in the first level of the search tree, so that the underlying algorithm *SearchMaxClique* can work with a better vertex ordering and a more reduced graph when searching these subgraphs with incremental MaxSAT reasoning.

Our new algorithm can solve all tested graphs efficiently and shows a performance that is superior over PMC and BBMCSP, refuting the opinion in the literature that sophisticated techniques such as MaxSAT reasoning are not useful for large sparse graphs. The empirical analysis suggests that the performance of incremental MaxSAT reasoning in LMC comes from the fact that the proposed preprocessing considerably increases the density of the graphs to be solved.

ACKNOWLEDGEMENTS

This work is supported by NSFC Grant No. 61370184, No. 61272014, and No. 61472147. The third author was supported by Mobility Grant PRX16/00215 of the Ministerio de Educación, Cultura y Deporte, the Generalitat de Catalunya grant AGAUR 2014-SGR-118, and the MINECO-FEDER project RASO TIN2015-71799-C2-1-P.

REFERENCES

- [1] L. Babel and G. Tinhofer, 'A branch and bound algorithm for the maximum clique problem', *Methods and Models of Operations Research*, **34**(3), 207–217, (1990).
- [2] B. Balasundaram, S. Butenko, and I.V. Hicks, 'Clique relaxations in social network analysis: The maximum k-plex problem', *Operations Research*, **59**(1), 133–142, (2011).
- [3] V. Batagelj and A. Mrvar, 'Pajek: a program for large network analysis', *Connections*, **21**(2), 47–57, (1998).
- [4] V. Batagelj and M. Zaversnik, 'An o(m) algorithm for cores decomposition of networks', *Eprint Arxiv Cs*, **1**(6), 34–37, (2003).
- [5] U. Benlic and J.K. Hao, 'Breakout local search for maximum clique problems', *Computers & Operations Research*, **40**(1), 192–206, (2013).
- [6] P. Berman and A. Pelc, 'Distributed probabilistic fault diagnosis for multiprocessor systems', in *Fault-Tolerant Computing, 1990. FTCS-20. Digest of Papers., 20th International Symposium*, pp. 340–346, (1990).
- [7] V. Boginski, S. Butenko, and P.M. Pardalos, 'Mining market data: A network approach.', *Computers & Operations Research*, **33**(11), 3171–3184, (2006).
- [8] R. Carraghan and P.M. Pardalos, 'An exact algorithm for the maximum clique problem', *Operations Research Letters*, **9**(6), 375–382, (1990).
- [9] M.S. Dawn, B. Earl, and Joel S.S., 'Optimal protein structure alignment using maximum cliques', *Operations Research*, **53**(3), 389–402, (2005).
- [10] T. Etzion and P.R.J. Ostergard, 'Greedy and heuristic algorithms for codes and colorings', *IEEE Transactions on Information Theory*, **44**(1), 382–388, (1998).
- [11] T. Fahle, 'Simple and fast: Improving a branch-and-bound algorithm for maximum clique', *Esa*, **2461**, 47–86, (2002).
- [12] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.
- [13] A. Grosso, M. Locatelli, and W. Pullan, 'Simple ingredients leading to very efficient heuristics for the maximum clique problem', *Journal of Heuristics*, **14**(6), 587–612, (2008).
- [14] D.J. Johnson and M.A. Trick, *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, Workshop, October 11-13, 1993*, American Mathematical Society, 1996.
- [15] J. Konc and D. Janezic, 'An improved branch and bound algorithm for the maximum clique problem', *Communications in Mathematical & in Computer Chemistry*, **58**(3), 569–590, (2007).
- [16] C.M. Li, Z.W. Fang, and K. Xu, 'Combining maxsat reasoning and incremental upper bound for the maximum clique problem', in *Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on*, pp. 939–946, (2013).
- [17] C.M. Li, H. Jiang, and R.C. Xu, 'Incremental maxsat reasoning to reduce branches in a branch-and-bound algorithm for maxclique', *Lecture Notes in Computer Science*, **8994**, 268–274, (2015).
- [18] C.M. Li and F. Manyà, 'MaxSAT, hard and soft constraints', in *Handbook of Satisfiability*, eds., Armin Biere, Hans van Maaren, and Toby Walsh, 613–631, IOS Press, (2009).
- [19] C.M. Li and Z. Quan, 'Combining graph structure exploitation and propositional reasoning for the maximum clique problem', in *Proceedings of the 2010 22nd IEEE International Conference on Tools with Artificial Intelligence - Volume 01*, pp. 344–351, (2010).
- [20] C.M. Li and Z. Quan, 'An efficient branch-and-bound algorithm based on maxsat for the maximum clique problem', in *Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, Usa, July*, pp. 128–133, (2010).
- [21] M.E.J. Newman, 'The structure and function of complex networks', in *SIAM Rev*, pp. 167–256, (2003).
- [22] P.R.J. Ostergard, 'A fast algorithm for the maximum clique problem', in *Discrete Appl. Math*, pp. 197–207, (2002).
- [23] W. Pullan and H.H. Hoos, 'Dynamic local search for the maximum clique problem', *J. Artif. Int. Res.*, **25**(1), 159–185, (2006).
- [24] W. Pullan, F. Mascia, and M. Brunato, 'Cooperating local search for the maximum clique problem', *Journal of Heuristics*, **17**(2), 181–199, (2011).
- [25] R.A. Rossi, D.F. Gleich, A.H. Gebremedhin, and M.M.A. Patwary, 'Parallel maximum clique algorithms with applications to network analysis and storage', *Eprint Arxiv*, (2013).
- [26] R.A. Rossi and K.A. Nesreen, 'The network data repository with interactive graph analytics and visualization', in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, (2015).
- [27] P.S. Segundo, L. Alvaro, and P.M. Pardalos, 'A new exact maximum clique algorithm for large and massive sparse graphs', *Computers & Operations Research*, **66**, 81–94, (2016).
- [28] P.S. Segundo, F. Matia, D. Rodriguez-Losada, and M. Hernando, 'An improved bit parallel exact maximum clique algorithm', *Optimization Letters*, **7**(3), 467–479, (2011).
- [29] P.S. Segundo, D. Rodríguez-Losada, and A. Jiménez, 'An exact bit-parallel algorithm for the maximum clique problem', *Computers & Operations Research*, **38**(2), 571–581, (2011).
- [30] S.B. Seidman, 'Network structure and minimum degree', *Social Networks*, **5**(3), 269–287, (1983).
- [31] E. Tomita and T. Kameda, 'An efficient branch-and-bound algorithm for finding a maximum clique with computational experiments', *Journal of Global Optimization*, **37**(1), 95–111, (2007).
- [32] E. Tomita, Y. Sutani, T. Higashi, S. Takahashi, and M. Wakatsuki, 'A simple and faster branch-and-bound algorithm for finding a maximum clique', in *WALCOM: Algorithms and Computation, 4th International Workshop, WALCOM 2010*, pp. 191–203, (2010).

An Efficient Approach for the Generation of Allen Relations

Kleanthi Georgala¹ and Mohamed Ahmed Sherif² and Axel-Cyrille Ngonga Ngomo³

Abstract. Event data is increasingly being represented according to the Linked Data principles. The need for large-scale machine learning on data represented in this format has thus led to the need for efficient approaches to compute RDF links between resources based on their temporal properties. Time-efficient approaches for computing links between RDF resources have been developed over the last years. However, dedicated approaches for linking resources based on temporal relations have been paid little attention to. In this paper, we address this research gap by presenting AEGLE, a novel approach for the efficient computation of links between events according to Allen’s interval algebra. We study Allen’s relations and show that we can reduce all thirteen relations to eight simpler relations. We then present an efficient algorithm with a complexity of $O(n \log n)$ for computing these eight relations. Our evaluation of the runtime of our algorithms shows that we outperform the state of the art by up to 4 orders of magnitude while maintaining a precision and a recall of 1.

1 INTRODUCTION

Over the past years, technological progress in hardware development and network infrastructures have led to the collection of large amounts of event data in scenarios as diverse as monitoring industrial plants [12], monitoring open SPARQL endpoints [22], implementing the Internet of Things (IoT) and Cloud Computing [13]. For example, the LSQ dataset [22] consists of more than 1.2 billion facts which describe more than 250 million query events on open SPARQL endpoints. The availability of such large collections of event data in Resource Description Framework (RDF)⁴ format as well as the uptake of semantic technologies (in particular RDF) to represent machine events [20] has consequently led to the need for interlinking these events, especially to support structured machine learning [11] (e.g., predictive maintenance for machine data or discovering sequences of query patterns that a triple store is often faced with) over these datasets.

Given that the computation of links is the fourth principle of Linked Data,⁵ a large number of frameworks have been developed to facilitate the computation of links between knowledge bases (see [14] for a survey). Still, to the best of our knowledge, only one approach has been developed for computing temporal links between events [23]. The approach presented in [23] is based on the Multi-

Block algorithm [8] and employs multi-dimensional blocking to reduce the number of comparisons necessary to compute temporal relations. However, our evaluation suggests that this approach does not scale to larger number of events.

In this paper, we hence address the problem of computing temporal relations between events efficiently. To this end, we rely on Allen’s Interval Algebra [1] as it encompasses all primitive temporal relations between events. Our approach, dubbed AEGLE (Allen’s intErval alGebra for Link discovEry), relies on two insights: First, the 13 Allen relations can be reduced to 8 simpler relations that all compare exactly either the beginning or the end of an event with the beginning or end of another event. The second insight behind our approach is that given that time is ordered, we can reduce the problem of detecting such relations to the problem of finding matching entities in two sorted lists. As this problem has a complexity of $O(n \log n)$, our approach should scale well even for larger datasets. Importantly, our method achieves 100% precision and recall as it computes all temporal relations between events from a source set S and a target set T . The main contributions of our work are thus as follows:

- We show how the 13 Allen relations can be reduced to 8 atomic relations and how these 8 relations can be combined using set theory to reconstruct the 13 Allen relations.
- We provide an efficient approach to computing each of the 8 atomic relations aforementioned.
- We evaluate the runtime of our approach using real and synthetic data and show that we outperform the state of the art by up to 4 orders of magnitude.

The rest of this paper is organised as follows: Section 2 includes the basic notation and preliminaries behind link discovery (LD)⁶ and Allen’s Interval Algebra. Section 3 describes our approach by (1) defining the set of atomic relations we use to compute temporal relations and (2) showing how to derive more complex relations from the set of atomic relations derived previously. In Section 4, we present a systematic comparison of our approach with the state of the art. Finally, we give an overview of the existing related work and conclude with a brief summary of our work and future plans.

2 PRELIMINARIES

In this section, we present the concepts and notation that are necessary to understand the rest of the paper. First, we introduce the LD problem by providing a formal definition akin to that introduced in [16]. Thereafter, we provide the notation for the basic relations between intervals as introduced in [1].

⁶ We use the term “link discovery” to signify the computation of links of particular types between pairs of resources represented in RDF. Never do we use this term in the sense of mining links between nodes in a graph.

¹ AKSW Research Group, University of Leipzig, Germany, email: georgala@informatik.uni-leipzig.de

² AKSW Research Group, University of Leipzig, Germany, email: sherif@informatik.uni-leipzig.de

³ AKSW Research Group, University of Leipzig, Germany, email: ngonga@informatik.uni-leipzig.de

⁴ <https://www.w3.org/RDF/>

⁵ <http://www.w3.org/DesignIssues/LinkedData.html>

2.1 Link Discovery

Throughout this paper, we deal with facts represented in RDF. Each fact is a triple $(c, p, o) \in (\mathcal{R} \cup \mathcal{B}) \times \mathcal{P} \times (\mathcal{R} \cup \mathcal{B} \cup \mathcal{L})$, where

1. c is the subject of the triple (i.e., what the triple is mainly about),
2. p is the predicate of the triple (i.e., the relation that the subject has with the object),
3. o is the object of the triple (that which is to be related to the subject through the predicate),
4. \mathcal{R} is the set of all RDF resources, where each resource stands for a thing from the real world, e.g., an event,
5. $\mathcal{P} \subseteq \mathcal{R}$ is the set of all RDF properties, which are binary predicates,
6. \mathcal{B} is the set of all RDF blank nodes, which basically model existential semantics and
7. \mathcal{L} is the set of all literals, i.e., of all data types (e.g., time points).

An example of a fact would be $(:E1, :begin, 0.1)$, which states that the event $:E1$ begins at the point 0.1 in time.

We call a set of triples a knowledge base (KB). Given two sets of resources S and T from two (not necessarily distinct) KBs as well as a binary relation R , the main goal of LD is to discover the set $M = \{(s, t) \in S \times T : R(s, t)\}$. We call M a mapping. Naive approaches towards this goal are quadratic in complexity as they have to compare every $s \in S$ with every $t \in T$, which is clearly impracticable for large S and T . In this work, we thus consider the efficient computations of temporal relations between events. Hence, we assume that each of the resources in S and T considered in the subsequent portion of this paper describes an event v with a beginning time denoted $b(v)$ and an end time denoted $e(v)$. Note that we assume that $b(v) < e(v)$ throughout this work.

2.2 Allen's Interval Algebra

Allen's Interval Algebra [1] is a widely known time interval calculus, which provides a set of 13 "distinct, exhaustive, and qualitative" relations between time intervals [2]. Table 1 illustrates this set of relations and shows a set of six relations between two time intervals X and Y , their corresponding symbols along with the symbols of their inverse relation. The *equal* relation is symmetric.

2.3 Link Discovery between Events

An event can be modelled as a time interval because we assume that its description always includes a begin time property and an end time property. Thus, an event instance s can be described as pair of time points $(b(s), e(s))$, where $b(s) < e(s)$. Formally, computing the temporal relations between events can thus be reduced to computing the following mappings M :

- if $R = bf$, then $M = \{(s, t) \in S \times T : (b(s) < b(t)) \wedge (b(s) < e(t)) \wedge (e(s) < b(t)) \wedge (e(s) < e(t))\}$
- if $R = bfi$, then $M = \{(s, t) \in S \times T : (b(s) > b(t)) \wedge (b(s) > e(t)) \wedge (e(s) > b(t)) \wedge (e(s) > e(t))\}$
- if $R = m$, then $M = \{(s, t) \in S \times T : (b(s) < b(t)) \wedge (b(s) < e(t)) \wedge (e(s) = b(t)) \wedge (e(s) < e(t))\}$
- if $R = mi$, then $M = \{(s, t) \in S \times T : (b(s) > b(t)) \wedge (b(s) = e(t)) \wedge (e(s) > b(t)) \wedge (e(s) > e(t))\}$
- if $R = f$, then $M = \{(s, t) \in S \times T : (b(s) > b(t)) \wedge (b(s) < e(t)) \wedge (e(s) > b(t)) \wedge (e(s) = e(t))\}$
- if $R = fi$, then $M = \{(s, t) \in S \times T : (b(s) < b(t)) \wedge (b(s) < e(t)) \wedge (e(s) > b(t)) \wedge (e(s) = e(t))\}$

Table 1. Allen's Interval Algebra

Relation	Notation	Inverse	Illustration
X before Y	$bf(X, Y)$	$bfi(X, Y)$	
X meets Y	$m(X, Y)$	$mi(X, Y)$	
X finishes Y	$f(X, Y)$	$fi(X, Y)$	
X starts Y	$st(X, Y)$	$sti(X, Y)$	
X during Y	$d(X, Y)$	$di(X, Y)$	
X equal Y	$eq(X, Y)$	$eq(X, Y)$	
X overlaps with Y	$ov(X, Y)$	$ovi(X, Y)$	

- if $R = st$, then $M = \{(s, t) \in S \times T : (b(s) = b(t)) \wedge (b(s) < e(t)) \wedge (e(s) > b(t)) \wedge (e(s) < e(t))\}$
- if $R = sti$, then $M = \{(s, t) \in S \times T : (b(s) = b(t)) \wedge (b(s) < e(t)) \wedge (e(s) > b(t)) \wedge (e(s) > e(t))\}$
- if $R = d$, then $M = \{(s, t) \in S \times T : (b(s) > b(t)) \wedge (b(s) < e(t)) \wedge (e(s) > b(t)) \wedge (e(s) < e(t))\}$
- if $R = di$, then $M = \{(s, t) \in S \times T : (b(s) < b(t)) \wedge (b(s) < e(t)) \wedge (e(s) > b(t)) \wedge (e(s) > e(t))\}$
- if $R = eq$, then $M = \{(s, t) \in S \times T : (b(s) = b(t)) \wedge (b(s) < e(t)) \wedge (e(s) > b(t)) \wedge (e(s) = e(t))\}$
- if $R = ov$, then $M = \{(s, t) \in S \times T : (b(s) < b(t)) \wedge (b(s) < e(t)) \wedge (e(s) > b(t)) \wedge (e(s) < e(t))\}$
- if $R = ovi$, then $M = \{(s, t) \in S \times T : (b(s) > b(t)) \wedge (b(s) < e(t)) \wedge (e(s) > b(t)) \wedge (e(s) > e(t))\}$

3 APPROACH

3.1 Overview

As we have now introduced the necessary notations and concepts behind LD and Allen's Interval Algebra, we can proceed to explain our approach for the rapid computation of temporal links between events in detail. The main goal of our approach is to compute all Allen's interval relations (as illustrated in Table 1) between two sets of atomic events efficiently. The main insight underlying this work is that we can reduce the computation of the 13 relations to the computation and combinations of a mere 8 atomic relations and thus reduce the overall computation time of Allen relations. We use this insight to devised means to compute all interval relations efficiently by reducing all of Allen's relations to re-usable atomic relations that can be computed efficiently. We then combine the results of these atomic relations to compute Allen's relations. While doing so, we ensure that we achieve 100% accuracy in retrieving all possible Allen relations between resources in the given sets of resources S and T .

3.2 AEGLE

The main idea behind our approach is to represent each relation of Table 1 as a Boolean combination of atomic relations. By computing each of the atomic relations only once and only if needed, we can decrease the overall runtime of the computation of a given set of Allen relations.

As described in Section 2.3, each atomic event s can be described using two time points $b(s)$ and $e(s)$. To compose the atomic interval relations, we define all possible binary relations between the begin and end points of two event resources $s = (b(s), e(s))$ and $t = (b(t), e(t))$ as follows:

- Atomic relations between $b(s)$ and $b(t)$:
 - $BB^1(s, t) \Leftrightarrow (b(s) < b(t))$
 - $BB^0(s, t) \Leftrightarrow (b(s) = b(t))$
 - $BB^{-1}(s, t) \Leftrightarrow (b(s) > b(t)) \Leftrightarrow \neg(BB^1(s, t) \vee BB^0(s, t))$
- Atomic relations between $b(s)$ and $e(t)$:
 - $BE^1(s, t) \Leftrightarrow (b(s) < e(t))$
 - $BE^0(s, t) \Leftrightarrow (b(s) = e(t))$
 - $BE^{-1}(s, t) \Leftrightarrow (b(s) > e(t)) \Leftrightarrow \neg(BE^1(s, t) \vee BE^0(s, t))$
- Atomic relations between $e(s)$ and $b(t)$:
 - $EB^1(s, t) \Leftrightarrow (e(s) < b(t))$
 - $EB^0(s, t) \Leftrightarrow (e(s) = b(t))$
 - $EB^{-1}(s, t) \Leftrightarrow (e(s) > b(t)) \Leftrightarrow \neg(EB^1(s, t) \vee EB^0(s, t))$
- Atomic relations between $e(s)$ and $e(t)$:
 - $EE^1(s, t) \Leftrightarrow (e(s) < e(t))$
 - $EE^0(s, t) \Leftrightarrow (e(s) = e(t))$
 - $EE^{-1}(s, t) \Leftrightarrow (e(s) > e(t)) \Leftrightarrow \neg(EE^1(s, t) \vee EE^0(s, t))$

Out of Table 1, we can derive how each of Allen's relations can be reduced to a Boolean combination of a subset of the relations above as follows:

- $bf(s, t) \Leftrightarrow BB^1(s, t) \wedge BE^1(s, t) \wedge EB^1(s, t) \wedge EE^1(s, t)$.
Now given that $b(s) < e(s)$ and $b(t) < e(t)$ and by virtue of the transitivity of $<$, we get
 1. $e(s) < b(t) \Rightarrow b(s) < b(t)$,
 2. $e(s) < b(t) \Rightarrow b(s) < e(t)$ (by virtue of 1.) and
 3. $e(s) < b(t) \Rightarrow e(s) < e(t)$.
 Hence $bf(s, t) = EB^1(s, t)$.
- $bfi(s, t) \Leftrightarrow BB^{-1}(s, t) \wedge BE^{-1}(s, t) \wedge EB^{-1}(s, t) \wedge EE^{-1}(s, t)$. Now given that $b(s) < e(s)$ and $b(t) < e(t)$ and by virtue of the transitivity of $<$, we get
 1. $b(s) > e(t) \Rightarrow b(s) > b(t)$,
 2. $b(s) > e(t) \Rightarrow e(s) > b(t)$ (by virtue of 1.) and
 3. $b(s) > e(t) \Rightarrow e(s) > e(t)$.
 Hence $bfi(s, t) = BE^{-1} = \neg(BE^1(s, t) \vee BE^0(s, t))$.
- $m(s, t) \Leftrightarrow BB^1(s, t) \wedge BE^1(s, t) \wedge EB^0(s, t) \wedge EE^1(s, t)$.
Now given that $b(s) < e(s)$ and $b(t) < e(t)$ and by virtue of the transitivity of $<$, we get
 1. $e(s) = b(t) \Rightarrow b(s) < b(t)$,
 2. $e(s) = b(t) \Rightarrow e(s) < e(t)$ and

3. $e(s) = b(t) \Rightarrow b(s) < e(t)$ (by virtue of 1.).

Hence $mi(s, t) = EB^0(s, t)$.

- $mi(s, t) \Leftrightarrow BB^{-1}(s, t) \wedge BE^0(s, t) \wedge EB^{-1}(s, t) \wedge EE^{-1}(s, t)$.
Now given that $b(s) < e(s)$ and $b(t) < e(t)$ and by virtue of the transitivity of $<$, we get
 1. $b(s) = e(t) \Rightarrow b(s) > b(t)$,
 2. $b(s) = e(t) \Rightarrow e(s) > e(t)$ and
 3. $b(s) = e(t) \Rightarrow e(s) > b(t)$ (by virtue of 1.).
 Hence $mi(s, t) = BE^0(s, t)$.
- $f(s, t) \Leftrightarrow BB^{-1}(s, t) \wedge BE^1(s, t) \wedge EB^{-1}(s, t) \wedge EE^0(s, t)$.
Now given that $b(s) < e(s)$ and $b(t) < e(t)$ and by virtue of the transitivity of $<$, we get
 1. $b(s) > b(t) \Rightarrow e(s) > b(t)$ and
 2. $e(s) = e(t) \Rightarrow b(s) < e(t)$
 Hence $f(s, t) = \{EE^0(s, t) \wedge BB^{-1}(s, t)\} = \{EE^0(s, t) \wedge \neg(BB^0(s, t) \vee BB^1(s, t))\}$.
- $fi(s, t) \Leftrightarrow BB^1(s, t) \wedge BE^1(s, t) \wedge EB^{-1}(s, t) \wedge EE^0(s, t)$.
Now given that $b(s) < e(s)$ and $b(t) < e(t)$ and by virtue of the transitivity of $<$, we get
 1. $b(s) < b(t) \Rightarrow b(s) < e(t)$ and
 2. $e(s) = e(t) \Rightarrow e(s) > b(t)$
 Hence $fi(s, t) = \{BB^1(s, t) \wedge EE^0(s, t)\}$.
- $st(s, t) \Leftrightarrow BB^0(s, t) \wedge BE^1(s, t) \wedge EB^{-1}(s, t) \wedge EE^1(s, t)$.
Now given that $b(s) < e(s)$ and $b(t) < e(t)$ and by virtue of the transitivity of $<$, we get
 1. $b(s) = b(t) \Rightarrow b(s) < e(t)$ and
 2. $e(s) < e(t) \Rightarrow e(s) > b(t)$
 Hence $st(s, t) = \{BB^0(s, t) \wedge EE^1(s, t)\}$.
- $sti(s, t) \Leftrightarrow BB^0(s, t) \wedge BE^1(s, t) \wedge EB^{-1}(s, t) \wedge EE^{-1}(s, t)$.
Now given that $b(s) < e(s)$ and $b(t) < e(t)$ and by virtue of the transitivity of $<$, we get
 1. $b(s) = b(t) \Rightarrow e(s) > b(t)$ and
 2. $b(s) = b(t) \Rightarrow b(s) < e(t)$
 Hence $sti(s, t) = \{BB^0(s, t) \wedge EE^{-1}(s, t)\} = \{BB^0(s, t) \wedge \neg(EE^0(s, t) \vee EE^1(s, t))\}$.
- $d(s, t) \Leftrightarrow BB^{-1}(s, t) \wedge BE^1(s, t) \wedge EB^{-1}(s, t) \wedge EE^1(s, t)$.
Now given that $b(s) < e(s)$ and $b(t) < e(t)$ and by virtue of the transitivity of $<$, we get
 1. $b(s) > b(t) \Rightarrow e(s) > b(t)$ and
 2. $e(s) < e(t) \Rightarrow b(s) < e(t)$
 Hence $d(s, t) = \{EE^1(s, t) \wedge BB^{-1}(s, t)\} = \{EE^1(s, t) \wedge \neg(BB^0(s, t) \vee BB^1(s, t))\}$.
- $di(s, t) \Leftrightarrow BB^1(s, t) \wedge BE^1(s, t) \wedge EB^{-1}(s, t) \wedge EE^{-1}(s, t)$.
Now given that $b(s) < e(s)$ and $b(t) < e(t)$ and by virtue of the transitivity of $<$, we get
 1. $e(s) > e(t) \Rightarrow e(s) > b(t)$ and
 2. $b(s) < b(t) \Rightarrow b(s) < e(t)$
 Hence $di(s, t) = \{BB^1(s, t) \wedge EE^{-1}(s, t)\} = \{BB^1(s, t) \wedge \neg(EE^0(s, t) \vee EE^1(s, t))\}$.
- $eq(s, t) \Leftrightarrow BB^0(s, t) \wedge BE^1(s, t) \wedge EB^{-1}(s, t) \wedge EE^0(s, t)$.
Now given that $b(s) < e(s)$ and $b(t) < e(t)$ and by virtue of the transitivity of $<$, we get

1. $e(s) = e(t) \Rightarrow e(s) > b(t)$ and
2. $b(s) = b(t) \Rightarrow b(s) < e(t)$

Hence $eq(s, t) = \{BB^0(s, t) \wedge EE^0(s, t)\}$.

- $ov(s, t) \Leftrightarrow BB^1(s, t) \wedge BE^1(s, t) \wedge EB^{-1}(s, t) \wedge EE^1(s, t)$.
Now given that $b(s) < e(s)$ and $b(t) < e(t)$ and by virtue of the transitivity of $<$, we get

1. $b(s) < b(t) \Rightarrow b(s) < e(t)$.

Hence $ov(s, t) = \{BB^1(s, t) \wedge EB^{-1}(s, t) \wedge EE^1(s, t)\} = \{(BB^1(s, t) \wedge EE^1(s, t)) \wedge \neg(EB^0(s, t) \vee EB^1(s, t))\}$.

- $ovi(s, t) \Leftrightarrow BB^{-1}(s, t) \wedge BE^1(s, t) \wedge EB^{-1}(s, t) \wedge EE^{-1}(s, t)$.
Now given that $b(s) < e(s)$ and $b(t) < e(t)$ and by virtue of the transitivity of $<$, we get

1. $e(s) > e(t) \Rightarrow e(s) > b(t)$.

Hence $ovi(s, t) = \{BB^{-1}(s, t) \wedge BE^1(s, t) \wedge EE^{-1}(s, t)\} = \{(BE^1(s, t) \wedge \neg(BB^0(s, t) \vee BB^1(s, t))) \wedge \neg(EE^0(s, t) \vee EE^1(s, t))\}$.

Clearly, we hence only need to compute the 8 atomic relations $EB^0, EB^1, EE^0, EE^1, BB^0, BB^1, BE^0$ and BE^1 to be able to generate all of Allen's relations. In the following, we explicate our approach to computing these 8 relations efficiently.

3.3 Algorithm

Algorithm 1: AEGLE

Input: source S , target T , set of Allen relations \mathcal{ATR}

Output: Set of mappings \mathcal{M}

```

1  $\mathcal{M} \leftarrow \emptyset$ 
2  $\mathcal{A} \leftarrow \emptyset$ 
3 foreach  $rel \in \mathcal{ATR}$  do
4    $requiredRelations \leftarrow getAtmRelations(rel)$ 
5    $atomics(rel) \leftarrow \emptyset$ 
6   foreach  $atomicRel \in requiredRelations$  do
7     if  $\mathcal{A}$  does not contain  $atomicRel$  then
8        $a \leftarrow computeAtmRelation(atomicRel, S, T)$ 
9        $\mathcal{A}.put(atomicRel, a)$ 
10     $atomics(rel).put(atomicRel, \mathcal{A}.get(atomicRel))$ 
11   $M \leftarrow computeRelation(atomics(rel))$ 
12   $\mathcal{M}.add(M)$ 
13 Return  $\mathcal{M}$ 

```

Given a set \mathcal{ATR} of Allen relations that are to be computed, the basic idea behind our approach is to begin by detecting the subset of the 8 atomic relations that needs to be computed and to compute each of these relations exactly once. Algorithm 1 describes how the idea was implemented. Our approach, AEGLE, takes two sets of events, S and T , and the set \mathcal{ATR} as input. The algorithm returns a set of mappings \mathcal{M} , of which each corresponds to exactly one of the relations in \mathcal{ATR} .

We begin by initialising the final set of mappings \mathcal{M} in line 1 and the map \mathcal{A} in line 2. \mathcal{A} includes the labels of atomic relations as keys and their corresponding mapping as values. During the first step of our algorithm, for each $rel \in \mathcal{ATR}$, AEGLE retrieves the set of required atomic relations in line 4 by calling the function $getAtmRelations(rel)$. This function is responsible for retrieving the set of the labels of the atomic relations that are required to

compute rel based on the rules defined in Section 3.2. For each required $atomicRel$ of the current rel , the algorithm checks if the mapping is already computed (line 7). If not, it invokes the function $computeAtmRelation$ to compute the appropriate atomic relations in line 8 and places the resulting mapping along with the atomic relation label in \mathcal{A} . Then, it retrieves the mapping from \mathcal{A} and places it in the $atomics(rel)$ map needed to compute the mapping of rel . Each Allen's relation described in \mathcal{ATR} constructs its own $atomics(rel)$ map that has the labels of the requisite atomic relations as keys and their corresponding mappings as values. Finally, the algorithm computes the mapping M of rel by calling the function $computeRelation$ (line 11) and adds the resulting set of links in \mathcal{M} (line 12).

The time-critical portion of the execution lies in the computation of the atomic relations. The idea underlying our approach to computing these relations is that one can reduce their computation to the problem of finding pairs of matching elements in two sorted lists. For example, to compute BB^0 , one needs to (1) sort the list of elements of S and T according to the time at which they began (guaranteed time complexity: $O(|S| \log |S|)$ resp. $O(|T| \log |T|)$), (2) search for the elements of the smaller set in the larger set ($O(\min(|S|, |T|) \log(\max(|S|, |T|)))$). This leads to an overall complexity of $O(n \log n)$. The complexity is the same for the computation of all relations.

Algorithm 2: $computeAtmRelations(atomicRel, S, T)$ for $atomicRel = BB^0$

Output: mapping of BB^0 AM

```

1  $AM \leftarrow \emptyset$ 
2  $sources \leftarrow orderByDate(S, beginDate)$ 
3  $targets \leftarrow orderByDate(T, beginDate)$ 
4  $AM \leftarrow mapEvents(sources, targets, concurrent)$ 
5 Return  $AM$ 

```

Algorithm 3: $computeAtmRelations(atomicRel, S, T)$ for $atomicRel = EE^1$

Output: mapping of EE^1 AM

```

1  $AM \leftarrow \emptyset$ 
2  $sources \leftarrow orderByDate(S, endDate)$ 
3  $targets \leftarrow orderByDate(T, endDate)$ 
4  $AM \leftarrow mapEvents(sources, targets, predecessor)$ 
5 Return  $AM$ 

```

Algorithm 4: $orderByDate(S, dateType)$

Output: O

```

1 foreach  $s \in S$  do
2    $timeStamp \leftarrow s.getDate(dateType)$ 
3    $tempO \leftarrow \emptyset$ 
4   if  $O$  contains  $timeStamp$  then
5      $tempO \leftarrow O.get(timeStamp)$ 
6    $tempO \leftarrow tempO \cup s$ 
7    $O.put(timeStamp, tempO)$ 
8 Return  $O$ 

```

To illustrate the main procedure of Algorithm 1 (lines 3- 12), consider $\mathcal{ATR} = \{st, sti\}$ as example. The other relations are com-

Algorithm 5: $mapEvents(sources, targets, eventType)$ **Output:** mapped events $Events$

```

1  $Events \leftarrow \emptyset$ 
2 foreach  $sourceTimeStamp \in sources$  do
3   if  $eventType == concurrent$  then
4      $tempT \leftarrow targets.get(sourceTimeStamp)$ 
5   else
6      $tempT \leftarrow$ 
7        $targets.getHigher(sourceTimeStamp)$ 
8   if  $tempT \neq \emptyset$  then
9     foreach  $s \in sources.get(sourceTimeStamp)$  do
10       $Events.put(s, tempT)$ 
10 Return  $Events$ 

```

Algorithm 6: $computeRelation(atomics)$ for st **Output:** mapping M

```

1  $M \leftarrow \emptyset$ 
2 foreach  $s \in atomics.get(BB^0)$  do
3    $M1 \leftarrow atomics.get(BB^0).get(s)$ 
4    $tempEE1 \leftarrow atomics.get(EE^1)$ 
5   if  $tempEE1$  contains  $s$  then
6      $M2 \leftarrow tempEE1.get(s)$ 
7      $M.put(s, M1 \cap M2)$ 
8 Return  $M$ 

```

puted analogously. In line 8 of Algorithm 1, AEGLE calls $computeAtmRelations$ in order to generate the mappings for the required atomic relations for rel , where $rel = st$ and $requiredRelations = BB^0, EE^1$. Since \mathcal{A} is empty and the condition in line 7 holds, Algorithm 1 will call the function $computeAtmRelation$ for BB^0 and then for EE^1 .

For BB^0 , Algorithm 2 describes the necessary steps to compute the mapping of BB^0 . To begin with, Algorithm 2 invokes the function $orderByDate$ for the source S and the target T datasets, to order both complex event resources using the property $beginDate$. Algorithm 4 illustrates the procedure of ordering a complex event S given the value of a property $dateType$, in this case $beginDate$. The main idea of this function is to assert each atomic event $s \in S$ to the appropriate time-bucket, given its $dateType$ value. $orderByDate$ returns a map that has the unique $dateType$ values of the input KB S as keys and the set of events that correspond to each $dateType$ as values. Once $sources$ and $targets$ are retrieved (lines 2, 3 resp. of Algorithm 2), $computeAtmRelations$ calls the function $mapEvents$ using the label $concurrent$, that is responsible for matching each source event s with the set of target events with the same $b(s)$. In the function $mapEvents$ (Algorithm 5), for each source event s that belongs to a time-bucket with time-stamp $sourceTimeStamp$, the algorithm retrieves the appropriate subset of target events that have the same time-stamp (line 4), if any (line 7). Then, it constructs a mapping between each s and the matching set of target events (line 9). Finally, the mapping is returned to Algorithm 1 and it is placed in \mathcal{A} in line 9.

To continue, Algorithm 1 calls again $computeAtmRelations$ since the mapping of EE^1 is not contained as well in \mathcal{A} , following the procedure described in Algorithm 3. For EE^1 , $computeAtmRelations$ is going to order S and T by invoking the $orderByDate$ func-

tion that is going to order the event sources using the $endDate$ property. Once both $sources$ and $targets$ are retrieved (lines 2, 3 resp. of Algorithm 3), $mapEvents$ will be called with $eventType = predecessor$, in order to match each source $s \in S$ with the target events that were terminated after the source event s ended (line 6 of Algorithm 5). Finally, the mapping is returned to the main algorithm and it is placed in \mathcal{A} in line 9.

Once both mappings of BB^0 and EE^1 are retrieved and placed in $atomics(rel)$, Algorithm 1 calls $computeRelation$ for st . Algorithm 6 illustrates the procedure of computing st . For each source event s , the algorithm retrieves the set of targets with the same $b(s)$ from the $atomics$ set (line 3). Then, Algorithm 6 checks if there exists a set of targets with $endDate$ higher than $e(s)$ (line 5). If the condition holds, then $computeRelation$ retrieves the aforementioned set of targets (line 6) and based on the equation in Section 3.2, it computes the intersection between the two sub-sets of targets. The procedure is performed for each source instance and the final mapping M is returned in Algorithm 1 and placed in \mathcal{M} .

Then, the AEGLE proceeds into computing the sti relation, following the steps described above. However, since $sti(s, t) = \{BB^0(s, t) \wedge \neg(EE^0(s, t) \vee EE^1(s, t))\}$, the algorithm will only have to compute EE^0 and retrieve the mappings for BB^0 and EE^1 .

4 EVALUATION

The aim of our evaluation was to address the following questions:

- Q_1 : Does the reduction of Allen relations to 8 atomic relations influence the overall runtime of the approach?
- Q_2 : How does AEGLE perform when compared with the state of the art in terms of time efficiency?

To the best of our knowledge, only one other link discovery framework implements an approach for the discovery of temporal relations. In [23], the blocking approach underlying SILK was extended to deal with spatio-temporal data. We thus compared our approach with the SILK LD framework.

4.1 Experimental Setup

We evaluated our approach on two different sets of datasets (see Table 2 for their characteristics):

- The first set of datasets ($3KMachines$, $30KMachines$, $300KMachines$) was created by generating synthetic event data using information obtained from real logs generated by production machinery. To this end, we retrieved 30,000 events from production machines which covered a full day of event generation.⁷ Then, we computed the probability that an event began or ended at any given point in time. Finally, we constructed our synthetic datasets by generating a fixed number of events that maintained the probability of an event beginning or ending at a particular point in time.
- The second set of datasets ($3KQueries$, $30KQueries$, $300KQueries$) was obtained by collecting real event data from query logs of triple stores exposed on the Web. The data was retrieved from the SPARQL endpoint of the LSQ project [22].⁸ For each dataset, we performed a $SPARQL$ query against the LSQ endpoint and obtained a set of events from a set of consecutive days.

⁷ The source of the events cannot be disclosed due to legal reasons.

⁸ More information can be found at <http://aksw.github.io/LSQ/>

As evaluation measure, we computed the *runtime* of each of the atomic relations, the *time* required by our implementation to perform the *computeRelation* for each Allen Relation (Algorithm 6) and the *total runtime* required for computing all 13 relations. For SILK, we measured the time it required to compute each of the Allen relations.⁹

Table 2. Characteristics of data sets. Size stands for the number of events contained in the dataset.

Log Type	Dataset name	Size	Unique $b(s)$	Unique $e(s)$
Machinery	<i>3KMachines</i>	3,154	960	960
	<i>30KMachines</i>	28,869	960	960
	<i>300KMachines</i>	288,690	960	960
Query	<i>3KQueries</i>	3,888	3,636	3,638
	<i>30KQueries</i>	30,635	3,070	3,070
	<i>300KQueries</i>	303,991	184	184

We set the value of SILK’s block size to 1 ms.¹⁰ Each temporal relation implemented in SILK was given a maximum runtime of 6 hours. We will use the symbol *NA* to signify that a run did not terminate within 6 hours. For the sake of comparison, we also implemented a naive *baseline* for the *eq* relation. This naive implementation performs an exhaustive comparison of the events of *S* and *T* to compute *eq*. For each experiment, we linked each data source with itself, i.e., we set $S = T$. All experiments for all implementations were carried out on the same 20-core Linux Server running *OpenJDK* 64-Bit Server 1.8.0.74 on Ubuntu 14.04.4 LTS on Intel(R) Xeon(R) CPU E5-2650 v3 processors clocked at 2.30GHz. Each experiment was ran on exactly one core using 64 GB of RAM. We implemented AEGLE using Java 1.8.0_60 and the sorting algorithm described in *orderByDate* (Algorithm 4) was performed using the *MergeSort* algorithm [9] as implemented in Java 1.8.0_60 with a guaranteed time complexity $O(n \log n)$.

4.2 Results

To address Q_1 , we computed the execution runtime of all 8 atomic relations as described in Section 3.2. Table 4 shows the runtimes of the atomic relations as well as the total runtime required to run the full set of atomic relations. For our largest dataset *300KQueries*, our approach needs only 84.83s to compute all atomic relations. The maximum required runtime is achieved on the *300KMachines* dataset, where our algorithm needs approximately 7 min. As expected, the atomic relations which rely on equality (i.e., BB^0 , BE^0 , EB^0 , EE^0) require less time than the rest of the atomic relations.

Table 3. Total runtime of Allen Relation for all datasets for AEGLE and SILK. All runtimes are presented in seconds.

Log Type	Dataset Name	Total Runtime		
		AEGLE	AEGLE *	SILK
Machine	<i>3KMachines</i>	11.26	5.51	294.00
	<i>30KMachines</i>	1,016.21	437.79	29,846.00
	<i>300KMachines</i>	189,442.16	78,416.61	NA
Query	<i>3KQueries</i>	26.94	17.91	541.00
	<i>30KQueries</i>	988.78	463.27	33,502.00
	<i>300KQueries</i>	211,996.88	86,884.98	NA

⁹ To measure this time, we contacted the author of [23], who informed us that measuring the duration of the “Match Task” was the way to measure the runtime of his approach.

¹⁰ We contacted the authors of SILK’s temporal relation extension and were informed that this setting should return the best results.

Another interesting observation derived from Table 4 is the relation between the size of the data, the number of the unique $b(s)$ and $e(s)$ among the event sources and the execution runtime of each relation. In the *Machines* datasets, the distribution of beginning and end times is equal among the different sizes of data. As expected by virtue of the complexity of our approach, the total runtimes grow in accordance with $O(n \log n)$ with the increase of the data. From *Query* datasets, we notice that the number of unique $b(s)$ and $e(s)$ has a significant impact on the runtime of our approach. For example, even though *300KQueries* includes 10 times more data than *30KQueries*, *30KQueries* has a significantly higher number of unique $b(s)$ and $e(s)$ than *300KQueries*. Hence, AEGLE requires 15 secs less for *300KQueries* than for the *30KQueries* dataset. The benefits of our implementation can be noticed clearly when comparing AEGLE with the *baseline* (see Table 6). For the *eq* relation, we see that AEGLE is 470 times faster than the brute-force approach. We can thus answer Q_1 by stating that (1) both the number of unique events and the distribution of events across time have a significant influence on the overall runtime and (2) our approach improves the overall runtime of the computation of Allen relations significantly.

Tables 3, 5 and 6 provide us with the insights necessary to answer Q_2 . They show clearly that AEGLE outperforms SILK on all datasets in terms of time efficiency while achieving 100% precision and recall, i.e., while computing all the links that can be found. Therefore, our idea proves to be beneficial and time-efficient for the task of linking temporal data of various sizes.

In more detail, AEGLE requires 211,996.88 s to run the complete computation of Allen relations on our largest dataset (*300KQueries*), whereas SILK is unable to produce full results for any of the relations within the time frame of 280,800 s (3.25 days). *30KQueries* is the largest dataset for which SILK was able to produce links for the given time limit. Here, we observe that AEGLE is more than 33 times faster than SILK. Furthermore, Table 5 suggests that the most costly operations are carried out for the inverse relations. However, by relying on the semantics of Allen relations, we can refrain from computing inverse relations and have them inferred by any forward or backward chaining system. The results under AEGLE* in Table 3 show that overall, the total runtime for computing the seven Allen relations *bf*, *m*, *f*, *st*, *d*, *eq* and *ov* amounts to less than half of AEGLE’s runtime.

To conclude our answer for Q_2 , we studied what would happen if we computed each of the Allen relation individually, i.e., we ran 13 experiments where we set *AIR* to contain exactly one of the Allen relations. We used this setting to allow for a fine-granular comparison of our runtimes with SILK’s. The results of this experiment are shown in Table 6. Overall, we outperform SILK clearly even when computing each of the Allen relations on its own. This suggests that our core implementation for the computation of atomic relations is superior to the generic blocking scheme followed by SILK. This is especially clear when looking at the results on large datasets in more detail. For *30KQueries* for example, SILK needs 2,473 seconds while AEGLE only needs 0.45. The answer to Q_2 is hence that AEGLE outperforms the state of the art in all our experimental settings. Note that the total runtime of a relation is increased by the number of atomic relations involved in its computation when computed using AEGLE. As a result, AEGLE needs more time for the *ovi* relation (which is derived by combining 5 atomic relations) than for *eq* (2 atomic relations).

Table 4. Execution runtime of all 8 atomic relations for all datasets. All runtimes are presented in seconds.

Log Type	Dataset Name	BB^0	BB^1	BE^0	BE^1	EB^0	EB^1	EE^0	EE^1	Total runtime of atomic relations
Machine	<i>3KMachines</i>	0.02	0.41	0.02	0.41	0.02	0.41	0.02	0.42	1.73
	<i>30KMachines</i>	0.19	5.55	0.19	5.51	0.19	5.48	0.18	5.49	22.78
	<i>300KMachines</i>	2.70	95.55	2.14	92.26	3.39	115.66	2.13	94.4	408.23
Query	<i>3KQueries</i>	0.03	2.93	0.03	3.04	0.02	2.89	0.03	2.90	11.87
	<i>30KQueries</i>	0.19	24.5	0.19	26.28	0.21	23.85	0.19	23.80	99.22
	<i>300KQueries</i>	2.52	12.11	1.98	12.57	3.89	25.41	1.93	24.42	84.83

Table 5. Execution runtime of the 13 Allen Relations for all datasets for AEGLE and SILK and *baseline*. The runtimes reported for AEGLE are the times required to perform the set operations necessary to compute each relation. The overall runtimes (i.e., computation of required sets plus times for set operations) are presented in Table 6. All runtimes are presented in seconds.

Relation	Approach	Machine				Query		
		<i>3KMachines</i>	<i>30KMachines</i>	<i>300Machines</i>	<i>3KQueries</i>	<i>30KQueries</i>	<i>300KQueries</i>	
<i>bf</i>	AEGLE	0.00	0.00	0.05	0.00	0.00	0.03	
	SILK	22.00	2,511.00	NA	43.00	2,794.00	NA	
<i>bfi</i>	AEGLE	1.52	127.37	27,103.19	2.37	127.37	32,023.10	
	SILK	24.00	2,547.00	NA	42.00	2,961.00	NA	
<i>m</i>	AEGLE	0.00	0.00	0.03	0.00	0.00	0.00	
	SILK	23.00	2,219.00	NA	41.00	2,466.00	NA	
<i>mi</i>	AEGLE	0.00	0.00	0.03	0.00	0.00	0.00	
	SILK	23.00	2,290.00	NA	44.00	2,584.00	NA	
<i>f</i>	AEGLE	0.73	77.88	13,775.31	1.18	70.53	16,280.24	
	SILK	23.00	2,306.00	NA	41.00	2,531.00	NA	
<i>fi</i>	AEGLE	0.42	47.07	7,837.04	0.62	40.04	8,600.89	
	SILK	23.00	2,305.00	NA	43.00	2,535.00	NA	
<i>st</i>	AEGLE	0.21	29.48	4,849.29	0.34	22.70	5,796.87	
	SILK	21.00	2,166.00	NA	40.00	2,613.00	NA	
<i>sti</i>	AEGLE	0.74	76.14	14,063.02	1.19	69.69	16,270.20	
	SILK	21.00	2,226.00	NA	43.00	2,533.00	NA	
<i>d</i>	AEGLE	1.14	125.20	24,094.20	1.84	107.60	26,213.64	
	SILK	24.00	2,363.00	NA	41.00	2,546.00	NA	
<i>di</i>	AEGLE	1.20	125.04	24,083.00	1.83	108.50	26,149.58	
	SILK	23.00	2,293.00	NA	41.00	2,476.00	NA	
<i>eq</i>	AEGLE	0.01	0.40	45.01	0.00	0.06	344.10	
	SILK	23.00	2,250.00	NA	41.00	2,473.00	NA	
	<i>baseline</i>	2.05	171.10	23,436.30	3.15	196.09	31,452.54	
<i>ov</i>	AEGLE	1.70	182.04	35,244.48	2.68	163.16	38,165.31	
	SILK	22.00	2,181.00	NA	39.00	2,487.00	NA	
<i>ovi</i>	AEGLE	1.87	202.80	37,939.27	3.02	179.90	42,068.13	
	SILK	22.00	2,189.00	NA	42.00	2,503.00	NA	

5 RELATED WORK

Over the past few years, the problem of scalable and time-efficient Link Discovery has been addressed by several approaches and frameworks such as *LIMES*[16], *SILK* [26], *KnoFuss* [18] and *Zhishi.links* [19]. These tools incorporate declarative approaches towards LD, with *SILK* and *KnoFuss* using blocking techniques to identify links between KBs so as to avoid unnecessary comparisons between resources. *LIMES* reduces the time-complexity of the LD procedure by combining techniques such as *PPJoin+* [27] and *HR*³ [15] with set theoretical operators and planning algorithms [17]. *LIMES* provides both theoretical and practical guarantees of completeness and efficiency. A review comprising further LD approaches can be found in [14].

Up until now, only *SILK* provides temporal LD for RDF datasets

by incorporating the recently published work of Smeros et al.[23]. The authors of this paper used MultiBlock to develop an approach for the efficient computation of temporal links. As shown in Section 4.2, AEGLE is able to outperform this approach by 4 orders of magnitude.

In the field of stream reasoning and CEP on Linked Data, there has been a notable amount of research over querying temporal data. For example, *Continuous SPARQL (C-SPARQL)* [4] provides a syntactic and semantic extension of SPARQL to query RDF temporal data by defining a time window for processing events. *C-SPARQL* is able to incrementally re-materialise the input data, using partial static background knowledge. The novel idea behind *C-SPARQL* is the author's contribution to add an *expiration date* to each RDF triple in order to support fast deletion of events that are no longer valid. However, the use of time window frames for linking events prohibits the opportunity of linking previous events with current or future events.

Table 6. Execution runtime of all Allen Relations if computed individually. All runtimes are presented in seconds.

Relation	Approach	Machine			Query		
		3KMachines	30KMachines	300Machines	3KQueries	30KQueries	300KQueries
<i>bf</i>	AEGLE	0.41	5.48	115.71	2.89	23.86	25.44
	SILK	22.00	2,511.00	NA	43.00	2,794.00	NA
<i>bfi</i>	AEGLE	1.95	133.42	27,197.59	5.58	153.84	32,037.64
	SILK	24.00	2,547.00	NA	42.00	2,961.00	NA
<i>m</i>	AEGLE	0.02	0.19	3.42	0.02	0.21	3.89
	SILK	23.00	2,219.00	NA	41.00	2,466.00	NA
<i>mi</i>	AEGLE	0.02	0.20	2.17	0.03	0.19	1.98
	SILK	23.00	2,290.00	NA	44.00	2,584.00	NA
<i>f</i>	AEGLE	1.20	84.18	13,875.70	4.20	95.67	16,296.80
	SILK	23.00	2,306.00	NA	41.00	2,531.00	NA
<i>fi</i>	AEGLE	0.85	53.74	7,934.73	3.57	64.78	8,614.93
	SILK	23.00	2,305.00	NA	43.00	2,535.00	NA
<i>st</i>	AEGLE	0.66	35.23	4,946.39	3.29	46.70	5,823.81
	SILK	21.00	2,166.00	NA	40.00	2,613.00	NA
<i>sti</i>	AEGLE	1.20	83.71	14,162.25	4.13	94.39	16,299.07
	SILK	21.00	2,226.00	NA	43.00	2,533.00	NA
<i>d</i>	AEGLE	2.10	138.55	24,286.85	7.69	156.87	26,252.70
	SILK	24.00	2,363.00	NA	41.00	2,546.00	NA
<i>di</i>	AEGLE	2.15	138.47	24,275.08	7.67	157.75	26,188.04
	SILK	23.00	2,293.00	NA	41.00	2,476.00	NA
<i>eq</i>	AEGLE	0.05	0.79	49.84	0.05	0.45	348.51
	SILK	23.00	2,250.00	NA	41.00	2,473.00	NA
	baseline	2.05	171.10	23,436.30	3.15	196.09	31,452.54
<i>ov</i>	AEGLE	2.96	199.73	35,553.48	11.42	236.87	38,231.15
	SILK	22.00	2,181.00	NA	39.00	2,487.00	NA
<i>ovi</i>	AEGLE	3.16	222.27	38,226.32	11.97	257.59	42,121.68
	SILK	22.00	2,189.00	NA	42.00	2,503.00	NA

Similarly, *Streaming SPARQL* [5] provides an extension of semantics and algebraic functions of *SPARQL* that translates queries into logical algebra plans.

ETALIS is an open-source engine that is able to detect and report changes over events in near real time, by combining both static and streaming knowledge. It incorporates the *ETALIS Language for Events (ELE)* and *Event Processing SPARQL (EP-SPARQL)* [3]. The core of *ETALIS* is implemented in Prolog and incorporates the fundamentals of logic programming: an event is modeled by *ELE*, using logic facts and Prolog-style rules. In addition, *EP-SPARQL* was used to assist real-time complex event detection. In contrast to *C-SPARQL*, using this framework, the user is able to define time windows in the past.

A novel approach in the area of query processing over Linked Stream Data is *C-QUELS* [10]. In this work, the authors proposed a white-box approach for querying stream data efficiently. To this end, they define and use techniques such as query optimisation, caching and indexing. Similarly, *INSTANS* [21] (which is based on the Rete-algorithm) is able to process streams of RDF data and cache the data after the processing is over. Moreover, *INSTANS* is the only approach that supports the simultaneous processing of *SPARQL* queries, where the immediate results of a query can be used from other queries once stored. Additionally, another *SPARQL* query extension language is described in [25], where the authors proposed τ -*SPARQL* that combined with an index structure for temporal intervals achieves better runtime performance.

Most of these approaches focus on extending the semantics and

functions of *SPARQL*. To the best of our knowledge, the only *SPARQL* extension that incorporates Allen's Interval Algebra is *T-SPARQL* [7]. *T-SPARQL* is a temporal extension of *SPARQL* using the multi-temporal RDF database model of [6], using similar design characteristics as *TSQL2* [24]. To query an event KB, *T-SPARQL* enhances the `FILTER` field in order to identify links between monodimensional temporal data. *T-SPARQL* utilises operators that explicitly define the *bf*, *eq*, *ov*, *m* and *di* relations.

6 CONCLUSIONS AND FUTURE WORK

With the use of RDF to represent an ever-growing amount of event data (e.g., for predictive maintenance of industrial machinery) comes the need to compute temporal relations between events. We presented an approach based on the reduction of Allen relations to 8 atomic relations that can be computed efficiently. We showed that by using simple sorting, we can reduce the complexity of computing any of these relations to $O(n \log n)$. Our experiments showed that our approach outperforms the state of the art, which is based on multi-dimensional blocking. In future work, we will extend the scalability of our approach by providing dedicated solutions for load balancing within a parallel execution setting. Moreover, we will study the incremental computation of temporal links on streams of data.

REFERENCES

- [1] James F. Allen, 'Maintaining Knowledge About Temporal Intervals', *Commun. ACM*, **26**(11), 832–843, (November 1983).

- [2] Thomas A Alspaugh, 'Software Support for Calculations in Allen's Interval Algebra', (UCI-ISR-05-2), (February 2005).
- [3] Darko Anicic, Paul Fodor, Sebastian Rudolph, and Nenad Stojanovic, 'EP-SPARQL: a unified language for event processing and stream reasoning', in *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pp. 635–644, New York, NY, USA, (2011). ACM.
- [4] Davide Francesco Barbieri, Daniele Braga, Stefano Ceri, Emanuele Della Valle, and Michael Grossniklaus, 'Incremental Reasoning on Streams and Rich Background Knowledge', in *Proceedings of the 7th International Conference on The Semantic Web: Research and Applications - Volume Part I*, ESWC'10, pp. 1–15, Berlin, Heidelberg, (2010). Springer-Verlag.
- [5] Andre Bolles, Marco Grawunder, and Jonas Jacobi, 'Streaming SPARQL - Extending SPARQL to Process Data Streams', in *Proceedings of the 5th European Semantic Web Conference on The Semantic Web: Research and Applications*, ESWC'08, pp. 448–462, Berlin, Heidelberg, (2008). Springer-Verlag.
- [6] Fabio Grandi, 'Multi-temporal RDF Ontology Versioning', in *International Workshop on Ontology Dynamics*. CEUR-WS, (2009).
- [7] Fabio Grandi, 'T-SPARQL: A TSQL2-like Temporal Query Language for RDF', in *ADBIS (Local Proceedings)*, eds., Mirjana Ivanovic, Bernhard Thalheim, Barbara Catania, and Zoran Budimac, volume 639 of *CEUR Workshop Proceedings*, pp. 21–30. CEUR-WS.org, (2010).
- [8] Robert Isele, Anja Jentzsch, and Christian Bizer, 'Efficient Multidimensional Blocking for Link Discovery without losing Recall', in *Proceedings of the 14th International Workshop on the Web and Databases 2011, WebDB 2011, Athens, Greece, June 12, 2011*, (2011).
- [9] Donald Knuth, *The Art of Computer Programming*. Addison-Wesley, United States, 1968.
- [10] Danh Le-Phuoc, Minh Dao-Tran, Josiane Xavier Parreira, and Manfred Hauswirth, 'A Native and Adaptive Approach for Unified Processing of Linked Streams and Linked Data', in *Proceedings of the 10th International Conference on The Semantic Web - Volume Part I*, ISWC'11, pp. 370–388, Berlin, Heidelberg, (2011). Springer-Verlag.
- [11] Jens Lehmann and Pascal Hitzler, 'Concept learning in description logics using refinement operators', *Machine Learning*, **78**(1), 203–250, (2009).
- [12] M. Loskyll, J. Schlick, S. Hodek, L. Ollinger, T. Gerber, and B. Prvu, 'Semantic service discovery and orchestration for manufacturing processes', in *Emerging Technologies Factory Automation (ETFA), 2011 IEEE 16th Conference on*, pp. 1–8, (Sept 2011).
- [13] Nijat Mehdiyev, Julian Krumeich, David Enke, Dirk Werth, and Peter Loos, 'Determination of Rule Patterns in Complex Event Processing Using Machine Learning Techniques', in *Procedia Computer Science. Complex Adaptive Systems (CAS-15), July 2-4, San Jose, CA, USA*, volume 61, pp. 395–401. Elsevier, (2015).
- [14] Markus Nentwig, Michael Hartung, Axel-Cyrille Ngonga Ngomo, and Erhard Rahm, 'A survey of current Link Discovery frameworks', *Semantic Web*, (Preprint), 1–18, (2015).
- [15] Axel-Cyrille Ngonga Ngomo, 'Link Discovery with Guaranteed Reduction Ratio in Affine Spaces with Minkowski Measures', in *International Semantic Web Conference (1)*, pp. 378–393, (2012).
- [16] Axel-Cyrille Ngonga Ngomo, 'On link discovery using a hybrid approach', *Journal on Data Semantics*, **1**(4), 203–217, (December 2012).
- [17] Axel-Cyrille Ngonga Ngomo, 'HELIOS - Execution Optimization for Link Discovery', in *The Semantic Web - ISWC 2014 - 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I*, pp. 17–32. Springer, (2014).
- [18] Andriy Nikolov, Mathieu d'Aquin, and Enrico Motta, 'Unsupervised learning of link discovery configuration', in *9th Extended Semantic Web Conference (ESWC 2012)*, (2012).
- [19] Xing Niu, Shu Rong, Yunlong Zhang, and Haofen Wang, 'Zhishi.links results for oaei 2011', in *OM*, (2011).
- [20] Mikko Rinne, Eva Blomqvist, Robin Keskisärkkä, and Esko Nuutila, 'Event processing in RDF', in *Proceedings of the 4th International Conference on Ontology and Semantic Web Patterns-Volume 1188*, pp. 52–64. CEUR-WS.org, (2013).
- [21] Mikko Rinne, Esko Nuutila, and Seppo Törmä, 'INSTANS: High-Performance Event Processing with Standard RDF and SPARQL', in *Proceedings of the International Semantic Web Conference (ISWC) 2012 Posters & Demonstrations Track, Boston, USA, November 11-15, 2012*, (2012).
- [22] Muhammad Saleem, Muhammad Intizar Ali, Aidan Hogan, Qaiser Mehmood, and Axel-Cyrille Ngonga Ngomo, *LSQ: The Linked SPARQL Queries Dataset*, 261–269, Springer International Publishing, Cham, 2015.
- [23] Panayiotis Smeros and Manolis Koubarakis, 'Discovering Spatial and Temporal Links among RDF Data', in *Proceedings of the 25th World Wide Web Conference Workshop*, (2016).
- [24] Michael D. Soo and Richard T. Snodgrass, 'Temporal Data Types', in *The TSQL2 Temporal Query Language*, 119–148, (1995).
- [25] Jonas Tappolet and Abraham Bernstein, 'Applied Temporal RDF: Efficient Temporal Querying of RDF Data with SPARQL', in *Proceedings of the 6th European Semantic Web Conference on The Semantic Web: Research and Applications*, ESWC 2009 Heraklion, pp. 308–322, Berlin, Heidelberg, (2009). Springer-Verlag.
- [26] Julius Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov, 'Discovering and Maintaining Links on the Web of Data', in *Proceedings of the 8th International Semantic Web Conference*, ISWC '09, pp. 650–665, Berlin, Heidelberg, (2009). Springer-Verlag.
- [27] Chuan Xiao, Wei Wang, Xuemin Lin, and Jeffrey Xu Yu, 'Efficient similarity joins for near duplicate detection', in *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pp. 131–140, New York, NY, USA, (2008). ACM.

You Can't Always Forget What You Want: On the Limits of Forgetting in Answer Set Programming

Ricardo Gonçalves and Matthias Knorr and João Leite¹

Abstract. Selectively forgetting information while preserving what matters the most is becoming an increasingly important issue in many areas, including in knowledge representation and reasoning. Depending on the application at hand, forgetting operators are defined to obey different sets of desirable properties. Of the myriad of desirable properties discussed in the context of forgetting in Answer Set Programming, *strong persistence*, which imposes certain conditions on the correspondence between the answer sets of the program pre- and post-forgetting, and a certain independence from non-forgotten atoms, seems to best capture its essence, and be desirable in general. However, it has remained an open problem whether it is always possible to forget a set of atoms from a program while obeying strong persistence. In this paper, after showing that it is not always possible to forget a set of atoms from a program while obeying this property, we move forward and precisely characterise what can and cannot be forgotten from a program, by presenting a necessary and sufficient criterion. This characterisation allows us to draw some important conclusions regarding the existence of forgetting operators for specific classes of logic programs, to characterise the class of forgetting operators that achieve the correct result whenever forgetting is possible, and investigate the related question of determining what we can forget from some specific logic program.

1 Introduction

In this paper, we show that it is not always possible to forget some set of atoms from an answer set program while preserving all existing relations between the atoms not to be forgotten, and investigate the *when*, *what*, and *how* related to adequately forgetting a set of atoms from an answer set program.

Whereas keeping memory of information and knowledge has always been at the heart of research in Knowledge Representation and Reasoning, with tight connections to broader areas such as Databases and Artificial Intelligence, we have recently observed a growing attention being devoted to the complementary problem of *forgetting*.

Forgetting – or variable elimination – is an operation that allows the removal of *middle* variables no longer deemed relevant. It is most useful when we wish to eliminate (temporary) variables introduced to represent auxiliary concepts, with the goal of restoring the declarative nature of some knowledge base, or just to simplify it. Furthermore, it is becoming increasingly necessary to properly deal with legal and privacy issues, including, for example, the implementation of court orders to eliminate certain pieces of illegal information. Recent applications of forgetting to cognitive robotics [28, 29, 33], resolving

conflicts [20, 45, 12, 21], and ontology abstraction and comparison [42, 19, 17, 18], further witness its importance.

With its early roots in Boolean Algebra [24], forgetting has been extensively studied in the context of classical logic [3, 20, 22, 23, 30, 31, 43] and, more recently, in the context of logic programming, notably of Answer Set Programming (ASP). The non-monotonic rule-based nature of ASP called for the development of specific methods and techniques – just as it happened with other belief change operations such as revision and update, cf. [2, 8, 34, 35, 36, 6, 37] – resulting in a significant number of different forgetting operators [45, 12, 44, 40, 39, 16, 41, 7], obeying different sets of properties deemed desirable, and often defined for different classes of answer set programs. Such properties include the so-called *consequence persistence*, which requires that the answer sets of the result of forgetting correspond exactly to those of the original program, ignoring the atoms to be forgotten, or *existence* which requires that the result of forgetting belongs to the same class of programs admitted by the forgetting operator, so that the operator can be iterated, among many others. A complete picture of the existing forgetting operators and properties they obey can be found in a recent survey [15].

From observing the landscape of existing operators and properties, one can conclude that there cannot be a one-size-fits-all forgetting operator for ASP, but rather a family of operators, each obeying a specific set of properties. Furthermore, it is clear that not all properties bear the same relevance. Whereas some properties can be very important, such as *existence*, since it guarantees that we can use the same automated reasoners after forgetting, despite not being a property specific of forgetting operators, other properties are less important, sometimes perhaps even questionable, as discussed in [15].

There is nevertheless one property – *strong persistence* [16] – which seems to best capture the essence of forgetting in the context of ASP. The property of *strong persistence* essentially requires that all existing relations between the atoms not to be forgotten be preserved, captured by requiring that there be a correspondence between the answer sets of a program before and after forgetting a set of atoms, and that such correspondence be preserved in the presence of additional rules not containing the atoms to be forgotten. With a slight abuse of using notation that has not been introduced yet, an operator f is said to obey strong persistence if, for any program P and any set of atoms to be forgotten V , it holds that $\mathcal{AS}(f(P, V) \cup R) = \mathcal{AS}(P \cup R)_{\parallel V}$, for all programs R not containing atoms in V , where $f(P, V)$ denotes the result of forgetting V from P , $\mathcal{AS}(P)$ the answer sets of P , and $\mathcal{AS}(P)_{\parallel V}$ their restriction to atoms not in V .

Whereas it seems rather undisputed that *strong persistence* is a desirable property, it is not clear to what extent we can define operators that satisfy it. In [16], the authors propose an operator that obeys such

¹ NOVA LINCS, Departamento de Informática, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa, email: {rjrg|mkn|jleite}@fct.unl.pt.

property, but which is only defined for a restricted class of programs and can only be applied to forget a single atom from a program in a very limited range of situations.

In this paper, we investigate the limits of *forgetting* under *strong persistence*, by answering the following fundamental questions.

Can we always forget some set of atoms from an ASP while obeying strong persistence? This is perhaps the most fundamental open question that remained in [15], not being clear whether such operator does not exist, or simply no one had found it. As the reader can guess from the title of this paper, the answer to this question is negative: sometimes it is simply not possible to forget some set of atoms from a program, while maintaining the relevant relations between other atoms, since the atoms to be forgotten play a pivotal role. From this negative result, the following questions become central, which we will also address in this paper.

When can't we forget some set of atoms from an ASP while obeying strong persistence? We answer this by characterizing when a specific set of atoms cannot be forgotten from a specific program. We define a criterion (Ω) on a program and set of atoms which, when satisfied, implies that such atoms cannot be forgotten from the program.

When (and how) can we forget some set of atoms from an ASP while obeying strong persistence? We answer this by presenting a class of operators that satisfy strong persistence, among many other properties, and show that Ω is both sufficient and necessary to determine when some set of atoms can be forgotten from a program.

What can we forget from a specific ASP while obeying strong persistence? We answer this by providing a constructive definition of the sets of atoms that can be forgotten from a given program. While investigating the answer to this question, we uncover certain classes of programs from which we can always forget any single atom.

Throughout the paper, other relevant intermediate results are shown, and the main concepts illustrated with examples. After the next section with the background on ASP and on forgetting, the remainder of the paper is structured according to the questions above.

2 Forgetting in ASP

In this section, we recall the necessary notions on answer set programming and forgetting, following the presentation in [15].

We assume a propositional language $\mathcal{L}_{\mathcal{A}}$ over a *signature* \mathcal{A} , a finite set of propositional atoms². The *formulas* of $\mathcal{L}_{\mathcal{A}}$ are inductively defined using connectives \perp , \wedge , \vee , and \supset :

$$\varphi ::= \perp \mid p \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \varphi \supset \psi \quad (1)$$

where $p \in \mathcal{A}$. In addition, $\neg\varphi$ and \top are resp. shortcuts for $\varphi \supset \perp$ and $\perp \supset \perp$. Given a finite set S of formulas, $\bigvee S$ and $\bigwedge S$ denote resp. the disjunction and conjunction of all formulas in S . In particular, $\bigvee \emptyset$ and $\bigwedge \emptyset$ stand for resp. \perp and \top , and $\neg S$ and $\neg\neg S$ represent resp. $\{\neg\varphi \mid \varphi \in S\}$ and $\{\neg\neg\varphi \mid \varphi \in S\}$. Unless otherwise stated, we assume that the underlying signature for a particular formula φ is $\mathcal{A}(\varphi)$, the set of atoms appearing in φ .

HT-models Regarding the semantics of propositional formulas, we consider the monotonic logic here-and-there (HT) and equilibrium models [26]. An *HT-interpretation* is a pair $\langle H, T \rangle$ s.t. $H \subseteq T \subseteq \mathcal{A}$. The satisfiability relation in HT, denoted \models_{HT} , is recursively defined as follows for $p \in \mathcal{A}$ and formulas φ and ψ :

- $\langle H, T \rangle \models_{\text{HT}} p$ if $p \in H$;
- $\langle H, T \rangle \not\models_{\text{HT}} \perp$;

- $\langle H, T \rangle \models_{\text{HT}} \varphi \wedge \psi$ if $\langle H, T \rangle \models_{\text{HT}} \varphi$ and $\langle H, T \rangle \models_{\text{HT}} \psi$;
- $\langle H, T \rangle \models_{\text{HT}} \varphi \vee \psi$ if $\langle H, T \rangle \models_{\text{HT}} \varphi$ or $\langle H, T \rangle \models_{\text{HT}} \psi$;
- $\langle H, T \rangle \models_{\text{HT}} \varphi \supset \psi$ if both (i) $T \models \varphi \supset \psi$,³ and (ii) $\langle H, T \rangle \models_{\text{HT}} \varphi$ implies $\langle H, T \rangle \models_{\text{HT}} \psi$.

An *HT-interpretation* $\langle H, T \rangle$ is an *HT-model* of a formula φ if $\langle H, T \rangle \models_{\text{HT}} \varphi$. We denote by $\mathcal{HT}(\varphi)$ the set of all *HT-models* of φ . In particular, $\langle T, T \rangle \in \mathcal{HT}(\varphi)$ is an *equilibrium model* of φ if there is no $T' \subset T$ s.t. $\langle T', T \rangle \in \mathcal{HT}(\varphi)$.

Given two formulas φ and ψ , if $\mathcal{HT}(\varphi) \subseteq \mathcal{HT}(\psi)$, then φ *entails* ψ in HT, written $\varphi \models_{\text{HT}} \psi$. Also, φ and ψ are *HT-equivalent*, written $\varphi \equiv_{\text{HT}} \psi$, if $\mathcal{HT}(\varphi) = \mathcal{HT}(\psi)$.

The *V-exclusion* of a set of HT-interpretations \mathcal{M} , denoted $\mathcal{M}_{\parallel V}$, is $\{\langle X \setminus V, Y \setminus V \rangle \mid \langle X, Y \rangle \in \mathcal{M}\}$. Finally, determining if a formula has an HT-model is NP-complete [32].

Logic Programs An (*extended*) *logic program* P is a finite set of (*extended*) *rules*, i.e., formulas of the form

$$\bigwedge \neg\neg D \wedge \bigwedge \neg C \wedge \bigwedge B \supset \bigvee A, \quad (2)$$

where all elements in $A = \{a_1, \dots, a_k\}$, $B = \{b_1, \dots, b_l\}$, $C = \{c_1, \dots, c_m\}$, $D = \{d_1, \dots, d_n\}$ are atoms.⁴ Such rules r are also commonly written as

$$a_1 \vee \dots \vee a_k \leftarrow b_1, \dots, b_l, \text{not } c_1, \dots, \text{not } c_m, \\ \text{not not } d_1, \dots, \text{not not } d_n, \quad (3)$$

and we will use both forms interchangeably. Given r , we distinguish its *head*, $\text{head}(r) = A$, and its *body*, $\text{body}(r) = B \cup \neg C \cup \neg\neg D$, representing a disjunction and a conjunction.

Any set of (propositional) formulas is HT-equivalent to an (*extended*) logic program [4], which is why we can focus solely on these.

This class of logic programs, \mathcal{C}_e , includes a number of special kinds of rules r : if $n = 0$, then we call r *disjunctive*; if, in addition, $k \leq 1$, then r is *normal*; if on top of that $m = 0$, then we call r *Horn*, and *fact* if also $l = 0$. The classes of *disjunctive*, *normal* and *Horn programs*, \mathcal{C}_d , \mathcal{C}_n , and \mathcal{C}_H , are defined resp. as a finite set of disjunctive, normal, and Horn rules. We have $\mathcal{C}_H \subset \mathcal{C}_n \subset \mathcal{C}_d \subset \mathcal{C}_e$.

We now recall the *answer set semantics* [14] for logic programs. Given a program P and a set I of atoms, the *reduct* P^I is defined as $P^I = \{A \leftarrow B : r \text{ of the form (3) in } P, C \cap I = \emptyset, D \subseteq I\}$. A set I' of atoms is a model of P^I if, for each $r \in P^I$, $I' \models B$ implies $I' \models A$. I is minimal in a set S , denoted by $I \in \text{MIN}(S)$, if there is no $I' \in S$ s.t. $I' \subset I$. Then, I is an *answer set* of P iff I is a minimal model of P^I . Note that, for \mathcal{C}_n and its subclasses, this minimal model is in fact unique. The set of all answer sets of P is denoted by $\mathcal{AS}(P)$. Note that, for \mathcal{C}_d and its subclasses, all $I \in \mathcal{AS}(P)$ are pairwise incomparable. If P has an answer set, then P is *consistent*. Also, the *V-exclusion* of a set of answer sets \mathcal{M} , denoted $\mathcal{M}_{\parallel V}$, is $\{X \setminus V \mid X \in \mathcal{M}\}$. Two programs P_1, P_2 are *equivalent* if $\mathcal{AS}(P_1) = \mathcal{AS}(P_2)$ and *strongly equivalent* if $P_1 \equiv_{\text{HT}} P_2$, i.e., if $\mathcal{AS}(P_1 \cup R) = \mathcal{AS}(P_2 \cup R)$ for any $R \in \mathcal{C}_e$. It is well-known that answer sets and equilibrium models coincide [26], but since the former notion is frequently used in the literature and arguably easier to use, we will mainly rely on it. Also, determining if program P has an answer set is Σ_2^P -complete, and NP-complete if P is normal [5].

Forgetting The principal idea of forgetting in ASP is to remove or hide certain atoms from a given program, while preserving its se-

³ \models is the standard consequence relation from classical logic.

⁴ Extended logic programs [27] are actually more expressive, but this form is sufficient here.

² Often, the term propositional variable is used synonymously.

antics for the remaining atoms. As the result, rather often, a representative up to some notion of equivalence between programs is considered. In this sense, many notions of forgetting for logic programs are defined semantically, i.e., they introduce a class of operators that satisfy a certain semantic characterization. Each single operator in such a class is then a concrete function that, given a program P and a non-empty set of atoms V to be forgotten, returns a unique program, the result of forgetting about V from P . Formally, given a class of logic programs \mathcal{C} over \mathcal{A} , a *forgetting operator (over \mathcal{C})* is a partial function $f : \mathcal{C} \times 2^{\mathcal{A}} \rightarrow \mathcal{C}$ s.t. $f(P, V)$ is a program over $\mathcal{A}(P) \setminus V$, for each $P \in \mathcal{C}$ and $V \in 2^{\mathcal{A}} \setminus \emptyset$. We call $f(P, V)$ the *result of forgetting about V from P* . Whenever $\mathcal{C} = \mathcal{C}_e$, we leave \mathcal{C} implicit. Furthermore, f is called *closed* for $\mathcal{C}' \subseteq \mathcal{C}$ if, for every $P \in \mathcal{C}'$ and $V \in 2^{\mathcal{A}}$, we have $f(P, V) \in \mathcal{C}'$. A *class F of forgetting operators* is a set of forgetting operators. Often, F is defined for a (maximal) class of programs \mathcal{C} , denoted as a *class F of forgetting operators over \mathcal{C}* . The requirement for f being a partial function is a natural one given the existing literature, where some operators as well as classes of these are not closed for certain classes of programs.

Previous work on forgetting in ASP has introduced a variety of desirable properties. Unless stated otherwise, F is a class of forgetting operators, and \mathcal{C} the class of programs over \mathcal{A} of a given $f \in F$.

- (sC) F satisfies *strengthened Consequence* if, for each $f \in F$, $P \in \mathcal{C}$ and $V \subseteq \mathcal{A}$, we have $\mathcal{AS}(f(P, V)) \subseteq \mathcal{AS}(P)_{\parallel V}$.
- (wE) F satisfies *weak Equivalence* if, for each $f \in F$, $P, P' \in \mathcal{C}$ and $V \subseteq \mathcal{A}$, we have $\mathcal{AS}(f(P, V)) = \mathcal{AS}(f(P', V))$ whenever $\mathcal{AS}(P) = \mathcal{AS}(P')$.
- (SE) F satisfies *Strong Equivalence* if, for each $f \in F$, $P, P' \in \mathcal{C}$ and $V \subseteq \mathcal{A}$: if $P \equiv_{\text{HT}} P'$, then $f(P, V) \equiv_{\text{HT}} f(P', V)$.
- (W) F satisfies *Weakening* if, for each $f \in F$, $P \in \mathcal{C}$ and $V \subseteq \mathcal{A}$, we have $P \models_{\text{HT}} f(P, V)$.
- (PP) F satisfies *Positive Persistence* if, for each $f \in F$, $P \in \mathcal{C}$ and $V \subseteq \mathcal{A}$: if $P \models_{\text{HT}} P'$, with $P' \in \mathcal{C}$ and $\mathcal{A}(P') \subseteq \mathcal{A} \setminus V$, then $f(P, V) \models_{\text{HT}} P'$.
- (NP) F satisfies *Negative Persistence* if, for each $f \in F$, $P \in \mathcal{C}$ and $V \subseteq \mathcal{A}$: if $P \not\models_{\text{HT}} P'$, with $P' \in \mathcal{C}$ and $\mathcal{A}(P') \subseteq \mathcal{A} \setminus V$, then $f(P, V) \not\models_{\text{HT}} P'$.
- (SI) F satisfies *Strong (addition) Invariance* if, for each $f \in F$, $P \in \mathcal{C}$ and $V \subseteq \mathcal{A}$, we have $f(P, V) \cup R \equiv_{\text{HT}} f(P \cup R, V)$ for all programs $R \in \mathcal{C}$ with $\mathcal{A}(R) \subseteq \mathcal{A} \setminus V$.
- (Ec) F satisfies *Existence for \mathcal{C}* , i.e., F is *closed for a class of programs \mathcal{C}* if there exists $f \in F$ s.t. f is closed for \mathcal{C} .
- (CP) F satisfies *Consequence Persistence* if, for each $f \in F$, $P \in \mathcal{C}$ and $V \subseteq \mathcal{A}$, we have $\mathcal{AS}(f(P, V)) = \mathcal{AS}(P)_{\parallel V}$.
- (SP) F satisfies *Strong Persistence* if, for each $f \in F$, $P \in \mathcal{C}$ and $V \subseteq \mathcal{A}$, we have $\mathcal{AS}(f(P, V) \cup R) = \mathcal{AS}(P \cup R)_{\parallel V}$, for all programs $R \in \mathcal{C}$ with $\mathcal{A}(R) \subseteq \mathcal{A} \setminus V$.
- (wC) F satisfies *weakened Consequence* if, for each $f \in F$, $P \in \mathcal{C}$ and $V \subseteq \mathcal{A}$, we have $\mathcal{AS}(P)_{\parallel V} \subseteq \mathcal{AS}(f(P, V))$.

Throughout the paper, whenever we write that a single operator f obeys some property, we mean that the singleton class composed of that operator, $\{f\}$, obeys such property.

3 Can We Always Forget?

Among the desirable properties of classes of forgetting operators recalled in the previous section, *strong persistence* (SP) [16] is of particular interest, as it ensures that forgetting preserves all existing relations between all atoms occurring in the program, but the forgotten. In this sense, a class of operators satisfying (SP) removes the desired

atoms, but has no negative semantical effects on the remainder. The importance of (SP) is also witnessed by the fact that a class of operators that satisfies (SP) also satisfies all the other previously mentioned properties with the exception of (W) and (NP), which happen to be equivalent and can hardly be considered desirable [15].

However, determining a forgetting operator that satisfies (SP) is a difficult problem, since, for the verification whether a certain program P' should be the result of forgetting about V from P , none of the well-established equivalence relations can be used, i.e., neither equivalence nor strong equivalence hold in general between P and P' , not even relativized equivalence [10], even though it is close in spirit to the ideas of (SP). Hence, maybe not surprisingly, there is no known general class of operators that satisfies (SP) and which is closed (for the considered class of logic programs).

The two known positive results concerning the satisfiability of (SP) are the existence of several known classes of operators that satisfy (SP) *when restricted to Horn programs* [15], and the existence of one specific operator that, in a very restricted range of situations based on a non-trivial syntactical criterion, permits forgetting about V from P while satisfying (SP) [16]. However, the former result is probably of little relevance given the crucial role played by (default) negation in ASP, while the criterion required in the latter result is certainly too strong, excluding large classes of cases where forgetting about V from P is possible.

All this begs the question of whether there exists a forgetting operator, or a class of these, defined over a class of programs \mathcal{C} beyond the class of Horn programs, that satisfies (SP). The following theorem provides a negative answer to this question.

Theorem 1 *There is no forgetting operator over $\mathcal{C} \supseteq \mathcal{C}_n$ that satisfies (SP).*

Proof: Let \mathcal{C} be a class of programs with $\mathcal{C} \supseteq \mathcal{C}_n$ and suppose there exists f over \mathcal{C} that satisfies (SP). Then, for each $P \in \mathcal{C}$ and $V \subseteq \mathcal{A}$, we have $\mathcal{AS}(f(P, V) \cup R) = \mathcal{AS}(P \cup R)_{\parallel V}$, for all programs $R \in \mathcal{C}$ with $\mathcal{A}(R) \subseteq \mathcal{A} \setminus V$. Consider $P \in \mathcal{C}_n$:

$$a \leftarrow p \quad b \leftarrow q \quad p \leftarrow \text{not } q \quad q \leftarrow \text{not } p$$

We construct $\mathcal{HT}(f(P, \{p, q\}))$, the set of HT-models of the result of forgetting about $V = \{p, q\}$ from P .

We know that $\langle ab, ab \rangle^5$ must be part of $\mathcal{HT}(f(P, \{p, q\}))$, otherwise it would not be possible to obtain the answer set $\{a, b\}$ for the result when adding $R = \{a \leftarrow; b \leftarrow\}$ to $f(P, \{p, q\})$.

At the same time, since $\{a, b\}$ (modulo V) is not an answer set of the original program, $\langle X, ab \rangle \in \mathcal{HT}(f(P, \{p, q\}))$ for at least one $X \subset \{a, b\}$ to prevent $\{a, b\}$ from being an answer set of $f(P, \{p, q\})$. Consider the three alternatives:

- $\langle \emptyset, ab \rangle \notin \mathcal{HT}(f(P, \{p, q\}))$, as adding $R = \{a \leftarrow b; b \leftarrow a\}$, whose HT-models are $\{\langle \emptyset, \emptyset \rangle, \langle \emptyset, ab \rangle, \langle ab, ab \rangle\}$, yields one answer set $\{a, b\}$ for $P \cup R$ (modulo V) which the forgetting result has to preserve;
- $\langle a, ab \rangle \notin \mathcal{HT}(f(P, \{p, q\}))$, since adding $R = \{a \leftarrow\}$, whose HT-models include $\langle a, ab \rangle$, yields one answer set $\{a, b\}$ for $P \cup R$ (modulo V) which the forgetting result has to preserve;
- $\langle b, ab \rangle \notin \mathcal{HT}(f(P, \{p, q\}))$ symmetrically for $R = \{b \leftarrow\}$.

We derive a contradiction. ■

⁵ We follow a common convention and abbreviate sets in HT-interpretations such as $\{a, b\}$ with the sequence of its elements, ab .

Consequently, it is not always possible to forget a set of atoms from a given logic program satisfying the property **(SP)**. In the next section, we address the question of when it is not possible to forget.

4 When Can't We Forget?

Whereas Thm. 1 shows that in general it is not always possible to forget while satisfying **(SP)**, its proof provides some hints on why this is the case. Some atoms play an important role in the program, being pivotal in establishing the relations between the remaining atoms. Therefore, it is simply not possible to forget them and expect that the relations between other atoms be preserved. That is precisely what happens with the pair of atoms p and q in the program

$$a \leftarrow p \quad b \leftarrow q \quad p \leftarrow \text{not } q \quad q \leftarrow \text{not } p$$

presented in the proof of Thm. 1. It is simply not possible to forget them both and expect all the semantic relations between a and b to be kept. No program over atoms $\{a, b\}$ would have the same answer sets as those of the original program (modulo p and q), when both are extended with an arbitrary set of rules over $\{a, b\}$.

This observation immediately leads to one of the central questions here: under what circumstances is it not possible to forget about a given set of atoms V from P while satisfying **(SP)**? In particular, given a concrete program, which sets of atoms play such a pivotal role that they cannot be jointly forgotten without affecting the semantic relations between the remaining atoms in the original program?

To deal with these questions that no longer require the satisfaction of certain properties in general for all programs P and all sets of forgotten atoms V , but rather that **(SP)** holds for a concrete P and set V , we introduce the notion of a *forgetting instance*.

Definition 1 (Forgetting Instance) Let \mathcal{C} be a class of programs over \mathcal{A} . A (forgetting) instance (over \mathcal{C}) is a pair $\langle P, V \rangle$ s.t. $P \in \mathcal{C}$ and $V \subseteq \mathcal{A}$.

We also introduce a restriction of property **(SP)** to operators of forgetting and such forgetting instances.

Definition 2 (Strong Persistence for Forgetting Instance) A forgetting operator f over \mathcal{C} satisfies **(SP)** $_{\langle P, V \rangle}$ if $\mathcal{AS}(f(P, V) \cup R) = \mathcal{AS}(P \cup R)_{\parallel V}$, for all programs $R \in \mathcal{C}$ with $\mathcal{A}(R) \subseteq \mathcal{A} \setminus V$.

Also, f satisfies **(SP)** $_V$ if f satisfies **(SP)** $_{\langle P, V \rangle}$ for all $P \in \mathcal{C}$.

First, we focus on specific classes of programs.

Proposition 1 There is no forgetting operator over \mathcal{C}_n that satisfies **(SP)** $_V$ for any V .

Example 1 Consider forgetting about q from P :

$$p \leftarrow \text{not } q \quad q \leftarrow \text{not } p$$

The only correct result that satisfies the condition of **(SP)** is strongly equivalent to $\{p \leftarrow \text{not not } p\}$, which is not strongly equivalent to any normal program.

This example shows that even if we can forget about some V from a normal program, the result will in general be an extended program, and we would have to revert to using operators over this general class for subsequent forgetting operations.

We could also wonder whether considering \mathcal{C}_d instead of \mathcal{C}_n , in the former proposition, would yield better results, but to no avail.

Proposition 2 There is no forgetting operator over \mathcal{C}_d that satisfies **(SP)** $_V$ for any V .

This can be verified by considering P in Ex. 1, since there is no disjunctive program which is strongly equivalent to the presented result.

The reason why Props. 1 and 2 hold is tied to the fact that the result of forgetting is not within the class of programs considered. Since we are interested in more essential reasons why a set of atoms cannot be forgotten from a program, in what follows, unless otherwise stated, we will focus on forgetting operators over the entire class \mathcal{C}_e .

We now proceed with the introduction of a criterion (Ω) which will play a fundamental role in characterizing the instances for which we cannot expect forgetting operators to satisfy **(SP)** $_{\langle P, V \rangle}$.

Definition 3 (Criterion Ω) Let P be a program over \mathcal{A} and $V \subseteq \mathcal{A}$. An instance $\langle P, V \rangle$ satisfies criterion Ω if there exists $Y \subseteq \mathcal{A} \setminus V$ such that the set of sets

$$\mathcal{R}_{\langle P, V \rangle}^Y = \{R_{\langle P, V \rangle}^{Y, A} \mid A \in \text{Rel}_{\langle P, V \rangle}^Y\}$$

is non-empty and has no least element, where

$$R_{\langle P, V \rangle}^{Y, A} = \{X \setminus V \mid \langle X, Y \cup A \rangle \in \mathcal{HT}(P)\}$$

$$\text{Rel}_{\langle P, V \rangle}^Y = \{A \subseteq V \mid \langle Y \cup A, Y \cup A \rangle \in \mathcal{HT}(P) \text{ and}$$

$$\nexists A' \subset A \text{ such that } \langle Y \cup A', Y \cup A \rangle \in \mathcal{HT}(P)\}.$$

The following example illustrates how criterion Ω can be checked.

Example 2 Recall $P \in \mathcal{C}_n$ used in the proof of Thm. 1:

$$a \leftarrow p \quad b \leftarrow q \quad p \leftarrow \text{not } q \quad q \leftarrow \text{not } p$$

To check if the instance $\langle P, \{p, q\} \rangle$ satisfies criterion Ω , we need to inspect the HT-models of P , $\mathcal{HT}(P)$, which contains 15 elements:

$$\begin{array}{cccc} \langle ap, ap \rangle & \langle bq, abq \rangle & \langle b, abpq \rangle & \langle abp, abpq \rangle \\ \langle bq, bq \rangle & \langle abq, abq \rangle & \langle ab, abpq \rangle & \langle abq, abpq \rangle \\ \langle ap, abp \rangle & \langle \emptyset, abpq \rangle & \langle ap, abpq \rangle & \langle abpq, abpq \rangle \\ \langle abp, abp \rangle & \langle a, abpq \rangle & \langle bq, abpq \rangle & \end{array}$$

To prove that instance $\langle P, \{p, q\} \rangle$ satisfies Ω , we need to find $Y \subseteq \mathcal{A} \setminus V = \{a, b, p, q\} \setminus \{p, q\} = \{a, b\}$ such that $\mathcal{R}_{\langle P, V \rangle}^Y$ is non-empty and has no least element. For that, we only need to focus on those sets $Y' \subseteq \{a, b\}$ for which there exists $\langle H, T \rangle \in \mathcal{HT}(P)$ with $Y' = T \setminus \{p, q\}$, since for all other Y' the set $\mathcal{R}_{\langle P, V \rangle}^{Y'}$ is necessarily empty. In this case, such sets are $Y' = \{b\}$, since there is an HT-model of P of the form $\langle X, bq \rangle$, $Y' = \{a\}$, since there is an HT-model of P of the form $\langle X, ap \rangle$, and $Y' = \{a, b\}$, since there are HT-models of P of the form $\langle X, abp \rangle$, $\langle X, abq \rangle$ and $\langle X, abpq \rangle$. For $Y' = \{b\}$, $\mathcal{R}_{\langle P, V \rangle}^{Y'}$ has only one element and therefore necessarily a least one. The same holds for $Y' = \{a\}$.

We are left to inspect $Y' = \{a, b\}$. For such Y' we need to focus on HT-models of the form $\langle X, abp \rangle$, $\langle X, abq \rangle$ and $\langle X, abpq \rangle$.

Those of the form $\langle X, abpq \rangle$ are however not relevant (for property **(SP)**), since $\{a, b, p, q\}$ can never be an answer set of $P \cup R$ with $\mathcal{A}(R) \subseteq \{a, b\}$. This happens since, besides $\langle abpq, abpq \rangle$, $\langle ab, abpq \rangle$, $\langle abp, abpq \rangle$ and $\langle abq, abpq \rangle$ are also HT-models of P , and since any program R over $\{a, b\}$ cannot distinguish these HT-models, i.e., either all are HT-models of R or none is. Therefore, $\{p, q\} \notin \text{Rel}_{\langle P, V \rangle}^{\{a, b\}}$. Then, $\mathcal{R}_{\langle P, V \rangle}^{\{a, b\}}$ has only two elements, $R_{\langle P, V \rangle}^{\{a, b\}, \{p\}} = \{\{a, b\}, \{a\}\}$ and $R_{\langle P, V \rangle}^{\{a, b\}, \{q\}} = \{\{a, b\}, \{b\}\}$. Since these two sets are incomparable, $\mathcal{R}_{\langle P, V \rangle}^{\{a, b\}}$ has no least element. Therefore, taking $Y = \{a, b\}$, we conclude that $\langle P, \{p, q\} \rangle$ satisfies Ω .

Since criterion Ω heavily relies on HT-models, it helps to observe that, for all instances $\langle P, V \rangle$, any forgetting operator that satisfies **(SP)** $_{\langle P, V \rangle}$ produces a result whose HT-models are a subset of the HT-models of the original program (modulo the forgotten atoms).

Proposition 3 *If a forgetting operator f over \mathcal{C} satisfies $(\mathbf{SP})_{\langle P, V \rangle}$ for an instance $\langle P, V \rangle$ over \mathcal{C} then*

$$\mathcal{HT}(f(P, V)) \subseteq \mathcal{HT}(P)_{\parallel V}.$$

Proof: We first prove that $\langle Y, Y \rangle \in \mathcal{HT}(P)_{\parallel V}$ if $\langle Y, Y \rangle \in \mathcal{HT}(f(P, V))$. Let R be a program over $\mathcal{A}(P) \setminus V$ whose only HT-model of the form $\langle X, Y \rangle$ is $\langle Y, Y \rangle$. Then $Y \in \mathcal{AS}(f(P, V) \cup R)$. Since f satisfies $(\mathbf{SP})_{\langle P, V \rangle}$, we have that $Y \in \mathcal{AS}(P \cup R)_{\parallel V}$. Then, there exists $A \subseteq V$ such that $\langle Y \cup A, Y \cup A \rangle \in \mathcal{HT}(P \cup R)$. In particular $\langle Y \cup A, Y \cup A \rangle \in \mathcal{HT}(P)$, and hence $\langle Y, Y \rangle \in \mathcal{HT}(P)_{\parallel V}$.

Now let $\langle X, Y \rangle \in \mathcal{HT}(f(P, V))$ with $X \subset Y$. We then have that $\langle Y, Y \rangle \in \mathcal{HT}(f(P, V))$, and, as we proved above, $\langle Y, Y \rangle \in \mathcal{HT}(P)_{\parallel V}$. Let R be a program over $\mathcal{A}(P) \setminus V$ whose only HT-models of the form $\langle X', Y \rangle$ are $\langle X, Y \rangle$ and $\langle Y, Y \rangle$. Then, $Y \notin \mathcal{AS}(f(P, V) \cup R)$. Since f satisfies $(\mathbf{SP})_{\langle P, V \rangle}$, we have that $Y \notin \mathcal{AS}(P \cup R)_{\parallel V}$. Since R is a program over $\mathcal{A}(P) \setminus V$, the only HT-models of R of the form $\langle X', Y \cup A \rangle$ with $A \subseteq V$ are of the form $\langle X \cup A', Y \cup A \rangle$ or $\langle Y \cup A', Y \cup A \rangle$ with $A' \subseteq A$. It is easy to prove that there is $A' \subseteq A \subseteq V$ such that $\langle X \cup A', Y \cup A \rangle \in \mathcal{HT}(P)$. Otherwise, taking R' over $\mathcal{A}(P) \setminus V$ whose only HT-model of the form $\langle X, Y \rangle$ is $\langle Y, Y \rangle$, we would have that $Y \notin \mathcal{AS}(P \cup R')$, but $Y \in \mathcal{AS}(P \cup R')$. Therefore, $\langle X, Y \rangle \in \mathcal{HT}(P)_{\parallel V}$. ■

We are now ready to state that Ω is a sufficient condition to determine that some set of atoms V cannot be forgotten from a program P while satisfying *strong persistence*.

Theorem 2 *If $\langle P, V \rangle$ satisfies Ω , then no forgetting operator f satisfies $(\mathbf{SP})_{\langle P, V \rangle}$.*

Proof: Let $\langle P, V \rangle$ be an instance satisfying Ω , and assume that there is a forgetting operator f that satisfies $(\mathbf{SP})_{\langle P, V \rangle}$.

Since $\langle P, V \rangle$ satisfies Ω , we have that there is $Y \subseteq \mathcal{A} \setminus V$ such that the set $\mathcal{R}_{\langle P, V \rangle}^Y = \{R_{\langle P, V \rangle}^{Y, A} \mid A \in \text{Rel}_{\langle P, V \rangle}^Y\}$ – as defined in Def. 3 – has no least element.

First, we prove that $\langle Y, Y \rangle \in \mathcal{HT}(f(P, V))$. Assume it is not the case. Consider the program $R = \{a \leftarrow a \in Y\}$ over $\mathcal{A} \setminus V$. The only HT-model of R of the form $\langle X, Y \rangle$ is $\langle Y, Y \rangle$. Since $\mathcal{R}_{\langle P, V \rangle}^Y$ is non-empty, P must contain an HT-model of the form $\langle Y', Y' \rangle$ such that $Y = Y' \setminus V$ and there is no $\langle X', Y' \rangle$ with $X' \subset Y'$ and $Y = X' \setminus V$. Therefore, $Y \in \mathcal{AS}(P \cup R)_{\parallel V}$, but $Y \notin \mathcal{AS}(f(P, V) \cup R)$, contradicting the assumption that f satisfies $(\mathbf{SP})_{\langle P, V \rangle}$.

We now show that, for each $\langle X, Y \cup A \rangle \in \mathcal{HT}(P)$ such that $A \in \text{Rel}_{\langle P, V \rangle}^Y$ and $X \setminus V \notin \bigcap \mathcal{R}_{\langle P, V \rangle}^Y$, we have that $\langle X \setminus V, Y \rangle \notin \mathcal{HT}(f(P, V))$. Assume that $\langle X \setminus V, Y \rangle \in \mathcal{HT}(f(P, V))$. Consider a program R over $\mathcal{A} \setminus V$ whose HT-models over $\mathcal{A} \setminus V$ of the form $\langle X', Y \rangle$ are only $\langle Y, Y \rangle$ and $\langle X \setminus V, Y \rangle$. Since $X \setminus V \notin \bigcap \mathcal{R}_{\langle P, V \rangle}^Y$, there exists $A' \in \text{Rel}_{\langle P, V \rangle}^Y$ such that $X \setminus V \notin R_{\langle P, V \rangle}^{Y, A'}$. Therefore, there is no $A'' \subseteq A'$ such that $\langle (X \setminus V) \cup A'', Y \cup A' \rangle \in \mathcal{HT}(P)$. Since R is a program over $\mathcal{A} \setminus V$ then $\langle Y, Y \rangle \in \mathcal{HT}(R)$ implies that, for any $A'' \subseteq V$, we have that $\langle Y \cup A'', Y \cup A'' \rangle \in \mathcal{HT}(R)$ over \mathcal{A} . Then, $Y \cup A' \in \mathcal{AS}(P \cup R)$, which implies that $Y \in \mathcal{AS}(P \cup R)_{\parallel V}$. But $Y \notin \mathcal{AS}(f(P, V) \cup R)$, since $\langle X \setminus V, Y \rangle \in \mathcal{HT}(f(P, V) \cup R)$. This contradicts the assumption that f satisfies $(\mathbf{SP})_{\langle P, V \rangle}$.

Recall from Prop. 3 that $\mathcal{HT}(f(P, V)) \subseteq \mathcal{HT}(P)_{\parallel V}$. This, together with $\langle X, Y \rangle \notin \mathcal{HT}(f(P, V))$ for $X \notin \bigcap \mathcal{R}_{\langle P, V \rangle}^Y$, implies that $\{X \mid \langle X, Y \rangle \in \mathcal{HT}(f(P, V))\} \subseteq \bigcap \mathcal{R}_{\langle P, V \rangle}^Y$.

Since we are assuming that $\mathcal{R}_{\langle P, V \rangle}^Y$ has no least element, then, for each $A \in \text{Rel}_{\langle P, V \rangle}^Y$, there exists $X_A \in R_{\langle P, V \rangle}^{Y, A}$ s.t. $X_A \notin \bigcap \mathcal{R}_{\langle P, V \rangle}^Y$. Now take a program R over $\mathcal{A} \setminus V$ whose HT-models of the form

$\langle X', Y \rangle$ are exactly $\langle Y, Y \rangle$ and $\langle X_A, Y \rangle$ for each $A \in \text{Rel}_{\langle P, V \rangle}^Y$. Then we clearly have that $Y \notin \mathcal{AS}(P \cup R)_{\parallel V}$. Since each $\langle X_A, Y \rangle$ cannot belong to $\mathcal{HT}(f(P, V))$ because $X_A \notin \bigcap \mathcal{R}_{\langle P, V \rangle}^Y$, we have that $Y \in \mathcal{AS}(f(P, V) \cup R)$. This is a contradiction to the assumption that f satisfies $(\mathbf{SP})_{\langle P, V \rangle}$. Therefore, f cannot satisfy $(\mathbf{SP})_{\langle P, V \rangle}$. ■

Example 3 *Recall $P \in \mathcal{C}_n$ used in the proof of Thm. 1 and Ex. 2.*

$$a \leftarrow p \quad b \leftarrow q \quad p \leftarrow \text{not } q \quad q \leftarrow \text{not } p$$

Since $\langle P, \{p, q\} \rangle$ satisfies criterion Ω (cf. Ex. 2), no forgetting operator f satisfies $(\mathbf{SP})_{\langle P, \{p, q\} \rangle}$.

We can determine the complexity of testing Ω for a given instance.

Theorem 3 *The complexity of checking if an instance $\langle P, V \rangle$ satisfies criterion Ω is in Π_3^p .*

Proof: We need to find $R_{\langle P, V \rangle}^{Y, A_1}, R_{\langle P, V \rangle}^{Y, A_2} \in \mathcal{R}_{\langle P, V \rangle}^Y$ s.t. both are minimal in $\mathcal{R}_{\langle P, V \rangle}^Y$ and do not coincide to ensure that Ω is satisfied.

We can guess Y, A_1 and A_2 in polynomial time.

The test of whether some $R_{\langle P, V \rangle}^{Y, A} \in \mathcal{R}_{\langle P, V \rangle}^Y$ requires verifying whether $A \in \text{Rel}_{\langle P, V \rangle}^Y$. This can be solved in each case by verifying whether (a) $\langle Y \cup A, Y \cup A \rangle \in \mathcal{HT}(P)$ and whether (b) there is no proper subset A' of A such that $\langle Y \cup A', Y \cup A \rangle \in \mathcal{HT}(P)$. Testing whether (a) a given HT-interpretation is a HT-model can be done in PTIME, and (b) corresponds to a coNP problem with an NP oracle (for finding HT-models). Hence, this test can be done in Π_2^p .

Checking whether each $R_{\langle P, V \rangle}^{Y, A_i}$ is minimal corresponds to a coNP problem using a Σ_2^p oracle for guessing and checking suitable $R_{\langle P, V \rangle}^{Y, A} \in \mathcal{R}_{\langle P, V \rangle}^Y$. Hence, this test can be done in Π_3^p .

Finally, the test of whether the two $R_{\langle P, V \rangle}^{Y, A_i}$ do not coincide can be done by finding a single HT-interpretation $\langle X, Y \cup A_1 \rangle$ such that $X \setminus V \in R_{\langle P, V \rangle}^{Y, A_1}$ but no corresponding $\langle X', Y \cup A_2 \rangle \in \mathcal{HT}(P)$ such that $X' \setminus V \in R_{\langle P, V \rangle}^{Y, A_2}$. This test can be done in coNP with an NP oracle, hence in Π_2^p . Thus, the overall test is in Π_3^p . ■

Even though showing hardness for this result remains open, it seems clear that testing for Ω is non-trivial, the intuitive reason being that besides the complexity of determining HT-models, there are two minimizations to consider, one for $\text{Rel}_{\langle P, V \rangle}^Y$ and one for $\mathcal{R}_{\langle P, V \rangle}^Y$. Still, the complexity is lower than that of computing results of forgetting, which is commonly exponential [15] – *it's not easy to forget*. Nevertheless, if the HT-models have already been computed, checking Ω can be done in linear time on the number of HT-models.

Turning back to the reasons why we cannot forget some sets of atoms from a given program, we can observe – both in the proof of Thm. 1 as well as when the criterion Ω is satisfied – that they seem to be strongly connected to the presence of atoms to be forgotten that depend on themselves via an even (and non-zero) number of negations. This could raise the question of whether it is possible to forget about sets of atoms under certain restrictions. For example, we can verify that criterion Ω is not satisfied for forgetting only about p from the normal program used in the proof of Thm. 1, i.e., we can forget about p in this case. So it would seem that forgetting about certain sets of atoms from normal programs should be a more admissible problem. Unfortunately, this is not the case, since even if we were to delimit V , e.g., by restricting its size, in general, no forgetting operator exists that satisfies $(\mathbf{SP})_V$.

Proposition 4 *There is no forgetting operator over \mathcal{C}_e that satisfies $(\mathbf{SP})_V$ for any V of fixed size.*

Example 4 Consider forgetting about p from P :

$$a \leftarrow p \quad b \leftarrow \text{not } p \quad p \leftarrow \text{not not } p,$$

Even though we forget only about one atom p , the argument in the proof of Thm. 1 applies just as well, and no program $f(P, \{p\})$ exists whose HT-models match the requirements to satisfy $(\mathbf{SP})_V$.

5 When (and How) Can We Forget?

So far we have focused on what cannot be forgotten. We now turn our attention to what can be forgotten.

While Ω allows us to test when it is not possible to forget without sacrificing (\mathbf{SP}) , we do not yet know whether this is a necessary criterion. That being the case would ensure that whenever we forget about V from P , and Ω is not satisfied, $(\mathbf{SP})_{\langle P, V \rangle}$ could be obeyed. In this section, we investigate this matter with the aim of being able to determine when we can forget, and, if possible, to characterise those operators that can obtain the desirable result. To this end, we start by introducing a new class of forgetting operators that computes the HT-models that represent a result of forgetting about V from P .

Definition 4 (SP-Forgetting) Let $F_{\mathbf{SP}}$ be the class of forgetting operators defined by the following set:

$$\{f \mid \mathcal{HT}(f(P, V)) = \{\langle X, Y \rangle \mid Y \subseteq \mathcal{A}(P) \setminus V \wedge X \in \bigcap \mathcal{R}_{\langle P, V \rangle}^Y\}\}$$

As expected, the definition of this class of operators strongly relies on the sets of sets $\mathcal{R}_{\langle P, V \rangle}^Y$, where computing the intersections amounts to determining the least elements in each $\mathcal{R}_{\langle P, V \rangle}^Y$ if they exist, i.e., whenever criterion Ω is not satisfied.

Example 5 Consider the following normal program $P \in \mathcal{C}_n$.

$$a \leftarrow \text{not } p \quad p \leftarrow \text{not } b$$

Forgetting $\{p\}$ from P while satisfying (\mathbf{SP}) should preserve all dependencies between a and b . Such dependencies are internalized in the set of HT-models of P . In this case, all models in $\mathcal{HT}(P)$ are of the form $\langle X, Y \rangle$ with $Y = \{p\}$, $Y = \{a, p\}$, $Y = \{b, p\}$, $Y = \{a, b\}$ or $Y = \{a, b, p\}$. Some of these models are however not relevant from the point of view of (\mathbf{SP}) . For example, since $\langle b, bp \rangle \in \mathcal{HT}(P)$, we have that $\langle b, p \rangle$ can never be an answer set of $(P \cup R)$, for a program R over $\{a, b\}$. This is due to the fact that any R over $\{a, b\}$ cannot distinguish the models $\langle b, bp \rangle$ and $\langle bp, bp \rangle$, i.e., one is a model of R iff the other is. The same type of argument applies to $Y = \{a, b, p\}$ since $\langle ab, abp \rangle \in \mathcal{HT}(P)$. Therefore, $\{p\} \notin \text{Rel}_{\langle P, V \rangle}^{\{b\}}$ and $\{p\} \notin \text{Rel}_{\langle P, V \rangle}^{\{a, b\}}$, which intuitively means that models of P of the form $\langle X, bp \rangle$ and of the form $\langle X, abp \rangle$ are not relevant for the existence of models of the form $\langle X, b \rangle$ and $\langle X, ab \rangle$ in $f(P, \{p\})$, respectively. In the case of $Y = \{b\}$, we have that $\mathcal{R}_{\langle P, V \rangle}^{\{b\}} = \emptyset$, which means that $f(P, V)$ has no HT-model of the form $\langle X, \{b\} \rangle$. For the other possible models we have $\{p\} \in \text{Rel}_{\langle P, \{p\} \rangle}^{\emptyset}$, $\{a\} \in \text{Rel}_{\langle P, \{p\} \rangle}^{\{a\}}$ and $\{a, b\} \in \text{Rel}_{\langle P, \{p\} \rangle}^{\{a, b\}}$. For each such Y , the set $\mathcal{R}_{\langle P, V \rangle}^Y$ has only one element, thus the intersection is precisely that element. This immediately implies that the forgetting instance $\langle P, \{p\} \rangle$ does not satisfy criterion Ω . The resulting set of HT-models of $f(P, V)$ for $f \in F_{\mathbf{SP}}$, is $\{\langle \emptyset, \emptyset \rangle, \langle \emptyset, a \rangle, \langle a, a \rangle, \langle a, ab \rangle, \langle ab, ab \rangle\}$. This means that for any $f \in F_{\mathbf{SP}}$, the program $f(P, V)$ is strongly equivalent to $\{a \leftarrow \text{not not } b\}$.

The close connection between the definition of $F_{\mathbf{SP}}$ and criterion Ω is not mere coincidence. It turns out that every operator in $F_{\mathbf{SP}}$ in fact satisfies (\mathbf{SP}) for those instances $\langle P, V \rangle$ that do not satisfy Ω .

Theorem 4 Every $f \in F_{\mathbf{SP}}$ satisfies $(\mathbf{SP})_{\langle P, V \rangle}$ for every $\langle P, V \rangle$ that does not satisfy Ω .

Proof: Let $f \in F_{\mathbf{SP}}$ and let $\langle P, V \rangle$ be an instance such that $\langle P, V \rangle$ does not satisfy $(\mathbf{SP})_{\langle P, V \rangle}$. Recall that by construction, $\mathcal{HT}(f(P, V)) = \{\langle X, Y \rangle \mid Y \subseteq \mathcal{A}(P) \setminus V \wedge X \in \bigcap \mathcal{R}_{\langle P, V \rangle}^Y\}$. Let R be a program over $\mathcal{A}(P) \setminus V$.

We start by proving that $\mathcal{AS}(f(P, V) \cup R) \subseteq \mathcal{AS}(P \cup R)_{\parallel V}$. Suppose that $Y \in \mathcal{AS}(f(P, V) \cup R)$. Then, $\langle Y, Y \rangle \in \mathcal{HT}(f(P, V) \cup R)$ and there is no $X \subset Y$ such that $\langle X, Y \rangle \in \mathcal{HT}(f(P, V) \cup R)$. We can then conclude that there is no $X \subset Y$ with $X \in \bigcap \mathcal{R}_{\langle P, V \rangle}^Y$ and $\langle X, Y \rangle \in \mathcal{HT}(R)$. Since $\langle Y, Y \rangle \in \mathcal{HT}(f(P, V) \cup R)$, we can conclude that $\mathcal{R}_{\langle P, V \rangle}^Y$ is non-empty, and given that we are assuming that $\langle P, V \rangle$ does not satisfy Ω , the $\mathcal{R}_{\langle P, V \rangle}^Y$ has a least element, which is precisely $\bigcap \mathcal{R}_{\langle P, V \rangle}^Y$. Let $A \in \text{Rel}_{\langle P, V \rangle}^Y$ be such that $R_{\langle P, V \rangle}^{Y, A} = \bigcap \mathcal{R}_{\langle P, V \rangle}^Y$. Then, since $\langle Y, Y \rangle \in \mathcal{HT}(R)$ and $\mathcal{A}(R) \subseteq \mathcal{A}(P) \setminus V$, we have that $\langle Y \cup A, Y \cup A \rangle \in \mathcal{HT}(R)$. Therefore, $\langle Y \cup A, Y \cup A \rangle \in \mathcal{HT}(P \cup R)$. Since $A \in \text{Rel}_{\langle P, V \rangle}^Y$ and there is no $X \subset Y$ with $X \in \bigcap \mathcal{R}_{\langle P, V \rangle}^Y$ such that $\langle X, Y \rangle \in \mathcal{HT}(R)$, we can conclude that there is no $X \subset Y \cup A$ such that $\langle X, Y \cup A \rangle \in \mathcal{HT}(P \cup R)$. Therefore, $Y \cup A \in \mathcal{AS}(P \cup R)$, and so $Y \in \mathcal{AS}(P \cup R)_{\parallel V}$.

We now prove that $\mathcal{AS}(P \cup R)_{\parallel V} \subseteq \mathcal{AS}(f(P, V) \cup R)$. Suppose $Y \in \mathcal{AS}(P \cup R)_{\parallel V}$. Then there exists $A \subseteq V$ such that $\langle Y \cup A, Y \cup A \rangle \in \mathcal{HT}(P \cup R)$ and there is no $X \subset Y \cup A$ such that $\langle X, Y \cup A \rangle \in \mathcal{HT}(P \cup R)$. Therefore, $\mathcal{R}_{\langle P, V \rangle}^Y$ is non-empty and $A \in \text{Rel}_{\langle P, V \rangle}^Y$. Also, there is no $X' \in \bigcap \mathcal{R}_{\langle P, V \rangle}^Y$ such that $\langle X', Y \rangle \in \mathcal{HT}(R)$. Therefore, there is no $\langle X', Y \rangle \in \mathcal{HT}(f(P, V) \cup R)$ such that $X' \subset Y$. In order to conclude that $Y \in \mathcal{AS}(f(P, V) \cup R)$ we just need to prove that $\langle Y, Y \rangle \in \mathcal{HT}(f(P, V) \cup R)$. Since $\langle Y \cup A, Y \cup A \rangle \in \mathcal{HT}(R)$ and $\mathcal{A}(R) \subseteq \mathcal{A}(P) \setminus V$, we have that $\langle Y, Y \rangle \in \mathcal{HT}(R)$. Since $Y \in R_{\langle P, V \rangle}^{Y, A}$ for all $A \in \text{Rel}_{\langle P, V \rangle}^Y$, we clearly have that $Y \in \bigcap \mathcal{R}_{\langle P, V \rangle}^Y$ and therefore, by construction, $\langle Y, Y \rangle \in \mathcal{HT}(f(P, V))$. We can then conclude that $\langle Y, Y \rangle \in \mathcal{HT}(f(P, V) \cup R)$. Therefore, $Y \in \mathcal{AS}(f(P, V) \cup R)$. ■

It immediately follows from Thms. 2 and 4 that Ω is a necessary and sufficient criterion for determining whether it is possible to forget about V from P and preserve (\mathbf{SP}) .

Corollary 1 There is a forgetting operator f that satisfies $(\mathbf{SP})_{\langle P, V \rangle}$ iff $\langle P, V \rangle$ does not satisfy Ω .

Thus, given an instance $\langle P, V \rangle$, we can test whether Ω is not satisfied, i.e., whether we are allowed to forget V from P while preserving (\mathbf{SP}) , in which case we can compute the HT-models that characterise a result using the definition of $F_{\mathbf{SP}}$.

Also, whenever it is possible to forget, the class $F_{\mathbf{SP}}$ precisely characterises the result of forgetting.

Proposition 5 Let $\langle P, V \rangle$ be an instance that does not satisfy Ω . Then, for every forgetting operator f satisfying $(\mathbf{SP})_{\langle P, V \rangle}$ and every $f' \in F_{\mathbf{SP}}$ we have that $f(P, V) \equiv_{\mathbf{HT}} f'(P, V)$.

For the particular case of Horn programs, it is possible to simplify the construction of the result since the set of HT-models of the result of forgetting coincides with the set of HT-models of the original program, modulo the forgotten atoms. A side-effect of this is that the result is guaranteed to be itself a Horn program.

Proposition 6 Let f be in $F_{\mathbf{SP}}$. Then, for every $V \subseteq \mathcal{A}$:

$$\mathcal{HT}(f(P, V)) = \mathcal{HT}(P)_{\parallel V} \text{ for } P \in \mathcal{C}_H \text{ and } f(P, V) \in \mathcal{C}_H.$$

Proof: We know from [9] that the HT-models M of Horn programs are characterized by the following conditions:

- (A) $\langle X, Y \rangle \in M, Y \subseteq Y'$ and $\langle Y', Y' \rangle \in M \Rightarrow \langle X, Y' \rangle \in M$
- (B) $\langle X, Y \rangle \in M$ iff $X \subseteq Y, \langle X, X \rangle \in M$ and $\langle Y, Y \rangle \in M$
- (C) $\langle X, Y \rangle \in M$ and $\langle H, T \rangle \in M \Rightarrow \langle X \cap H, Y \cap T \rangle \in M$

We first prove that $\mathcal{HT}(f(P, V)) = \mathcal{HT}(P)_{\parallel V}$ for $P \in \mathcal{C}_H$. Let us prove the two inclusions. Consider $\langle X, Y \rangle \in \mathcal{HT}(f(P, V))$. Then $Y \subseteq \mathcal{A}(P) \setminus V$ and $X \in \bigcap \mathcal{R}_{(P, V)}^Y$, i.e., there is at least one $A \in \text{Rel}_{(P, V)}^Y$ such that $X \in R_{(P, V)}^{Y, A}$. Hence, there is $\langle X', Y \cup A \rangle \in \mathcal{HT}(P)$ with $X = X' \setminus V$.

Now consider $\langle X, Y \rangle \in \mathcal{HT}(P)_{\parallel V}$. Then, there exists $A' \subseteq A \subseteq V$ such that $\langle X \cup A', Y \cup A \rangle \in \mathcal{HT}(P)$. Let $A'' \in \text{Rel}_{(P, V)}^Y$. Then $\langle Y \cup A'', Y \cup A'' \rangle \in \mathcal{HT}(P)$. Using condition (C) we can conclude that $\langle (X \cup A') \cap (Y \cup A''), (Y \cup A) \cap (Y \cup A'') \rangle \in \mathcal{HT}(P)$, i.e., $\langle X \cup (A' \cap A''), Y \cup (A \cap A'') \rangle \in \mathcal{HT}(P)$. Since $Y \cup (A \cap A'') \subseteq Y \cup A''$, and using condition (A) we can conclude that $\langle X \cup (A' \cap A''), Y \cup A'' \rangle \in \mathcal{HT}(P)$, and consequently $X \in R_{(P, V)}^{Y, A'}$. Hence $X \in \bigcap \mathcal{R}_{(P, V)}^Y$, and therefore $\langle X, Y \rangle \in \mathcal{HT}(f(P, V))$.

We now prove that for $P \in \mathcal{C}_H$ we have $f(P, V) \in \mathcal{C}_H$. So, we need to prove that conditions (A), (B) and (C) hold for $\mathcal{HT}(f(P, V))$.

For condition (A) take $\langle X, Y \rangle \in \mathcal{HT}(f(P, V))$ and $\langle Y', Y' \rangle \in \mathcal{HT}(f(P, V))$ such that $Y \subseteq Y'$. We aim to prove that $\langle X, Y' \rangle \in \mathcal{HT}(f(P, V))$. By definition $X \in \mathcal{R}_{(P, V)}^Y$ and $Y' \in \mathcal{R}_{(P, V)}^{Y'}$. Therefore, $\langle X \cup A, Y \cup A' \rangle \in \mathcal{HT}(P)$ for some $A \subseteq A' \subseteq V$. Also, $\langle Y' \cup A'', Y' \cup A'' \rangle \in \mathcal{HT}(P)$ for every $A'' \in \text{Rel}_{(P, V)}^{Y'}$. Then, since $\mathcal{HT}(P)$ satisfies condition (B), we can easily conclude that $\langle X \cup (A \cap A''), Y' \cup A'' \rangle \in \mathcal{HT}(P)$ for every $A'' \in \text{Rel}_{(P, V)}^{Y'}$. Therefore, $X \in \bigcap \mathcal{R}_{(P, V)}^{Y'}$, hence $\langle X, Y' \rangle \in \mathcal{HT}(f(P, V))$.

For condition (B), first take $\langle X, Y \rangle \in \mathcal{HT}(f(P, V))$. We aim to prove that $\langle X, X \rangle \in \mathcal{HT}(f(P, V))$ and $\langle Y, Y \rangle \in \mathcal{HT}(f(P, V))$. Since $\langle X, Y \rangle \in \mathcal{HT}(f(P, V))$ we have that $X \in \bigcap \mathcal{R}_{(P, V)}^Y$. Then, $\langle X \cup A', Y \cup A'' \rangle \in \mathcal{HT}(P)$ for some $A' \subseteq A'' \subseteq V$. Using condition (B) for $\mathcal{HT}(P)$ we have that $\langle X \cup A', X \cup A' \rangle \in \mathcal{HT}(P)$ and $\langle Y \cup A'', Y \cup A'' \rangle \in \mathcal{HT}(P)$. Using again condition (B), we can conclude that for every $A''' \in \text{Rel}_{(P, V)}^X$ we have $\langle X \cup (A' \cap A'''), X \cup A''' \rangle \in \mathcal{HT}(P)$. Hence, $X \in \bigcap \mathcal{R}_{(P, V)}^X$, and therefore $\langle X, X \rangle \in \mathcal{HT}(f(P, V))$. In the same way, we can conclude that for every $A''' \in \text{Rel}_{(P, V)}^Y$ we have $\langle Y \cup (A'' \cap A'''), Y \cup A''' \rangle \in \mathcal{HT}(P)$. Hence, $Y \in \bigcap \mathcal{R}_{(P, V)}^Y$, and therefore $\langle Y, Y \rangle \in \mathcal{HT}(f(P, V))$.

For the converse, assume that $\langle X, X \rangle \in \mathcal{HT}(f(P, V))$ and $\langle Y, Y \rangle \in \mathcal{HT}(f(P, V))$. We aim to prove that $\langle X, Y \rangle \in \mathcal{HT}(f(P, V))$. Since $\langle X, X \rangle \in \mathcal{HT}(f(P, V))$ and $\langle Y, Y \rangle \in \mathcal{HT}(f(P, V))$ we have that $X \in \bigcap \mathcal{R}_{(P, V)}^X$ and $Y \in \bigcap \mathcal{R}_{(P, V)}^Y$.

Then, there is $A' \subseteq V$ such that $\langle X \cup A', X \cup A' \rangle \in \mathcal{HT}(P)$. Also, for every $A'' \in \text{Rel}_{(P, V)}^Y$ we have $\langle Y \cup A'', Y \cup A'' \rangle \in \mathcal{HT}(P)$. Since $X \subseteq Y$ and using condition (C) we have that $\langle X \cup (A' \cap A''), X \cup (A' \cap A'') \rangle \in \mathcal{HT}(P)$. Since $X \cup (A' \cap A'') \subseteq Y \cup A''$ we can use condition (B) to conclude that $\langle X \cup (A' \cap A''), Y \cup A'' \rangle \in \mathcal{HT}(P)$, therefore $X \in \bigcap \mathcal{R}_{(P, V)}^Y$. Then $\langle X, Y \rangle \in \mathcal{HT}(f(P, V))$.

For condition (C) take $\langle X, Y \rangle \in \mathcal{HT}(f(P, V))$ and $\langle H, T \rangle \in \mathcal{HT}(f(P, V))$. We aim to prove that $\langle X \cap H, Y \cap T \rangle \in \mathcal{HT}(f(P, V))$. Since $\langle X, Y \rangle \in \mathcal{HT}(f(P, V))$ and $\langle H, T \rangle \in \mathcal{HT}(f(P, V))$ we have that $X \in \bigcap \mathcal{R}_{(P, V)}^Y$ and $H \in \bigcap \mathcal{R}_{(P, V)}^T$. Then, there are $A, A', A'', A''' \subseteq V$ such that $\langle X \cup A, Y \cup A' \rangle \in \mathcal{HT}(P)$ and $\langle H \cup A'', T \cup A''' \rangle \in \mathcal{HT}(P)$. Using condition (C) for $\mathcal{HT}(P)$ we have that $\langle (X \cap H) \cup (A \cap A''), (Y \cap T) \cup (A' \cap A''') \rangle \in \mathcal{HT}(P)$. Therefore, it follows easily from condition (B)

that $\langle (X \cap H) \cup (A \cap A'' \cap A'''), (Y \cap T) \cup A''' \rangle \in \mathcal{HT}(P)$ for any $A''' \in \text{Rel}_{(P, V)}^{Y \cap T}$. Therefore $(X \cap H) \in \bigcap \mathcal{R}_{(P, V)}^{Y \cap T}$, and hence $\langle X \cap H, Y \cap T \rangle \in \mathcal{HT}(f(P, V))$. ■

We can now show that F_{SP} is closed in the general case and for Horn programs, but not for disjunctive or normal programs. This turns out to be similar as for previous classes of operators defined for the class of extended programs that are based on manipulating HT-models, namely HT-forgetting [41] and SM-forgetting [39].

Theorem 5 F_{SP} is closed for extended programs and Horn programs, but neither for disjunctive programs nor normal programs.

Proof: F_{SP} is naturally closed for \mathcal{C}_e , since $\langle X, Y \rangle \in \mathcal{HT}(f(P, V))$, by construction, implies $\langle Y, Y \rangle \in \mathcal{HT}(f(P, V))$. The negative results for $(\mathbf{E}_{\mathcal{C}_d})$ and $(\mathbf{E}_{\mathcal{C}_n})$ follow from Props. 1 and 2.

Regarding $(\mathbf{E}_{\mathcal{C}_H})$, by Prop. 6, $f(P, V) \in \mathcal{C}_H$ for $P \in \mathcal{C}_H$. ■

We state the result in the general case here, i.e., independently of whether we are allowed to forget or not, but these results obviously apply in exactly the same way if we restrict our attention to the cases where we can forget about some V from a given program P .

If we restrict our attention to the cases where we can forget, i.e., where the considered instance does not satisfy Ω , then most of the properties mentioned in Sec. 2 are satisfied.

Theorem 6 Restricted to instances $\langle P, V \rangle$ that do not satisfy Ω , F_{SP} satisfies (\mathbf{sC}) , (\mathbf{wE}) , (\mathbf{SE}) , (\mathbf{PP}) , (\mathbf{SI}) , (\mathbf{CP}) , (\mathbf{SP}) and (\mathbf{wC}) .

Proof: By Thm 4, every $f \in F_{\text{SP}}$ satisfies $(\mathbf{SP})_{\langle P, V \rangle}$ for every $\langle P, V \rangle$ that does not satisfy Ω . Hence, F_{SP} satisfies (\mathbf{SP}) under this restriction. Then, by Prop. 1 of [15], F_{SP} also satisfies (\mathbf{sC}) , (\mathbf{wE}) , (\mathbf{SE}) , (\mathbf{PP}) , (\mathbf{SI}) , (\mathbf{CP}) , (\mathbf{wC}) , but neither (\mathbf{W}) nor (\mathbf{NP}) . ■

The properties which are not satisfied – (\mathbf{W}) and (\mathbf{NP}) – have been proved orthogonal to (\mathbf{SP}) [15], hence of little relevance in our view.

Theorem 7 Let f be in F_{SP} , $P, P' \in \mathcal{C}_e$ and $V \subseteq \mathcal{A}$. Deciding if $P' \equiv_{\text{HT}} f(P, V)$ is in Π_2^p .

Proof: If $P' \not\equiv_{\text{HT}} f(P, V)$, then there exists an HT-interpretation $\langle X, Y \rangle$ such that either (a) $\langle X, Y \rangle \models_{\text{HT}} P'$ and $\langle X, Y \rangle \not\models_{\text{HT}} f(P, V)$, or (b) $\langle X, Y \rangle \not\models_{\text{HT}} P'$ and $\langle X, Y \rangle \models_{\text{HT}} f(P, V)$.

Consider (a). Checking whether $\langle X, Y \rangle \models_{\text{HT}} P'$ can be done in PTIME. Checking whether $\langle X, Y \rangle \not\models_{\text{HT}} f(P, V)$ amounts to verifying that $X \notin \bigcap \mathcal{R}_{(P, V)}^Y$. This holds if either $\bigcap \mathcal{R}_{(P, V)}^Y$ is empty or there is at least one $R_{(P, V)}^{Y, A} \in \mathcal{R}_{(P, V)}^Y$ such that $X \notin R_{(P, V)}^{Y, A}$. The first case corresponds to the co-problem of finding at least one $R_{(P, V)}^{Y, A} \in \mathcal{R}_{(P, V)}^Y$, which is in Π_2^p (see proof of Thm. 3). The second also requires finding such $R_{(P, V)}^{Y, A} \in \mathcal{R}_{(P, V)}^Y$, but also to ensure that $\langle X \cup V', Y \cup A \rangle$ is not an HT-model of P for any $V' \subseteq A$ using an NP-oracle. Hence, (a) is in Π_2^p . A similar argument holds for (b). ■

Still, computing the actual result (and not just its representation in terms of HT-models) is exponential [15]. This high computational complexity of computing a result, together with the fact that the test used in the proof of Thm. 7 does not check if Ω holds, justifies the use of our test for criterion Ω before proceeding with the operation.

6 What Can We Forget?

We now approach the problem from a different angle, and determine which sets of atoms can be forgotten from a specific program.

We begin with the case where the set of atoms to be forgotten is a singleton and the program normal. As it turns out, we can always forget single atoms from normal programs without having to test Ω .

Proposition 7 *There is a forgetting operator satisfying $(\mathbf{SP})_{\langle P, V \rangle}$, for every $P \in \mathcal{C}_n$ and every V such that $|V|=1$.*

Proof: Let $P \in \mathcal{C}_n$ and assume that $|V|=1$. We aim to prove that $\langle P, V \rangle$ does not satisfy Ω . Since $|V|=1$, for every $Y \subseteq \mathcal{A}(P) \setminus V$, the set $\mathcal{R}_{\langle P, V \rangle}^Y$ has at most two elements: $R_{\langle P, V \rangle}^{Y, \emptyset}$ and $R_{\langle P, V \rangle}^{Y, V}$. If $\mathcal{R}_{\langle P, V \rangle}^Y$ has at most one element then either the set is empty or it has a least element. Suppose both $R_{\langle P, V \rangle}^{Y, \emptyset}$ and $R_{\langle P, V \rangle}^{Y, V}$ belong to $\mathcal{R}_{\langle P, V \rangle}^Y$. Then both $\langle Y, Y \rangle$ and $\langle Y \cup V, Y \cup V \rangle$ are HT-models of P .

We now prove that $R_{\langle P, V \rangle}^{Y, \emptyset} \subseteq R_{\langle P, V \rangle}^{Y, V}$. Let $X \in R_{\langle P, V \rangle}^{Y, \emptyset}$. Then $\langle X, Y \rangle \in \mathcal{HT}(P)$. We can apply one of the conditions that characterize the possible classes of models of programs in \mathcal{C}_n : if $\langle H, T \rangle \in \mathcal{HT}(P)$ and $\langle T', T' \rangle \in \mathcal{HT}(P)$ such that $T \subseteq T'$, then $\langle H, T' \rangle \in \mathcal{HT}(P)$. Since $\langle X, Y \rangle \in \mathcal{HT}(P)$ and $\langle Y \cup V, Y \cup V \rangle \in \mathcal{HT}(P)$ we have that $\langle X, Y \cup V \rangle \in \mathcal{HT}(P)$, and therefore $X \setminus V = X \in R_{\langle P, V \rangle}^{Y, V}$. We can conclude that $R_{\langle P, V \rangle}^{Y, \emptyset} \subseteq R_{\langle P, V \rangle}^{Y, V}$, and so $R_{\langle P, V \rangle}^{Y, \emptyset}$ is the least element of $\mathcal{R}_{\langle P, V \rangle}^Y$. Therefore, $\langle P, V \rangle$ does not satisfy Ω for every $P \in \mathcal{C}_n$ and V such that $|V|=1$. ■

This result also holds for the class of disjunctive programs, whose proof, which we omit for lack of space, follows a similar strategy.

Proposition 8 *There is a forgetting operator satisfying $(\mathbf{SP})_{\langle P, V \rangle}$, for every $P \in \mathcal{C}_d$ and every V such that $|V|=1$.*

As indicated in Thm. 5, no operator in $\mathbf{F}_{\mathbf{SP}}$ is closed for normal or disjunctive programs, hence quite likely the result will not be applicable throughout an iterative process of forgetting one atom at a time. But for a one-time forgetting operation, they might be useful.

We now provide a general way to determine which sets of atoms can be forgotten from a given program.

Theorem 8 *Let P be a program. Then the set of sets of atoms*

$\mathcal{V}_P = \{V \subseteq \mathcal{A}(P) \mid \bigcap \mathcal{R}_{\langle P, V \rangle}^Y \in \mathcal{R}_{\langle P, V \rangle}^Y \text{ for every } \mathcal{R}_{\langle P, V \rangle}^Y \neq \emptyset\}$
is the set of all sets V of atoms for which it is possible to forget V from P while satisfying $(\mathbf{SP})_{\langle P, V \rangle}$.

Proof: Let P be a program. Using Cor. 1 we need to check that \mathcal{V}_P is exactly the set of all $V \subseteq \mathcal{A}(P)$ such that the instance $\langle P, V \rangle$ does not satisfies criterion Ω .

Let $V \in \mathcal{V}_P$. By definition of \mathcal{V}_P , we have that for every $Y \subseteq \mathcal{A}(P) \setminus V$ either $\mathcal{R}_{\langle P, V \rangle}^Y = \emptyset$ or $\bigcap \mathcal{R}_{\langle P, V \rangle}^Y \in \mathcal{R}_{\langle P, V \rangle}^Y$. Since clearly $\bigcap \mathcal{R}_{\langle P, V \rangle}^Y$ is the least element of $\mathcal{R}_{\langle P, V \rangle}^Y$, $\langle P, V \rangle$ cannot satisfy Ω .

Now let $V \notin \mathcal{V}_P$. Then, by definition of \mathcal{V}_P there exists $Y \subseteq \mathcal{A}(P) \setminus V$ such that $\mathcal{R}_{\langle P, V \rangle}^Y \neq \emptyset$ and $\bigcap \mathcal{R}_{\langle P, V \rangle}^Y \notin \mathcal{R}_{\langle P, V \rangle}^Y$. Suppose that $\mathcal{R}_{\langle P, V \rangle}^Y$ has a least element, call it L . Then $L \subseteq R$, for every $R \in \mathcal{R}_{\langle P, V \rangle}^Y$. Therefore $L \subseteq \bigcap \mathcal{R}_{\langle P, V \rangle}^Y$. Since $L \in \mathcal{R}_{\langle P, V \rangle}^Y$ we have that $\bigcap \mathcal{R}_{\langle P, V \rangle}^Y \subseteq L$. Thus $L = \bigcap \mathcal{R}_{\langle P, V \rangle}^Y$, which contradicts the fact that $\bigcap \mathcal{R}_{\langle P, V \rangle}^Y \notin \mathcal{R}_{\langle P, V \rangle}^Y$. Therefore, $\mathcal{R}_{\langle P, V \rangle}^Y$ has no least element, and we can conclude that $\langle P, V \rangle$ satisfies criterion Ω . ■

Thus, this result provides a general way to obtain all sets of atoms V that can be forgotten from a given program P while preserving (\mathbf{SP}) . Notably, all possible sets contained in \mathcal{V}_P have to be determined individually as neither sub- nor supersets are necessarily contained in the result. For example, for program P given in Ex. 2, \mathcal{V}_P contains $\{p\}$, $\{q\}$, and, e.g., $\{p, q, a\}$, but not $\{p, q\}$.

7 Conclusions

We have studied forgetting in ASP, focusing on what is perhaps its most crucial property – *strong persistence* (\mathbf{SP}) – which captures the essence of forgetting in ASP by ensuring that all semantic relations between the atoms not forgotten are preserved.

We began by answering an important open question, showing that it is not always possible to forget a set of atoms while obeying (\mathbf{SP}) .

Departing from this impossibility result, we conducted a thorough study of the limits of forgetting in ASP, including a necessary and sufficient criterion (Ω) to determine whether a particular set of atoms can be forgotten from a program while obeying (\mathbf{SP}) . Whereas at a technical level, criterion Ω is closely tied to certain conditions on the HT-models of the program at hand, it seems that what cannot be forgotten from a program are atoms used in rules that are somehow equivalent to *choice rules* [25], and those atoms are pivotal in the sense that they play an active role in determining the truth of other atoms in some answer sets i.e., there are rules whose bodies mention these atoms and they are true at least in some answer sets. Further investigating this conjecture is an interesting line of future work.

We have also introduced a new class of operators that allows us to show how to forget a set of atoms from a given program while preserving (\mathbf{SP}) . It is worth noting that of the many classes of operators investigated in [15], one, dubbed $\mathbf{F}_{\mathbf{Sas}}$, also satisfies (\mathbf{SP}) . However, $\mathbf{F}_{\mathbf{Sas}}$ is only partially defined, witnessed by the fact that the only known operator in $\mathbf{F}_{\mathbf{Sas}}$, dubbed $\mathbf{f}_{\mathbf{Sas}}$, is only defined for a non-standard class of programs (permitting double negation but no disjunctions), and can only be applied to forget about single atoms p if a sufficient (but not necessary) criterion, called *p-forgettable* (see [16]), holds, which is considerably stronger than Ω , hence unnecessarily excluding many possible cases. Nevertheless, for the sake of completeness, we formally relate such operator and the class $\mathbf{F}_{\mathbf{SP}}$.

Corollary 2 *Let $\mathbf{f}_{\mathbf{Sas}}$ be the operator defined for $\mathbf{F}_{\mathbf{Sas}}$ and P a program for which $\mathbf{f}_{\mathbf{Sas}}$ is defined. For any $\mathbf{f} \in \mathbf{F}_{\mathbf{SP}}$, if a single atom p is *p-forgettable* from P , then $\mathcal{HT}(\mathbf{f}(P, \{p\})) = \mathcal{HT}(\mathbf{f}_{\mathbf{Sas}}(P, \{p\}))$.*

We also provided a general condition to determine all sets of atoms that can be forgotten from a given program, as well as special cases in which a set of atoms can always be forgotten, namely singleton sets in the case of normal and disjunctive programs.

Left open, for future work, is the definition of a syntactic operator similar in style to that defined for strong as-forgetting [16], as well as its implementation. Also, we may investigate the relation of Ω to projections of answer sets [11] and investigate the limits of forgetting for semantics other than ASP, such as [41] based on the FLP-semantics [38], or [1, 16] based on the well-founded semantics [13].

ACKNOWLEDGEMENTS

We would like to thank the reviewers for their comments, which helped improve this paper. R. Gonçalves, M. Knorr and J. Leite were partially supported by FCT under strategic project NOVA LINCS (PEst/UID/CEC/04516/2013). R. Gonçalves was partially supported by FCT grant SFRH/BPD/100906/2014 and M. Knorr by FCT grant SFRH/BPD/86970/2012.

REFERENCES

- [1] José Júlio Alferes, Matthias Knorr, and Kewen Wang, 'Forgetting under the well-founded semantics', in *Procs. of LPNMR*, eds., Pedro Cabalar and Tran Cao Son, volume 8148 of *LNCS*, pp. 36–41. Springer, (2013).
- [2] José Júlio Alferes, João Alexandre Leite, Luís Moniz Pereira, Halina Przymusińska, and Teodor C. Przymusiński, 'Dynamic updates of non-monotonic knowledge bases', *The Journal of Logic Programming*, **45**(1-3), 43–70, (September/October 2000).
- [3] W. W. Bledsoe and Larry M. Hines, 'Variable elimination and chaining in a resolution-based prover for inequalities', in *Procs. of CADE*, eds., Wolfgang Bibel and Robert A. Kowalski, volume 87 of *LNCS*, pp. 70–87. Springer, (1980).
- [4] Pedro Cabalar and Paolo Ferraris, 'Propositional theories are strongly equivalent to logic programs', *TPLP*, **7**(6), 745–759, (2007).
- [5] Evgeny Dantsin, Thomas Eiter, Georg Gottlob, and Andrei Voronkov, 'Complexity and expressive power of logic programming', *ACM Comput. Surv.*, **33**(3), 374–425, (2001).
- [6] James P. Delgrande, Torsten Schaub, Hans Tompits, and Stefan Woltran, 'A model-theoretic approach to belief change in answer set programming', *ACM Trans. Comput. Log.*, **14**(2), 14, (2013).
- [7] James P. Delgrande and Kewen Wang, 'A syntax-independent approach to forgetting in disjunctive logic programs', in *Procs. of AAAI*, eds., Blai Bonet and Sven Koenig, pp. 1482–1488. AAAI Press, (2015).
- [8] Thomas Eiter, Michael Fink, Giuliana Sabbatini, and Hans Tompits, 'On properties of update sequences based on causal rejection', *Theory and Practice of Logic Programming (TPLP)*, **2**(6), 721–777, (2002).
- [9] Thomas Eiter, Michael Fink, Hans Tompits, and Stefan Woltran, 'Simplifying logic programs under uniform and strong equivalence', in *Procs. of LPNMR*, eds., Vladimir Lifschitz and Ilkka Niemelä, volume 2923 of *LNCS*, pp. 87–99. Springer, (2004).
- [10] Thomas Eiter, Michael Fink, and Stefan Woltran, 'Semantical characterizations and complexity of equivalences in answer set programming', *ACM Trans. Comput. Log.*, **8**(3), (2007).
- [11] Thomas Eiter, Hans Tompits, and Stefan Woltran, 'On solution correspondences in answer-set programming', in *Procs. of IJCAI*, eds., Leslie Pack Kaelbling and Alessandro Saffiotti, pp. 97–102. Professional Book Center, (2005).
- [12] Thomas Eiter and Kewen Wang, 'Semantic forgetting in answer set programming', *Artif. Intell.*, **172**(14), 1644–1672, (2008).
- [13] Allen Van Gelder, Kenneth A. Ross, and John S. Schlipf, 'The well-founded semantics for general logic programs', *J. ACM*, **38**(3), 620–650, (1991).
- [14] Michael Gelfond and Vladimir Lifschitz, 'Classical negation in logic programs and disjunctive databases', *New Generation Comput.*, **9**(3-4), 365–385, (1991).
- [15] Ricardo Gonçalves, Matthias Knorr, and João Leite, 'The ultimate guide to forgetting in ASP', in *Procs. of KR*, eds., Chitta Baral, James Delgrande, and Frank Wolter. AAAI, (2016).
- [16] Matthias Knorr and José Júlio Alferes, 'Preserving strong equivalence while forgetting', in *Procs. of JELIA*, eds., Eduardo Fermé and João Leite, volume 8761 of *LNCS*, pp. 412–425. Springer, (2014).
- [17] Boris Konev, Michel Ludwig, Dirk Walther, and Frank Wolter, 'The logical difference for the lightweight description logic EL', *J. Artif. Intell. Res. (JAIR)*, **44**, 633–708, (2012).
- [18] Boris Konev, Carsten Lutz, Dirk Walther, and Frank Wolter, 'Model-theoretic inseparability and modularity of description logic ontologies', *Artif. Intell.*, **203**, 66–103, (2013).
- [19] Roman Kontchakov, Frank Wolter, and Michael Zakharyashev, 'Logic-based ontology comparison and module extraction, with an application to dl-lite', *Artif. Intell.*, **174**(15), 1093–1141, (2010).
- [20] Jérôme Lang, Paolo Liberatore, and Pierre Marquis, 'Propositional independence: Formula-variable independence and forgetting', *J. Artif. Intell. Res. (JAIR)*, **18**, 391–443, (2003).
- [21] Jérôme Lang and Pierre Marquis, 'Reasoning under inconsistency: A forgetting-based approach', *Artif. Intell.*, **174**(12-13), 799–823, (2010).
- [22] Javier Larrosa, 'Boosting search with variable elimination', in *Procs. of CP*, ed., Rina Dechter, volume 1894 of *LNCS*, pp. 291–305. Springer, (2000).
- [23] Javier Larrosa, Enric Morancho, and David Niso, 'On the practical use of variable elimination in constraint optimization problems: 'still-life' as a case study', *J. Artif. Intell. Res. (JAIR)*, **23**, 421–440, (2005).
- [24] C. I. Lewis, *A survey of symbolic logic*, University of California Press, 1918. Republished by Dover, 1960.
- [25] Vladimir Lifschitz, 'What is answer set programming?', in *Procs. of AAAI*, eds., Dieter Fox and Carla P. Gomes, pp. 1594–1597. AAAI Press, (2008).
- [26] Vladimir Lifschitz, David Pearce, and Agustín Valverde, 'Strongly equivalent logic programs', *ACM Trans. Comput. Log.*, **2**(4), 526–541, (2001).
- [27] Vladimir Lifschitz, Lappoon R. Tang, and Hudson Turner, 'Nested expressions in logic programs', *Ann. Math. Artif. Intell.*, **25**(3-4), 369–389, (1999).
- [28] Fangzhen Lin and Raymond Reiter, 'How to progress a database', *Artif. Intell.*, **92**(1-2), 131–167, (1997).
- [29] Yongmei Liu and Ximing Wen, 'On the progression of knowledge in the situation calculus', in *Procs. of IJCAI*, ed., Toby Walsh, pp. 976–982. IJCAI/AAAI, (2011).
- [30] Aart Middeldorp, Satoshi Okui, and Tetsuo Ida, 'Lazy narrowing: Strong completeness and eager variable elimination', *Theor. Comput. Sci.*, **167**(1&2), 95–130, (1996).
- [31] Yves Moinard, 'Forgetting literals with varying propositional symbols', *J. Log. Comput.*, **17**(5), 955–982, (2007).
- [32] David Pearce, Hans Tompits, and Stefan Woltran, 'Characterising equilibrium logic and nested logic programs: Reductions and complexity', *TPLP*, **9**(5), 565–616, (2009).
- [33] David Rajaratnam, Hector J. Levesque, Maurice Pagnucco, and Michael Thielscher, 'Forgetting in action', in *Procs. of KR*, eds., Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter. AAAI Press, (2014).
- [34] Chiaki Sakama and Katsumi Inoue, 'An abductive framework for computing knowledge base updates', *Theory and Practice of Logic Programming (TPLP)*, **3**(6), 671–713, (2003).
- [35] Martin Slota and João Leite, 'Robust equivalence models for semantic updates of answer-set programs', in *Procs. of KR*, eds., Gerhard Brewka, Thomas Eiter, and Sheila A. McIlraith, pp. 158–168. AAAI Press, (2012).
- [36] Martin Slota and João Leite, 'A unifying perspective on knowledge updates', in *Procs. of JELIA*, eds., Luis Fariñas del Cerro, Andreas Herzig, and Jérôme Mengin, volume 7519 of *LNAI*, pp. 372–384. Springer, (2012).
- [37] Martin Slota and João Leite, 'The rise and fall of semantic rule updates based on se-models', *TPLP*, **14**(6), 869–907, (2014).
- [38] Mirosław Truszczynski, 'Reducts of propositional theories, satisfiability relations, and generalizations of semantics of logic programs', *Artif. Intell.*, **174**(16-17), 1285–1306, (2010).
- [39] Yisong Wang, Kewen Wang, and Mingyi Zhang, 'Forgetting for answer set programs revisited', in *Procs. of IJCAI*, ed., Francesca Rossi. IJCAI/AAAI, (2013).
- [40] Yisong Wang, Yan Zhang, Yi Zhou, and Mingyi Zhang, 'Forgetting in logic programs under strong equivalence', in *Procs. of KR*, eds., Gerhard Brewka, Thomas Eiter, and Sheila A. McIlraith, pp. 643–647. AAAI Press, (2012).
- [41] Yisong Wang, Yan Zhang, Yi Zhou, and Mingyi Zhang, 'Knowledge forgetting in answer set programming', *J. Artif. Intell. Res. (JAIR)*, **50**, 31–70, (2014).
- [42] Zhe Wang, Kewen Wang, Rodney W. Topor, and Jeff Z. Pan, 'Forgetting for knowledge bases in DL-Lite', *Ann. Math. Artif. Intell.*, **58**(1-2), 117–151, (2010).
- [43] Andreas Weber, 'Updating propositional formulas', in *Expert Database Conf.*, pp. 487–500, (1986).
- [44] Ka-Shu Wong, *Forgetting in Logic Programs*, Ph.D. dissertation, The University of New South Wales, 2009.
- [45] Yan Zhang and Norman Y. Foo, 'Solving logic program conflict through strong and weak forgettings', *Artif. Intell.*, **170**(8-9), 739–778, (2006).

Solving Set Optimization Problems by Cardinality Optimization with an Application to Argumentation

Wolfgang Faber¹ and Mauro Vallati¹ and Federico Cerutti² and Massimiliano Giacomin³

Abstract.

Optimization—minimization or maximization—in the lattice of subsets is a frequent operation in Artificial Intelligence tasks. Examples are subset-minimal model-based diagnosis, nonmonotonic reasoning by means of circumscription, or preferred extensions in abstract argumentation. Finding the optimum among many admissible solutions is often harder than finding admissible solutions with respect to both computational complexity and methodology. This paper addresses the former issue by means of an effective method for finding subset-optimal solutions. It is based on the relationship between cardinality-optimal and subset-optimal solutions, and the fact that many logic-based declarative programming systems provide constructs for finding cardinality-optimal solutions, for example maximum satisfiability (MaxSAT) or weak constraints in Answer Set Programming (ASP). Clearly each cardinality-optimal solution is also a subset-optimal one, and if the language also allows for the addition of particular restricting constructs (both MaxSAT and ASP do) then all subset-optimal solutions can be found by an iterative computation of cardinality-optimal solutions. As a showcase, the computation of preferred extensions of abstract argumentation frameworks using the proposed method is studied.

1 INTRODUCTION

In Artificial Intelligence, the task of set optimization, in the sense of finding a set that is minimal or maximal with respect to set inclusion, frequently occurs. There are famous examples such as Circumscription [20] or Model-based Diagnosis that involve set minimization. Computing preferred extensions of abstract argumentation frameworks [12] is an example that involves set maximization.

Often, set optimization is an element that creates difficulties in implementation and representation. For example, McCarthy had to resort to Second-order logic for defining Circumscription of First-order theories [20]. Also for computing preferred extensions, relatively sophisticated techniques are required, for instance QBFs rather than propositional formulas [2].

There is another notion of set optimization, finding a set that has minimal or maximal cardinality, which has more readily available system support nowadays. The most prominent examples of languages and systems that support cardinality optimization are MaxSAT, Constraint Programming, and Answer Set Programming (ASP).

In this paper, we show how set optima can be computed by a general algorithm that employs cardinality optimizing subroutines, provided that the underlying languages allow for expressing simple constraints. Algorithm MCSes in Figure 2 of [19], which computes Minimal Correction Sets⁴ of a propositional formula, bears a few similarities to our algorithms. However, it is also different in several respects, most notably it is formulated for solving one particular problem only, assumes propositional formulas as the representation formalism, and does not employ a cardinality optimization oracle explicitly. To the best of our knowledge a general method applicable in a variety of representation formalisms and for unspecific set optimization problems settings has not been proposed in the literature before. We develop two instantiations of the general method, for MaxSAT and for ASP, and show that they are suitable.

There are two more recent software tools that also support computing set optimization problems when the underlying language is ASP: **asprin** [9] and **D-FLAT²** [8]. The scope of the tool **asprin** is actually reasoning with preferences, but as a special case one can express preferences such that only set optimal solutions remain. The required preferences come with the predefined library of **asprin**. The underlying algorithms of **asprin** are very different from those presented in this paper. **D-FLAT²** builds on dynamic programming and exploits tree decomposition in order to solve set optimization problems and is therefore also very different from the method that we will present in this paper. In the realm of ASP, the tool **metasp** [17] can be seen as a predecessor of **asprin**, which does not seem to be maintained any longer. It relies on reification of rules and exploits a programming pattern known as saturation for set optimization, which is also very different from the method described in this paper.

We then turn our attention to a showcase application, computing preferred extensions of abstract argumentation frameworks. Dung's theory of abstract argumentation [12] is a unifying framework able to encompass a large variety of specific formalisms in the areas of nonmonotonic reasoning, logic programming and computational argumentation. It is based on the notion of argumentation framework (*AF*), consisting of a set of *arguments* and a binary *attack* relation between them. Arguments can thus be represented by nodes of a directed graph, and attacks by arcs. The nature of arguments is left unspecified: it can be anything from logical statements to informal natural language text. For instance, [24] shows how argumentation can be efficiently used for supporting critical thinking and intelligence analysis in military-sensitive contexts.

Different *argumentation semantics* declare the criteria to deter-

¹ School of Computing and Engineering, University of Huddersfield, Huddersfield, UK. email: n.surname@hud.ac.uk

² Cardiff University, School of Computer Science & Informatics, Cardiff, UK. email: ceruttif@cardiff.ac.uk

³ Università degli Studi di Brescia, Brescia, Italy. email: massimiliano.giacomin@unibs.it

⁴ It is worth noticing that algorithms exploiting minimal correction sets have been proposed for computing argumentation semantics extensions, in particular for semi-stable, ideal, and eager semantics [25], but not for preferred semantics which is our main test-case in this paper.

mine which arguments emerge as “justified” among conflicting ones, by identifying a number of *extensions*, i.e. sets of arguments that can “survive the conflict together”. In [12] four “traditional” semantics were introduced, namely *complete*, *grounded*, *stable*, and *preferred* semantics. For a complete overview of subsequently proposed alternative semantics, the interested reader is referred to [3].

The main computational problems in abstract argumentation include *decision*—e.g. determine if an argument is in all the extensions prescribed by a semantics—and *construction* problems, and turn out to be computationally intractable for most argumentation semantics [13]. In this paper we focus on the *extension enumeration* problem, i.e. constructing *all* extensions for a given *AF*: its solution provides complete information about the justification status of arguments and allows for solving the other problems as well.

Our general method allowed for the definition and implementation of two novel algorithms for enumerating preferred extensions: **prefMaxSAT** (based on a MaxSAT solver) and **prefASP** (based on an ASP solver). Both are evaluated using benchmarks from the International Competition on Computational Models of Argumentation (ICMA2015). We report on a variety of experiments: the first focuses on **prefASP** and starts with comparing the use of different solver configurations for **prefASP**, followed by a comparison of **prefASP** to **asprin** and **D-FLAT²**, and eventually comparing **prefASP** to the dedicated argumentation solver **ASPARTIX-V**. Eventually we compare **prefASP** and **prefMaxSAT** to the ICMA2015 competition winner **Cegartix**. The experiments show that despite their conceptual simplicity, our software tools are competitive with the best available ones.

2 ABSTRACT METHODOLOGY

The proposed methodology applies to a variety of knowledge representation formalisms, we therefore consider an abstract setting. We define a knowledge base \mathcal{K} to be associated with a set $\sigma(\mathcal{K})$ of solutions, and we assume that each $s \in \sigma(\mathcal{K})$ is a set. We also use a set restriction operator \downarrow_O such that $s \downarrow_O = s \cap O$, the idea being that $s \downarrow_O$ identifies the solution elements that are relevant for optimization. We also assume a composition operator $\mathcal{K}_1 \circ \mathcal{K}_2$ to be present that allows to compose two knowledge bases \mathcal{K}_1 and \mathcal{K}_2 . We next define a few optimization criteria for solutions of knowledge bases.

Definition 1 Let \mathcal{K} be a knowledge base and R be a set (of elements occurring in solutions of \mathcal{K}). Define:

$$\begin{aligned} S_R^{max}(\mathcal{K}) &= \{s \mid s \in \sigma(\mathcal{K}), \nexists s' \in \sigma(\mathcal{K}) : s' \downarrow_R \supset s \downarrow_R\} \\ S_R^{min}(\mathcal{K}) &= \{s \mid s \in \sigma(\mathcal{K}), \nexists s' \in \sigma(\mathcal{K}) : s' \downarrow_R \subset s \downarrow_R\} \\ C_R^{max}(\mathcal{K}) &= \{s \mid s \in \sigma(\mathcal{K}), \nexists s' \in \sigma(\mathcal{K}) : |s' \downarrow_R| > |s \downarrow_R|\} \\ C_R^{min}(\mathcal{K}) &= \{s \mid s \in \sigma(\mathcal{K}), \nexists s' \in \sigma(\mathcal{K}) : |s' \downarrow_R| < |s \downarrow_R|\} \end{aligned}$$

While $S_R^{min}(\mathcal{K})$ and $S_R^{max}(\mathcal{K})$ occur in diverse applications of knowledge representation and reasoning, it often happens that the computational complexity of these tasks increases (under standard assumptions) compared to $S_R^{min}(\mathcal{K})$ and $S_R^{max}(\mathcal{K})$. For example, deciding whether $\sigma(\mathcal{K}) = \emptyset$ is co-NP-complete (in the size of \mathcal{K}) if \mathcal{K} is represented using a propositional formula and $\sigma(\mathcal{K})$ is the set of satisfying assignments, where each assignment is represented as the set of true variables. In this setting, computing $S_R^{max}(\mathcal{K})$ is then Σ_2^P -hard, as showing the optimality of a solution may require exponentially many co-NP checks, while $C_R^{max}(\mathcal{K})$ is in Δ_2^P , requiring at most a polynomial number of co-NP checks. Note that this does

not necessarily have practical consequences, because at the moment all known algorithms to solve these problems require at least exponential time.

There are also representational repercussions. Still assuming \mathcal{K} to be a propositional formula, one cannot find a propositional formula of polynomial size that encodes any of $S_R^{min}(\mathcal{K})$, $S_R^{max}(\mathcal{K})$, $C_R^{max}(\mathcal{K})$, and $C_R^{min}(\mathcal{K})$ (if $NP \neq \Sigma_2^P$, which is currently unknown, but often conjectured). If \mathcal{K} has been modelled using ASP, it is easy to encode $C_R^{max}(\mathcal{K})$ and $C_R^{min}(\mathcal{K})$ because of the availability of weak constraints (or optimization constructs). In fact, one can use ASP also for encoding $S_R^{min}(\mathcal{K})$ and $S_R^{max}(\mathcal{K})$, because ASP can express all problems in Σ_2^P . We will discuss this further in Section 3.

In this paper, we relate $S_R^{max}(\mathcal{K})$ to $C_R^{max}(\mathcal{K})$ (and $S_R^{min}(\mathcal{K})$ to $C_R^{min}(\mathcal{K})$). We first observe that each cardinality optimal solution is also subset optimal.

Observation 1 For any knowledge base \mathcal{K} and set R , $C_R^{max}(\mathcal{K}) \subseteq S_R^{max}(\mathcal{K})$ and $C_R^{min}(\mathcal{K}) \subseteq S_R^{min}(\mathcal{K})$.

This observation holds because if any $s \in C_R^{max}(\mathcal{K})$ were not in $S_R^{max}(\mathcal{K})$, then there would be some $s' \in \sigma(\mathcal{K})$ such that $s' \downarrow_R \supset s \downarrow_R$ and clearly $|s' \downarrow_R| > |s \downarrow_R|$ then holds (and symmetrically for minimization).

This implies that when the task is to compute one subset optimal solution, one can instead safely compute one cardinality optimal solution. When, however, the computational task involves an enumeration of all subset optimal solutions, one is faced with incompleteness, as not all subset optima are cardinality optimal.

Example 1 Let \mathcal{K}_1 be such that $\sigma(\mathcal{K}_1) = \{\{a, b\}, \{b\}, \{c\}\}$ and let $R_1 = \{a, b, c\}$. Then $S_{R_1}^{max}(\mathcal{K}_1) = \{\{a, b\}, \{c\}\}$ while $C_{R_1}^{max}(\mathcal{K}_1) = \{\{a, b\}\}$.

This can be overcome by an iterative approach, in which first cardinality optimal solutions are computed. In the next stage, the knowledge base is extended in a way that it no longer admits the solutions already found or any subsets (for maximization) or supersets (for minimization) thereof.

Definition 2 Given a knowledge base \mathcal{K} , a set R , and a set $S \subseteq \sigma(\mathcal{K})$, let $\mathcal{N}^{\subseteq}(\mathcal{K}, R, S)$ denote a knowledge base such that

$$\sigma(\mathcal{K} \circ \mathcal{N}^{\subseteq}(\mathcal{K}, R, S)) = \sigma(\mathcal{K}) \setminus \{s' \mid s' \downarrow_R \subseteq s \downarrow_R \wedge s \in S\}.$$

Symmetrically, let $\mathcal{N}^{\supseteq}(\mathcal{K}, R, S)$ denote a knowledge base such that

$$\sigma(\mathcal{K} \circ \mathcal{N}^{\supseteq}(\mathcal{K}, R, S)) = \sigma(\mathcal{K}) \setminus \{s' \mid s' \downarrow_R \supseteq s \downarrow_R \wedge s \in S\}.$$

It depends on the formalism used for the knowledge base, whether $\mathcal{N}^{\subseteq}(\mathcal{K}, R, S)$ and $\mathcal{N}^{\supseteq}(\mathcal{K}, R, S)$ can be created, and in particular whether they can be represented in a concise way. It also depends on the formalism whether there is a uniform way of encoding $\mathcal{N}^{\subseteq}(\mathcal{K}, R, S)$ and $\mathcal{N}^{\supseteq}(\mathcal{K}, R, S)$, or whether one has to rely on a representation that depends on the structure of \mathcal{K} .

The iterative approach is formalized for subset maximality in Algorithm 1 and for subset minimality in Algorithm 2. Note that practical algorithms will usually not collect all solutions in the output because of space considerations, but rather output them immediately as they are computed.

Theorem 1 For a knowledge base \mathcal{K} and set R , Algorithm 1 computes $S_R^{max}(\mathcal{K})$, and Algorithm 2 computes $S_R^{min}(\mathcal{K})$.

Algorithm 1 Enumerating $S_R^{max}(\mathcal{K})$ by means of $C_R^{max}(\mathcal{K})$

```

1: Input:  $\mathcal{K}, R$ 
2: Output:  $S_R^{max}(\mathcal{K})$ 
3:  $\mathcal{K}_i := \mathcal{K}$ 
4:  $S := \emptyset$ 
5:  $S_i := C_R^{max}(\mathcal{K}_i)$ 
6: while  $S_i \neq \emptyset$  do
7:    $S := S \cup S_i$ 
8:    $\mathcal{K}_i := \mathcal{K}_i \circ \mathcal{N}^{\subseteq}(\mathcal{K}_i, R, S_i)$ 
9:    $S_i := C_R^{max}(\mathcal{K}_i)$ 
10: end while
11: return  $S$ 

```

Algorithm 2 Enumerating $S_R^{min}(\mathcal{K})$ by means of $C_R^{min}(\mathcal{K})$

```

1: Input:  $\mathcal{K}, R$ 
2: Output:  $S_R^{min}(\mathcal{K})$ 
3:  $\mathcal{K}_i := \mathcal{K}$ 
4:  $S := \emptyset$ 
5:  $S_i := C_R^{min}(\mathcal{K}_i)$ 
6: while  $S_i \neq \emptyset$  do
7:    $S := S \cup S_i$ 
8:    $\mathcal{K}_i := \mathcal{K}_i \circ \mathcal{N}^{\supseteq}(\mathcal{K}_i, R, S_i)$ 
9:    $S_i := C_R^{min}(\mathcal{K}_i)$ 
10: end while
11: return  $S$ 

```

Proof 1 (Sketch) We show that Algorithm 1 computes $S_R^{max}(\mathcal{K})$. Symmetric arguments prove that Algorithm 2 computes $S_R^{min}(\mathcal{K})$.

We first observe that each assignment of variable S_i contains only elements of $S_R^{max}(\mathcal{K})$. When variable S_i is first initialized in line 5 of Algorithm 1, Observation 1 guarantees the claim. For each later assignment, by construction only $s \in \sigma(\mathcal{K})$ are assigned, and any such s is such that $\nexists s' \in \sigma(\mathcal{K}) : s' \downarrow_R \supset s \downarrow_R$ (otherwise $|s' \downarrow_R| > |s \downarrow_R|$ would hold). It is also clear that the algorithm terminates (if $\sigma(\mathcal{K})$ is finite).

Now observe that each $s \in S_R^{max}(\mathcal{K})$ is assigned once to S_i in Algorithm 1. Indeed, the first assignment contains all elements in $S_R^{max}(\mathcal{K})$ that are of maximum cardinality, the next iteration contains all elements in $S_R^{max}(\mathcal{K})$ of the next-highest cardinality, and so forth down to the elements of $S_R^{max}(\mathcal{K})$ of least cardinality in the last assignment. In this way, all elements of $S_R^{max}(\mathcal{K})$ will be contained in S when Algorithm 1 terminates.

It should be pointed out that Algorithms 1 and 2 also work when instead of $C_R^{max}(\mathcal{K}_i)$ (or $C_R^{min}(\mathcal{K}_i)$) any non-empty subset thereof is assigned to S_i in lines 5 and 9. In particular, for the instantiation of the framework using MaxSAT that will be discussed in Section 3.1, many MaxSAT solvers calculate only one solution, rather than all.

Let us note that the number of subcalls to $C_R^{max}(\mathcal{K}_i)$ (resp., $C_R^{min}(\mathcal{K}_i)$) is at most $|S_R^{max}(\mathcal{K})|$ (resp., $|S_R^{min}(\mathcal{K})|$). The cardinality of these sets can be exponential in \mathcal{K} in the worst case. That also means that in the worst case an exponential number of knowledge bases $\mathcal{N}^{\supseteq}(\mathcal{K}_i, R, S_i)$ (or $\mathcal{N}^{\subseteq}(\mathcal{K}_i, R, S_i)$) are composed to \mathcal{K} , which could lead to a use of exponential space. Note however, that this only occurs if there is an exponential number of solutions to be generated by the algorithm. This only occurs if $s_{1 \downarrow R} \not\subseteq s_{2 \downarrow R}$ and $s_{2 \downarrow R} \not\subseteq s_{1 \downarrow R}$ holds for almost all solutions s_1 and s_2 of $\sigma(\mathcal{K})$.

Note that there is also a contrast to more traditional algorithms that invoke a co-NP oracle call for each $s \in \sigma(\mathcal{K})$, especially if they run a test on each subset of a found solution. In that setting, the number of

subcalls that take exponential time will usually be much greater than $|S_R^{min}(\mathcal{K})|$ (resp., $|S_R^{max}(\mathcal{K})|$). We view this feature of our algorithm as one of the main advantages over more traditional methods.

Algorithm 1 bears some similarities to Algorithm MCSes in Figure 2 of [19]. Algorithm MCSes computes Minimal Correction Sets of a propositional formula. Apart from the fact that it solves a much more specific problem and assumes a specific knowledge representation formalism. In fact, it iteratively increases the cardinality of the (relevant portion of the) solution to be computed and enforces the cardinality by means of formulas thus imitating the behaviour of a MaxSAT. The clauses that Algorithm MCSes adds follow the same idea of $\mathcal{N}^{\subseteq}(\mathcal{K}_i, R, S_i)$, and correspond to $\mathcal{N}_{sat}^{\subseteq}(\mathcal{K}_i, R, S_i)$ that will be defined in Section 3.1.

3 CONCRETIZATIONS USING MaxSAT AND ASP

We now show how to instantiate the abstract method described in Section 2 using MaxSAT and ASP.

3.1 SAT and MaxSAT

In Boolean satisfiability (SAT) [7], one asks for satisfying assignments of a propositional formula φ . We will assume that the propositional variables are taken from the infinite set \mathcal{L} and use the standard connectives $\neg, \wedge, \vee, \supset$. SAT solvers usually assume that φ is represented in conjunctive normal form, as a set of clauses.

In this setting, the terminology of Section 2 is instantiated as follows: \mathcal{K} is a propositional formula φ over \mathcal{L} , or a set of clauses over \mathcal{L} . The solutions $\sigma(\mathcal{K})$ are the satisfying assignments of \mathcal{K} , represented as sets of propositional variables that are true in the respective assignment. The operation $\mathcal{K}_1 \circ \mathcal{K}_2$ is either $\mathcal{K}_1 \wedge \mathcal{K}_2$ if $\mathcal{K}_1, \mathcal{K}_2$ are formulae, or $\mathcal{K}_1 \cup \mathcal{K}_2$ if $\mathcal{K}_1, \mathcal{K}_2$ are sets of clauses. In the sequel, we will assume the clause notation.

In order to encode $C_R^{max}(\mathcal{K})$ and $C_R^{min}(\mathcal{K})$, one can make use of weighted MaxSAT [18]. In weighted MaxSAT, one can assign a numerical weight to some of the clauses (known as soft clauses), and these clauses do not necessarily have to be satisfied. The solutions of a MaxSAT problem are determined among all assignments that satisfy the non-weighted clauses (hard clauses), and are those that maximize the sum of weights of the soft clauses.

Definition 3 Given a set of clauses \mathcal{K} and $R \subset \mathcal{L}$, we define the following MaxSAT problem

$$C_{R,sat}^{max}(\mathcal{K}) \equiv \mathcal{K} \cup \{1 : \{r\} \mid r \in R\}$$

consisting of hard clauses \mathcal{K} and soft clauses $\{r\}$ (unit clauses) for each $r \in R$, all with the same positive weight 1 (any other weight could be used, as long it is the same for all soft clauses).

In a similar way,

$$C_{R,sat}^{min}(\mathcal{K}) \equiv \mathcal{K} \cup \{1 : \{\neg r\} \mid r \in R\}$$

consists of hard clauses \mathcal{K} and soft clauses $\{\neg r\}$ (unit clauses) for each $r \in R$, again all with the same weight.

It is clear that $C_R^{max}(\mathcal{K})$ corresponds to the solutions of $C_{R,sat}^{max}(\mathcal{K})$ and $C_R^{min}(\mathcal{K})$ corresponds to the solutions of $C_{R,sat}^{min}(\mathcal{K})$.

For obtaining $S_R^{max}(\mathcal{K})$ and $S_R^{min}(\mathcal{K})$, one would have to resort to Quantified Boolean Formulae (QBFs), which we do not discuss in detail in this paper.

Example 2 An instantiation of Example 1 would be the $\mathcal{K}_1^{sat} = \{\{\neg a, \neg c\}, \{\neg b, \neg c\}, \{b, c\}\}$, with $\sigma(\mathcal{K}_1^{sat}) = \{\{a, b\}, \{b\}, \{c\}\}$. Then, with $R_1 = \{a, b, c\}$, $C_{R_1, sat}^{max}(\mathcal{K}_1^{sat}) = \{\{\neg a, \neg c\}, \{\neg b, \neg c\}, \{b, c\}, 1 : \{a\}, 1 : \{b\}, 1 : \{c\}\}$, and the only optimal solution of $C_{R_1, sat}^{max}(\mathcal{K}_1^{sat})$ is $\{a, b\}$, which is equal to $C_{R_1}^{max}(\mathcal{K}_1^{sat})$.

Finally, we consider how to encode $\mathcal{N}^{\subseteq}(\mathcal{K}, R, S)$ and $\mathcal{N}^{\supseteq}(\mathcal{K}, R, S)$ in SAT. For $\mathcal{N}^{\subseteq}(\mathcal{K}, R, S)$, the idea is to require for each solution in S that at least an element of R not in that solution should be true. This inhibits the solution and any subset (restricted to R) of it. For $\mathcal{N}^{\supseteq}(\mathcal{K}, R, S)$, we require for each solution in S that at least one variable in R should be false. In this way the solution itself and any superset is inhibited.

Definition 4 Given a set of clauses \mathcal{K} , $R \subset \mathcal{L}$, and $S \subseteq \sigma(\mathcal{K})$, let

$$\mathcal{N}_{sat}^{\subseteq}(\mathcal{K}, R, S) \equiv \bigwedge_{s \in S} \bigvee_{v \in R \setminus s} v$$

and in a similar way

$$\mathcal{N}_{sat}^{\supseteq}(\mathcal{K}, R, S) \equiv \bigwedge_{s \in S} \bigvee_{v \in R \cap s} \neg v .$$

Note that these formulas are in clause form. It is clear that $\mathcal{N}_{sat}^{\subseteq}(\mathcal{K}, R, S)$ and $\mathcal{N}_{sat}^{\supseteq}(\mathcal{K}, R, S)$ restrict the solutions in the way required by $\mathcal{N}^{\subseteq}(\mathcal{K}, R, S)$ and $\mathcal{N}^{\supseteq}(\mathcal{K}, R, S)$.

Example 3 Continuing Example 3, if $S_1 = \{\{a, b\}\}$, then $\mathcal{N}_{sat}^{\subseteq}(\mathcal{K}_1^{sat}, R_1, S_1) = \{\{c\}\}$, therefore the only optimal solution of $C_{R_1, sat}^{max}(\mathcal{K}_1^{sat}) \cup \mathcal{N}_{sat}^{\subseteq}(\mathcal{K}_1^{sat}, R_1, S_1)$ is $\{c\}$.

3.2 Answer Set Programming

In Answer Set Programming (ASP) one asks for the answer sets (often also called stable models) of a logic program. The full language specification can be found at <https://www.mat.unical.it/aspcomp2013/ASPstandardization>, below we provide a brief overview of the concepts relevant to this work.

The basic constructs in ASP logic programs are of the form

$$h_1 \mid \dots \mid h_k : - b_1, \dots, b_m, \text{not } b_{m+1}, \dots, \text{not } b_n.$$

where $0 \leq k$, $0 \leq m \leq n$ and the h_i and b_j are function-free first-order atoms. When $k > 0$, it is called a rule, otherwise a constraint. If $k = 1$ and $m = n = 0$, the rule is called a fact. The part left of the construct $:$ is called head, the part right of it is called body. Sets of rules and constraints are called programs.

Programs with variables are thought of as shorthand for their ground (variable-free) versions with respect to the Herbrand universe of the program. Answer sets are defined on ground programs: they are Herbrand models of the program, which satisfy an additional stability condition.

The language of ASP consists of quite a lot more constructs. One relevant for this paper is the weak constraint, which takes the form

$$:\sim b_1, \dots, b_m, \text{not } b_{m+1}, \dots, \text{not } b_n.$$

An interpretation that satisfies all literals to the right of $:\sim$ will incur a (uniform) penalty. Answer sets of programs with weak constraints are then those answer sets of the weak-constraint-free portion that minimize the penalties incurred by weak constraints.

For ASP, the terminology of Section 2 is instantiated as follows: \mathcal{K} in this setting is a weak-constraint-free program, and $\sigma(\mathcal{K})$ is the set of its answer sets. The operation $\mathcal{K}_1 \circ \mathcal{K}_2$ simply is the set union of the two programs \mathcal{K}_1 and \mathcal{K}_2 .

In ASP, it is possible to encode $C_R^{max}(\mathcal{K})$ and $C_R^{min}(\mathcal{K})$ by means of weak constraints, similar to the MaxSAT approach described earlier.

Definition 5 Given a set of clauses \mathcal{K} and $R \subset \mathcal{L}$, we define

$$C_{R, asp}^{max}(\mathcal{K}) = \mathcal{K} \cup \{:\sim \text{not } r. \mid r \in R\}$$

and in a similar way

$$C_{R, asp}^{min}(\mathcal{K}) = \mathcal{K} \cup \{:\sim r. \mid r \in R\} .$$

It is easy to verify that $C_R^{max}(\mathcal{K})$ corresponds to the answer sets of $C_{R, sat}^{max}(\mathcal{K})$ and $C_R^{min}(\mathcal{K})$ corresponds to the answer sets of $C_{R, sat}^{min}(\mathcal{K})$.

ASP also allows for encoding $S_R^{max}(\mathcal{K})$ and $S_R^{min}(\mathcal{K})$, but this requires rather involved, and often ad-hoc programs. A general approach has been presented in [17], but it relies on reification techniques, which is often detrimental for performance.

Let us now consider how to obtain $\mathcal{N}^{\subseteq}(\mathcal{K}, R, S)$ and $\mathcal{N}^{\supseteq}(\mathcal{K}, R, S)$ in ASP. For $\mathcal{N}^{\subseteq}(\mathcal{K}, R, S)$, one requires for each solution in S that not all elements of R outside that solution should be false. This inhibits the solution itself and any subset (restricted to R) of it. For $\mathcal{N}^{\supseteq}(\mathcal{K}, R, S)$, we require for each solution in S that not all of its elements in R should be true. In this way the solution itself and any superset is inhibited.

Definition 6 Given a set of clauses \mathcal{K} , $R \subset \mathcal{L}$, and $S \subseteq \sigma(\mathcal{K})$, let

$$\mathcal{N}_{asp}^{\subseteq}(\mathcal{K}, R, S) = \{:- \text{not } a_1, \dots, \text{not } a_n. \mid s \in S, R \setminus s = \{a_1, \dots, a_n\}\}$$

and in a similar way

$$\mathcal{N}_{asp}^{\supseteq}(\mathcal{K}, R, S) = \{:- a_1, \dots, a_n. \mid s \in S, R \cap s = \{a_1, \dots, a_n\}\} .$$

Again, it is easy to verify that $\mathcal{N}_{asp}^{\subseteq}(\mathcal{K}, R, S)$ and $\mathcal{N}_{asp}^{\supseteq}(\mathcal{K}, R, S)$ restrict the answer sets in the way required by $\mathcal{N}^{\subseteq}(\mathcal{K}, R, S)$ and $\mathcal{N}^{\supseteq}(\mathcal{K}, R, S)$.

4 METHODOLOGY SHOWCASE: ABSTRACT ARGUMENTATION

In this section we show how to use the proposed methodology for enumerating all preferred extensions of abstract argumentation frameworks. After a short background on abstract argumentation, we introduce a MaxSAT-based solver (**prefMaxSAT**) and an ASP-based solver (**prefASP**) that employ the methods described in Section 3.

4.1 Background on Abstract Argumentation

An argumentation framework [12] consists of a set of arguments⁵ and a binary attack relation between them.

⁵ In this paper we consider only *finite* sets of arguments: see [4] for a discussion on infinite sets of arguments.

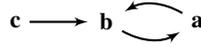


Figure 1. The $AF \Gamma_M$ for the hypertension problem

Definition 7 An argumentation framework (AF) is a pair $\Gamma = \langle \mathcal{A}, \mathcal{R} \rangle$ where \mathcal{A} is a set of arguments and $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$. We say that \mathbf{b} attacks \mathbf{a} iff $\langle \mathbf{b}, \mathbf{a} \rangle \in \mathcal{R}$, also denoted as $\mathbf{b} \rightarrow \mathbf{a}$. The set of attackers of an argument \mathbf{a} will be denoted as $\mathbf{a}^- \triangleq \{\mathbf{b} : \mathbf{b} \rightarrow \mathbf{a}\}$, the set of arguments attacked by \mathbf{a} will be denoted as $\mathbf{a}^+ \triangleq \{\mathbf{b} : \mathbf{a} \rightarrow \mathbf{b}\}$.

Each AF has an associated directed graph where the vertices are the arguments, and the edges are the attacks.

As an intuitive example from [6], let \mathbf{a} be the argument ‘‘Patient has hypertension so prescribe diuretics,’’ \mathbf{b} : ‘‘Patient has hypertension so prescribe betablockers,’’ and \mathbf{c} : ‘‘Patient has emphysema which is a contraindication for betablockers.’’ Intuitively, assuming that only one treatment is possible at the very same time, \mathbf{a} attacks \mathbf{b} and vice versa, while \mathbf{c} suggests that \mathbf{b} should not be the case (\mathbf{c} attacks \mathbf{b}). Therefore, let $\Gamma_M = \langle \mathcal{A}_M, \mathcal{R}_M \rangle$ such that, $\mathcal{A}_M = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$ and $\mathcal{R}_M = \{\langle \mathbf{c}, \mathbf{b} \rangle, \langle \mathbf{b}, \mathbf{a} \rangle, \langle \mathbf{a}, \mathbf{b} \rangle\}$. Γ_M is depicted in Fig. 1.

The basic properties of conflict-freeness, acceptability, and admissibility of a set of arguments are fundamental for the definition of argumentation semantics.

Definition 8 Given an $AF \Gamma = \langle \mathcal{A}, \mathcal{R} \rangle$:

- a set $S \subseteq \mathcal{A}$ is a conflict-free set of Γ if $\nexists \mathbf{a}, \mathbf{b} \in S$ s.t. $\mathbf{a} \rightarrow \mathbf{b}$;
- an argument $\mathbf{a} \in \mathcal{A}$ is acceptable with respect to a set $S \subseteq \mathcal{A}$ of Γ if $\forall \mathbf{b} \in \mathcal{A}$ s.t. $\mathbf{b} \rightarrow \mathbf{a}$, $\exists \mathbf{c} \in S$ s.t. $\mathbf{c} \rightarrow \mathbf{b}$;
- the function $F_\Gamma : 2^{\mathcal{A}} \rightarrow 2^{\mathcal{A}}$ such that $F_\Gamma(S) = \{\mathbf{a} \mid \mathbf{a} \text{ is acceptable w.r.t. } S\}$ is called the characteristic function of Γ ;
- a set $S \subseteq \mathcal{A}$ is an admissible set of Γ if S is a conflict-free set of Γ and every element of S is acceptable with respect to S of Γ .

In the $AF \Gamma_M$ of Fig. 1, $\{\mathbf{a}\}$ is an admissible set because it is conflict-free (there is no such attack $\langle \mathbf{a}, \mathbf{a} \rangle$) and each element of the set (i.e. \mathbf{a}) is defended against the attack it receives (i.e. \mathbf{a} is attacked by \mathbf{b} , but, in turn, \mathbf{b} is attacked by \mathbf{a}).

An argumentation semantics S prescribes for any $AF \Gamma$ a set of extensions, denoted as $\mathcal{E}_S(\Gamma)$, namely a set of sets of arguments satisfying the conditions dictated by S . Here we recall the definition of the grounded semantics, denoted as \mathcal{GR} , and of the preferred semantics, denoted as \mathcal{PR} .

Definition 9 Given an $AF \Gamma = \langle \mathcal{A}, \mathcal{R} \rangle$:

- a set $S \subseteq \mathcal{A}$ is the grounded extension of Γ , if S is the least (w.r.t. set inclusion) fixed point of the characteristic function F_Γ ;
- a set $S \subseteq \mathcal{A}$ is a preferred extension of Γ , i.e. $S \in \mathcal{E}_{\mathcal{PR}}(\Gamma)$, if S is a maximal (w.r.t. \subseteq) admissible set of Γ .

While $\{\mathbf{a}\}$ is an admissible set for Γ_M , it is not a preferred extension. In fact, $\{\mathbf{c}, \mathbf{a}\}$ is also an admissible set which contains $\{\mathbf{a}\}$. Since there are no admissible supersets of $\{\mathbf{c}, \mathbf{a}\}$, it is therefore maximal and thus a preferred extension, the only one for Γ_M .

4.2 Admissible Extensions in SAT and ASP

As discussed in [5, 11] the search for admissible sets can be encoded using propositional logic formulae.

Definition 10 Given an $AF \Gamma = \langle \mathcal{A}, \mathcal{R} \rangle$, \mathcal{L} a set of propositional variables, and $v : \mathcal{A} \rightarrow \mathcal{L}$, let $admsat_\Gamma$ be

$$\bigwedge_{\mathbf{a} \in \mathcal{A}} \left(\left(v(\mathbf{a}) \supset \bigwedge_{\mathbf{b} \in \mathbf{a}^-} \neg v(\mathbf{b}) \right) \wedge \left(v(\mathbf{a}) \supset \bigwedge_{\mathbf{b} \in \mathbf{a}^-} \bigvee_{\mathbf{c} \in \mathbf{b}^-} v(\mathbf{c}) \right) \right)$$

The models of $admsat_\Gamma$ correspond to the admissible sets of Γ .

For the example $AF \Gamma_M$, $admsat_{\Gamma_M}$ is

$$\begin{aligned} & (v(\mathbf{a}) \supset \neg v(\mathbf{b})) \wedge (v(\mathbf{a}) \supset (v(\mathbf{a}) \vee v(\mathbf{c}))) \wedge \\ & (v(\mathbf{b}) \supset (\neg v(\mathbf{a}) \wedge \neg v(\mathbf{c}))) \wedge (v(\mathbf{b}) \supset (v(\mathbf{b}) \wedge \perp)) \wedge \\ & (v(\mathbf{c}) \supset \top) \wedge (v(\mathbf{c}) \supset \top) \end{aligned}$$

For ASP, an encoding for admissible extensions is rather straightforward, see [14, 11].

Definition 11 Given an $AF \Gamma = \langle \mathcal{A}, \mathcal{R} \rangle$, for each $\mathbf{a} \in \mathcal{A}$ a fact

$$\text{arg}(\mathbf{a}).$$

is created and for each $(\mathbf{a}, \mathbf{b}) \in \mathcal{R}$ a fact

$$\text{att}(\mathbf{a}, \mathbf{b}).$$

is created (this corresponds to the *apx* file format in the ICCMA competition). Together with the program

$$\begin{aligned} \text{in}(\mathbf{X}) & : \text{not out}(\mathbf{X}), \text{arg}(\mathbf{X}). \\ \text{out}(\mathbf{X}) & : \text{not in}(\mathbf{X}), \text{arg}(\mathbf{X}). \\ & : \text{in}(\mathbf{X}), \text{in}(\mathbf{Y}), \text{att}(\mathbf{X}, \mathbf{Y}). \\ \text{defeated}(\mathbf{X}) & : \text{in}(\mathbf{Y}), \text{att}(\mathbf{Y}, \mathbf{X}). \\ \text{not_defended}(\mathbf{X}) & : \text{att}(\mathbf{Y}, \mathbf{X}), \text{not defeated}(\mathbf{Y}). \\ & : \text{in}(\mathbf{X}), \text{not_defended}(\mathbf{X}). \end{aligned}$$

we form $admasp_\Gamma$ and there is a one-to-one correspondence between answer sets of $admasp_\Gamma$ and admissible extensions.

Different from the SAT encoding, the ASP encoding only changes facts for different AF s, the main program remains equal.

4.3 Preferred Extensions via Algorithm 1

We can now use our methodology in order to step from admissible to preferred extensions. Indeed, if \mathcal{K} encodes admissible extensions of an AF and R is the language part encoding the extensions, then $S_R^{max}(\mathcal{K})$ encodes preferred extensions. We can then use Algorithm 1 to compute them. The two concretizations then give rise to two solvers, **prefMaxSAT** and **prefASP**, described in the following.

For **prefMaxSAT**, given an $AF \Gamma = \langle \mathcal{A}, \mathcal{R} \rangle$ and $v : \mathcal{A} \rightarrow \mathcal{L}$, we use Algorithm 1 with input \mathcal{K} being $admsat_\Gamma$ and input R being the image of v , in the following referred to as $img(v)$. In lines 5 and 9 of Algorithm 1, we use a MaxSAT solver for obtaining one solution of $C_{img(v), sat}^{max}(\mathcal{K}_i)$. In line 8, $\mathcal{N}_{sat}^{\subseteq}(\mathcal{K}_i, img(v), S_i)$ is used.

For **prefASP**, given an $AF \Gamma = \langle \mathcal{A}, \mathcal{R} \rangle$, we use Algorithm 1 with input \mathcal{K} being $admasp_\Gamma$ and input R being $\{\text{in}(\mathbf{a}) \mid \mathbf{a} \in \mathcal{A}\}$, in the following referred to as $I(\mathcal{A})$. In lines 5 and 9 of Algorithm 1, we use an ASP solver for obtaining all answer sets of $C_{I(\mathcal{A}), asp}^{max}(\mathcal{K}_i)$. In line 8, $\mathcal{N}_{asp}^{\subseteq}(\mathcal{K}_i, I(\mathcal{A}), S_i)$ is used.

Apart from the encodings and underlying solvers, there is also a difference in the fact that **prefASP** computes all cardinality-optimal solutions in one go, while **prefMaxSAT** computes one at a time.

5 EXPERIMENTAL ANALYSIS

In order to evaluate the efficiency of the introduced algorithms, we have carried out an experimental analysis where performance is analyzed from different perspectives. After describing the general setup, we first report on an experiment for choosing a parameter setting in the backend solver of **prefASP**. Next we report on a comparison of **prefASP** with **asprin** and **D-FLAT²**. As discussed in the Introduction, these systems also support set optimization in an ASP setting, but use very different underlying algorithms. This is followed by a comparison to dedicated argumentation systems, in which we compare **prefASP** with **ASPARTIX-V**, since **ASPARTIX-V** is also based on ASP. Finally, we compare both **prefASP** and **prefMaxSAT** with **Cegartix**, which is the state-of-the-art solver, in the sense that it won the ICCMA2015 competition. These comparisons are deliberately split up into small pairings, in order to have a crisper picture of the relative performance measure. Indeed, the IPC score depends on the solvers considered in the comparison, while PAR10 and coverage are specific to a single solver. Experiments do not need to be re-run for presenting results together: however, comparisons would be less informative given the changes in the IPC score.

5.1 Experimental Settings

prefMaxSAT has been implemented in C++, and exploits the AS-Pino MaxSAT solver [1]. It should be noted that **prefMaxSAT** can be used with any MaxSAT system supporting the DIMACS format. The ASP-based algorithm **prefASP** has been implemented as a bash script using basic tools like `sed` and `grep` and exploits `clingo 4.5.2` [16] as ASP solver.

The experiments were conducted on a cluster with computing nodes equipped with 2.5 GHz IntelTM Core 2 Quad Processors, 4 GB of RAM and Linux operating system. A cutoff of 900 seconds—wallclock time—was imposed to compute the preferred extensions for each *AF*. For each involved solver we recorded the overall result: success (if it finds each preferred extension), crashed, timed-out or ran out of memory. In fact, in our experimental evaluation all the unsuccessful runs are due to time-out. Experiments have been conducted on the ICCMA2015 benchmarks [23], which is a set of randomly generated 192 *AF*s. They have been generated considering three different graph models, in order to provide different levels of complexity. More details can be found on the ICCMA website.⁶

The performance measures reported in this paper are the Penalised Average Runtime and the International Planning Competition (IPC) score.

The Penalised Average Runtime (PAR score) is a real number which counts (i) runs that fail to find all the preferred extensions as ten times the cutoff time (PAR10) and (ii) runs that succeed as the actual runtime. PAR scores are commonly used in automated algorithm configuration, algorithm selection, and portfolio construction, because using them allows runtime to be considered while still placing a strong emphasis on high instance set coverage.

The IPC score, borrowed from the Planning community, is defined as follows:

- For each test case (in our case, each test *AF*) let T^* be the best execution time among the compared systems (if no system produces the solution within the time limit, the test case is not considered valid and ignored).

- For each valid case, each system gets a score of $1/(1 + \log_{10}(T/T^*))$, where T is its execution time, or a score of 0 if it fails in that case. Runtimes below 1 second get by default the maximal score of 1.

The IPC score for a system is the sum of its scores over all the valid test cases.

It should be noted that the IPC score depends on the ensemble of tested systems, that is, it is a relative measure, which depends on the experimental context. Indeed, in the following, the IPC scores of **prefASP** vary depending on the different experimental settings. In contrast to this, the PAR10 score of **prefASP** remains equal, as this is an absolute score.

5.2 Comparison of **prefASP** Using Different Solver Configurations

Clingo offers several different solver configurations (inherited from the clasp solver used in `clingo`) which correspond to different heuristic and search setups, see [15] for detailed explanations. As a first analysis, we investigated how robust **prefASP** is with respect to these configurations, and, as a by-product, we determined the best-performing configuration to be used for comparison with other systems.

Table 1. Comparison of different Clingo solver configurations, that can be exploited by **prefASP**, on the ICCMA2015 benchmarks. Results are shown in terms of IPC score (maximum achievable is 192.0), percentages of success (% Success) and PAR10.

	IPC score	% Success	PAR10
Crafty	183.7	100.0	23.1
Frumpy	178.6	99.5	75.0
Jumpy	172.0	99.5	82.3
Many	177.4	99.5	84.8

Table 1 shows a comparison of the different solver configurations, in terms of IPC score, percentage of successfully analysed *AF*s and PAR10, on the ICCMA2015 benchmarks. It can be observed that all configurations perform relatively similar to each other, implying that the particular chosen configuration is not critical for the performance of **prefASP**.

However, there is one winning configuration: the **Crafty** configuration allows **prefASP** to enumerate preferred extensions of all the considered *AF*s, and to provide solutions faster. This configuration is geared towards “crafted” problems, which also makes sense in the context of the considered benchmarks. Therefore, in the rest of the experimental analysis the **Crafty** configuration will be considered for **prefASP**.

5.3 Comparison with Existing General Algorithms

We now turn to general tools that allow for easy representation and effective solution of subset optimization problems. To the best of our knowledge, the only tools of this kind are **asprin** [9] (with its predecessor **metasp**) and **D-FLAT²** [8], which we have discussed in the Introduction.

Actually, **D-FLAT²** uses the computation of preferred extensions as an example. When we followed the instruction provided by the

⁶ <http://argumentationcompetition.org>

authors⁷, we were able to compute the preferred extensions of small *AF*s, but the system was already struggling with resource consumption on medium sized instances. On the ICCMA2015 benchmarks, the system ran out of memory very quickly, and we did not obtain any solutions for any of the ICCMA2015 benchmarks. This is probably due to the fact that ICCMA instances have large tree-width and **D-FLAT²** relies heavily on tree decomposition. For this reason, in the remainder of this section we focus our comparison on **asprin**.

As **asprin** is based on ASP, the most natural comparison is against the ASP implementation of the proposed approach, namely **prefASP**. For **asprin**, we used clingo 4.5.2, the same version that is used as a backend for **prefASP**.

As input to **asprin** we use the program $admasp_{\Gamma}$ in Definition 11 together with the following preference definition:

```
#preference(p1, superset){
    in(X) : arg(X)
}.
#optimize(p1).
```

which makes **asprin** compute those answer sets, which are subset maximal for atoms with the predicate `in`.

Table 2. Comparison of **prefASP** and **asprin**, on the ICCMA2015 benchmarks. Results are shown in terms of IPC score (maximum achievable is 192.0), percentages of success (% Success) and PAR10.

	IPC score	% Success	PAR10
prefASP	191.2	100.0	23.1
asprin	157.8	100.0	44.9

The results of comparison between **asprin** and **prefASP** performed on the ICCMA2015 benchmarks for enumerating preferred extensions are shown in Table 2. Results indicate that the proposed **prefASP** system is significantly faster: **prefASP** achieves an IPC score of 191.2 versus 157.8 of **asprin**. According to the results, **prefASP** is the fastest system on 187 of the considered *AF*s. This is also confirmed by the PAR10 scores; on average **asprin** is about 20 seconds slower than **prefASP**, while in terms of coverage, both the considered systems are able to successfully analyse all the 192 *AF*s of the benchmark set. With regard to the observed performance difference, our conjecture is that **asprin** proves properties for more subsets of each answer set/extension than **prefASP** has to.

5.4 Comparison with Abstract Argumentation Algorithms Based on the Same Approaches

According to the results of ICCMA2015 [23], **ASPARTIX-V** [22] is the ASP-based abstract argumentation solver that showed the best performance in the preferred enumeration track.⁸ Table 3 presents the results of a comparison between **prefASP** and **ASPARTIX-V** performed on the ICCMA2015 benchmarks. Presented results indicate that **prefASP** is faster, both in terms of IPC and PAR10 scores. Remarkably, **prefASP** is able to successfully analyse a larger number of *AF*s (100.0% against 94.0%).

⁷ **D-FLAT²** software and instructions have been retrieved from <https://github.com/hmarkus/dflat-2> in March 2016.

⁸ We are not aware of any existing solver able to enumerate preferred extensions in a way directly comparable to **prefMaxSAT**.

Table 3. Comparison of **prefASP** and **ASPARTIX-V**, the ASP-based abstract argumentation solver that showed the best performance in the preferred enumeration track, on the ICCMA2015 benchmarks. Results are shown in terms of IPC score (maximum achievable is 192.0), percentages of success (% Success) and PAR10.

	IPC score	% Success	PAR10
prefASP	171.3	100.0	23.1
ASPARTIX-V	148.5	94.0	630.9

At a closer look, it is noticeable that—among ICCMA2015 frameworks—the *AF*s with a very large grounded extension and many nodes in general⁹ are very challenging for **ASPARTIX-V**, while the proposed **prefASP** solver is able to quickly and effectively analyse also such large frameworks.

5.5 Comparison with the State of the Art Solver

In this analysis we compare **prefASP** and **prefMaxSAT** with the winner of the the ICCMA2015 track on enumerating preferred extensions, **Cegartix** [10]. Table 4 shows the performance of considered solvers in terms of IPC score, percentage of successfully analysed *AF*s and PAR10. **prefASP** performs significantly better than **prefMaxSAT**. This is possibly due to the fact that each preferred extension results from an execution of the MaxSAT solver; and a final run is needed in order to demonstrate that no other extensions exist. Therefore, the number of MaxSAT calls is exactly the number of preferred extensions plus one. The generated MaxSAT formulae tend to be large on the considered benchmarks; therefore, the added overhead can be remarkable.

Table 4. Comparison of **prefMaxSAT** and **prefASP** with the winner of the track of ICCMA2015 on enumerating preferred extensions, **Cegartix**. Results are shown in terms of IPC score (maximum 192.0), percentages of success and PAR10.

	IPC score	% Success	PAR10
prefASP	161.7	100.0	23.1
prefMaxSAT	115.3	85.0	1423.8
Cegartix	188.9	100.0	15.2

Interestingly, the performance of **prefASP** is comparable to the performance of **Cegartix**; according to PAR10, **prefASP** needs on average 8 seconds more to enumerate the preferred extensions. Moreover, by re-running the top solvers that took part in this track of ICCMA2015, we observed that **prefASP** would have been ranked second. This is an impressive achievement, considering that the described algorithm: (i) is very general, in the sense that it does not exploit any argumentation-specific knowledge; (ii) is very easy to implement, particularly in the ASP configuration; and (iii) has been implemented as a prototype, without attention on software engineering techniques for improving performance.

6 CONCLUSIONS AND FUTURE WORK

We have proposed a general methodology for solving subset optimality problems by means of iteratively solving cardinality optimality problems. This approach is motivated by the availability of efficient

⁹ <http://argumentationcompetition.org/2015/results.html>

systems that support finding cardinality optimal solutions, namely MaxSAT solvers and ASP solvers supporting weak constraints.

As a methodology showcase we have produced two prototype systems, **prefMaxSAT** and **prefASP**, for enumerating preferred extensions of abstract argumentation frameworks. While the algorithms are general and easy to implement, an experimental analysis showed that they are competitive with the state-of-the-art system, which is specialized for this particular problem. On this showcase, our methods also prove higher performance than the existing general methods for computing subset optimal solutions of answer set programs, viz. **asprin** and **D-FLAT²**.

There are quite many opportunities for future work. As discussed in [19], retain information between calls could help the performance, and we might investigate it in future work. However, this impact on the modularity of the concrete approach—i.e., a solver must be integrated in the framework—and can potentially reduce the generality of the overall framework.

Apart from tuning the prototype implementations of **prefMaxSAT** and **prefASP** to improve their performance, we intend to apply the methodology also to other application domains. Diagnosis or minimal model computation are immediate candidates. Another possibility is integrating our algorithm into a system like **asprin**, or one that supports the same input language.

The methodology would also allow for computing Σ_3^P -hard problems when using ASP, which would be interesting to explore, as it would give rise to alternatives to implementations relying on QBFs. It would also be worthwhile to explore whether the general methodology can be used also with formalisms different from MaxSAT and ASP, which would open entirely new avenues.

Given that there are a few similarities to Algorithm MCSes in [19], a comparison with the CAMUS system described in that paper could be attempted. Looking at the other direction, evaluating whether some implementation techniques in this area, e.g. in [21], can be generalized to our less specialized setting could be pursued as well.

Acknowledgements

The authors would like to acknowledge the use of the University of Huddersfield Queensgate Grid in carrying out this work.

The authors also thank the anonymous reviewers for their helpful comments.

REFERENCES

- [1] Mario Alviano, Carmine Dodaro, and Francesco Ricca, ‘A maxsat algorithm using cardinality constraints of bounded size’, in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pp. 2677–2683, (2015).
- [2] Ofer Arieli and Martin W.A. Caminada, ‘A QBF-based formalization of abstract argumentation semantics’, *Journal of Applied Logic*, **11**(2), 229–252, (2013).
- [3] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin, ‘An introduction to argumentation semantics’, *Knowledge Engineering Review*, **26**(4), 365–410, (2011).
- [4] Pietro Baroni, Federico Cerutti, Paul E. Dunne, and Massimiliano Giacomin, ‘Automata for Infinite Argumentation Structures’, *Artificial Intelligence*, **203**(0), 104–150, (2013).
- [5] Philippe Besnard and Sylvie Doutre, ‘Checking the acceptability of a set of arguments’, in *Proc. of the 10th Int. Workshop on Non-Monotonic Reasoning (NMR 2004)*, pp. 59–64, (2004).
- [6] Philippe Besnard and Anthony Hunter, ‘Constructing argument graphs with deductive arguments: a tutorial’, *Argument & Computation*, **5**(1), 5–30, (2014).
- [7] *Handbook of Satisfiability*, eds., Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, volume 185 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2009.
- [8] Bernhard Bliem, Günther Charwat, Markus Hecher, and Stefan Woltran, ‘D-FLAT²: Subset minimization in dynamic programming on tree decompositions made easy’, in *Eighth Workshop on Answer Set Programming and Other Computing Paradigms (ASPOCP 2015)*, (2015).
- [9] Gerhard Brewka, James P. Delgrande, Javier Romero, and Torsten Schaub, ‘asprin: Customizing answer set preferences without a headache’, in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI 2015)*, pp. 1467–1474, (2015).
- [10] Rémi Brochenin, Thomas Linsbichler, Marco Maratea, Johannes Peter Wallner, and Stefan Woltran, ‘Abstract solvers for Dung’s argumentation frameworks’, in *Proceedings of the International Workshop on Theory and Applications of Formal Argument (TAFE)*, (2015).
- [11] Günther Charwat, Wolfgang Dvořák, Sarah A. Gaggl, Johannes P. Wallner, and Stefan Woltran, ‘Methods for solving reasoning problems in abstract argumentation – A survey’, *Artificial Intelligence*, **220**, 28–63, (2015).
- [12] Phan M Dung, ‘On the Acceptability of Arguments and Its Fundamental Role in Nonmonotonic Reasoning, Logic Programming, and n-Person Games’, *Artificial Intelligence*, **77**(2), 321–357, (1995).
- [13] Paul E. Dunne and Michael Wooldridge, ‘Complexity of abstract argumentation’, in *Argumentation in AI*, eds., I Rahwan and G Simari, chapter 5, 85–104, Springer-Verlag, (2009).
- [14] Uwe Egly, Sarah Alice Gaggl, and Stefan Woltran, ‘Answer-set programming encodings for argumentation frameworks’, *Argument & Computation*, **1**(2), 147–177, (2010).
- [15] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, Javier Romero, and Torsten Schaub, ‘Progress in clasp series 3’, in *Proceedings of the Thirteenth International Conference on Logic Programming and Non-monotonic Reasoning (LPNMR 2015)*, Lecture Notes in Artificial Intelligence, pp. 368–383, (2015).
- [16] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub, ‘Clingo = ASP + control: Preliminary report’, in *Technical Communications of the Thirtieth International Conference on Logic Programming (ICLP 2014)*, (2014).
- [17] Martin Gebser, Roland Kaminski, and Torsten Schaub, ‘Complex optimization in answer set programming’, *Theory and Practice of Logic Programming*, **11**(4-5), 821–839, (2011).
- [18] Chu Min Li and Felip Manyà, ‘Maxsat, hard and soft constraints.’, *Handbook of Satisfiability*, **185**, 613–631, (2009).
- [19] Mark H Liffiton and Karem A Sakallah, ‘Algorithms for computing minimal unsatisfiable subsets of constraints’, *Journal of Automated Reasoning*, **40**(1), 1–33, (2008).
- [20] John McCarthy, ‘Circumscription - A form of non-monotonic reasoning’, *Artificial Intelligence*, **13**(1-2), 27–39, (1980).
- [21] Antonio Morgado, Mark Liffiton, and Joao Marques-Silva, ‘Maxsat-based mcs enumeration’, in *Hardware and Software: Verification and Testing*, 86–101, Springer Berlin Heidelberg, (2012).
- [22] Matthias Thimm and Serena Villata, ‘System descriptions of the first international competition on computational models of argumentation (ICMA-15)’, *arXiv preprint arXiv:1510.05373*, (2015).
- [23] Matthias Thimm, Serena Villata, Federico Cerutti, Nir Oren, Hannes Strass, and Mauro Vallati, ‘Summary report of the first international competition on computational models of argumentation’, *AI Magazine*, **37**(1), 102, (2016).
- [24] Alice Toniolo, Timothy J. Norman, Anthony Etuk, Federico Cerutti, Robin Wentao Ouyang, Mani Srivastava, Nir Oren, Timothy Dropps, John A. Allen, and Paul Sullivan, ‘Agent Support to Reasoning with Different Types of Evidence in Intelligence Analysis’, in *Proceedings of the 14th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2015)*, (2015).
- [25] Johannes Peter Wallner, Georg Weissenbacher, and Stefan Woltran, ‘Advanced sat techniques for abstract argumentation’, in *Proceedings of Computational Logic in Multi-Agent Systems: 14th International Workshop (CLIMA XIV)*, pp. 138–154, (2013).

The Need for Knowledge Extraction: Understanding Harmful Gambling Behavior with Neural Networks

Chris Percy², Artur S. d'Avila Garcez¹, Simo Dragičević²,
Manoel V. M. França¹, Greg Slabaugh¹ and Tillman Weyde¹

Abstract. Responsible gambling is a field of study that involves supporting gamblers so as to reduce the harm that their gambling activity might cause. Recently in the literature, machine learning algorithms have been introduced as a way to predict potentially harmful gambling based on patterns of gambling behavior, such as trends in amounts wagered and the time spent gambling. In this paper, neural network models are analyzed to help predict the outcome of a partial proxy for harmful gambling behavior: when a gambler “self-excludes”, requesting a gambling operator to prevent them from accessing gambling opportunities. Drawing on survey and interview insights from industry and public officials as to the importance of interpretability, a variant of the knowledge extraction algorithm TREPAN is proposed which can produce compact, human-readable logic rules efficiently, given a neural network trained on gambling data. To the best of our knowledge, this paper reports the first industrial-strength application of knowledge extraction from neural networks, which otherwise are black-boxes unable to provide the explanatory insights which are crucially required in this area of application. We show that through knowledge extraction one can explore and validate the kinds of behavioral and demographic profiles that best predict self-exclusion, while developing a machine learning approach with greater potential for adoption by industry and treatment providers. Experimental results reported in this paper indicate that the rules extracted can achieve high fidelity to the trained neural network while maintaining competitive accuracy and providing useful insight to domain experts in responsible gambling.

Keywords. Neural Networks; Knowledge Extraction; Gambling; Problem Gambling.

1 INTRODUCTION

Responsible gambling is a recent and complex field of study that investigates how to best support gamblers, so as to reduce the harm that their gambling activity might cause

[17, 20]. Account-based internet gambling revolutionized the amount of data available to identify early warning signs of potentially harmful behavior [12]. However, the quantity of data simultaneously opens up questions of how best to interpret the data and its results: specifically, how to transform raw gambling session data into meaningful, descriptive variables of behavior, called behavioral markers, and how then to relate those descriptive variables to an individual who is potentially at risk.

Through gambling platforms that permit individuals to self-exclude, it is intended that individuals might recognize that they are at risk of losing control during gambling sessions and instruct the gambling platform to deactivate or block their account for a certain period of time. Leveraging anonymized gambling data made available by industry leaders and research partners IGT, in this paper we explore how such behavior could be explained through the use of neural networks and knowledge extraction. We extend and apply for the first time the TREPAN knowledge extraction algorithm [4] to the problem of predicting self-exclusion from gambling. We then evaluate the knowledge extracted and its importance in the context of this industrial application, in relation to its fidelity to the neural networks from which it was extracted, but also in terms of interpretability in comparison with other machine learning methods such as Bayesian networks and random forests, as used in [15].

In [15], the performances of neural networks trained with backpropagation, random forests, logistic regression, and Bayesian networks were evaluated and compared systematically on the same IGT dataset used here. Although the test set accuracy of the regression model was the lowest (72%), this model may be preferred by industry leaders because of the importance assigned to interpretability. Random forests achieved the highest accuracy (87%), with Bayesian networks in second and neural networks third. But the random forests were very difficult to interpret, consisting of 200 decision trees of unlimited depth. Much simpler than the random forest, the neural network used was a single-hidden layer perceptron with 33 inputs, 17 hidden neurons and 2 outputs, one for self-excluding players and the other for the control group players. But with more than 500

¹ Department of Computer Science, City University London, United Kingdom. Emails: {a.garcez, manoel.franca, t.e.veyde, gregory.slabaugh.1}@city.ac.uk

² Data Analytics Group, BetBuddy London, United Kingdom. Emails: cwspercy@gmail.com, simo@bet-buddy.com

weights, the neural network was also a black-box. The Bayesian network, which used the K2 algorithm, included 360 separately defined conditional probabilities, and also failed to instigate useful insight about the problem when shown to industry experts. In contrast, the TREPAN extension proposed in this paper, when applied to the neural network model above, produced a compact decision tree with a low loss of accuracy that was easily interpretable by experts, as discussed in detail in what follows. As a result, as a baseline, in this paper, we also apply a decision tree directly onto the data, and confirm that the use of the neural network is not redundant. This will also be discussed in more detail in the next section.

In practice, there are two important benefits of being able to predict self-exclusion events. The first is improved player protection. A common motivation, although not the sole motivation, for self-exclusion is concern over one's gambling behavior and the potential for unhealthy levels of gambling. By identifying individuals whose play pattern approximates those who have subsequently chosen to self-exclude, or by identifying individuals in advance of a self-exclusion, the gambling operator can choose to share information or advice with the player that may support healthy engagement with the gambling platform. For this to happen effectively, interpretability of results is important. Alternatively, the operator may choose to restrict marketing activity or platform activities for that player for a certain period of time. The second benefit is more stable, long-term revenue flows to gambling operators, since gamblers that might use their platform less intensively than before may do so with greater security and satisfaction.

Machine learning algorithms have only recently been applied to this field of study as a way of predicting potentially harmful gamblers [15, 17]. In this paper, differently from the related work report in the next section, we are interested in revealing through knowledge extraction different aspects of harmful gambling behavior: which kinds of profiles fit into problematic gambling? Which attributes explain players who have such profiles? Such questions are motivated by survey and interview insight from a responsible gambling conference in Vancouver in which gambling operators, treatment providers and public policy officials set out the need for effective interpretation of such complex machine learning algorithms (New Horizons in Responsible Gambling, February 2016). We show that TREPAN can produce human-readable logic rules when applied to neural networks trained on gambling data. The rules obtained have high fidelity to the trained neural networks without much loss in accuracy. The decision tree produced by TREPAN, from which the rules can be read, additionally are found to summarize the key behaviors that are good predictors of self-exclusion, notably players who flagged highly on bet variability and intensity.

The rest of the paper is structured as follows: in Section 2, we present and discuss some of the most relevant literature regarding understanding gambling behavior and knowledge extraction from machine learning models. Section 3 introduces our methodology, discusses the changes made to TREPAN, and applies knowledge extraction to gambling behavior understanding using neural networks. Section 4 presents our empirical results

comparing rule fidelity and accuracy to that of the neural networks but also with the direct application of decision trees to the data. Section 5 discusses the interpretability of our empirical results following direct feedback from industry leaders and gambling regulators, and concludes with the need for algorithm interpretation and directions for future work.

2 RELATED WORK

In current literature, despite significant research analyzing problem gambling more generally, studies using machine learning have been limited to prediction tasks, i.e. how well machine learning techniques can predict harmful gambling behavior. This paper builds on the literature by explaining the perceived value of knowledge extraction from black-box machine learning models and by adapting and applying a knowledge extraction technique to an industry dataset, and evaluating it both quantitatively and qualitatively through domain expert feedback.

Application to Clinical Analysis of Problem Gambling: In [2], a pathways model provides a framework with which to assess the effectiveness of machine learning models to support clinical analysis of problematic gambling behavior. The pathways model describes three possible pathways to gambling addiction: behaviourally conditioned problem gamblers (pathway 1), emotionally vulnerable problem gamblers (pathway 2), and antisocial, impulsive problem gamblers (pathway 3). In [8] it is argued that it is not possible to link pathway 2 (emotionally vulnerable problem gamblers) with data and behavioral insight extracted from game play. For example, while age is the variable used most in training the random forest predictions [15], there is no indication of its influence or dependence on other variables, or what value ranges are most relevant to the predictions. Similarly, behaviors associated with antisocial, impulsive problem gamblers (pathway 3), are also arguably very difficult to identify purely from analysing the patterns of play. However, in [8] it is argued that insights from behavioral data could provide evidence of gamblers at risk of becoming behaviourally conditioned problem gamblers (pathway 1), notably due to heavy or excessive gambling and loss chasing. For example, problem gamblers often fluctuate between regular, heavy and excessive gambling because of conditioning, distorted cognitions surrounding the probability of winning, or a series of bad judgments or poor decision-making. In [9] it is also noted that wager increase is an indicator of problem gambling behaviour.

Predicting Harmful Gambling Behaviour with Machine Learning: In [15], data obtained from the gambling operator IGT is used to describe internet gambling self-excluders in terms of their demographic and behavioral characteristics. Data analysis approaches and methods for improving the accuracy of predicting self-excluders are developed by hand towards inferred behavior models. Differently, this paper develops this by using artificial neural networks and TREPAN on the same IGT dataset to

describe, rather than predict, self-excluders through knowledge extraction.

Supervised machine learning models were evaluated in [20] in the context of predicting which gamblers could be at risk of problem gambling. Their results suggest useful but general methods and techniques for building models that can predict gamblers at risk of harm. While they propose benchmarks for building such models, specific techniques and the variables that could prove to be good predictors of problem or at-risk gambling are not investigated.

Building on the work from the live action sports betting dataset available from the Division on Addiction public domain, in [17] nine supervised learning methods are assessed to determine which data mining methods are most effective at identifying disordered internet sports gamblers. The supervised learning methods include logistic regression, regularized general linear models (GLM), neural networks, support vector machines (SVM) and random forests, with results ranging from 62% to 67% with random forests the highest performing technique.

Knowledge Extraction from Neural Networks:

Considerable interest and research was devoted to knowledge extraction around the turn of the century [1,4,13]. Recently, with a renewed interest in neural networks as a result of their successful application in a range of big data problems, the importance of knowledge extraction has also been highlighted, e.g. [5, 16, 18, 19, 21, 22]. In particular, there is a sense of knowledge extraction being needed to help organize the research in neural networks, though a better understanding of the strengths and limitation of the various models, but also to transfer knowledge from a source to a target domain in the context of transfer learning applications. In general, extraction methods continue to be classified as either decompositional (where the network is broken apart and its weight vectors are used by the extraction algorithm) or pedagogical (where the network or learning model is treated as an oracle to which queries are posed and answers are obtained). One of the early decompositional methods, TREPAN [4], is still nowadays one of the most successful extraction methods.

In [4], the TREPAN algorithm is proposed for the extraction of decision trees from trained neural networks. TREPAN is originally an M-of-N propositional tree inducer which uses a learned neural network as oracle to form a set of examples S , possibly distinct from the examples used to train the neural network, from which a decision tree is built recursively in the usual way, based on an information gain heuristic. M-of-N rules are of the form: if any M out of concepts A_1, A_2, \dots, A_n are *true* then concept B is *true*. In the next section, our approach for understanding gambling behavior relies on TREPAN with a few small but important modifications (discussed in the next section) to generate a decision tree which is then seen as a model for predicting gambling self-excluders.

As an example of a pedagogical approach, in [10, 13], a knowledge extraction algorithm is presented that is based on a partial ordering on the set of input vectors of the network, which is used to define a number of pruning rules and

simplification rules that interact with such an ordering and allow sound knowledge extraction of rules in certain cases. Although provably sound, such a pedagogical extraction approach may generate far too many logical rules or take too long to compute in the case of large networks. In this paper, we take a more practical perspective.

The Need for Accountability and ‘Human in the Loop’ Oversight of Algorithms: In [6] it was stated that while algorithms can encode power to organizations, they also first can stand in tension with transparency. The types of questions and challenges addressed in [6] are: what is the basis for a prioritization decision? Is it fair and just, or discriminatory? What are the criteria built into a ranking, classification, or association, and are they politicized or biased in some consequential way? What are the limits to measuring and operationalizing the criteria used by the algorithm? How has the algorithm been tuned to privilege false positive or false negative errors? Does that tuning benefit one set of stakeholders over another? What are the potential biases of the training data used in a classifying algorithm? What types of parameters or data were used to initiate the algorithm?

By developing and being able to describe the machine learning algorithms used for predicting self-exclusion, answers to many of the questions posed above can be articulated. For example, the model input variables can be described and computations explained, model technical configurations can be defined, thresholds for flagging risk outlined, and limitations of the dependent variable as a proxy for predicting harm investigated. However, because of the complex nature of how algorithms use the data to obtain results, even a clear explanation of the above arguably still does not provide sufficient transparency as to why the algorithms produce the predictions they do.

In [6], it is further argued that autonomous decision-making is the crux of algorithmic power. Sometimes, though, the outcomes are important (or messy and uncertain) enough that a human operator makes the final decision. In the context of the gambling industry, this has important connotations. In [11], it is stated that ‘human in the loop’ algorithms enable industry and academics to understand, challenge, and improve models. In the context of the analysis of player communications data, for example, such as customer emails and online chats, operators can categorize the frequency, intensity, and complexity of such interactions to see if any gambling *red flags* emerge. In this example [11], adding a human-in-the-loop check after the algorithm's result has increased the chances of identifying harm correctly while at the same time not inconveniencing the core player base.

3 METHODOLOGY

The IGT dataset is based on gambling behavior data made available by IGT from 2009 to 2011 for a sample of 669 control group players and 176 qualifying self-excluders who self-excluded for at least six months. The spin-level play data is manipulated in a number of ways to identify behavior markers that represent known aspects of risk, such as how much time gamblers spend online and how much they bet.

For full details of how the dataset was developed and behavioral markers generated, please see [15].

Our proposed approach for understanding gambling behavior through neural networks is composed of three steps: gambling data analysis, neural network training with backpropagation, and knowledge extraction using TREPAN. In the first step, gambling data analysis, an evaluation of variable relevance and redundancy is carried out on the IGT gambling data using the variable ranking approach mRMR [7]. Further, over-sampling was carried out using the SMOTE algorithm [3], which has been shown to perform better than other methods of dealing with dataset rebalancing. We have applied the optimal SMOTE level to achieve an approximately 50:50 split between control group and self-excluding cohorts. Then, network training is carried out using standard backpropagation with momentum and early stopping [15]. Finally, following network training, in order to perform knowledge extraction, a modified version of the TREPAN algorithm is applied to the trained neural network. We have adapted TREPAN in order to allow its efficient application to the domain of gambling behavior prediction, as follows: tree generation has been simplified with only the *maximum-size* criterion being used for stopping the process, and the search heuristic for best M-of-N split now takes into account the size of M by subtracting M/N from the original heuristic value for a given split. In this way, smaller values of M are preferred (larger values of M are penalized), leading to rules with fewer antecedents than the standard TREPAN and, hopefully, a better interpretability, as such rules should be easier to read.

In the above process, we are interested in evaluating the amount of accuracy loss which is expected when TREPAN is applied to produce an interpretable model from a neural network. The existing trade-off between accuracy and comprehensibility is well-known [1]. In this specific domain of application, gambling behavior, further work might also investigate whether other measures of relevance, in addition to accuracy, might be appropriate. As usual, we are also interested in evaluating the fidelity of the extracted model (decision tree) w.r.t. the neural network, as opposed to w.r.t. the data itself, as in the case of accuracy values. The next section contains the experimental results with such evaluations.

4 EXPERIMENTAL RESULTS

In this section, we present the results on knowledge extraction. We have used ten-fold cross validation and, below, we compare the performance of neural networks, TREPAN (including fidelity to the network), and decision trees obtained directly from the data.

Results show that extracted rules have highly competitive accuracy in comparison with the trained neural network, at around 79% vs an original 80%. A high fidelity rate of 87% of the TREPAN tree to the original neural network was achieved. Accuracy results of both the neural network and extracted TREPAN tree shown an improvement in relation

to [17], suggesting that this approach is competitive in relation to previous analyses in the same domain.

In assessing whether extracting a decision tree from the neural network via TREPAN is a worthwhile process, we compare performance of the TREPAN tree with two decision trees created directly from the data, one with unlimited height and one restricted to the same height of the decision tree created by TREPAN. The decision trees were constructed in H2O with 10-fold cross validation, using the same approach to over-sampling as done for the neural networks.

Method	Accuracy [Fidelity for TREPAN]	Decision Tree Leaves	Decision Tree Height
Neural Network	79.8%	-	-
TREPAN	78.8% [87.4%]	9	5
Decision Tree (unlimited height)	77.3%	168	20
Decision Tree (max height 5)	75.1%	25	5

Table 1: Predictive accuracies, i.e. average test-set performance post SMOTE for the trained neural network and the extracted TREPAN tree in comparison with decision trees.

The results in Table 1 show an improvement in accuracy by TREPAN in comparison with both decision trees. More importantly, there is also a significant improvement in decision tree simplicity (and, hence, the likelihood of human readability) via the TREPAN approach. Even the decision tree with a restricted height of five layers, as produced naturally by the TREPAN algorithm, has a far more complicated decision structure, resulting in 25 separate routes that result in a label prediction, as opposed to the 9 leaves deployed by the TREPAN model. However, it is still necessary to determine whether the simpler structure of the TREPAN model is indeed human-readable in a meaningful sense and whether it can thus be subjected to human validation and application.

In assessing the readability of the TREPAN output, Figure 1 contains a visual representation of the entire TREPAN decision tree. While the extracted decision tree remains somewhat complex, it is possible to read it and infer certain conclusions from the split nodes, enabling further validation and exploration by industry professionals and domain experts, as follows:

The majority of self-excluders are identified due to either a weak flag on “Variability” or a strong flag on “Intensity” (at least a 22% increase in the number of bets placed). However, players flagging both on “Variability” and strongly on “Intensity” must also be men (approximately 80% of the sample are) and flagging high on “Frequency” to be assessed as self-excluders. This is a minor part of the decision tree as only 3% to 5% of the samples will flag on both “Variability” and “Intensity”. Otherwise, you are likely to be a control group player regardless of your score on

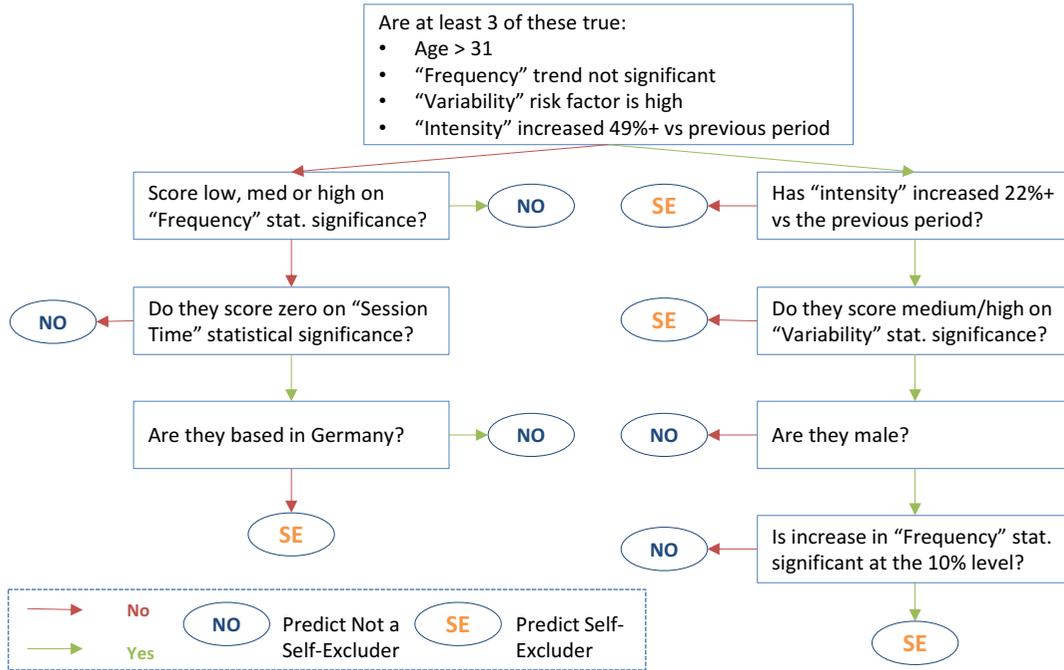


Figure 1: TREPAN decision tree. Phrases in double quotation marks refer to behavioral Risk Factors which were input to the neural network obtained by pre-processing the raw spin-level play data from IGT. For instance, “Trajectory” refers to a player’s total amount of money wagered over an active gambling day and how it changes over time, “Variability” refers to the standard deviation in total amounts wagered over time, “Intensity” refers to the number of bets placed per day, “Frequency” refers to the share of calendar days on which gambling takes place, and “Session Time” refers to the time spent gambling online [15].

“Frequency”, unless you are based outside Germany in which case you could be assessed as a self-excluding player, provided you have also flagged, even if only very mildly, on the “Session Time” risk factor (although some 75% of players in the sample were based in Germany).

The first two bullet points in the root node of the decision tree in Figure 1 suggest that “Variability” and “Intensity” are more important risk factors for identifying self-excluders than “Session Time” or “Frequency”. These other risk factors still play a role in predicting self-exclusion, but it is a more nuanced role based on the interactions between risk factors and demographic circumstances. How risk factors work together to better isolate self-excluder behavioral patterns would benefit from further examination, considering that risk factors are (in theory at least) often correlated with each other, e.g. someone who spends more time online than before is also likely to be wagering more money in total and placing more bets.

The influence of Germany-based players on the model is small, since it appears at the lowest level of the decision tree with only a small proportion of the samples still present to be differentiated at that node. Nonetheless, the presence of Germany in the model remains hard to interpret. It could be a quirk of the sample used for this analysis, or it could reflect some cultural or structural feature of online gambling in that market relative to other European gambling markets.

Industry Insight on the Importance of Algorithm Interpretability: While presenting a related paper at the 2016 New Horizons in Responsible Gambling conference

[14], we polled the audience and conducted interviews to explore the importance of knowledge extraction and algorithm interpretability (see Table 2 for a summary of the polling data). By a combination of show of hands and smartphone-based electronic polling, the audience was made up of approximately 40% representatives from gambling operators and casinos, 25% regulators and policy officials, 25% academics and other attendees, and 10% clinicians and problem gambling treatment providers. Depending on the question, the sample size was 35 to 40 respondents.

Q: Which would you prefer: an algorithm that assesses problematic play that is 90% accurate which you cannot properly understand or explain OR one that is 75% accurate which you fully understand or explain?

	Total (incl. show of hands)	Gambling operators	Treatment providers	Regulators/policy makers	Academics/Others
Sample Size	40	8	2	6	6
Prefer 90% accurate	20%	13%	-	-	17%
Prefer 75% accurate	70%	75%	50%	67%	83%
It depends /Unsure	10%	13%	50%	33%	-

Q: Is it more important to have model-level interpretability or individual-level interpretability?					
	Total (incl. show of hands)	Gambling operators	Treatment providers	Regulators/policy makers	Academics/Others
Sample Size	35	8	2	5	7
Model-level	20%	25%	-	-	14%
Individual-level	26%	25%	-	-	29%
Both	46%	50%	100%	100%	43%
Neither /Unsure	9%	-	-	-	14%

Table 2: Audience Polling Results from New Horizons in Responsible Gambling Conference, Vancouver, February 2016. Note that audience members contributing via show of hands were not segmented into one of the four categories.

Respondents were asked whether they would prefer a responsible gambling assessment algorithm that provided a 90% accurate assessment of problem gambling risk that they could not explain or understand, or a model that provided a 75% accurate assessment that was fully interpretable and accountable. Only 20% chose the more accurate model, but 70% preferred to sacrifice 15 percentage points of accuracy for greater interpretability; 10% were uncertain or felt it depended on the circumstances, which means overall a significant majority for the interpretable model.² This pattern was broadly consistent among gambling operators, policy officials and treatment providers.

We also explored the value of two different types of interpretation for the audience, namely model-level and individual interpretability. Model-level interpretability entails understanding which inputs and their values are most important for determining a prediction in the model. It does not change from gambler to gambler. Examples of model-level interpretability are coefficients in a logistic regression or the analysis of the TREPAN decision tree above. Such interpretability enables users to challenge, test and gain confidence in the model (since models and model development techniques are known to be imperfect) and understand its strengths and flaws. It might also enable policy makers to take industry-level action based on model insights (e.g. identify if some casino games are very high risk) or point towards ways to simplify the model and get similar accuracy, which might matter for real-time prediction systems or online learning.

The second type is individual-level interpretability, where one can explain specifically why a particular individual was given a particular risk assessment score, what factors contributed most to it, and what the individual might change to not gain such a risk assessment in future. This is

² We compared votes for 'more accurate' vs. those 'explainable' + 'undecided' in a one-sided Binomial test ($n=40$), which yielded $p<0.01$ with null hypothesis of random choice with equal probabilities.

important when one wants to explain a particular decision to someone, since providing a more detailed explanation might help gamblers accept the assessment and take action accordingly. An example of individual-level interpretability is the frequency analyses that can be done in the case of random forests, which are difficult to interpret as a whole but easy to analyse in individual cases by comparing the outcomes of the various decision trees.

In our survey, 46% of respondents said that both model-level and individual-level interpretability were important to them. Of those who said that only one type of interpretation was important, there was a slight preference for individual-level interpretability at 26% vs 20% for model-level. Only 8% were not sure or felt that neither was important. Each level had a significant majority voting for its importance.³ All regulators and public policy officials indicated that both types of interpretability were important.

While this polling evidence presents a clear picture of importance of interpretability, and indeed a willingness to sacrifice some accuracy in favour of clearer models, subsequent interviews with treatment providers identified an insightful way to use both types of models to better protect gamblers. The most accurate model, even if as a black-box, can be used to most accurately identify which gamblers require a responsible gambling intervention, for instance with a responsible gambling message about the player setting a limit, a cessation of email marketing by the operator, or a conversation over a cup of tea in the case of venue-based players. In the case of messaging or a conversation, while the black box can best identify which gambler would benefit from such an intervention, the less accurate but more understandable model can be used to determine the content of these interactions, thus helping the player appreciate and come to terms with their patterns of behaviour and determine what action they might take in the future. The assumption in this approach is that, even with discrepancies in accuracy, both models are fundamentally exploring and interpreting the same underlying features of a person's gambling behavior. Alternatively, we can see that, provided descriptions of a gambler's behavior are relevant and easy to understand, they can be used by professionals in conversations or interventions, even if they are not the most precise predictors of the associated machine learning model.

5 DISCUSSION AND CONCLUSION

In the neural network used here to classify gambling self-exclusion, each input can affect the network's output via 34 different possible routes with an overall model parametrized by over 500 weights. Such a complex set-up, while fully determined mathematically, constitutes a black box from the point of view of human readability. While the model

³ For each level, we applied a one-sided Binomial test ($n=35$), i.e. to 'both' + 'individual' vs. 'model' + 'undecided' votes yielding $p<0.01$, and to 'both' + 'model' vs. 'individual' + 'undecided' votes yielding $p<0.05$ against random choice with equal probabilities.

produces both a predictive assessment of each player and a view of how accurate that assessment is likely to be on average, it fails to permit challenge and validation by domain experts and it fails to allow users to explain risk assessments to gamblers, which might offer the potential for such explanations to lead to changes in behavior.

The reduced form decision tree generated by applying a small variation of the TREPAN knowledge extraction algorithm resolves these concerns with only very minimal loss of overall accuracy (approx. 1 percentage point) and with 87% overall fidelity to the neural network model. The reduced form tree remains somewhat complex, with a height of 5 and 8 distinct decision points at split nodes, showing interactions between the input risk factors that are not necessarily straightforward to interpret. Nonetheless, the reduced form model is human readable and can be translated into a series of statements that are meaningful to domain experts. Simultaneously, it is possible to trace the route of an individual gambler through the decision tree to identify at which points they become more or less likely to be assessed as a self-excluder. Industry leaders and regulators have also indicated that it would be possible to use a simplified, interpretable model, such as the TREPAN decision tree, alongside a more complicated, more accurate model, in order to best serve gambling clients.

However, in [15] random forests were the highest performing method on the same dataset used here, with a lower standard deviation in accuracy across the ten-fold cross-validation and an even balance between sensitivity and specificity, unlike the unhelpful bias towards specificity in the case of the neural network. On this basis, we suggest that a TREPAN decision tree derived from this neural network may not be an optimal approach and its human-readable conclusions should not be treated as fixed. We hope that further work will apply the TREPAN approach to models such as random forests as well. As a result, this paper demonstrates the industry need for different forms of model interpretability and knowledge extraction, alongside the effectiveness of TREPAN as a tool in enabling this requirement to be met.

In what concerns the challenge of using the knowledge extraction results obtained here in a clinical capacity, it is important to note that the players' likelihood of self-excluding was analyzed here by modelling their play data. One could not assess, for example, whether any of the gamblers in our study suffered from poor coping or problem solving skills, or negative family experiences, or suffered from behavioral problems such as substance abuse, which are important elements in pathways 2 and 3 [2]. In order to enable that, one would have to augment the model with data which is much more difficult to obtain, requiring interviews and observations from gambling operator staff in a physical casino or retail environment, for example. That would enable a more detailed assessment, however, of any negative consequences our players may suffer as a result of their gambling behaviors.

In this paper, we have evaluated how well neural networks can be used to classify but more importantly

describe and therefore explain self-excluding gamblers. Understanding gambling behavior can lead to gambling studies that develop new techniques and models capable of managing and helping gambling operators or treatment providers handle problematic players. As our model shows, there are a number of factors that might indicate problematic gambling behavior, and we highlight the value of further investigation on why and how these factors can be used in practice to support gamblers.

ACKNOWLEDGEMENTS

This work is partially supported by Innovate UK and the EPSRC under grant number EP/M50712X/1.

REFERENCES

- [1] I. R. Andrews, J. Diederich, and A. Tickle, Survey and critique of techniques for extracting rules from trained artificial neural networks, *Knowledge Based Systems*, 8(6), 373-389, 1995
- [2] Blaszczynski, A., and Nower, L. (2002). A pathways model of problem and pathological gambling. *Addiction*, 97, 487-499
- [3] Bowyer, K., Chawla, N., and Hall, L., Kegelmeyer P. (2002). SMOTE: Synthetic Minority Over sampling Technique. *Journal Of Artificial Intelligence Research*, 16, pages 321-357.
- [4] Craven, M., and Shavlik, J. (1996). Extracting Tree-Structured Representations of Trained Networks. *Advances in Neural Information Processing Systems*, 37-45.
- [5] M. G. Augasta and T. Kathirvalavakumar, Reverse engineering the neural networks for rule extraction in classification problems, *Neural Processing Letters*, vol. 35, no. 2, pp. 131-150, 2012.
- [6] Diakopoulos, N. (2013). Algorithmic Accountability Reporting: On the Investigation of Black Boxes (Available at http://towcenter.org/wp-content/uploads/2014/02/78524_Tow-Center-Report-WEB-1.pdf, last accessed 2016/03/08)
- [7] Ding, C., and Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology*, 3(2), 185-205.
- [8] Dragičević, S., Tsogas, G., & Kudic, A. (2011). Analysis of casino online gambling data in relation to behavioural risk markers for high-risk gambling and player protection. *International Gambling Studies*, 11 (3), 377-391.
- [9] Ferris and Wynne (2001). *The Canadian Problem Gambling Index: Final report*. Ottawa, ON: Canadian Centre on Substance Abuse
- [10] Franca, M. V. M., Zaverucha, G., and Garcez, A. S. D. (2014). Fast relational learning using bottom clause propositionalization with artificial neural networks. *Machine Learning*, 94(1), 81-104.
- [11] Dragičević, S. (2016). Human-in-the-Loop Machine Learning and Responsible Gambling Analytics (available at: <http://bet-buddy.com/blog/human-in-the-loop-machine-learning-and-responsible-gambling-analytics/>, last accessed 2016/03/22)
- [12] Gainsbury, S., Wood, R. (2011). Internet gambling policy in critical comparative perspective: The

- effectiveness of existing regulatory frameworks. *International Gambling Studies*, 11(3), 309-32.
- [13] Garcez, A. S. D., Broda, K., and Gabbay, D. (2001). Symbolic knowledge extraction from trained neural networks: A sound approach. *Artificial Intelligence*, 125(1-2), 155–207.
- [14] Percy, C. (2016). Can ‘BlackBox’ responsible gambling algorithms be understood by users? A real-world example. New Horizons in Responsible Gambling conference paper, Vancouver, February 2016.
- [15] Percy, C., Franca, N., Dragičević, S., and Garcez, A. S. D. (2016). Predicting online gambling self-exclusion: an analysis of the performance of supervised machine learning models. *International Gambling Studies*. DOI:10.1080/14459795.2016.1151913.
- [16] K. Odajima, Y. Hayashi, G. Tianxia, and R. Setiono, Greedy rule generation from discrete data and its use in neural network rule extraction, *Neural Networks*, vol. 21, no. 7, pp. 1020–1028, 2008.
- [17] Philander, K. S. (2013). Identifying high risk online gamblers: a comparison of data mining procedures. *International Gambling Studies*. DOI: 10.1080/14459795.2013.841721
- [18] Karim, A. and Zhou, S. X-TREPAN: a multi class regression and adapted extraction of comprehensible decision tree in artificial neural networks. arXiv:1508.07551, Aug, 2015.
- [19] R. Setiono, B. Baesens, and C. Mues, “Recursive neural network rule extraction for data with mixed attributes,” *IEEE Trans. Neural Networks*, vol. 19, no. 2, pp. 299–307, 2008.
- [20] T. Schellinck and T. Schrans (2011). Intelligent design: How to model gambler risk assessment by using loyalty tracking data. *Journal of Gambling Issues*: 26(1), 51-68.
- [21] A. Hara and Y. Hayashi, Ensemble neural network rule extraction using RE-RX algorithm. International Joint Conference on Neural Networks (IJCNN), Brisbane, Australia, June 10-15, 2012, 2012.
- [22] T. A. Etchells and P. J. G. Lisboa, Orthogonal search-based rule extraction (OSRE) for trained neural networks: a practical and efficient approach, *IEEE Trans. Neural Networks*, vol. 17, no. 2, pp. 374–384.

Strategy Representation and Reasoning in the Situation Calculus

Liping Xiong and Yongmei Liu¹

Abstract. Strategy representation and reasoning has been one of the most active research areas in AI and multi-agent systems. Representative strategic logics are ATL and the more expressive Strategy Logic SL which reasons about strategies explicitly. In this paper, by a simple extension of the situation calculus with a strategy sort, we develop a general framework for strategy representation and reasoning for complete information games. This framework can be used to compactly represent both concurrent and turn-based possibly infinite game structures, specify the internal structure of strategies, reason about strategies explicitly, and reason about strategic abilities of coalitions under commitments to strategy specifications. We show that our framework is strictly more expressive than SL, and inspired by the work of De Giacomo *et al.* on bounded action theories, give a decidable fragment of our framework.

1 INTRODUCTION

Strategy representation and reasoning has been one of the most active research areas in AI and multi-agent systems, and many strategic logics have been proposed recently. From the representation side, the following are some desiderata of a strategic logic: modeling strategic abilities of coalitions; representation and reasoning about strategies explicitly and even the internal structures of strategies; capability to deal with both concurrent games and turn-based games; compact representation of game structures, even infinite ones. Yet hardly any existing strategic logic has all these desiderata.

Most strategic logics are built upon Alternating-time Temporal Logic (ATL) [2] where formula $\langle\langle A \rangle\rangle\varphi$ expresses that coalition A can ensure temporal formula φ holds no matter what the other agents do. However, strategies are treated implicitly in ATL.

To model strategies explicitly, there are mainly two approaches. The first approach is to treat a strategy as an explicit first-order object in which a strategy is a function from states (or sequences of states) to actions [43, 45, 6, 30]. In particular, based on the work of [6] which introduces first-order quantifications over strategies, Mogavero *et al.* [30] propose Strategy Logic SL, which is a very expressive logic for strategic reasoning and strictly contains ATL*, an extension of ATL. Yet this approach cannot model the internal structure of strategies. The second approach is to treat a strategy as a program so that program connectives can be used to obtain combined strategies from simple ones [44, 33, 41].

Nonetheless, the above strategic logics represent game structures with concrete game models which suffer from the state explosion problem. For example, in the Chess game, there are almost 10^{30} states. To model games compactly, the Game Description Language (GDL) has been proposed as a practical language for encoding the

rules of arbitrary finite games [16]. Based on GDL, Zhang and Thielscher [47, 48] use propositional modal logic formulas to represent strategies, and introduce prioritised connectives for combining strategies. However, their works can only model turn-based games, and cannot model strategic abilities of coalitions.

Other than modal logics, another main family of logics in AI is action formalisms. A prominent example of action formalisms is the situation calculus [34], which is a first-order language with some second-order ingredients suitable for reasoning about action and change. Based on the situation calculus, a logic programming language Golog [22] has been designed for high-level agent control. There have been a few works [37, 17, 10] studying strategic reasoning in the situation calculus. However, all these works deal with turn-based games. The first work does not support ATL-like reasoning, the second one focuses on the coordination problem, and the third one does not support explicit reasoning about strategies.

In this paper, we propose a framework based on the situation calculus for representation and reasoning about strategies for complete information games with all the four desiderata. We first propose a simple extension of the situation calculus with a strategy sort, which can be used to compactly represent both concurrent and turn-based possibly infinite game structures, and to reason about strategies explicitly. We show that SL can be embedded into the extended situation calculus. Then we use a simple fragment of Golog as a strategy specification language, and define the strategy verification and synthesis problems. We illustrate our logical framework with examples of both turn-based and concurrent games. Finally, inspired by the work of De Giacomo *et al.* on bounded action theories [8], we give a decidable fragment of our extended situation calculus.

2 PRELIMINARIES

In this section, we first introduce the situation calculus, and then review the syntax and semantics of Strategy Logic.

2.1 The situation calculus

The *situation calculus* [34] is a many-sorted first-order logic language (with some second-order elements) specifically designed for representing dynamically changing worlds. There are three disjoint sorts: *situation* for situations, *action* for actions, and *object* for everything else. Intuitively, a situation is a finite sequence of actions. In this language, the constant S_0 is used to denote the initial situation; the binary function $do(a, s)$ is used to denote the successor situation of s resulting from performing action a , and $do([a_1, a_2, \dots, a_k], s)$ is used as a shorthand for $do(a_k, \dots, do(a_2, do(a_1, s)))$; the binary predicate $Poss(a, s)$ means that action a is possible in situation s . Actions can be parameterized, e.g., $repair(r, x)$ represents robot r repairing object x . There are relational and functional fluents whose

¹ Department of Computer Science, Sun Yat-sen University, China, email: xionglp3@mail2.sysu.edu.cn & ymliu@mail.sysu.edu.cn.

values vary from situation to situation. These fluents are denoted by symbols that take a situation term as their last argument. There are also situation-independent predicates and functions. Finally, there is a binary predicate \sqsubseteq on situations: $s \sqsubseteq s'$ means that s' is the result of some sequence of actions being performed in s . We use $s \sqsubseteq s'$ as a shorthand for $s \sqsubseteq s' \vee s = s'$. We say that a situation s is executable if it is possible to perform the actions in s one by one:

$$Exec(s) \doteq \forall a, s'. do(a, s') \sqsubseteq s \supset Poss(a, s').$$

In this language, an application domain is specified by a basic action theory (BAT) \mathcal{D} consisting of five disjoint parts:

1. Σ , the foundational axioms of the situation calculus:
 - $\forall P.P(S_0) \wedge \forall a, s[P(s) \supset P(do(a, s))] \supset (\forall s)P(s)$,
 - $do(a, s) = do(a', s') \supset a = a' \wedge s = s'$,
 - $\neg s \sqsubseteq S_0 \wedge (s \sqsubseteq do(a, s') \equiv s \sqsubseteq s')$.
2. \mathcal{D}_{ap} , a precondition axiom for each action function specifying when the action can be legally performed.
3. \mathcal{D}_{ss} , a successor state axiom for each fluent which describes how fluent values change between situations.
4. \mathcal{D}_{una} , unique name axioms for actions.
5. \mathcal{D}_{S_0} , axioms describing the initial situation S_0 .

2.2 Strategy logic (SL)

Strategy Logic [30] is a very expressive logic for reasoning explicitly about strategies in multi-agent concurrent systems; it strictly contains ATL^* , and can express the existence of deterministic multi-player Nash equilibria that cannot be expressed in ATL^* .

We fix an SL signature $\langle AP, AG, Var \rangle$, here AP is a finite non-empty set of atoms, $AG = \{1, \dots, n\}$ is a finite non-empty set of agents, and Var is a countable set of variables.

Definition 1 *SL formulas are built inductively as follows:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \bigcirc\varphi \mid \varphi\mathcal{U}\varphi \mid \langle\langle x \rangle\rangle\varphi \mid (i, x)\varphi,$$

where $p \in AP$, $i \in AG$, and $x \in Var$.

Syntactically, SL extends linear-time temporal logic LTL [32] (\bigcirc means *next*, \mathcal{U} means *until*) with two operators. Intuitively, $\langle\langle x \rangle\rangle$ and (i, x) mean “there exists a strategy x ”, and “bind agent i to the strategy associated with variable x ” respectively. We use \top (resp. \perp) to represent true (resp. false). We use the universal quantifier $\llbracket x \rrbracket$ as the dual of $\langle\langle x \rangle\rangle$, i.e., $\llbracket x \rrbracket = \neg\langle\langle x \rangle\rangle\neg$, which means “for all strategies x ”. For a formula φ , we let $free(\varphi)$ denote the set of *free agents and variables* of φ , and we omit its formal definition here. For example, $free(\langle\langle x \rangle\rangle(1, x)(2, y) \bigcirc p) = \{y\} \cup (AG - \{1, 2\})$.

Like ATL, the semantics of SL is based on the notion of concurrent game structures.

Definition 2 *A concurrent game structure (CGS) is a tuple $\mathcal{G} = \langle AC, W, \lambda, \tau, w^0 \rangle$, where AC and W are finite non-empty sets of actions and states respectively; $w^0 \in W$ is a designated initial state; $\lambda : W \rightarrow 2^{AP}$ is a labeling function; and $\tau : W \times AC^{AG} \rightarrow W$ is a transition function mapping a state and a decision (i.e., a function from AG to AC) to a new state. We also denote AC^{AG} as DC .*

A CGS can be viewed as a multi-player game where players perform concurrent actions strategically.

To define the semantics of SL, we begin with some definitions and notations. A track h in a CGS \mathcal{G} is a finite state sequence $w_0 w_1 \dots w_k$ in W s.t. for all i , $0 \leq i < k$, there exists $d \in DC$ s.t. $w_{i+1} = \tau(w_i, d)$. We let $Trk(\mathcal{G})$ denote the set of all tracks in \mathcal{G} beginning from the initial state w^0 .

A strategy in \mathcal{G} is a function $f : Trk(\mathcal{G}) \rightarrow AC$. Let $Str(\mathcal{G})$ denote the set of all strategies in \mathcal{G} . Intuitively, a strategy is a plan for

an agent which contains the choice of action for any track starting from the initial state.

Like a variable assignment in first-order logic, a strategy assignment is a partial function $\chi : AG \cup Var \rightarrow Str(\mathcal{G})$, mapping agents and variables to strategies. We use $dom(\chi)$ to denote the domain of χ . If $AG \subseteq dom(\chi)$, χ is called complete. We use $\chi[x/f]$ to denote an assignment exactly like χ except that it maps x to f .

Given a complete strategy assignment χ , it determines a unique infinite state sequence $w_0 w_1 w_2 \dots$ and a unique infinite decision sequence $d^1 d^2 \dots$ as follows: $w_0 = w^0$, and for each $j \geq 1$, d^j is the decision associated to the track $w_0 \dots w_{j-1}$, i.e., for each $i \in AG$, $d^j(i) = \chi(i)(w_0 \dots w_{j-1})$, and $w_j = \tau(w_{j-1}, d^j)$. We use $l(\chi)$ to denote this infinite state sequence, and let $l_j(\chi)$ denote the j -th state on the sequence.

Definition 3 *Given a CGS $\mathcal{G} = \langle AC, W, \lambda, \tau, w^0 \rangle$, an SL formula φ , a complete strategy assignment χ with $free(\varphi) \subseteq dom(\chi)$, and a state $w = l_k(\chi)$ for some $k \geq 0$, the relation $\mathcal{G}, w, \chi \models \varphi$ is inductively defined as follows:*

- $\mathcal{G}, w, \chi \models p$ iff $p \in \lambda(w)$;
- $\mathcal{G}, w, \chi \models \neg\varphi$ iff $\mathcal{G}, w, \chi \not\models \varphi$;
- $\mathcal{G}, w, \chi \models \varphi_1 \wedge \varphi_2$ iff $\mathcal{G}, w, \chi \models \varphi_1$ and $\mathcal{G}, w, \chi \models \varphi_2$;
- $\mathcal{G}, w, \chi \models \bigcirc\varphi$ iff $\mathcal{G}, l_{k+1}(\chi), \chi \models \varphi$;
- $\mathcal{G}, w, \chi \models \varphi_1 \mathcal{U} \varphi_2$ iff there is an index $k' \in \mathbb{N}$ with $k \leq k'$ such that $\mathcal{G}, l_{k'}(\chi), \chi \models \varphi_2$ and, for all indexes $j \in \mathbb{N}$ with $k \leq j < k'$, it holds that $\mathcal{G}, l_j(\chi), \chi \models \varphi_1$;
- $\mathcal{G}, w, \chi \models \langle\langle x \rangle\rangle\varphi$ iff there exists a strategy $f \in Str(\mathcal{G})$ such that $\mathcal{G}, w, \chi[x/f] \models \varphi$;
- $\mathcal{G}, w, \chi \models (i, x)\varphi$ iff $\mathcal{G}, w, \chi[i/\chi(x)] \models \varphi$.

An SL formula φ is called a sentence if $free(\varphi)$ is empty. Clearly, for a sentence φ , whether $\mathcal{G}, w, \chi \models \varphi$ does not depend on χ , hence we omit χ . An SL sentence φ is valid if for any CGS \mathcal{G} , we have $\mathcal{G}, w^0 \models \varphi$. Denote $\mathcal{G}, w^0 \models \varphi$ as $\mathcal{G} \models \varphi$.

Example 1 The following are some SL sentences:

1. $\varphi_1 : \langle\langle x \rangle\rangle\llbracket y \rrbracket(1, x)(2, x)(3, y) \bigcirc p$, which intuitively says that agents 1 and 2 can share a strategy to ensure $\bigcirc p$ no matter what strategy agent 3 takes;
2. $\varphi_2 : \langle\langle x \rangle\rangle\llbracket y \rrbracket\langle\langle z \rangle\rangle(1, x)(2, y)(3, z) \bigcirc p$;
3. $\varphi_3 : \langle\langle x \rangle\rangle\llbracket y \rrbracket(1, x)(2, y) \bigcirc p \supset \langle\langle x \rangle\rangle\langle\langle y \rangle\rangle(1, x)(2, y) \bigcirc p$ is valid.

Finally, we introduce an important property of SL that is invariant under local isomorphism, defined as follows:

Definition 4 [28] *Let $\mathcal{G}_1 = \langle AC_1, W_1, \lambda_1, \tau_1, w_1^0 \rangle$ and $\mathcal{G}_2 = \langle AC_2, W_2, \lambda_2, \tau_2, w_2^0 \rangle$ be two CGSs. Then \mathcal{G}_1 and \mathcal{G}_2 are locally isomorphic iff there is a relation $\sim \subseteq W_1 \times W_2$, and a function $g : \sim \rightarrow 2^{AC_1 \times AC_2}$ s.t. the following hold:*

1. $w_1^0 \sim w_2^0$;
2. for all $w_1 \in W_1$ and $w_2 \in W_2$, if $w_1 \sim w_2$ then
 - (a) $\lambda_1(w_1) = \lambda_2(w_2)$;
 - (b) for all $a_1 \in AC_1$, there is $a_2 \in AC_2$ s.t. $(a_1, a_2) \in g(w_1, w_2)$;
 - (c) for all $a_2 \in AC_2$, there is $a_1 \in AC_1$ s.t. $(a_1, a_2) \in g(w_1, w_2)$;
 - (d) for all $(d^1, d^2) \in \hat{g}(w_1, w_2)$, it holds that $\tau_1(w_1, d^1) \sim \tau_2(w_2, d^2)$, where $\hat{g} : \sim \rightarrow 2^{DC_1 \times DC_2}$ mapping pairs of states in \sim to relations between decisions such that $(d^1, d^2) \in \hat{g}(w_1, w_2)$ iff, for all $i \in AG$, it holds that $(d^1(i), d^2(i)) \in g(w_1, w_2)$.
3. $\sim \cap (\{\tau_1(w_1, d) : d \in DC_1\} \times \{\tau_2(w_2, d) : d \in DC_2\})$ is a bijective function, for all $w_1 \in W_1$ and $w_2 \in W_2$ with $w_1 \sim w_2$.

Theorem 1 [28] *For SL, it is invariant under local-isomorphism. That is, if two CGSs \mathcal{G}_1 and \mathcal{G}_2 are locally isomorphic, then for any SL sentence φ , $\mathcal{G}_1 \models \varphi$ iff $\mathcal{G}_2 \models \varphi$.*

3 AN EXTENSION OF THE SITUATION CALCULUS

In this section, we present a simple extension of the situation calculus with a strategy sort, denote as \mathcal{L}_{ext} , which can be used to compactly represent both concurrent and turn-based possibly infinite game structures, and to reason about strategies explicitly.

We fix a set of agents $AG = \{1, \dots, n\}$. We introduce two additional sorts: a sort for *joint actions* and a second-order sort for *s-strategies*. Intuitively, a joint action is an n -ary vector of actions, one action for each agent. A strategy is a function from situations to actions. Let $A \subseteq AG$. A collective strategy of coalition A is a function from A to strategies. A joint strategy is a collective strategy of AG . We use variables d, d', \dots for joint actions, g, g', \dots for strategies, g_A, g'_A, \dots for collective strategies of coalition A , and g_{all}, g'_{all}, \dots for joint strategies. We treat g_A the same as the set of strategy variables $\{g_i \mid i \in A\}$.

We introduce a function $joint(a_1, \dots, a_n)$ which maps n actions into a joint action, and n projection functions $pr_i(d)$, $1 \leq i \leq n$, which maps a joint action into its i -th component. For simplicity, we write $joint(a_1, \dots, a_n)$ as $\langle a_1, \dots, a_n \rangle$, and write $pr_i(d)$ as d_i . Situations are now sequences of joint actions; so the first argument of the function do and the predicate $Poss$ is of the joint action sort.

The set Σ of foundational axioms for situations is the same as before except that we replace each action variable with a joint action variable. We also add to Σ the following concerning joint actions:

- $\forall d \exists a_1, \dots, a_n. d = \langle a_1, \dots, a_n \rangle$;
- $\langle a_1, \dots, a_n \rangle = \langle a'_1, \dots, a'_n \rangle \supset a_1 = a'_1 \wedge \dots \wedge a_n = a'_n$;
- $pr_i(\langle a_1, \dots, a_n \rangle) = a_i, i = 1, \dots, n$.

Reiter [34] presents an account of true concurrency where a concurrent action is modeled as a possibly infinite set of simple actions. Our account of concurrent actions as joint actions can be considered as a special case of Reiter's account. ConGolog [7] is an extension of Golog with a rich account of interleaved concurrency. In contrast to interleaved concurrency, our framework is able to deal with the issues of action precondition interaction and concurrent effect specification, which are discussed in [34]. Two simple actions may each be possible, their preconditions may be jointly consistent, yet intuitively they should not be concurrently possible, *e.g.*, each of two robots can walk through a narrow door, but they cannot walk through the door at the same time; also, the effect of concurrently executing two simple actions, *e.g.*, lifting the two ends of a table, might be different from executing the two actions in sequence. Both examples can be specified in our framework.

In the absence of the precondition interaction problem, we introduce n predicates $Poss_i(a, s)$, meaning that it is possible for agent i to perform action a in situation s , and let $Poss(\langle a_1, \dots, a_n \rangle, s) \equiv \bigwedge_{i=1}^n Poss_i(a_i, s)$. To represent turn-based games, we introduce an action *nop* meaning doing nothing, and n fluents $turn_i(s)$, $i = 1, \dots, n$, meaning that it's agent i 's turn to make a move. We have $Poss_i(nop, s) \equiv \neg turn_i(s)$.

Let g_A be a collective strategy of coalition A . The abbreviation $s \sqsubseteq_{g_A} s'$ is used to represent the formula

$$s \sqsubseteq_{g_A} s' \wedge \forall s'' \forall d [s \sqsubseteq do(d, s'') \sqsubseteq s' \supset \bigwedge_{i \in A} d_i = g_i(s'')].$$

Intuitively, this means that s is a subhistory of s' , and on the way from s to s' , each agent i in A performs actions according to strategy g_i . Further, we introduce the abbreviation:

$$s \leq_{g_A} s' \doteq s \sqsubseteq_{g_A} s' \wedge Exec(s').$$

We remark that our idea of extending the situation calculus with joint actions and strategies originates from the literature. To reason about general games, Schiffel and Thielscher [35, 36] introduce joint

actions into the situation calculus. To study ability and knowing how in the situation calculus, Lespérance *et al.* [19] introduce strategies, which they call *action choice functions*. Yet De Giacomo *et al.* [11] present another approach to modeling concurrent game structures in the situation calculus. They introduce a subsort *move* of the *object* sort, representing agents' moves. They assume only one action function $tick(m_1, \dots, m_n)$, representing that each agent i performs move m_i simultaneously.

To illustrate the expressiveness of our language for representing strategic properties, we adapt an example from [30] concerning the existence of deterministic multi-player Nash equilibria:

Example 2 Consider two agents: agent 1 has the temporal goal $\diamond p$ ("eventually p "), and agent 2 has the temporal goal $\diamond q$. Then, we can express the existence of a strategy profile (g_1, g_2) that is a Nash equilibrium for the two agents wrt their goals by the following sentence:

$$\begin{aligned} \exists g_1, g_2. \{ & (\exists g)(\exists s)[p(s) \wedge S_0 \leq_{(g, g_2)} s] \supset \\ & (\exists s)[p(s) \wedge S_0 \leq_{(g_1, g_2)} s] \} \wedge \\ & \{ (\exists g)(\exists s)[q(s) \wedge S_0 \leq_{(g_1, g)} s] \supset \\ & (\exists s)[q(s) \wedge S_0 \leq_{(g_1, g_2)} s] \} \end{aligned}$$

Informally, this asserts that each agent has the "best" strategy once the strategy of the other agent has been fixed.

In the following, we illustrate with examples the expressiveness of our situation calculus language for compactly representing possibly infinite games.

Example 3 Consider the Chomp game [31]. As shown in Figure 1(a), cookies are laid out on a $m \times n$ or $\omega \times \omega$ grid, here ω is the least infinite ordinal. The cookie in the top left position (0,0) is poisoned. Two players take turns making moves: at each move, a player is required to eat a remaining cookie, together with all cookies to the right and/or below it. The loser is the player who has no choice but to eat the poisoned cookie.

Consider representing the Chomp game with concurrent game structures. The ordinal version cannot be represented with a CGS. A CGS representation of an $m \times m$ game would use m^2 actions and at least 2^m states. Hence a CGS representation of the game needs exponential space.

We now formalize this game in the situation calculus. Fluent $ck(j, k, s)$ means that there is a cookie at position (j, k) . Action $eat(j, k)$ means eating the cookie at position (j, k) together with all cookies to the right and/or below it. We use $size(m, n, s)$ to denote the formula $\forall j, k. ck(j, k, s) \equiv j < m \wedge k < n$, meaning that the current cookies form an $m \times n$ grid. In addition to the second-order axiomatization of Peano arithmetic, we have the following axioms:

$$\begin{aligned} Poss_i(eat(j, k), s) & \equiv turn_i(s) \wedge ck(j, k, s), i = 1, 2; \\ turn_i(do(d, s)) & \equiv \neg turn_i(s), i = 1, 2; \\ ck(j, k, do(d, s)) & \equiv ck(j, k, s) \wedge \neg \exists j', k'. [(j' \leq j \\ & \vee k' \leq k) \wedge (d_1 = eat(j', k') \vee d_2 = eat(j', k'))]; \\ turn_1(S_0) \wedge \neg turn_2(S_0); \\ (\forall j, k) ck(j, k, S_0) & \vee (\exists m, n) size(m, n, S_0). \end{aligned}$$

We use \mathcal{D}_{ch} to denote the BAT of Chomp, and we use \mathcal{D}_{mm} (resp. \mathcal{D}_{2m}) to denote \mathcal{D}_{ch} with the last axiom replaced by $\exists m. size(m, m, S_0)$ (resp. $\exists m. size(2, m, S_0)$). The abbreviation below says that agent i wins in situation s :

$$Win_i(s) \doteq \neg turn_i(s) \wedge size(1, 1, s).$$

It's easy to prove by a non-constructive proof that for Chomp game, player 1 always has a winning strategy. So we have

$$\mathcal{D}_{ch} \models \exists g_1 \forall g_2 \exists s. S_0 \leq_{(g_1, g_2)} s \wedge Win_1(s).$$

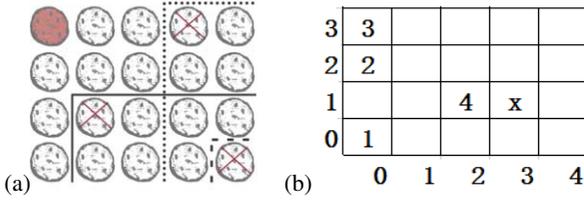


Figure 1. The Chomp and Thieves Games

Finally, we present an example of concurrent games.

Example 4 In a two-dimensional world, three thieves (agents 1-3) attempt to steal a treasure, which is protected by a guard (agent 4), and can only be lifted by at least two agents. As shown in Figure 1(b), the treasure is located at (3,1); initially, the thieves are located at (0,0), (0,2), (0,3) respectively, and the guard is located at (2,1). Any agent can stay at his current position. The thieves can move *right*, *up*, or *down* a unit, when they are not caught by the guard. A thief is caught by the guard if they share the same location. The guard can move *left*, *up*, or *down* a unit; he can also move *right* a unit when he is to the left of the treasure. We use fluent $loc_i(p, q, s)$ to represent that in situation s , agent i is located at (p, q) ; use fluent $cat_i(s)$ to mean that thief i is caught by the guard; and use fluent $win(s)$ to mean that the thieves successfully get the treasure. For illustration purpose, we only present some axioms of the BAT of this game, denoted by \mathcal{D}_{tg} :

$$\begin{aligned}
Poss_i(right, s) &\equiv \neg cat_i(s), i = 1, 2, 3; \\
Poss_4(right, s) &\equiv \exists p, q. loc_4(p, q, s) \wedge p < 2; \\
loc_i(p, q, do(d, s)) &\equiv \phi_i(p, q, d, s), \text{ where } \phi_i(p, q, d, s) \text{ is} \\
&loc_i(p, q, s) \wedge d_i = stay \vee \\
&loc_i(p+1, q, s) \wedge d_i = left \vee \dots, i = 1, 2, 3, 4; \\
cat_i(do(d, s)) &\equiv \exists p, q. \phi_i(p, q, d, s) \wedge \phi_4(p, q, d, s) \vee \\
&cat_i(s), i = 1, 2, 3; \\
win(do(d, s)) &\equiv d_1 = left \wedge d_2 = left \vee d_2 = left \wedge \\
&d_3 = left \vee d_1 = left \wedge d_3 = left \vee win(s); \\
loc_1(0, 0, S_0) \wedge \neg cat_1(S_0).
\end{aligned}$$

Finally, we illustrate that in our situation calculus language, we can specify different notions of strategies, such as memoryless and perfect recall strategies. We first introduce two abbreviations:

1. $Eqst(s, s')$ which says that the states of s and s' are the same [26]:

$$Eqst(s, s') \doteq \bigwedge_{i=1}^n \forall \vec{x}_i. (F_i(\vec{x}_i, s) \equiv F_i(\vec{x}_i, s')) \wedge \bigwedge_{j=1}^m \forall \vec{y}_j. (f_j(\vec{y}_j, s) = f_j(\vec{y}_j, s'))$$

Here $F_i(\vec{x}_i, s)$, $1 \leq i \leq n$ are the finitely many relational fluents, and $f_j(\vec{y}_j, s)$, $1 \leq j \leq m$ are the finitely many functional fluents.

2. $Eqlev(s, s')$ which says that s and s' are situations at the same level, *i.e.*, situations resulting from performing the same number of joint actions:

$$Eqlev(s, s') \doteq \forall P. P(S_0, S_0) \wedge \forall s_1, s'_1, d, d'. [P(s_1, s'_1) \supset P(do(d, s_1), do(d', s'_1))] \supset P(s, s').$$

Then we can describe different notions of strategies according to different statewise memory abilities (ignoring actions) as follows:

- Statewise memoryless strategy:
 $M_0(g) \doteq \forall s, s'. Eqst(s, s') \supset g(s) = g(s').$
- Statewise perfect recall strategy:
 $PR(g) \doteq \forall s, s'. [\forall s_1, s'_1. s_1 \sqsubseteq s \wedge s'_1 \sqsubseteq s' \wedge Eqlev(s_1, s'_1) \supset Eqst(s, s')] \wedge Eqlev(s, s') \supset g(s) = g(s').$
- Statewise perfect recall strategy wrt agent j :
 $PR_j(g) \doteq \forall s, s'. [\forall s_1, s'_1, d, d'. do(d, s_1) \sqsubseteq s \wedge do(d', s'_1) \sqsubseteq s' \wedge Eqlev(s_1, s'_1) \wedge d_j = d'_j \supset Eqst(s, s')] \wedge Eqlev(s, s') \supset g(s) = g(s').$

- K -bounded-memory strategy (take the example of $K = 1$):
 $M_1(g) \doteq \forall s, s'. Eqst(s, s') \wedge \forall s_1, s'_1, d, d'. [do(d, s_1) = s \wedge do(d', s'_1) = s' \supset Eqst(s_1, s'_1)] \supset g(s) = g(s').$

4 EMBEDDING SL INTO THE SITUATION CALCULUS

In this section, we show that Strategy Logic can be embedded into our extended situation calculus, *i.e.*, our framework is expressive enough to contain SL. With this result, we establish formal connections between our work and existing works.

Firstly, we give an encoding of a concurrent game structure \mathcal{G} into a BAT $\mathcal{D}_{\mathcal{G}}$; then we present a translation function which maps an SL sentence φ into a situation calculus formula φ' ; finally, we prove that for any CGS \mathcal{G} and SL sentence φ , $\mathcal{G}, w^0 \models \varphi$ iff $\mathcal{D}_{\mathcal{G}} \models \varphi'$.

We fix an SL signature $\langle AP, AG, Var \rangle$. Given a CGS $\mathcal{G} = \langle AC, W, \lambda, \tau, w^0 \rangle$, where $AC = \{a^0, \dots, a^{m-1}\}$ and $W = \{w^0, \dots, w^{l-1}\}$, the vocabulary of our situation calculus language includes the following: a set of action constants $\{ac_0, \dots, ac_{m-1}\}$, and a set of state constants $\{sc_0, \dots, sc_{l-1}\}$; a functional fluent $State(s)$, representing the state of the game in situation s ; for each $p \in AP$, a relational fluent $p(s)$, denoting that p holds in situation s .

We let $\phi(x, d, y)$ denote the following formula, which says y is the state resulting from performing joint action d in state x , *i.e.*, $\phi(x, d, y)$ represents the transition function τ :

$$\forall \{x = sc_j \wedge d = \langle ac_{j_1}, \dots, ac_{j_n} \rangle \wedge y = sc_k \mid j < l, j_1 < m, \dots, j_n < m, \tau(w^j, \langle a_{j_1}, \dots, a_{j_n} \rangle) = w^k\}.$$

Definition 5 Given a CGS $\mathcal{G} = \langle AC, W, \lambda, \tau, w^0 \rangle$, the BAT $\mathcal{D}_{\mathcal{G}}$ includes the following domain-specific axioms:

- (A0) $sc_j \neq sc_k, ac_j \neq ac_k, j \neq k$;
- (A1) $Poss_i(ac_j, s) \equiv \top, i = 1, \dots, n, j < m$;
- (A2) $(State(do(d, s)) = x) \equiv \phi(State(s), d, x)$;
- (A3) $p(do(d, s)) \equiv \bigvee \{\phi(State(s), d, sc_j) \mid p \in \lambda(w^j)\}, p \in AP$;
- (A4) $State(S_0) = sc_0$;
- (A5) $\bigwedge \{p(S_0) \mid p \in \lambda(w^0)\} \wedge \bigwedge \{\neg p(S_0) \mid p \notin \lambda(w^0)\}$.

Here A0 consists of the unique name axioms for states and actions. A1 says that each action is always possible. A2 is the successor state axiom for fluent $State(s)$, saying that x is the state after executing joint action d in situation s iff x is the state resulting from performing d in the state of s . A3 is the successor state axiom for fluent $p(s)$, saying that p holds after doing d in s iff the new state is one of the states where p holds. A4 states that the initial state is w^0 and A5 specifies the initial truth values of $p \in AP$ according to $\lambda(w^0)$.

Given an SL formula φ , a joint strategy g_{all} , and a situation s , we define a situation calculus formula $T(\varphi, g_{all}, s)$, which intuitively means that when the agents commit to the joint strategy g_{all} , φ holds in situation s . We use Var as the set of strategy variables in our situation calculus language. Let $x \in Var$. We let $g_{all}[i/x]$ denote the joint strategy exactly like g_{all} except that agent i adopts strategy x . Finally, we use $g_{all}(s)$ to represent the joint action at situation s , *i.e.*, $\langle g_1(s), \dots, g_n(s) \rangle$.

Definition 6 Given an SL formula φ , a joint strategy g_{all} , and a situation s , we define a situation calculus formula $T(\varphi, g_{all}, s)$ inductively as follows:

- $T(p, g_{all}, s) = p(s)$, for each $p \in AP$;
- $T(\neg \varphi, g_{all}, s) = \neg T(\varphi, g_{all}, s)$;
- $T(\varphi_1 \wedge \varphi_2, g_{all}, s) = T(\varphi_1, g_{all}, s) \wedge T(\varphi_2, g_{all}, s)$;

- $T(\bigcirc\varphi, g_{all}, s) = T(\varphi, g_{all}, do(g_{all}(s), s));$
- $T(\varphi_1 \mathcal{U} \varphi_2, g_{all}, s) = \exists s'. s \leq_{g_{all}} s' \wedge T(\varphi_2, g_{all}, s') \wedge \forall s'' [s \sqsubseteq s'' \sqsubset s' \supset T(\varphi_1, g_{all}, s'')];$
- $T(\langle\langle x \rangle\rangle\varphi, g_{all}, s) = \exists x. T(\varphi, g_{all}, s);$
- $T(\langle i, x \rangle\varphi, g_{all}, s) = T(\varphi, g_{all}[i/x], s).$

Given the semantics of SL (Definition 3), the above translation is quite straightforward.

Example 5 We illustrate the above translation with the SL sentence $\varphi : \langle\langle x \rangle\rangle \langle\langle y \rangle\rangle \langle\langle z \rangle\rangle (1, x)(2, y)(3, z) \bigcirc p$. Let g_{all} be the joint strategy $\langle u, v, w \rangle$. Then we have:

1. $T(\bigcirc p, g_{all}, s) = p(do(\langle u(s), v(s), w(s) \rangle, s));$
2. $T(\langle\langle x \rangle\rangle \langle\langle y \rangle\rangle \langle\langle z \rangle\rangle (1, x)(2, y)(3, z) \bigcirc p, g_{all}, s) = p(do(\langle x(s), y(s), z(s) \rangle, s));$
3. $T(\varphi, g_{all}, s) = \exists x \forall y \exists z. p(do(\langle x(s), y(s), z(s) \rangle, s)).$

Note that if φ is a sentence, then $T(\varphi, g_{all}, s)$ does not depend on g_{all} and it does not have free strategy variables. In this case, we omit g_{all} and simply write $T(\varphi, s)$.

We now state the embedding theorem as follows:

Theorem 2 Given a CGS \mathcal{G} and an SL sentence φ , we have $\mathcal{G}, w^0 \models \varphi$ iff $\mathcal{D}_{\mathcal{G}} \models T(\varphi, S_0)$.

A few remarks are in order before we prove the theorem. First of all, the significance of Theorem 2 lies with that it formally shows that SL can be embedded into the extended situation calculus and it establishes the correctness of our translation function $T(\varphi, g_{all}, s)$. Secondly, indeed, when encoding a CGS as a BAT, the states are represented explicitly, and the size of our encoding is of the same order as the size of the CGS. One may question that this brute-force encoding does not make use of the advantage of the situation calculus in compactly representing large state spaces. However, this is the best we can do given an arbitrary CGS. To represent a specific game structure, we might very well have a more succinct representation in the situation calculus than using a CGS. But given an arbitrary CGS where states are individual objects rather than being factored into atomic features, we cannot restore the original game structure and give a succinct representation. Finally, the extended situation calculus is more expressive than SL, which does not provide a way to compactly represent game structures. In SL, we can only discuss whether a game represented by a CGS has a certain property. In contrast, in the situation calculus, we can discuss properties of a class of games represented by a BAT. For example, \mathcal{D}_{ch} represents the class of Chomp games, either of any finite size or infinite, and \mathcal{D}_{mm} the class of Chomp games with square grids of any size.

To prove the theorem, we first prove a lemma.

A strategy assignment in a model M of $\mathcal{D}_{\mathcal{G}}$ is a partial function from Var to the set of strategies in M . Given a CGS $\mathcal{G} = \langle AC, W, \lambda, \tau, w^0 \rangle$, a joint strategy g_{all} , a complete strategy assignment χ in \mathcal{G} , and a state $w = l_k(\chi)$ for some $k \geq 0$, we construct a model $\bar{\mathcal{G}}$ of $\mathcal{D}_{\mathcal{G}}$, a strategy assignment $\bar{\chi}$ in $\bar{\mathcal{G}}$ s.t. $g_{all} \subseteq dom(\bar{\chi})$, and a situation \bar{w} in $\bar{\mathcal{G}}$ as follows.

The action and object domains of $\bar{\mathcal{G}}$ are AC and W respectively. We also call an object a state. The joint action domain of $\bar{\mathcal{G}}$ is the set of n -ary vectors of actions, and the situation domain of $\bar{\mathcal{G}}$ is the set of finite sequences of joint actions. We interpret ac_j as a^j , sc_j as w^j , and S_0 as the empty sequence. The \sqsubseteq predicate and the do , $joint$ and $projection$ functions get their natural interpretations. We interpret $Poss_i(ac_j, s)$, the fluents $State(s)$ and $p(s)$ according to axioms A1-A5 of Definition 5. Clearly, $\bar{\mathcal{G}}$ is a model of $\mathcal{D}_{\mathcal{G}}$. Intuitively, there is a one-to-one correspondence between the situations of $\bar{\mathcal{G}}$ and the tracks of \mathcal{G} starting from the initial state. So there is a one-to-one correspondence between strategies of $\bar{\mathcal{G}}$ and \mathcal{G} .

For a strategy κ in \mathcal{G} , we define a strategy $\bar{\kappa}$ in $\bar{\mathcal{G}}$ as follows. Let μ be a situation in $\bar{\mathcal{G}}$. Then there exist joint actions d^1, \dots, d^k s.t. μ is $[d^1, \dots, d^k]$. We get a track $h = w_0 \dots w_k$ in \mathcal{G} s.t. $w_0 = w^0$, $w_j = \tau(w_{j-1}, d^j)$, $1 \leq j \leq k$. Let $\bar{\kappa}(\mu) = \kappa(h)$. We now define $\bar{\chi}$ as follows. For any $i \in AG$, $\bar{\chi}(g_i) = \bar{\kappa}$, where $\kappa = \chi(i)$; and for any $x \in AG \cup (Var - g_{all})$, $\bar{\chi}(x) = \bar{\kappa}$, if $\chi(x)$ is defined and $\kappa = \chi(x)$, otherwise $\bar{\chi}(x)$ is undefined. As discussed before Definition 3, χ determines a unique infinite state sequence $w_0 w_1 w_2 \dots$ and a unique infinite decision sequence $d^1 d^2 \dots$. Since $w = l_k(\chi)$, i.e., $w = w_k$, we let \bar{w} be the situation $[d^1, \dots, d^k]$. By induction on φ :

Lemma 1 Given an SL formula φ , a CGS \mathcal{G} , a joint strategy g_{all} , a complete strategy assignment χ in \mathcal{G} , and a state $w = l_k(\chi)$ for some $k \geq 0$, we have $\mathcal{G}, w, \chi \models \varphi$ iff $\bar{\mathcal{G}}, \bar{w}, \bar{\chi} \models T(\varphi, g_{all}, s)$.

Note that any model of $\mathcal{D}_{\mathcal{G}}$ is isomorphic to $\bar{\mathcal{G}}$. Hence Theorem 2 follows immediately from Lemma 1.

Therefore, this extended situation calculus is at least as expressive as SL. In the following, we show that our extended situation calculus is strictly more expressive than SL.

Example 6 Consider the following sentence φ_0 in our extended situation calculus: $\exists x, y. x(S_0) \neq y(S_0) \wedge \forall z \exists s, s'. S_0 \sqsubseteq_{(x,z)} s \wedge S_0 \sqsubseteq_{(y,z)} s' \wedge p(s) \wedge p(s')$. Intuitively, this sentence means that agent 1 has two different strategies where agent 1 performs different actions at S_0 to ensure that eventually p holds.

Proposition 1 There exist two CGSs \mathcal{G}_1 and \mathcal{G}_2 such that no SL sentence can distinguish between \mathcal{G}_1 and \mathcal{G}_2 but φ_0 distinguishes between $\bar{\mathcal{G}}_1$ and $\bar{\mathcal{G}}_2$.

Proof: Given a signature $\langle \{p\}, \{1, 2\}, Var \rangle$, let $\mathcal{G}_1 = \langle AC, W, \lambda, \tau_1, w_0 \rangle$ and $\mathcal{G}_2 = \langle AC, W, \lambda, \tau_2, w_0 \rangle$, where $AC = \{a_0, a_1, a_2\}$, $W = \{w_0, w_1, w_2\}$. $\lambda(w_1) = \{p\}$, and $\lambda(w_0) = \lambda(w_2) = \emptyset$. $\tau_1(w_0, (a_0, *)) = \tau_1(w_0, (a_1, *)) = \tau_2(w_0, (a_0, *)) = \tau_2(w_0, (a_1, a_1)) = w_1$, $\tau_1(w_0, (a_2, *)) = \tau_2(w_0, (a_2, *)) = \tau_2(w_0, (a_1, a_0)) = \tau_2(w_0, (a_1, a_2)) = w_2$, $\tau_1(w_i, (*, *)) = \tau_2(w_i, (*, *)) = w_i$, $i = 1, 2$, where $*$ denotes any action. We can easily see that $\bar{\mathcal{G}}_1$ satisfies φ_0 , but $\bar{\mathcal{G}}_2$ does not. However, \mathcal{G}_1 and \mathcal{G}_2 are locally isomorphic, because we can let $\sim = \{(s_i, s_i) : i = 0, 1, 2\}$, function $g : \sim \rightarrow 2^{AC \times AC}$, $g(s_i, s_i) = \{(a_0, a_0), (a_1, a_1), (a_2, a_2)\}$. By Theorem 1, no SL sentence can distinguish between \mathcal{G}_1 and \mathcal{G}_2 . ■

By Theorem 2 and Proposition 1, we get

Corollary 1 The extended situation calculus is strictly more expressive than SL.

5 STRATEGY SPECIFICATION

In this section, we propose a simple strategy specification language based on Golog, and define the strategy verification and synthesis problems.

A situation-suppressed formula φ is a situation calculus formula with all situation arguments suppressed, and $\varphi[s]$ denotes the formula obtained from φ by taking s as the situation arguments of all fluents mentioned in φ .

Definition 7 Strategy specifications are defined inductively as follows: $\delta ::= (\varphi?; \alpha) \mid (\alpha; \varphi?) \mid (\delta \mid \delta) \mid (\pi x. \delta(x))$, where φ is a situation-suppressed formula, and α is an action term.

Intuitively, $\varphi?; \alpha$ means that if φ holds then performing α , $\alpha; \varphi?$ means performing α so that φ holds, $\delta_1 \mid \delta_2$ represents nondeterministic choice of two strategies, and $\pi x. \delta(x)$ stands for nondeterministic choice of strategy arguments.

Note that our strategy specification language does not use the sequence or iteration constructs of Golog. As argued by van Benthem

[42], such a flat language often suffices for the purpose of strategy specifications: a strategy prescribes one move at a time, subject to local conditions; then local iterations make little sense, and local sequences only make sense when a player has consecutive turns.

The formal semantics of strategy specifications is defined by an abbreviation $Does_i(\delta, s, a)$, which intuitively means that action a forms a legal execution of δ by agent i in situation s .

Definition 8 $Does_i(\delta, s, a)$ is defined inductively as:

- $Does_i(\varphi?; \alpha, s, a) = \varphi[s] \wedge Poss_i(\alpha, s) \wedge a = \alpha;$
- $Does_i(\alpha; \varphi?, s, a) = \forall d. (Poss(d, s) \wedge d_i = \alpha \supset \varphi[do(d, s)]) \wedge Poss_i(\alpha, s) \wedge a = \alpha;$
- $Does_i(\delta_1 | \delta_2, s, a) = Does_i(\delta_1, s, a) \vee Does_i(\delta_2, s, a);$
- $Does_i(\pi x. \delta(x), s, a) = \exists x. Does_i(\delta(x), s, a).$

We now formally define the notion that a strategy g satisfies a specification δ for agent i .

Definition 9 The condition of specification δ for agent i in situation s , denoted $Cond_i(\delta, s)$, is defined inductively as:

- $Cond_i(\varphi?; \alpha, s) = \varphi[s] \wedge Poss_i(\alpha, s);$
- $Cond_i(\alpha; \varphi?, s, a) = \forall d. (Poss(d, s) \wedge d_i = \alpha \supset \varphi[do(d, s)]) \wedge Poss_i(\alpha, s);$
- $Cond_i(\delta_1 | \delta_2, s) = Cond_i(\delta_1, s) \vee Cond_i(\delta_2, s);$
- $Cond_i(\pi x. \delta(x), s) = \exists x. Cond_i(\delta(x), s).$

Definition 10 Given a strategy g and a specification δ for agent i , $Sat_i(g, \delta) \doteq \forall s. Cond_i(\delta, s) \supset Does_i(\delta, s, g(s)).$

Intuitively, $Sat_i(g, \delta)$ means that strategy g satisfies specification δ for agent i , that is, for any situation s , under the condition of δ in s , action $g(s)$ forms a legal execution of δ .

In the following, according to Sat_i , we give two relations between two strategy specifications.

Definition 11 Given a BAT \mathcal{D} , and a strategy constant f for agent i , two strategy specifications δ, δ' are equivalent with respect to f and i , if $\mathcal{D} \models Sat_i(f, \delta) \equiv Sat_i(f, \delta')$ holds. Call δ, δ' are equivalent w.r.t. agent i , if $\mathcal{D} \models \forall g. Sat_i(g, \delta) \equiv Sat_i(g, \delta')$.

Two strategy specifications δ, δ' for agent i are called disjoint, if $\mathcal{D} \models \forall s. \neg(Cond_i(\delta, s) \wedge Cond_i(\delta', s)).$

Intuitively, two specifications δ, δ' are equivalent with the strategy f for agent i , which means these two specifications satisfy the f at the same time; and two specifications equivalent w.r.t. agent i means that they specify the same set of strategies for agent i . Two specifications for agent i are disjoint, which means the conditions of two specifications for i are inconsistent.

For any disjoint specifications δ_1, δ_2 for agent i , a strategy for i satisfies $\delta_1 | \delta_2$ iff the strategy satisfies both δ_1 and δ_2 .

Proposition 2 For disjoint strategy specifications δ_1, δ_2 for agent i , given any strategy constant f for i , we have

$$\mathcal{D} \models Sat_i(f, \delta_1 | \delta_2) \equiv Sat_i(f, \delta_1) \wedge Sat_i(f, \delta_2).$$

In fact, any strategy specification for agent i has an equivalent canonical specification for it.

Definition 12 A strategy specification δ is called standard if the δ is of the following form, $\delta_1 | \dots | \delta_n$, in which for any $i \in \{1, \dots, n\}$, δ_i is like the following $\varphi_1?; a, a; \varphi?$ or $\pi \vec{x}. \varphi_2(\vec{x})?; a(\vec{x})$, where φ_1 and $\varphi_2(\vec{x})$ are uniform situation suppressed formulas.

Proposition 3 Given a basic action theory \mathcal{D} and any strategy specification δ for agent i , there must exist an equivalent standard specification δ' for agent i .

A collective strategy specification δ_A for coalition A is a function from A to strategy specifications. Below we formalize the concept of winning strategies. Recall we use g_{all} to denote a joint strategy.

Definition 13 (Winning strategy) Given a basic action theory \mathcal{D} , a situation-calculus formula $\varphi(g_{all}, s)$, a coalition A , and a collective strategy specification δ_A for A , we say that δ_A is a winning strategy for A wrt φ if $\mathcal{D} \models \forall g_{all}. \bigwedge_{i \in A} Sat_i(g_i, \delta_i) \supset \varphi(g_{all}, S_0)$ holds.

Here $\varphi(g_{all}, s)$ is a general situation-calculus formula whose free variables are among g_{all} and s . An example of such a formula is $\exists s. S_0 \leq_{g_{all}} s \wedge \phi(s)$, which means that via the joint strategy g_{all} , ϕ eventually holds at some executable situation. Other examples are $\forall s. S_0 \leq_{g_{all}} s \wedge \phi(s)$, meaning that via g_{all} , ϕ always holds, and $\phi(do(g_{all}(s), s))$, meaning that via g_{all} , ϕ holds next.

According to Propositions 2 and 3, when we want to find a winning strategy for A wrt φ , we can try attempt to combine those disjoint specifications of the forms $\varphi?; a$ or $a(\vec{x})?; \varphi(\vec{x})?$ to get one standard specification δ_A , then verify whether δ_A is a winning strategy.

We now illustrate strategy specifications with the Chomp, Thieves games, and the Nim game.

Example 3 Cont'd. For the chomp game, although the first player always has a winning strategy, nobody has been able to describe one that applies for all rectangular grids. Yet winning strategies can be specified when the grid is square or only has two finite rows or columns. When the grid is square, player 1's winning strategy is as follows: first do $eat(1, 1)$, and on subsequent moves, if player 2 does $eat(0, k)$ (resp. $eat(k, 0)$), then react with action $eat(k, 0)$ (resp. $eat(0, k)$). When the grid has two rows and m columns, player 1's winning strategy is this: begin with $eat(1, m - 1)$, and on subsequent moves, if player 2 does action $eat(0, k)$ (resp. $eat(1, k)$), then respond with $eat(1, k - 1)$ (resp. $eat(0, k + 1)$). We now present specifications for the above strategies. We first give an abbreviation: $Prev(i, a, s) \doteq \exists s' \exists d. s = do(d, s') \wedge d_i = a$, saying that a is the last action performed by agent i in situation s .

Let $\delta' = \delta_1 | \delta_2 | \delta_3$, and $\delta'' = \delta_4 | \delta_5 | \delta_6$, where

- $\delta_1 = \top?; eat(1, 1).$
- $\delta_2 = \pi k. Prev(2, eat(0, k)) \wedge k > 0?; eat(k, 0).$
- $\delta_3 = \pi k. Prev(2, eat(k, 0)) \wedge k > 0?; eat(0, k).$
- $\delta_4 = \pi k. k \geq 1 \wedge size(2, k)?; eat(1, k - 1).$
- $\delta_5 = \pi k. Prev(2, eat(0, k)) \wedge k > 0?; eat(1, k - 1).$
- $\delta_6 = \pi k. Prev(2, eat(1, k))?; eat(0, k + 1).$

Here, these six strategy specifications are pairwise disjoint. Both δ' and δ'' are standard. The following properties say that δ' (resp. δ'') is a winning strategy for player 1 for any $m \times m$ (resp. $2 \times m$) game:

- $\mathcal{D}_{mm} \models \forall g_{all}. Sat(g_1, \delta') \supset \exists s. S_0 \leq_{g_{all}} s \wedge Win_1(s);$
- $\mathcal{D}_{2m} \models \forall g_{all}. Sat(g_1, \delta'') \supset \exists s. S_0 \leq_{g_{all}} s \wedge Win_1(s).$

Example 4 Cont'd. The following is a winning strategy for the thieves. Each thief behaves as follows: moves right when he is to the left of the treasure and no matter where the guard is, moves up (resp. down) when he is to the due south (resp. north) of the treasure, and lifts the treasure when he gets to its location. In this way, at least two thieves will escape the guard and seize the treasure.

For $i = 1, 2, 3$, let $\delta_i = \delta_i^1 | \delta_i^2 | \delta_i^3 | \delta_i^4$, where

- $\delta_i^1 = \varphi_i^1?; right$, where $\varphi_i^1 = \exists p, q. loc_i(p, q) \wedge p \leq 2.$
- $\delta_i^2 = \varphi_i^2?; up$, where $\varphi_i^2 = \exists q. loc_i(3, q) \wedge q < 1.$
- $\delta_i^3 = \varphi_i^3?; down$, where $\varphi_i^3 = \exists q. loc_i(3, q) \wedge q > 1.$
- $\delta_i^4 = loc_i(3, 1)?; lift.$

Here for each $i \in \{1, 2, 3\}$, $\{\delta_i^j : j = 1, 2, 3, 4\}$ are pairwise disjoint, and each δ_i is standard. The following says that when each thief i adopts strategy δ_i , they are guaranteed to win:

$$\mathcal{D}_{tg} \models \forall g_{all}. \bigwedge_{i=1}^3 Sat(g_i, \delta_i) \supset \exists s. S_0 \leq_{g_{all}} s \wedge win(s).$$

Example 7 Consider the Nim game [13]. The Nim game is played as follows. There are three piles of chips containing k_1, k_2 , and k_3 chips

respectively. Two players take turns moving. Each move consists of selecting one of the three piles and removing one or more chips from it as desired. The winner is the player who removes the last chip.

We now formalize this game in the situation calculus. There is a fluent $rm(k, j, s)$ meaning that there are k chips remaining in the j th pile, $j = 1, 2, 3$. There is an action $take(k, j), k > 0$, meaning taking k chips from the j th piles. Here we also give a situation-independent function $bin(k_1, k_2, k_3)$, which is their addition without carry in base 2 calling it as the nim-sum of them. For example, for $(5, 7, 9)$, $bin(5, 7, 9) = 101 + 111 + 1001 = 1011$.

In [5], Bouton shows that in the Nim game with three piles, the first player has a winning strategy iff for the initial chips (k_1, k_2, k_3) , their addition without carry in base 2 is not zero.

In addition to the second-order axiomatization of Peano arithmetic, we have the following axioms:

$$\begin{aligned} Poss_i(take(k, j), s) &\equiv turn_i(s) \wedge \exists k'. rm(k', j, s) \wedge k' \geq k, \\ turn_i(do(d, s)) &\equiv \neg turn_i(s), \quad i = 1, 2, j = 1, 2, 3; \\ rm(k, j, do(d, s)) &\equiv \exists k', k''. rm(k', j, s) \wedge k = k' - k'' \wedge \varphi_0(k'', j) \\ &\quad \vee (rm(k, j, s) \wedge \neg \exists k'. \varphi_0(k', j)); \end{aligned}$$

$$turn_1(S_0) \wedge \neg turn_2(S_0),$$

where $\varphi_0(k, j) \doteq d_1 = take(k, j) \vee d_2 = take(k, j)$. We use \mathcal{D}_{nim} to denote the BAT of the Nim game. Two abbreviations are

- $\varphi_1(s) \doteq \exists k_1, k_2, k_3. \bigwedge_{j=1}^3 rm(k_j, j, s) \wedge bin(k_1, k_2, k_3) = 0$.
 - $\varphi_2(s) \doteq \exists k_1, k_2, k_3. \bigwedge_{j=1}^3 rm(k_j, j, s) \wedge \neg bin(k_1, k_2, k_3) = 0$.
- Intuitively, they mean that in the situation s , the addition of the remaining chips in each pile without carry in base 2 is 0 or not 0.

Denote $\mathcal{D}_{nim} \cup \varphi_1(S_0)$ as \mathcal{D}_{nim1} , denote $\mathcal{D}_{nim} \cup \varphi_2(S_0)$ as \mathcal{D}_{nim2} . The abbreviation below says that agent i wins in situation s : $Win_i(s) \doteq executable(s) \wedge \neg turn_i(s) \wedge \bigwedge_{j=1}^3 rm(0, j, s)$.

Then we can give a strategy specification δ_i ($i=1,2$):

$\pi k, j. take(k, j); \varphi_1?$. Intuitively, δ_i means that if agent i can ensure φ_1 holds next by performing $take(k, j)$, then he does this action.

The following properties say that δ_1 (resp. δ_2) is a winning strategy for player 1 for game with the initial nim-sum is zero (resp. not zero):

- $\mathcal{D}_{nim1} \models \forall g_{all}. Sat(g_1, \delta_1) \supset \exists s. S_0 \leq_{g_{all}} s \wedge Win_1(s)$;
- $\mathcal{D}_{nim2} \models \forall g_{all}. Sat(g_2, \delta_2) \supset \exists s. S_0 \leq_{g_{all}} s \wedge Win_2(s)$.

For $(5, 7, 9)$, initially, $bin(5, 7, 9) \neq 0$, to make zero, we should take 7 chips away from the 3rd pile, such that the remaining $bin(5, 7, 2) = 0$, and in the next, no matter what player 2 does, the remaining addition will be not zero.

Finally, we formalize the strategy verification and synthesis problems in our framework.

Definition 14 (Strategy verification) Given a BAT \mathcal{D} , a situation-calculus formula $\varphi(g_{all}, s)$, a coalition A , and a collective strategy specification δ_A for A , verify if δ_A is a winning strategy for A wrt φ .

Definition 15 (Strategy synthesis) Given a basic action theory \mathcal{D} , a situation-calculus formula $\varphi(g_{all}, s)$, a coalition A , generate a winning strategy for A wrt φ .

6 A DECIDABLE FRAGMENT

In this section, we give a decidable fragment of our extended situation calculus inspired by the work of De Giacomo *et al.* [8, 9].

In [8], De Giacomo *et al.* define a notion of bounded action theory in the situation calculus, where the theory entails that in all situations, the number of fluent atoms which hold is bounded by a constant, and then prove that verification of an expressive class of first-order μ -calculus temporal properties in such theories is decidable. Roughly, they prove their result by focussing on the active domain of situations, *i.e.*, the set of objects occurring in the extension of some fluent,

which is bounded. Essentially, they abstract situations whose active domains are isomorphic into a single state, and by abstracting also actions, they obtain an abstract finite transition system that satisfies exactly the same formulas of their query language.

We strengthen the notion of bounded action theory with the further restriction that the action theory should entail that in all situations, the number of actions executable by any agent is bounded. By similar ideas as in [8], we prove that verification of SL-like situation calculus formulas in such theories is decidable. The further restriction is due to the need of strategic reasoning.

We assume that there are no functions other than constants and there are no situation-independent predicates, there is a finite set \mathcal{F} of relational fluents, and there is a finite set of action functions. Moreover, we assume that the terms of the object sort are in fact a countably infinite set \mathcal{N} of standard names. We restrict our attention to *standard interpretations* of the object sort, where there is a bijection between the set of objects and \mathcal{N} . As shown in [20], this restriction can be captured by the set of unique name axioms for constants in \mathcal{N} , which we denote by \mathcal{D}_{uno} . We use $\mathcal{D} \models_{uno} \phi$ to denote $\mathcal{D} \cup \mathcal{D}_{uno} \models \phi$. We say that \mathcal{D}_{S_0} is a database, if it consists of axioms, one for each fluent F , of the form $\forall \vec{x}. F(\vec{x}, S_0) \equiv \vec{x} = \vec{c}_1 \vee \dots \vee \vec{x} = \vec{c}_m$, where each \vec{c}_i is a vector of constants from \mathcal{N} .

Definition 16 We say that an action theory \mathcal{D} is strictly bounded by a constant b if \mathcal{D} entails that in any executable situation s , for each fluent F , the number of $F(\vec{x}, s)$ which holds is less than b ; and for each agent i and each action type A , the number of $Poss_i(A(\vec{x}), s)$ which holds is less than b .

We now define the class \mathcal{SL} of SL-like situation calculus formulas. Recall that in Definition 6, we define a translation function $T(\varphi, g_{all}, s)$ which maps an SL formula φ into a situation calculus formula. Firstly, we extend SL to SL^+ so that in the base case, instead of an atom p , we have a situation-suppressed sentence φ , which we call a state sentence, and we let $T(\varphi, g_{all}, s) = \varphi[s]$. Note that a state sentence is a first-order sentence. Now we say ϕ is an SL-like situation calculus formula if it is $T(\varphi, g_{all}, \sigma)$ for some SL^+ formula φ and situation term σ . Note that because of our extension of SL to SL^+ , \mathcal{SL} is more expressive than SL. Recall that if φ is a sentence, then $T(\varphi, g_{all}, s)$ does not depend on g_{all} and it does not have free strategy variables. In this case, we simply write $T(\varphi, g_{all}, s)$ as $\varphi[s]$.

Theorem 3 Verifying if $\mathcal{D} \models_{uno} \phi$, where \mathcal{D} is a strictly bounded action theory with an initial database about S_0 , and ϕ is an SL-like situation calculus sentence, is decidable.

We prove the theorem by showing that the verification problem can be reduced to the model checking problem of SL, which is decidable [29]. Suppose that \mathcal{D} is a strictly bounded action theory with an initial database, and φ is an SL^+ sentence. Then \mathcal{D} has a unique standard model \mathcal{M} whose domain for the object (resp. situation) sort is \mathcal{N} (resp. \mathcal{S}). Thus $\mathcal{D} \models_{uno} \varphi[S_0]$ iff $\mathcal{M} \models \varphi[S_0]$. The following are the main steps of the proof.

1. \mathcal{M} induces a concurrent transition system \mathcal{G} with an infinite number of states, each of which is associated with a first-order structure with domain \mathcal{N} s.t. \mathcal{M} satisfies $\varphi[S_0]$ iff \mathcal{G} satisfies φ .
2. $\hat{\varphi}$ is constructed from φ via converting each state sentence of φ into an equivalent domain-independent one.
3. Based on \mathcal{D} , a finite domain $\hat{\Delta} \subseteq \mathcal{N}$ can be constructed, and \mathcal{G} can be abstracted into a concurrent transition system $\hat{\mathcal{G}}$ with a finite number of states each of which is associated with a first-order structure with domain $\hat{\Delta}$ s.t. \mathcal{G} satisfies φ iff $\hat{\mathcal{G}}$ satisfies $\hat{\varphi}$. Note that $\hat{\mathcal{G}}$ can be directly constructed from \mathcal{D} .

4. By using the domain $\hat{\Delta}$, $\hat{\mathcal{G}}$ can be ground into a CGS, and $\hat{\varphi}$ can be ground into an SL sentence, hence whether $\hat{\mathcal{G}}$ satisfies $\hat{\varphi}$ can be checked via SL model checking.

Firstly, we observe that the evaluation of any $\varphi(\vec{x}, s)$, a uniform FO formula of the situation calculus, does not depend on s , but only on the corresponding interpretation $I_s = \langle \Delta, \cdot^{I_s} \rangle$ of \mathcal{F} . For any fluent F in \mathcal{F} , $F^{I_s} = \{\vec{u} \mid \mathcal{M} \models F(\vec{u}, s)\}$. By successor state axioms, for any joint action d , if $I_{s_1} = I_{s_2}$, then $I_{do(d, s_1)} = I_{do(d, s_2)}$. For any joint action d , by operating on I_s , we evaluate the situation-suppressed successor states axioms to generate another $I_{do(d, s)}$.

After this, considering all situations in \mathcal{S} and joint actions, we can define a CGS-like system CGS $\mathcal{G} = \langle \Delta, Q, q_0, \{\kappa_i\}_{i \in AG}, \tau, \lambda \rangle$. Here the object domain is $\Delta = \mathcal{N}$; $Q = \mathcal{S}$; q_0 is just S_0 ; κ_i is a function mapping a state to a set of actions which can be executed by agent i in the state, here $\kappa_i(s)$ is bounded; τ is a transition function $\tau(s, d) = s'$ iff $s' = do^M(d, s)$, and $d_i \in \kappa_i(s)$; and $\lambda : Q \rightarrow Int_{\Delta}^{\mathcal{F}}$ is the function $\lambda(s) = I_s$. \mathcal{G} retains all the information necessary to evaluate Φ on \mathcal{M} . Next the semantics of SL^+ according to CGSs is defined as in Definition 3 except for the state sentence φ : $\mathcal{G}, \chi, w \models \varphi$ iff $\lambda(w) \models \varphi$.

Lemma 2 *If \mathcal{G} is as above, then for any SL^+ sentence φ , and the corresponding $\mathcal{S}\mathcal{L}$ sentence $\varphi[S_0]$, $\mathcal{M} \models \varphi[S_0]$ iff $\mathcal{G} \models \varphi$.*

Thus, to check whether $\mathcal{D}_{uno} \models \varphi[S_0]$, we can check whether $\mathcal{G} \models \varphi$. However, the \mathcal{G} is infinite. In the following, we should abstract this \mathcal{G} to a finite CGS $\hat{\mathcal{G}}$ under the strict boundedness.

Let $adom(I)$ denote the active domain of I , i.e., the set of all objects occurring in some fluent extension F^I . And denote by $\mathcal{C} \subseteq \mathcal{N}$ the finite set of all constants, appearing in \mathcal{D} . We say that I and J are ad-isomorphism [8], if there exists a bijection $i : adom(I) \cup \mathcal{C} \rightarrow adom(J) \cup \mathcal{C}$, s.t., $i(c) = c$, $c \in \mathcal{C}$, and $\vec{\sigma} \in F^I$ iff $i(\vec{\sigma}) \in F^J$. The notion of A-local-isomorphism \approx is like local-isomorphism in Definition 4, except that if two states are A-local-isomorphism, then their corresponding interpretations should be ad-isomorphism, and in any state, the bisimulation relations g_i on actions are given for each agent i , for instance, for $s_1 \approx s_2$, for any $a_1 \in \kappa_i(s_1)$, there is a $a_2 \in \kappa_i(s_2)$, such that $(a_1, a_2) \in g_i(s_1, s_2)$.

A first-order sentence φ is said to be *domain-independent* [1] if for each interpretation $I = \langle \Delta, \cdot^I \rangle$, $I \models \varphi$ iff $\hat{I} \models \varphi$, where $\hat{I} = \langle adom(I), \cdot^I \rangle$. And an SL^+ sentence is said *domain-independent* if so are all of its first-order components. Then we can prove that A-local-isomorphism CGSs preserve domain-independent SL^+ sentence as in [28].

Lemma 3 *Given two A-local-isomorphism CGSs $\mathcal{G}, \mathcal{G}'$, and a domain-independent SL^+ sentence φ , $\mathcal{G} \models \varphi$ iff $\mathcal{G}' \models \varphi$.*

In [25], Libkin shows, through syntactic manipulations, any FO sentence φ can be effectively turned into a logically equivalent domain-independent formula φ' . By adapting his approach to each FO component of an $\mathcal{S}\mathcal{L}$ sentence, we can generalise this result.

Lemma 4 *For any $\mathcal{S}\mathcal{L}$ sentence Φ , there exists a logically equivalent domain-independent $\mathcal{S}\mathcal{L}$ sentence Φ' .*

Next, we construct the finite CGS abstraction $\hat{\mathcal{G}}$ of the induced \mathcal{G} as the tuple $\langle \hat{\Delta}, \hat{Q}, \hat{q}_0, \{\hat{\kappa}_i\}_{i \in AG}, \hat{\tau}, \hat{\lambda} \rangle$. Here the finite object domain is $\hat{\Delta} \subseteq \mathcal{N}$, satisfying $\mathcal{C} \subseteq \hat{\Delta}$ and $|\hat{\Delta}| > |\mathcal{C}| + \beta + n\beta_1 + \eta$. Here $\beta = \sum_{F \in \mathcal{F}} b \cdot a_F$, with a_F the arity of situation suppressed fluent F , to ensure A-local-isomorphism, each state of \mathcal{G} must have a matching ad-isomorphism state in $\hat{\mathcal{G}}$; $\beta_1 = \sum_{A \in \mathcal{A}} a_A$, a_A is the arity of action function A ; and η is the maximal number of distinct variables occurring in the righthand side of action precondition and successor state axioms to abstract all the possible combinations of the objects used in some action, and mentioned in the axioms. Let $\hat{q}_0 = \langle \star, I_0 \rangle$,

here \star is a special symbol, and $I_0 = \langle \hat{\Delta}, \cdot^{I_0} \rangle$ is a FO interpretation over $\hat{\Delta}$. For any $\vec{\sigma} \in \hat{\Delta}^{|\vec{\sigma}|}$, and $F \in \mathcal{F}$, $\vec{\sigma} \in F^{I_0}$ iff $\mathcal{M}_0 \models F(\vec{\sigma})$, then in I_0 , by boundness, for each action type $A(\vec{x})$ and agent i , there exist bounded number instances $\vec{\sigma} \in \hat{\Delta}^{|\vec{x}|}$, s.t., $I_0 \models \phi_{A,i}(\vec{\sigma})$, here $Poss_i(A(\vec{x}), s) \equiv \phi_{A,i}(\vec{x}, s)$. So let $A(\vec{\sigma}) \in \kappa_i(\langle \star, I_0 \rangle)$. And there exist bounded executable joint actions for I_0 , after executing each such d , we get another interpretation I , then we give another state $\langle d, I \rangle$, and have $\hat{\tau}(\langle \star, I_0 \rangle, d) = \langle d, I \rangle$. By applying this procedure recursively, due to the strictly boundness and $\hat{\Delta}$ finite, we can generate all such finite $\langle d, I \rangle$, give κ_i and the transition function $\hat{\tau}$. Finally, define $\lambda(\langle d, I \rangle) = I$. Then the property below holds.

Lemma 5 *\mathcal{G} and $\hat{\mathcal{G}}$ are A-local-isomorphism.*

In fact, when the initial situation is incomplete, the verification of $\mathcal{S}\mathcal{L}$ is also decidable. More details are referred to [8].

7 CONCLUSION

In this paper, by a simple extension of the situation calculus with a strategy sort, we have developed a general framework for strategy representation and reasoning for complete information games. We have shown that Strategy Logic can be embedded into our framework. The framework can be used to compactly represent both concurrent and turn-based possibly infinite game structures, specify the structure of strategies, reason about strategies explicitly, and reason about strategic abilities of coalitions under strategy specifications. In a companion paper [46], we have extended this framework to deal with strategic reasoning for incomplete information games.

Action formalisms like the situation calculus and modal logics have been two main families of logics in AI. Van Benthem [40] proposes the idea that the situation calculus and modal logics meet and merge. Recent works in this direction, e.g., [27, 12], connect dynamic epistemic logic to the situation calculus. Based on earlier works, this paper presents another work in this line connecting strategic logics to the situation calculus. Compared with modal logics for strategic reasoning, the most distinguished feature of our work is expressiveness and compactness in representing game structures and strategies. Many of existing works based on multi-agent situation calculus, e.g., [18, 4], are mainly concerned with epistemic reasoning. Others [37, 14, 17, 10] deal with strategic reasoning, but they either focus on different issues such as the computation of Nash equilibria or the coordination problem or do not support ATL-like or SL-like reasoning about strategies.

In this paper, we focus on the representation side of our strategy formalism. From the reasoning side, our framework is highly undecidable since it is based on an extension of the situation calculus with a second-order strategy sort. Nonetheless, it can serve as the theoretic foundation for strategy verification and synthesis based on first-order theorem proving and fix-point computation techniques. In the software engineering community, there has been considerable progress on program verification via automatic discovery of loop invariants [3, 15, 38]. In recent years, progress has also been made on automated reasoning in the situation calculus and automatic verification of Golog programs [23, 24]. In the planning community, various techniques such as generating and testing [21], and object abstraction [39], have been developed for planning with loops. Inspired by these works, we would like to explore automatic strategy verification and synthesis.

ACKNOWLEDGEMENTS

We thank the anonymous reviewers for helpful comments. This work received support from the Natural Science Foundation of China under Grant No. 61572535.

REFERENCES

- [1] S. Abiteboul, R. Hull, and V. Vianu, *Foundations of Databases*, Addison-Wesley, 1995.
- [2] R. Alur, T.A. Henzinger, and O. Kupferman, 'Alternating-time temporal logic', *J. ACM*, **49**(5), 672–713, (2002).
- [3] T. Ball, R. Majumdar, T.T. Millstein, and S.K. Rajamani, 'Automatic predicate abstraction of C programs', in *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, pp. 203–213, (2001).
- [4] V. Belle and G. Lakemeyer, 'Reasoning about imperfect information games in the epistemic situation calculus', in *Proc. of AAI'2010*.
- [5] C.L. Bouton, 'Nim, a game with a complete mathematical theory', *Annals of Mathematics*, **3**(1), 35–39, (1901).
- [6] K. Chatterjee, T.A. Henzinger, and N. Piterman, 'Strategy logic', *Inf. Comput.*, **208**(6), 677–693, (2010).
- [7] G. De Giacomo, Y. Lespérance, and H.J. Levesque, 'Congolog, a concurrent programming language based on the situation calculus', *Artif. Intell.*, **121**(1-2), 109–169, (2000).
- [8] G. De Giacomo, Y. Lespérance, and F. Patrizi, 'Bounded situation calculus action theories and decidable verification', in *Proc. of KR'2012*.
- [9] G. De Giacomo, Y. Lespérance, and F. Patrizi, 'Bounded situation calculus action theories', *Artif. Intell.*, **237**, 172–203, (2016).
- [10] G. De Giacomo, Y. Lespérance, and A.R. Pearce, 'Situation calculus based programs for representing and reasoning about game structures', in *Proc. of KR'2010*.
- [11] G. De Giacomo, Y. Lespérance, and A.R. Pearce, 'Synchronous games in the situation calculus', in *Proc. of AAMAS'2015*.
- [12] L. Fang and Y. Liu, 'Multiagent knowledge and belief change in the situation calculus', in *Proc. of AAI'2013*.
- [13] T.S. Ferguson, *Game Theory*, UCLA, 2008.
- [14] A. Finzi and T. Lukasiewicz, 'Game-theoretic agent programming in golog', in *Proc. of ECAI'2004*.
- [15] C. Flanagan and S. Qadeer, 'Predicate abstraction for software verification', in *Conference Record of POPL 2002: The 29th SIGPLAN-SIGACT Symposium on Principles of Programming Languages*.
- [16] M.R. Genesereth, N. Love, and B. Pell, 'General game playing: Overview of the AAI competition', *AI Magazine*, **26**(2), 62–72, (2005).
- [17] H Ghaderi, H.J. Levesque, and Y. Lespérance, 'A logical theory of coordination and joint ability', in *Proc. of AAI'2007*.
- [18] R.F. Kelly and A.R. Pearce, 'Asynchronous knowledge with hidden actions in the situation calculus', *Artif. Intell.*, **221**, 1–35, (2015).
- [19] Y. Lespérance, H.J. Levesque, F. Lin, and R.B. Scherl, 'Ability and knowing how in the situation calculus', *Studia Logica*, **66**(1), 165–186, (2000).
- [20] H.J. Levesque, 'A completeness result for reasoning with incomplete first-order knowledge bases', in *Proc. of KR'1998*.
- [21] H.J. Levesque, 'Planning with loops', in *Proc. of IJCAI'2005*.
- [22] H.J. Levesque, R. Reiter, Y. Lespérance, F. Lin, and R.B. Scherl, 'GOLOG: A logic programming language for dynamic domains', *J. Log. Program.*, **31**(1-3), 59–83, (1997).
- [23] N. Li, Y. Fan, and Y. Liu, 'Reasoning about state constraints in the situation calculus', in *Proc. of IJCAI'2013*.
- [24] N. Li and Y. Liu, 'Automatic verification of partial correctness of golog programs', in *Proc. of IJCAI'2015*.
- [25] L. Libkin, 'Embedded finite models and constraint databases', in *In Finite Model Theory and Its Applications*, 257–338, Springer, (2007).
- [26] F. Lin and H.J. Levesque, 'What robots can do: Robot programs and effective achievability', *Artif. Intell.*, **101**(1-2), 201–226, (1998).
- [27] Y. Liu and H.J. Levesque, 'Incorporating action models into the situation calculus', in *Johan van Benthem on Logic and Information Dynamics, Outstanding Contributions to Logic*, eds., Alexandru Baltag and Sonja Smets, volume 5, chapter 21, 569–590, Springer, (2014).
- [28] F. Mogavero, *Logics in Computer Science - A Study on Extensions of Temporal and Strategic Logics*, volume 3 of *Atlantis Studies in Computing*, Atlantis Press, 2013.
- [29] F. Mogavero, A. Murano, G. Perelli, and M.Y. Vardi, 'Reasoning about strategies: On the model-checking problem', *ACM Trans. Comput. Log.*, **15**(4), 34:1–34:47, (2014).
- [30] F. Mogavero, A. Murano, and M.Y. Vardi, 'Reasoning about strategies', in *Proc. of FSTTCS'2010*.
- [31] R.J. Nowakowski, *Games of No Chance*, Cambridge University Press, 1998.
- [32] A. Pnueli, 'The temporal logic of programs', in *Proc. of FOCS'1977*.
- [33] R. Ramanujam and S.E. Simon, 'Dynamic logic on games with structured strategies', in *Proc. of KR'2008*.
- [34] R. Reiter, *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*, The MIT Press, 2001.
- [35] S. Schiffel and M. Thielscher, 'Reasoning about general games described in GDL-II', in *Proc. of AAI'2011*.
- [36] S. Schiffel and M. Thielscher, 'Representing and reasoning about the rules of general games with imperfect information', *J. Artif. Intell. Res.*, **49**, 171–206, (2014).
- [37] O. Schulte and J.P. Delgrande, 'Representing von neumann-morgenstern games in the situation calculus', *Ann. Math. Artif. Intell.*, **42**(1-3), 73–101, (2004).
- [38] S. Srivastava and S. Gulwani, 'Program verification using templates over predicate abstraction', in *Proc. of PLDI'2009*.
- [39] S. Srivastava, N. Immerman, and S. Zilberstein, 'A new representation and associated algorithms for generalized planning', *Artif. Intell.*, **175**(2), 615–647, (2011).
- [40] J. van Benthem, 'McCarthy variations in a modal key', *Artif. Intell.*, **175**(1), 428–439, (2011).
- [41] J. van Benthem, 'Reasoning about strategies', in *Computation, Logic, Games, and Quantum Foundations. The Many Facets of Samson Abramsky - Essays Dedicated to Samson Abramsky on the Occasion of His 60th Birthday*, pp. 336–347, (2013).
- [42] J. van Benthem, *Logic in Games*, The MIT Press, 2014.
- [43] W. van der Hoek, W. Jamroga, and M. Wooldridge, 'A logic for strategic reasoning', in *Proc. of AAMAS'2005*.
- [44] J. van Eijck, 'PDL as a multi-agent strategy logic', in *Proc. of TARK'2013*.
- [45] D Walther, W van der Hoek, and M. Wooldridge, 'Alternating-time temporal logic with explicit strategies', in *Proc. of TARK'2007*.
- [46] L. Xiong and Y. Liu, 'Strategy representation and reasoning for incomplete information concurrent games in the situation calculus', in *Proc. of IJCAI'2016*.
- [47] D. Zhang and M. Thielscher, 'A logic for reasoning about game strategies', in *Proc. of AAI'2015*.
- [48] D. Zhang and M. Thielscher, 'Representing and reasoning about game strategies', *J. Philosophical Logic*, **44**(2), 203–236, (2015).

Exploiting MUS Structure to Measure Inconsistency of Knowledge Bases

Said Jabbour¹ and Lakhdar Sais¹

Abstract. Measuring inconsistency is recognized as an important research issue for quantifying and handling inconsistencies in knowledge bases. Several logic-based inconsistency measures have been proposed. Minimal unsatisfiable and maximal satisfiable subsets are at the heart of the syntactic measures, while semantic inconsistency measures are often based on some paraconsistent semantics. In order to design interesting measures faithful to human rationality, many properties have been introduced to reach this goal. In this paper, we propose a new property called sub-additivity allowing to push further the ability to reorder knowledge bases according to their inconsistency degree. After pointing out the limitations of several measures to satisfy the sub-additivity property, we present a new measure based on a fine exploitation of the internal structure of the knowledge base, namely the structure of its associated minimal unsatisfiable subsets. Then, we show how its computation can be formulated as a nonlinear optimization problem. Finally, we prove that the new measure satisfies all the required properties while highlighting its interesting features.

1 Introduction

Reasoning about inconsistent knowledge bases (KBs) is one of the fundamental topics attracting growing interest from the AI community. It aims to quantify the amount of inconsistency, useful for guiding inconsistency resolving [14, 7, 5]. Its interest has been highlighted in several domains including software specifications [3], belief merging [33], news reports [18], integrity constraints [11], and multi-agents systems [20, 21].

These last years, several inconsistency measures have been proposed. Some of them [18, 12, 19, 26], focus on minimal unsatisfiable subsets (MUSes), pointed out as the elementary unit circumscribing inconsistency. Dually, maximal satisfiable subsets are also exploited in this context. Indeed, the maximal satisfiable subsets can be derived from the set of minimal unsatisfiable subsets using minimal hitting sets of the set of MUSes. They represent the different possible ways of restoring consistency of an inconsistent knowledge base. Among these measures, $I_{MI}(K)$ stated in [19] is defined as the number of MUSes of a knowledge base K , while considering the knowledge base with a high number of MUSes as the most inconsistent one. Unfortunately, such approach considers the contribution of each MUS to the whole inconsistency as independent from its possible interaction with the remaining MUSes. Recent work focuses on MUSes interaction through their intersections and consider that the inconsistency should take into consideration such overlaps. This observation leads

to a new active research issue with several new inconsistency measures [21, 23]. These measures have been motivated from the cooperative multi-agents perspective [23]. A classical problem of inconsistency assessment related to MUSes structure, can be represented in the following knowledge bases:

$$\begin{aligned} K_1 &= \{p, \wedge \neg p, q, \neg q\}; \\ K_2 &= \{p, \wedge \neg p \wedge q, \neg q\} \end{aligned}$$

K_1 contains two disjoint MUSes while K_2 holds interconnected MUSes. The recurrent questions are: if K_2 has the same inconsistency as K_1 ? Otherwise, How the inconsistency degree of K_2 is far from the one of K_1 ?

Works related to MUSes dependencies have been reported in many studies of inconsistency handling. In [36], hitting set based inconsistency measure is introduced. In [4], the authors highlighted dependence relation between MUSes by showing that inconsistency resolving is MUSes structure dependent. To facilitate the description of justificatory structure, in [2] the authors introduce a graph-based framework for capturing and analyzing relationships between justifications in OWL ontologies.

In parallel, and in order to consolidate or to justify the proposed measures, several properties have been proposed to judge the rationality of the proposed measures. Such properties were analyzed and discussed by many authors (see for example [6]). Among the properties that attracted much more attention, we can cite the dominance property [19] requiring a non decreasing inconsistency value when a consistent formula of a knowledge base is replaced with one of its logical consequences. A weaker form has been characterized using prime implicates, a canonical logic based representation of knowledge bases [22].

Following this research trend, in this paper, we propose a new approach to measure inconsistency. First, a new property called sub-additivity is proposed. Based on a finer analysis of the connections between MUSes, it allows us to push further the possibility to reorder knowledge bases with respect to their inconsistency degree. Secondly, we exploit the relationships between MUSes to design a new inconsistency metric satisfying the new property and some of the well-known rationality properties.

The contributions of this paper can be summarized as follows:

1. We introduce a new property called sub-additivity, and we discuss its relevance to the problem of measuring inconsistency.
2. We point out the limitations of the previous inconsistency measures and we propose a new one exploiting the inner-structure of the knowledge base, namely the structure of the graph-based representation of the set of MUSes.
3. We provide different ways for computing or approximating the inconsistency value, while providing results on some knowledge

¹ CRIL, Univ. Artois & CNRS, F-62300 Lens, France, email: {jabbour, sais@cril.fr}

bases.

The rest of this paper is organized as follows. After some preliminary definitions and notations, we briefly recall different proposals for measuring inconsistency relevant to the present work. In Section 3, after motivating the limitation of ind-additivity, we introduce our new property called sub-additivity. In Section 4, we introduce a new measure exploiting dependencies between MUSes. Section 5, provides a discussion of the properties of the introduced measure. In Section 6, we formulate the computation of our new inconsistency measure as an optimization problem and provide approximation. Finally, we give related work in Section 7, before concluding and giving some directions for future work in Section 8.

2 Preliminaries

A propositional language \mathcal{L} is built over a finite set of propositional symbols \mathcal{P} using classical logical connectives. We use letters (possibly indexed) p, q, r, \dots to represent atoms in \mathcal{P} , and Greek letters α, β, \dots to denote formulae. The special letter \perp denotes falsity (contradiction), while \top denotes validity.

A literal is an atom p or its negation $\neg p$. A clause C is a disjunction of literals $(p_1 \vee \dots \vee p_n)$. A formula α in *conjunctive normal form* (CNF) is a conjunction of clauses. Let $Var(\alpha)$ denotes the set of variables occurring in α . A knowledge base (in short KB) K is a finite set of propositional formulae. For a set S , we denote by $|S|$ its cardinality. We say that K is *inconsistent* if $K \vdash \perp$, where \vdash is the classical consequence relation.

Minimal unsatisfiable subsets are often used to analyze inconsistency. Formally, this concept is defined as follows:

Definition 1 (MUS). *Let K be a knowledge base and $M \subseteq K$. M is a Minimal Unsatisfiable (or Inconsistent) Subset (MUS) of K iff $M \vdash \perp$ and $\forall M' \subsetneq M, M' \not\vdash \perp$. The set of all minimal inconsistent subsets of K is denoted $MUSes(K)$.*

A formula $\alpha \in K$ is called a *free formula* iff there is no $M \in MUSes(K)$ such that $\alpha \in M$. The class of free formulae of K is written $free(K) = K \setminus \bigcup MUSes(K)$, and its complement is named *unfree formulae* set: $unfree(K) = K \setminus free(K)$.

At the same time, we can define the *Maximal Satisfiable (Consistent) Subset* (MSS), and Hitting set as follows:

Definition 2 (MSS). *Let K be a knowledge base and M a subset of K . M is a maximal satisfiable (consistent) subset (MSS) of K iff $M \not\vdash \perp$ and $\forall \alpha \in K \setminus M, M \cup \{\alpha\} \vdash \perp$.*

Definition 3. *H is a hitting set of a set of sets Ω if $\forall S \in \Omega, H \cap S \neq \emptyset$. A hitting set H is irreducible (or minimal) if there is no other hitting set H' s.t $H' \subset H$. A minimum hitting set is an irreducible hitting set with the smallest cardinality.*

Many researchers have focused their effort to elaborate rational properties [19, 21, 6] that an ideal inconsistency measure I must satisfy. Among such properties we can cite the following ones:

- *Consistency:* $I(K) = 0$ iff K is consistent.
- *MinInc:* $I(M) = 1$ if $M \in MUSes(K)$.
- *Independence:* $I(K \cup \{\alpha\}) = I(K)$ if $\alpha \in free(K \cup \{\alpha\})$.
- *Monotonicity:* if $K \subseteq K'$, then $I(K) \leq I(K')$.

- *Additivity:* $I(K_1 \cup \dots \cup K_n) = \sum_{i=1}^n I(K_i)$ if $MUSes(K_1 \cup \dots \cup K_n) = MUSes(K_1) \uplus \dots \uplus MUSes(K_n)$, where \uplus is a disjoint union over a family of sets.
- *Ind-Additivity:* $I(K_1 \cup \dots \cup K_n) = \sum_{i=1}^n I(K_i)$ if $MUSes(K_1 \cup \dots \cup K_n) = MUSes(K_1) \uplus \dots \uplus MUSes(K_n)$, and $unfree(K_i) \cap unfree(K_j) = \emptyset$ for $1 \leq i < j \leq n$.

The additivity property has attracted much more attention recently, allowing the introduction of additional conditions under which the inconsistency of the whole base can be obtained by summing up the inconsistency of its sub-bases i.e. by decomposing the whole base into sub-bases satisfying some requirements. In [19], the condition simply states that the MUSes of the knowledge base form a partition of the considered base (additivity). Such condition has been revisited in [21] by requiring a second condition over the independence of the formulae of the considered knowledge bases leading to the so called Ind-additivity. This last property allows to define the lower bound for standard inconsistency measures. Such lower bound has been characterized using the notion of closed set packing of MUSes (see Definition 4).

Definition 4 (Closed Set Packing of MUSes). *Let $S = \{M_1, \dots, M_n\}$ a subset of MUSes of K . S is a closed set packing of MUSes of K iff:*

- $\forall M_i, M_j \in S, M_i \cap M_j = \emptyset$
- $MUSes(M_1 \cup \dots \cup M_n) = S$

Closed Set Packing of MUSes of K [23] is defined as a set packing of MUSes (pairwise disjoint set of MUSes) that is closed by union. The closure means that the only MUSes that can be built using the formulae involved in the union of the elements of S is S itself.

Maximum cardinality of closed set packings of MUSes of K namely $\mu(K)$ is proved to be a lower bound of measures satisfying MinInc, Monotony, and Ind-additivity i.e., $\mu(K) \leq I(K)$ for all K . Such lower bound has been used in [21] as a new measure called I_{CC} : $I_{CC}(K) = \mu(K)$.

3 MUS Structure: Motivation and Properties

Rational properties like ind-additivity aims to simplify the computation of the inconsistency value when some conditions on the structure of MUSes are gathered. However, it lacks properties to better discriminate between inconsistent knowledge bases that do not satisfy the requirements.

To motivate our study, let us consider all the possible interactions of three MUSes as depicted in Figure 1. The inconsistency values of each knowledge base K_i ($1 \leq i \leq 6$) according to I_{MI} and I_{CC} are: $I_{MI}(K_i) = 3$ for all $i \in \{1, \dots, 6\}$, $I_{CC}(K_i) = 1$ for all $i \in \{1, 5, 6\}$, $I_{CC}(K_i) = 2$ for all $i \in \{2, 3\}$, and $I_{CC}(K_4) = 3$.

Clearly, I_{MI} and I_{CC} fail to discriminate between these different knowledge bases. Indeed, I_{MI} considers all the knowledge bases $\{K_1, \dots, K_6\}$ as equally inconsistent, while for I_{CC} measure, the two knowledge bases K_1 and K_5 admit the same inconsistency degree. This clearly highlights the inability of these measures to take into account dependencies between MUSes. Intuitively, formulas with disjoint MUSes are more inconsistent than those with connected MUSes. Indeed, if the set of MUSes are disjoint, then the minimum number of formulas that we need to remove to recover consistency coincides with I_{MI} . Consequently, a fine-grained measure should try

to capture the degree of connectivity between MUSes. From the different configuration depicted in Figure 1, K_1 should be less inconsistent than K_3 for example. An observation on the interactions between MUSes in K_1 and K_5 , suggests that K_1 is less inconsistent than K_5 , as it contains two disjoint MUSes, contrary to K_5 .

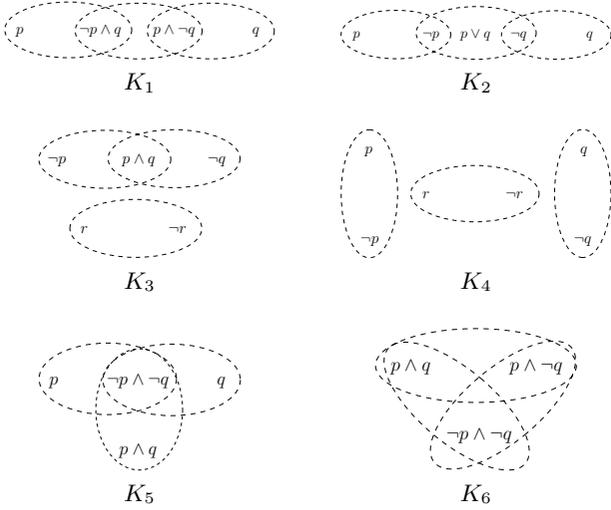


Figure 1. Interactions Between Three MUSes

Based on this observation, we propose a new property called sub-additivity in order to gain insights about the importance of the underlying structure of the MUSes.

Definition 5 (Sub-Additivity). An inconsistency measure I is called sub-additive if for any set of knowledge bases $\{K_1, \dots, K_n\}$ s.t. $MUSes(K_1 \cup \dots \cup K_n) = MUSes(K_1) \uplus \dots \uplus MUSes(K_n)$ and $unfree(K_1 \cup \dots \cup K_n) \neq unfree(K_1) \uplus \dots \uplus unfree(K_n)$, then $I(K_1 \cup \dots \cup K_n) < \sum_{i=1}^n I(K_i)$.

Let us note that the condition $unfree(K_1 \cup \dots \cup K_n) \neq unfree(K_1) \uplus \dots \uplus unfree(K_n)$ in Definition 5, is equivalent to the following condition: there exists i and j ($1 \leq i < j \leq n$) such that $unfree(K_i) \cap unfree(K_j) \neq \emptyset$.

The rationale behind Definition 5 is that the knowledge base with intersecting MUSes is less inconsistent than those with disjoint MUSes.

Proposition 1. Neither I_{MI} nor I_{CC} satisfies the sub-additivity.

Proof. 1) I_{MI} : let us consider the simple knowledge base $K = \{p, \neg p \wedge q, \neg q\}$. By considering $K_1 = \{p, \neg p \wedge q\}$ and $K_2 = \{\neg p \wedge q, \neg q\}$. We have $MUSes(K) = MUSes(K_1) \uplus MUSes(K_2)$ and $unfree(K_1) \cap unfree(K_2) \neq \emptyset$. However $I_{MI}(K) = 2 \not\leq I_{MI}(K_1) + I_{MI}(K_2) = 2$

2) I_{CC} : Let $K = \{p, \neg p, p \vee q, \neg q, q\}$ be a knowledge base. By considering $K_1 = \{p, \neg p, p \vee q, \neg q\}$ and $K_2 = \{\neg q, q\}$. We have $MUSes(K) = MUSes(K_1) \uplus MUSes(K_2)$ and $unfree(K_1) \cap unfree(K_2) = \{\neg q\}$ but $I_{CC}(K) = 2 \not\leq I_{CC}(K_1) + I_{CC}(K_2) = 2$. \square

The inconsistency measures I_{MI} and I_{CC} are not the only ones violating the sub-additivity property. In fact, let us remark that hitting sets based inconsistency measure I_{hs} [36] does not satisfy sub-additivity. Indeed, let us consider $K = \{p, \neg p, p \vee q, \neg q, q\}$ where

$K_1 = \{p, \neg p, p \vee q, \neg q\}$ and $K_2 = \{\neg q, q\}$. We have $I_{hs}(K) \not\leq I_{hs}(K_1) + I_{hs}(K_2)$ where $I_{hs}(K)$ is roughly equivalent to the cardinality of the minimum hitting set of the MUSes hypergraph. Furthermore, let us consider the MSS based inconsistency measure $I_M(K)$ defined as the maximum cardinality of the MSSs of K (see Definition 6). I_M does not satisfy sub-additivity.

Definition 6 ([14]). Let K be a knowledge base. I_M is defined as:

$$I_M(K) = |MSSs(K)| + |SelfC(K)| - 1$$

$SelfC(K)$ is the set of self contradictory formulas i.e., $\alpha \in K$ such that $\{\alpha\} \vdash \perp$.

Indeed, by considering the knowledge base $K = \{p, q, \neg p \wedge \neg q, r_1, \dots, r_n, \neg r_1 \vee \dots \vee \neg r_n, s_1, \dots, s_n, \neg s_1 \vee \dots \vee \neg s_n\}$ where $K_1 = \{p, \neg p \wedge \neg q, r_1, \dots, r_n, \neg r_1 \vee \dots \vee \neg r_n\}$ and $K_2 = \{q, \neg p \wedge \neg q, s_1, \dots, s_n, \neg s_1 \vee \dots \vee \neg s_n\}$, we have $I_M(K) = 2 \times n \times n - 1$, while $I_M(K_1) = I_M(K_2) = 4 \times n - 1$. Consequently, $I_M(K) \not\leq I_M(K_1) + I_M(K_2)$.

Intuitively, an inconsistency measure satisfying the sub-additivity property can be defined as the one counting the number of unfree formulas.

Proposition 2. $I_{unfree}(K) = |unfree(K)|$ satisfies the sub-additivity property.

Proof. Let $\{K_1, K_2, \dots, K_n\}$ be a set of knowledge bases such that $MUSes(K_1 \cup \dots \cup K_n) = MUSes(K_1) \uplus \dots \uplus MUSes(K_n)$ and there exists $1 \leq i \neq j \leq n$ such that $unfree(K_i) \cap unfree(K_j) \neq \emptyset$. Consequently, $I_{unfree}(K_1 \cup \dots \cup K_n) < \sum_{i=1}^n I_{unfree}(K_i)$. Indeed, as $unfree(K_i) \cap unfree(K_j) \neq \emptyset$, then by summing $I_{unfree}(K_i)$ and $I_{unfree}(K_j)$, the unfree formulae shared by K_i and K_j are counted twice. \square

In addition, I_{unfree} satisfies the ind-additivity, consistency, and free formula independence. However, the MinInc property is violated. This is clearly a problem, since the violation of MinInc makes I_{unfree} not enough informative about internal conflicts. Indeed, a knowledge base with several MUSes can be considered less inconsistent than a knowledge base with a single MUS.

Let us show how a measure satisfying a sub-additivity property behaves on some MUSes whose associated hypergraph is of particular structure. Let us note K^n a knowledge base with n MUSes $\{M_1, \dots, M_n\}$. We exhibit two categories of MUSes structures. The first category corresponds to MUSes hypergraph forming a star i.e. there exists a subset $S \subset K^n$ such that $\forall i (1 \leq i < j \leq n)$, $M_i \cap M_j = S$. If the MUSes hypergraph associated to K^n is a star (Figure 3.a), then the sub-additivity can be simplified using inequality (1). Indeed, if we decompose K^n to K' and K'' such that $MUSes(K^n) = MUSes(K') \uplus MUSes(K'')$, then the two hypergraphs of MUSes of K' and K'' are also stars. Consequently, there exists m such that $K' = K^m$ and $K'' = K^{n-m}$. As the MUSes hypergraphs of K' and K'' are disjoint stars, then the inconsistency degree of K^n must be less than the cumulated inconsistency degree of K' and K'' .

Consequently, for knowledge bases whose MUSes hypergraph forms a star, and for any measure I considering only the number of MUSes or their interactions, satisfying sub-additivity, we derive the following inequality:

$$I(K^n) < I(K^m) + I(K^{n-m}) \quad (1)$$

The measure I satisfying the sub-additivity property obtained using inequality (1), defines a well known (strictly) sub-additive sequence of nonnegative terms $(I(K^n))$. It is known that such kind of sequences are bounded below and converges to $\frac{I(K^n)}{n} = \inf\{\frac{I(K^n)}{n}, n \in \mathbb{N}\}$ [9].

Let us remark that I_{MI} has an additive behavior on a knowledge base K^n whose MUSes hypergraph forms a star. From the above discussion, measures of the form $I(K^n) = an + b$ where a and b are strictly positives integers, satisfy the sub-additivity and Monotony properties when the MUSes of K^n form a star. To satisfy *MinInc* property, the positive integers a and b must satisfy the following constraints: $a + b = 1$, $a > 0$ and $b > 0$.

In the second category, we consider knowledge bases K^n whose MUSes $\{M_1, \dots, M_n\}$ form a chain (Figure 3.c) i.e. $\forall i(1 \leq i < n), M_i \cap M_{i+1} \neq \emptyset$. Here again, an inconsistency measure satisfying sub-additivity property verifies inequality (1).

Similarly, measures of the form $I(K^n) = an + b$ where $a + b = 1$, $a > 0$ and $b > 0$ satisfies sub-additivity, monotony and MinInc properties. Note that, if ind-additivity is required, the value of a must be greater than $\frac{1}{2}$. Indeed, there are $\lfloor \frac{n+1}{2} \rfloor$ MUSes forming a closed set packing of MUS. Consequently, we must have $I(K_n) \geq \lfloor \frac{n+1}{2} \rfloor$.

Then, $a \geq \frac{\lfloor \frac{n+1}{2} \rfloor - 1}{n-1}, \forall n > 2$.

We have characterized sub-additive measures for knowledge bases with MUSes hypergraph of particular structure. In the next section we provide a measure that satisfy several required properties.

4 On Sub-Additive Inconsistency Measure

In this section, we present a new measure that satisfies several standard properties, namely MinInc, Monotony, Ind-additivity, and Sub-additivity. Up to now, we argued that MUSes, MSSes, or Hitting sets based measures fail to satisfy sub-additivity property.

Let us recall that the principle behind sub-additivity is that a knowledge base with n disjoint MUSes is more inconsistent than a knowledge base with n non disjoint MUSes. Consequently, to satisfy sub-additivity property, an inconsistency measure must consider the interactions between MUSes. The main idea behind our proposed sub-additive based inconsistency measure, is to first quantify each MUS of a knowledge base with the same inconsistency value. At first, the contribution of each MUS to the whole inconsistency is equal to one. Each time two MUSes share a formula, one of them must decrease its inconsistency accordingly. In other words, there contributions to inconsistency must be different.

Before presenting our measure, we define the graph representation associated to the MUSes of a knowledge base K .

Definition 7 (MUSes Graph). Let K be a knowledge base. We define the graph representation $G_{mus}^K(V, E)$ of $MUSes(K)$ as the graph where

- $V = MUSes(K)$ and
- $(M, M') \in E$ iff $M \cap M' \neq \emptyset$

Let K be a knowledge base and $S \subseteq MUSes(K)$, we say that S is a *MUS Cover (MC)* of K iff S is a vertex cover of $G_{mus}^K(V, E)$ i.e., $\forall (v, w) \in E$, we have $v \in S$ or $w \in S$.

Example 1. Let us consider a MUSes hypergraph of K and its associated MUS graph depicted in Figure 2. $MC_1 = \{M_1, M_2, M_4\}$ and $MC_2 = \{M_2, M_3, M_5\}$ are two MUS cover of K .

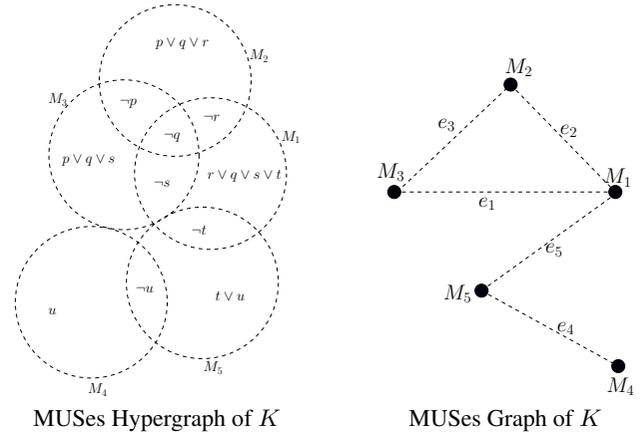


Figure 2. From MUSes Hypergraph to Graph of MUSes

It is widely known that a set of vertices is a vertex cover if and only if its complement is an independent set. So, if S is a MUS cover of K , then $MUSes(K) \setminus S$ is a pairwise disjoint set of MUSes i.e. an independent set of the graph of MUSes.

When two MUSes share formulas, their inconsistency must be set to different values in order to satisfy the sub-additivity property. To this end, let us first define an edge partition of the MUSes graph of K .

Definition 8. Let K be a knowledge base. We define an edge partition of G_{mus}^K as:

$$P_e = \bigsqcup_{M \in V} P_e(M) \text{ s.t. } P_e(M) \subseteq \{(M, M') \in E\}$$

From the definition of a partition of the edges of G_{mus}^K , each edge between two MUSes M and M' belongs either to $P_e(M)$ or to $P_e(M')$. Additionally, an edge partition P_e may contain empty sets and its size $|P_e| = |MUSes(K)|$. Such partition allows us to quantify the contribution of each MUS M as a function of the number of edges in $P_e(M)$.

Proposition 3. Let K be a knowledge base and P_e an edge partition of G_{mus}^K . $\{M \in MUSes(K) \mid P_e(M) \neq \emptyset\}$ is a MUS cover of K .

Proof. By definition, for each edge $a = (M, M')$, M or M' belongs to $\{M \in MUSes(K) \mid P_e(M) \neq \emptyset\}$. □

Now, we define the inconsistency measure of an edge partition as follows:

Definition 9. Let P_e be an edge partition of G_{mus}^K . We define the inconsistency measure of P_e as:

$$Inc^f(P_e) = \sum_{M \in V} f(|P_e(M)|)$$

where f is a strictly decreasing function over \mathbb{N} such that

$$\begin{cases} f(0) = 1 \\ \lim_{n \rightarrow +\infty} f(n) = c, \quad 0 < c < 1 \end{cases}$$

As f is a strictly decreasing function over the size of the elements of the edges partition, this means that when an edge is added to

$P_e(M)$, the contribution of M to the inconsistency measure of P_e decreases. In other words, the inconsistency degree of a MUS M depends on the number of edges attributed to $P_e(M)$. It is important to observe that the maximum inconsistency value $Inc^f(P_e)$ is reached for disjoint set of MUSes. In this last case, $Inc^f(P_e) = I_{MI}$.

Different functions f can be defined. For example, one can consider $f_1(n) = 1 - c + c^{n+1}$ or $f_2(n) = 1 - c + \frac{c}{n+1}$ s.t. $0 < c < 1$.

Example 2. Let us consider again the example depicted in Figure 2. $P_e(M_1) = \{e_1\}$, $P_e(M_2) = \{e_2\}$, $P_e(M_3) = \{e_3\}$, $P_e(M_4) = \{e_4\}$, and $P_e(M_5) = \{e_5\}$ is an edge partition. Using f_1 , we have $Inc^f(P_e) = 5(1 - c + c^2)$

Now, we are ready to define our inconsistency measure of given knowledge base.

Definition 10. Let K be a knowledge base. The inconsistency of K is defined as:

$$Inc(K) = \max_{P_e} Inc^f(P_e)$$

Let us provide the intuition behind the application of the maximum function over edge partitions. Assume we have n pairwise disjoint MUSes namely $\{M_1, \dots, M_n\}$ and an additional MUS M_{n+1} admitting a non empty intersection with the n first MUSes. Among the possible edge partitions we have P_e where $\forall i(1 \leq i \leq n)$ $P_e(M_i) = \{(M_i, M_{n+1})\}$ and $P_e(M_{n+1}) = \emptyset$. Another edge partition P'_e can be defined as $\forall i(1 \leq i \leq n)$, $P'_e(M_i) = \emptyset$ and $P'_e(M_{n+1}) = \{(M_1, M_{n+1}), \dots, (M_n, M_{n+1})\}$. We have $Inc^f(P_e) = n + 1 - f(1)n$ and $Inc^f(P'_e) = n + 1 - f(n)$. As monotony is required, we should have $Inc(K) \geq n$. $Inc^f(P_e)$ fails to satisfy such requirement when n is large enough. In contrast, with P'_e , the monotony property is satisfied. As it will be proved later, choosing the maximum inconsistency value among all the edge partitions is a key point towards the satisfaction of all the required properties.

Example 3. Let us consider again the example of Figure 2. Using f_1 , there are two edge partitions maximizing the inconsistency. The first one is P_e defined as follows: $P_e(M_1) = \{e_1, e_2, e_5\}$, $P_e(M_2) = \{e_3\}$, $P_e(M_3) = \emptyset$, $P_e(M_4) = \{e_4\}$, and $P_e(M_5) = \emptyset$. The second edge partition P'_e can be obtained from P_e by permuting M_2 and M_3 . The maximum value is obtained either with P_e or P'_e :

$$Inc(K) = 5 - 3c + 2c^2 + c^3$$

Let us now take another example described in Section 3 (see Figure 1). Applying our new measure, we obtain:

$$Inc(K_1) = 3 - c + c^3 \quad Inc(K_2) = 3 - c + c^3$$

$$Inc(K_3) = 3 - c + c^2 \quad Inc(K_4) = 3$$

$$Inc(K_5) = 3 - 2c + c^2 + c^3 \quad Inc(K_6) = 3 - 2c + c^2 + c^3$$

Our inconsistency measure allows to reorder the set of knowledge bases $\{K_1, \dots, K_6\}$ in the following way:

$$Inc(K_5) = Inc(K_6) < Inc(K_1) = Inc(K_2) < Inc(K_3) < Inc(K_4)$$

As we can see, K_4 is the most inconsistent one, while K_5 and K_6 are the least inconsistent ones. Furthermore $Inc(K_1) < Inc(K_3)$ and $Inc(K_2) < Inc(K_3)$ are required to satisfy the sub-additivity property. Our measure do not make distinction between K_5 and K_6

or K_1 and K_2 . Indeed, the MUSes graph of K_5 and K_6 are very similar, as well as for K_1 and K_2 .

We now show that Inc satisfy all required properties.

Proposition 4. Inc satisfies Consistency, Free Formula Independence, MinInc, Monotony, Ind-Additivity and Sub-Additivity properties.

Proof.

Consistency: If the knowledge base is consistent, the only partition P_e is the empty set. According to Definition 9, $Inc^f(P_e) = 0$. Consequently, $Inc(K) = 0$.

Free Formula Independence: Our measure consider only the MUSes, so the free formula independence is satisfied i.e., $Inc(K) = Inc(\text{unfree}(K))$.

MinInc: For a knowledge base K with a single MUS M , we have a single edge partition P_e s.t. $P_e(M) = \emptyset$. Consequently $Inc(K) = 1$.

Monotony: Let K be a knowledge base and α a formula. There is two cases:

- $\alpha \in \text{free}(K)$: $Inc(K) = Inc(K \cup \{\alpha\})$
- $\alpha \in \text{unfree}(K)$: Let $\{M_1, \dots, M_n\} = \text{MUSes}(K)$, $\{M'_1, \dots, M'_m\} = \text{MUSes}(K \cup \{\alpha\}) \setminus \text{MUSes}(K)$ and P_e an edge partition of K . P_e can be extended to an edge partition P'_e of $K \cup \{\alpha\}$. P'_e is such that if $a = (M_i, M'_j) \in E$ or $a = (M'_i, M'_j) \in E$, then $a \in P'_e(M'_j)$. Consequently, $Inc^f(P'_e) = Inc^f(P_e) + \sum_{i=1}^m f(M'_i)$. As $f(M'_i) > 0$ for all $0 < i < m$, then $Inc^f(P'_e) > Inc^f(P_e)$. Consequently, $Inc(K \cup \{\alpha\}) > Inc(K)$.

Ind-Additivity: If $\{K_1, \dots, K_n\}$ satisfies the conditions of the application of ind-Additivity then, each P_e an edge partition of $K_1 \cup \dots \cup K_n$ can be decomposed into disjoint edge partitions of K_1, K_2, \dots , and K_n . Consequently, $Inc(K_1 \cup \dots \cup K_n) = \sum_{i=1}^n Inc(K_i)$

Sub-Additivity: Let K_1, \dots, K_n such that $\text{MUSes}(K_1 \cup \dots \cup K_n) = \bigsqcup_{i=1}^n \text{MUSes}(K_i)$ and there exists i, j such that $K_i \cap K_j \neq \emptyset$. Performing $\sum_{i=1}^n Inc(K_i)$ is equivalent to remove in the MUSes graph G_{mus} of $(K_1 \cup \dots \cup K_n)$ the edges S linking the MUSes of K_i to MUSes of K_j . Let us note G'_{mus} such obtained graph. Let P_e be an edge partition of G_{mus} . Let us consider P'_e such that if $P_e(M) \in P_e$ then $P_e(M) \setminus S \in P'_e$. P'_e is an edge partition of G'_{mus} . As $S \neq \emptyset$, we have $Inc^f(P_e) < Inc^f(P'_e)$. Finally, P'_e is an edges partition of G'_{mus} , $Inc(K_1 \cup \dots \cup K_n) < \sum_{i=1}^n Inc(K_i)$. \square

Finally, in Proposition 5, we show that our inconsistency measure is bounded by the cardinality of the maximum independent set of the graph of MUSes.

Proposition 5. Let K be a knowledge base. We have

$$Inc(K) \geq \max_{S \in IS(G_{mus}^K)} (|S|)$$

where $IS(G_{mus}^K)$ is the set of independent sets of G_{mus}^K .

Proof. Let S be an independent set of G_{mus}^K . Then, there is no M, M' in S such that $M \cap M' \neq \emptyset$. Then it is possible to build an

edge partition P_e such that $P_e(M) = \emptyset$ for all $M \in S$. It is sufficient to attribute an edge (M, M') to $P_e(M')$ if $M \in S$ and $M' \in MUSes(K) \setminus S$. As $f(M) = 1$ when $P_e(M) = \emptyset$, then $Inc(K) \geq \max_{S \in IS(G_{mus}^K)} (|S|)$. \square

5 Discussion

Let us analyze the behavior of Inc inconsistency measure through its MUSes dependencies. We focus our study on three hypergraph classes: stars and chains (Figure 3.a and 3.c), and near disjoint (Figure 3.b). As f , we use the f_1 function defined in Section 4.

Stars: Let K a knowledge base with a set of MUSes $\{M_1, \dots, M_n\}$. We suppose that the MUSes hypergraph of K is a star. The MUSes graph G_{mus}^K is a clique. In this case, one edge partition P_e maximizing $Inc(K)$ is: $P_e(M_1) = \emptyset$ and $P_e(M_i) = \{(M_i, M_1), \dots, (M_i, M_{i-1})\}$. By simplifying $Inc(K)$, we obtain the following result:

$$Inc(K) = Inc^{f_1}(P_e) = (1 - c)n + c + \frac{c^2}{1 - c}(1 - c^{n-1})$$

Let us note that asymptotically, for large number of MUSes, the average contribution of each MUS tends to $(1 - c)$ since $\lim_{n \rightarrow +\infty} \frac{Inc(K)}{n} = 1 - c$.

Chains: Suppose now that G_{mus}^K is a chain and contains an odd number of nodes. The partition P_e maximizing $Inc^{f_1}(P_e)$ is: $P_e(M_i) = \emptyset$, if $i = 2k + 1$ and $P_e(M_i) = \{(M_{i-1}, M_i), (M_i, M_{i+1})\}$ if $i = 2k$. Then, the inconsistency value can be expressed as:

$$Inc(K) = Inc^{f_1}(P_e) = 1 + (1 - \frac{c}{2} + \frac{c^3}{2})(n - 1)$$

Asymptotically, we obtain $\lim_{n \rightarrow +\infty} \frac{Inc(K)}{n} = 1 - \frac{c}{2} + \frac{c^3}{2}$, the average contribution of each MUS to the inconsistency when n is large enough. It is worth noticing that, for chains MUSes hypergraph, the average contribution of each MUS is higher than for stars MUSes hypergraph ($1 - \frac{c}{2} + \frac{c^3}{2} > 1 - c$). This is rational since chains are less connected than stars (MUS graph is a clique). Similar reasoning can be applied for G_{mus}^K with an even number of nodes.

Near disjoint: Let us consider the MUSes $\{M_1, \dots, M_{n+1}\}$ of a knowledge base K such that $\{M_1, \dots, M_n\}$ are pairwise disjoint and M_{n+1} has a non empty intersection with a subset $S \subseteq \{M_1, \dots, M_n\}$ such that $|S| = k$. This near disjoint MUSes hypergraph is depicted in Figure 3).b. We obtain:

$$Inc(K) = n + 1 - c + c^{k+1}$$

For a fixed value of n , Inc suggests that the inconsistency value grows inversely to that of k . In other words, the inconsistency grows as the connections between MUSes decreases.

6 Computing Inconsistency Value

In this section, we are interested in computing $Inc(K)$. We provide a formulation as an optimization problem allowing the computation of $Inc(K)$, when the function f is fixed (see Definition 10 and 9). Let us recall that our approach is based on a partition of the set of edges over the set of vertices (MUSes) such that an edge $e = (M, M')$ is attributed either to M or to M' . The contribution of each MUS depends on the number of edges attributed to it. Formally, to seek for an optimal solution, we associate to each edge $e = (M, M')$

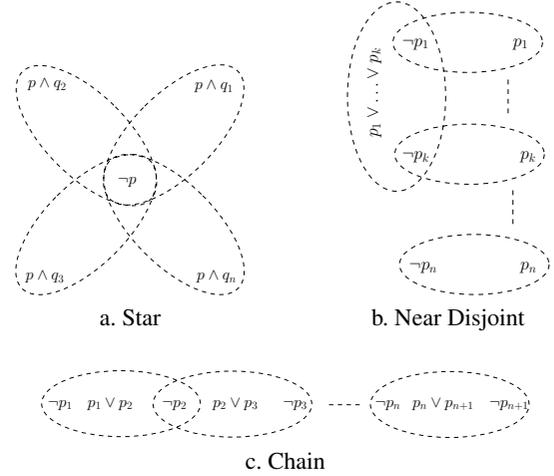


Figure 3. MUSes Hypergraph Classes

of G_{mus}^K , a new variable x_e . If e is attributed to M then x_e is true, otherwise it is false and it is associated with M' . This is equivalent to the attribution of x_e to M and $\neg x_e$ to M' . Then the global formulation can be obtained by considering the function used in Definition 9.

To illustrate the formulation of Definition 10 as an optimization problem, we consider the example of Figure 2. Let x_{e_i} ($1 \leq i \leq 5$) be the variable associated to edge e_i as depicted in Figure 2 (right hand side).

Using the function f_1 , computing $Inc(K)$, is equivalent to the maximization of the objective function depicted in Equation (2) obtained in the following way. We recall that $f_1(n) = 1 - c + c^{n+1}$. Let $M_1 < M_2 < \dots < M_5$ a total ordering on the set of MUSes and $\Gamma_e(M)$ the set of edges connected to M in the MUSes graph. The edge partition can be expressed as $P_e(M_i) = \{e \in \Gamma_e(M_i) | e \notin P_e(M_j), j < i\}$. In our example, the size of the elements of the edges partition can then be expressed as follows: $|P_e(M_1)| = x_{e_1} + x_{e_2} + x_{e_5}$, $|P_e(M_2)| = \neg x_{e_2} + x_{e_3}$, $|P_e(M_3)| = \neg x_{e_1} + \neg x_{e_3}$, $|P_e(M_4)| = x_{e_4}$ and $|P_e(M_5)| = \neg x_{e_4} + \neg x_{e_5}$. Then the objective function (Equation 2) can be derived from $Inc^f(P_e) = \sum_{i=1}^5 f_1(|P_e(M_i)|)$.

Similarly, for the function f_2 , we obtain the objective function described in Equation (3). Let us remark that the expression of Definition 10 as an optimization problem leads to a nonlinear objective function as illustrated by both Equations (2) and (3). Solving nonlinear optimization problems is generally more difficult than for linear problems. One can prove that Equation 2 can be linearized. A classical approach to find such optimum is to use a branch and bound like procedure.

As using an exact optimization method is computationally more costly, we propose in the sequel, an approximation of the $Inc(K)$ value. If the set of MUSes can be obtained in reasonable amount of time, one can use minimum vertex cover problem as an approximation of $Inc(K)$. Indeed, to maximize $Inc(K)$, one have to minimize the set of MUSes M with $P_e(M) = \emptyset$ which is equivalent to the problem of finding the minimum vertex cover of the MUSes graph. Computing the minimum vertex cover is known to be an NP-Hard

2.

$$Inc^{f1}(P_e) = 5(1 - c) + c^{1+x_{e_1}+x_{e_2}+x_{e_5}} + c^{1+\neg x_{e_2}+x_{e_3}} + c^{1+\neg x_{e_1}+\neg x_{e_3}} + c^{1+x_{e_4}} + c^{1+\neg x_{e_4}+\neg x_{e_5}} \quad (2)$$

$$Inc^{f2}(P_e) = 5(1 - c) + \frac{1}{1 + x_{e_1} + x_{e_2} + x_{e_5}} + \frac{1}{1 + \neg x_{e_2} + x_{e_3}} + \frac{1}{1 + \neg x_{e_1} + \neg x_{e_3}} + \frac{1}{1 + x_{e_4}} + \frac{1}{1 + \neg x_{e_4} + \neg x_{e_5}} \quad (3)$$

problem. It can be formulated easily as a linear program.

Problem: Minimum Vertex Cover

$$\text{minimize } \sum_{v \in V} x_v \text{ (minimize the total cost)}$$

subject to

$$x_u + x_v \geq 1 \forall \{u, v\} \in E \text{ (cover every edge)}$$

$$x_v \in \{0, 1\} \forall v \in V \text{ (} x_v = 1 \Leftrightarrow v \text{ in the vertex cover)}$$

Another alternative consists in approximating the minimum vertex cover using a greedy approach as shown in Algorithm 1. This algorithm is linear and allows to compute a reasonable approximation of the value of $Inc(K)$. For instance, using the greedy algorithm, the value obtained for the knowledge base of Figure 2 corresponds exactly to the optimal inconsistency value $Inc(K)$.

Algorithm 1 Inc(K): Greedy Approximation Approach

Require: A graph $G_{mus}^k = (V, E)$

Ensure: An edge partition P_e

```

1:  $P_e \leftarrow \emptyset$ 
2:  $val \leftarrow 0$ 
3: for  $M \in V$  do
4:    $E_d(M) \leftarrow \{e \mid e = (M, M') \in E\}$ 
5: end for
6: repeat
7:    $node \leftarrow \max_{M \in V} |E_d(M)|$ 
8:    $E \leftarrow E \setminus P_e(M)$ 
9:    $V \leftarrow V \setminus \{node\}$ 
10:   $val \leftarrow val + f(|P_e(M)|)$ 
11:   $P_e(node) \leftarrow E_d(node)$ 
12:  for  $M' \in V$  do
13:     $P_e(M') \leftarrow P_e(M') \setminus P_e(node)$ 
14:  end for
15: until  $(V = \emptyset)$ 
16: return  $val$ 

```

Algorithm 1 describes a greedy approach that aims to find an approximation of $Inc(K)$ based on vertex cover. First, the edges are distributed over vertices (MUSes). If $e = (M, M') \in E$, then e is associated temporarily to M and M' . At each iteration, the most connected vertex $node$ is chosen and P_e is updated accordingly. Then, the edges of this node are removed. The process is iterated until the set of vertices V becomes empty.

7 Related Work

In this section, we provide a brief overview of some works related to inconsistency measures. Several inconsistency measures have been

proposed over the years. An interesting work has been performed recently in [37] to compare a large set of inconsistency measures in terms of their ability to discriminate between knowledge bases. However, it is common to partition the set of approaches into three separate classes. The first one includes those based on either minimal inconsistent subsets [19, 28, 29, 21, 1, 24], or maximal consistent subsets [13, 8]. The second one [10, 16, 17, 30, 18, 12, 27, 38, 26, 15], focus on the semantics of the language, often based on some multi-valued semantics [32]. For example, in [39], the authors take the ratio of the propositions appearing in a minimal inconsistent subset wrt. the total number of propositions as the inconsistency value. The third one is based on probabilistic models [25, 8].

Among original approaches, one can cite also the one of [19]. It exploits the Shapley value, originally introduced in cooperative game, to analyze and quantify the amount of inconsistency that can be imputed to each formula in a given knowledge base. Usually, inconsistency measures can be partitioned according to their dependence on syntax or semantics. Semantic based measures aim to compute the proportion of the language that is affected by the inconsistency. The inconsistency measures belonging to this class are often based on some paraconsistent semantics and, thus, syntax independent, because we can still find paraconsistent models for inconsistent KBs. Whilst, syntax based approaches are concerned with the minimal number of formulas that cause inconsistencies. An overview of inconsistency measures for classical logics can be found in [13].

There is also related work on inconsistency measures in the context of quantitative logics. In particular, several works have extended existing inconsistency measures for classical frameworks to the probabilistic setting, while investigating their properties. One can quote for example the family of inconsistency metrics, proposed by [25, 8, 31, 35], based on the quantification of the minimal adjustments in the degrees of certainty (i.e., probabilities) of the statements necessary to make the knowledge base consistent. In [34], another inconsistency measure for probabilistic conditional logic is proposed. It is based on generalized divergence which is a specific distance for probability functions.

8 Conclusion and Future Work

In this paper we developed a novel approach to measure inconsistency in a knowledge base. A new property called sub-additivity is introduced providing a way to finely reorder knowledge bases. We showed that the classical approaches based on MUSes and their variants fail to satisfy the sub-additivity property. Then, we propose a new measure that exploits connections between MUSes and satisfies several properties including sub-additivity.

Our results are clearly of great interest. First, sub-additivity push further the issue of comparing the inconsistency of different knowledge bases. Secondly, MUS dependencies have been proven to be a key point for the design of more rational inconsistency measures. Several challenges need to be tackled in the future. Finding a better approximation of the inconsistency measure without enumerating all the MUSes is an important research issue. Intersections between MUSes can be more finely analyzed to improve the proposed mea-

sure. Finally, as a short term issue, we plan to analyze our measures in the light of the rational properties proposed in [6].

REFERENCES

- [1] Meriem Ammoura, Badran Raddaoui, Yakoub Salhi, and Brahim Oukacha, 'On measuring inconsistency using maximal consistent sets', in *ECSQARU*, pp. 267–276, (2015).
- [2] Samantha Bail, Matthew Horridge, Bijan Parsia, and Ulrike Sattler, 'The justificatory structure of the NCBO biportal ontologies', in *Proceedings of ISWC'2011*, pp. 67–82, (2011).
- [3] Ana Belén Barragans-Martinez, José Pazos Arias, and Ana Fernandez Vilas, 'On measuring levels of inconsistency in multi-perspective requirements specifications', in *PRISE*, pp. 21–30, (2004).
- [4] Salem Benferhat, Didier Dubois, and Henri Prade, 'A local approach to reasoning under inconsistency in stratified knowledge bases', in *ECSQARU*, pp. 36–43, (1995).
- [5] Salem Benferhat and Laurent Garcia, 'Handling locally stratified inconsistent knowledge bases', *Studia Logica*, **70**(1), 77–104, (2002).
- [6] Philippe Besnard, 'Revisiting postulates for inconsistency measures', in *JELIA*, pp. 383–396, (2014).
- [7] Sylvie Coste-Marquis and Pierre Marquis, 'Recovering consistency by forgetting inconsistency', in *JELIA*, pp. 113–125, (2008).
- [8] Dragan Doder, Miodrag Raskovic, Zoran Markovic, and Zoran Ognjanovic, 'Measures of inconsistency and defaults', *Int. J. Approx. Reasoning*, **51**(7), 832–845, (2010).
- [9] M. Fekete, 'Über die verteilung der wurzeln bei gewissen algebraischen gleichungen mit ganzzahligen koeffizienten', *Mathematische Zeitschrift*, **17**(1), 228–249, (1923).
- [10] John Grant, 'Classifications for inconsistent theories', *Notre Dame Journal of Formal Logic*, **19**(3), 435–444, (1978).
- [11] John Grant and Anthony Hunter, 'Measuring inconsistency in knowledgebases', *J. Intell. Inf. Syst.*, **27**(2), 159–184, (2006).
- [12] John Grant and Anthony Hunter, 'Analysing inconsistent first-order knowledgebases', *Artif. Intell.*, **172**(8-9), 1064–1093, (2008).
- [13] John Grant and Anthony Hunter, 'Measuring consistency gain and information loss in stepwise inconsistency resolution', in *ECSQARU*, pp. 362–373, (2011).
- [14] John Grant and Anthony Hunter, 'Measuring the good and the bad in inconsistent information', in *IJCAI*, pp. 2632–2637, (2011).
- [15] John Grant and Anthony Hunter, 'Distance-based measures of inconsistency', in *ECSQARU*, pp. 230–241, (2013).
- [16] Anthony Hunter, 'Measuring inconsistency in knowledge via quasi-classical models', in *AAAI*, pp. 68–73, (2002).
- [17] Anthony Hunter, 'Evaluating significance of inconsistencies', in *IJCAI*, pp. 468–478, (2003).
- [18] Anthony Hunter, 'How to act on inconsistent news: Ignore, resolve, or reject', *Data Knowl. Eng.*, **57**(3), 221–239, (2006).
- [19] Anthony Hunter and Sébastien Konieczny, 'On the measure of conflicts: Shapley inconsistency values', *Artif. Intell.*, **174**(14), 1007–1026, (2010).
- [20] Anthony Hunter, Simon Parsons, and Michael Wooldridge, 'Measuring inconsistency in multi-agent systems', *Unstliche Intelligenz*, **28**(3), 169–178, (2014).
- [21] Saïd Jabbour, Yue Ma, and Badran Raddaoui, 'Inconsistency measurement thanks to MUS-decomposition', in *AAMAS*, pp. 877–884, (2014).
- [22] Saïd Jabbour, Yue Ma, Badran Raddaoui, and Lakhdar Saïs, 'Prime implicates based inconsistency characterization', in *ECAI*, pp. 1037–1038, (2014).
- [23] Saïd Jabbour, Yue Ma, Badran Raddaoui, Lakhdar Sais, and Yakoub Salhi, 'On structure-based inconsistency measures and their computations via closed set packing', in *AAMAS*, pp. 1749–1750, (2015).
- [24] Saïd Jabbour, Badran Raddaoui, and Lakhdar Sais, 'Inconsistency-based ranking of knowledge bases', in *ICAART*, pp. 414–419, (2015).
- [25] Kevin Knight, 'Measuring inconsistency', *J. Philosophical Logic*, **31**(1), 77–98, (2002).
- [26] Yue Ma, Guilin Qi, and Pascal Hitzler, 'Computing inconsistency measure based on paraconsistent semantics', *J. Log. Comput.*, **21**(6), 1257–1281, (2011).
- [27] Yue Ma, Guilin Qi, Guohui Xiao, Pascal Hitzler, and Zuoquan Lin, 'Computational complexity and anytime algorithm for inconsistency measurement', *Int. J. Software and Informatics*, **4**(1), 3–21, (2010).
- [28] Kedian Mu, Weiru Liu, and Zhi Jin, 'A general framework for measuring inconsistency through minimal inconsistent sets', *Knowl. Inf. Syst.*, **27**(1), 85–114, (2011).
- [29] Kedian Mu, Weiru Liu, and Zhi Jin, 'Measuring the blame of each formula for inconsistent prioritized knowledge bases', *J. Log. Comput.*, **22**(3), 481–516, (2012).
- [30] Carlos A. Oller, 'Measuring coherence using lp-models', *J. Applied Logic*, **2**(4), 451–455, (2004).
- [31] David Picado-Muñoz, 'Measuring and repairing inconsistency in probabilistic knowledge bases', *Int. J. Approx. Reasoning*, **52**(6), 828–840, (2011).
- [32] Graham Priest, 'The logic of paradox', *J. Philosophical Logic*, **8**(1), 219–241, (1979).
- [33] Guilin Qi, Weiru Liu, and David A. Bell, 'Measuring conflict and agreement between two prioritized belief bases', in *IJCAI*, pp. 552–557, (2005).
- [34] W. Rodder and L. Xu, 'Elimination of inconsistent knowledge in the probabilistic expert system-shell SPIRIT (in German)', in *Operations Research Proceedings - Springer-Verlag*, pp. 260–265, (2001).
- [35] Matthias Thimm, 'Inconsistency measures for probabilistic logics', *Artif. Intell.*, **197**, 1–24, (2013).
- [36] Matthias Thimm, 'Towards large-scale inconsistency measurement', in *KI 2014*, pp. 195–206, (2014).
- [37] Matthias Thimm, 'On the expressivity of inconsistency measures', *Artif. Intell.*, **234**, 120–151, (2016).
- [38] Guohui Xiao, Zuoquan Lin, Yue Ma, and Guilin Qi, 'Computing inconsistency measurements under multi-valued semantics by partial MaxSAT solvers', in *KR*, (2010).
- [39] Guohui Xiao and Yue Ma, 'Inconsistency measurement based on variables in minimal unsatisfiable subsets', in *ECAI*, pp. 864–869, (2012).

Gaining Insight by Structural Knowledge Extraction

Pietro Cottone, Salvatore Gaglio, Giuseppe Lo Re, Marco Ortolani¹

Abstract. The availability of increasingly larger and more complex datasets has boosted the demand for systems able to analyze them automatically. The design and implementation of effective systems requires coding knowledge about the application domain inside the system itself; however, the designer is expected to intuitively grasp the most relevant features of the raw data as a preliminary step.

In this paper we propose a framework to get useful insight about a set of complex data, and we claim that a shift in perspective may be of help to tackle with the unaddressed goal of representing knowledge by means of the structure inferred from the collected samples. We will present a formulation of knowledge extraction in terms of Grammatical Inference (GI), an inductive process able to select the best grammar consistent with the samples, and a proof-of-concept application in a scenario of mobility data.

1 Introduction

Knowledge extraction has represented one of the most interesting challenges in Artificial Intelligence for the past decades [1]. Massive collections of data regarding the most disparate aspects of users' lives have become readily available for machine processing, deeply changing the nature of the problem. Nowadays the main concern is not just the necessity of accurate predictive models, but above all the demand for early provision of reliable insights to experts. The main issue regards the choice of the most appropriate tools and features to extract information from high-dimensional, incomplete and noisy datasets. Researchers have become increasingly more aware that “measuring” does not seamlessly translate into “understanding”, and their primary goal is to make sense of data by letting models *emerge* from the collected samples, rather than deducing them from pre-set assumptions. In this context, an essential requirement is the ability to build models that may be interrogated in order to improve representation and comprehension about the nature of data. A vast literature has investigated the interpretability of models and results produced by learning algorithms: approaches of increasing complexity have provided more and more accurate results, at the cost of less transparent representations [2]. Often, predictions supplied by these methods help the user in choosing the best option among several available ones; without interpretable models, this process can not provide any remarkable insight to support and explain the

decision. We claim that an approach to knowledge extraction that highlights the structural information can alleviate this problem. Specifically, we propose to represent the meaningful information by means of the *structure* inferred from the collected samples. Our definition of structural knowledge refers to the taxonomy proposed in [3], where three different types of knowledge are singled out:

- *declarative* knowledge expresses the awareness about some items, events or concepts. It is the knowledge about “*knowing that*”, which allows us to identify and describe an item or a concept, but does not enable us to use them;
- *procedural* knowledge describes how learners use or apply the former type of knowledge; it is about “*knowing how*” to do something.
- *structural* knowledge mediates the translation of declarative into procedural knowledge and facilitates the application of the latter; it refers to how concepts within a domain are interrelated; it is the knowledge about “*knowing why*”.

We note that *structural* knowledge is significantly different from *structured* knowledge, in that the latter typically refers to a description through entities and relationships; in other words, the focus is on how knowledge itself is organized. On the other hand, structural knowledge deals with the type of knowledge to be acquired, rather than the way it is organized. The emphasis is on the organization and structure of the objects of the analysis, and this will be the topic of our discussion.

In the present paper, the process of automatic extraction of this type of knowledge from raw data will rely on concepts and methods from *Algorithmic Learning Theory* (ALT), whose main subject is the study of formal languages and automata. Unlike its statistical counterpart, ALT does not require any specific constraints on the statistical properties of the available data, so it is well suited for cases when no a-priori hypotheses can be formulated. Its most interesting peculiarity is that the obtained knowledge is syntactically driven, hence intrinsically structural. Thus, representations obtained through algorithmic approaches can point out interesting relationships among the key elements of a dataset, implicitly suggesting what the most relevant features are. In particular, we will make use of *Grammatical Inference* (GI) [4], an inductive process able to select the best grammar (according to a metric) that is consistent with the samples. Instead of being represented in a vectorial space, we will thus regard our input as strings generated by an unknown grammar [5]; our claim is that GI can be successfully applied in order to get relevant insights about the hidden structure embedded in large collections of

¹ DICGIM – University of Palermo, Italy, email: *first-name.lastname@unipa.it*

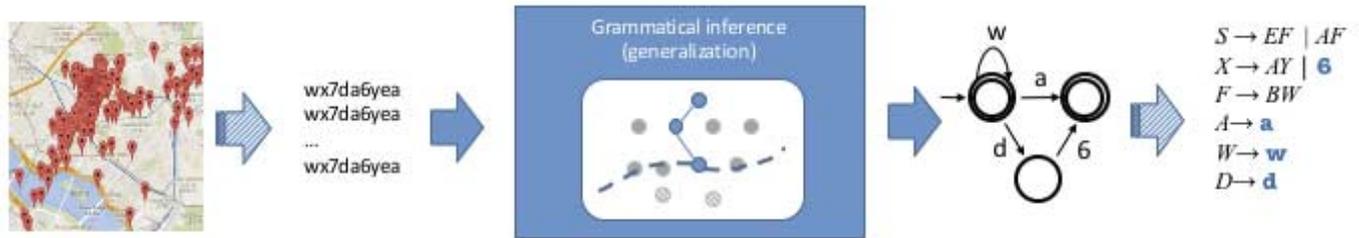


Figure 1. From data to grammars: an overview of the proposed approach.

data, enabling the user to pose new kinds of questions, taking advantage of the generative models obtained by the inductive process. Thanks to their recursive nature, grammars are also able to make recurrent relations among data explicit at different granularities.

In order to highlight the potential of the suggested approach, GI, and more specifically inference of regular languages [6], has been applied to the problem of inferring mobility models. In this context, multi-scale analysis allows us to grasp a more significant insight into data, and to get a better representation of user mobility habits, according to the traversed routes.

The remainder of the paper is structured as follows. Section 2 contains a very brief survey of methods for coping with high-dimensional and complex datasets. In Section 3 our approach based on GI will be described, followed by a case-study application to mobility data in Section 4. Finally, we will present our conclusions in Section 5.

2 Preliminaries: coping with dimensionality

Learning from experience is a key point in the design of intelligent agents. Over the years, this issue has been addressed in different ways, depending on the available devices, algorithms, and data, beginning with expert systems, probabilistic graphical models, and other statistical approaches. It soon became apparent, however, that one of the most relevant challenges was the selection of features from unlabeled data, so a lot of effort has been devoted during the last decade to the creation of systems able to perform this task automatically. Notable examples of this class of methods fall under the name of *Deep Learning*, and it has been shown that their finest performance is comparable to the best hand-engineered systems [7], [8]. A strong theoretical limitation, however, is represented by the well-known *No Free Lunch* theorem; one of its formulations informally states that “any two optimization algorithms are equivalent when their performance is averaged across all possible problems” [9]; in other words, there is no possible general criterion for choosing the optimal parameters of a method when absolutely no prior knowledge about the problem is available, except raw data [10]. If models are to be regarded as “black-boxes”, there is no reasonably efficient way to choose among several of them, when all choices fit the data comparably well.

The most recent technological advances have once more complicated the nature of the problem; it is now possible to

perform measurements regarding the most disparate aspects of users’ lives at previously inconceivable rates; moreover such data are highly heterogeneous, so the obtained datasets are typically high-dimensional and possibly incomplete. One of the most common examples is the massive volume of data with diverse features collected in smart environments [11], where pervasive networks of sensing devices are deployed, in order to support users in controlling the monitored environments [12], [13]. The peculiar challenges related to the analysis of this kind of data has given rise to a specific branch of AI named *Ambient Intelligence* (AmI), specifically aimed at exploiting the information about the environment state in order to personalize it, adapting the environment to users’ preferences [14].

In this context, most traditional approaches to data mining are not viable to handle the complexity of the new collections especially because they fail to provide useful insight into the real nature of data [15]. Very high dimensionality is hardly manageable by a human mind so, lacking support from the machine, designers are effectively prevented from grasping the most important features to consider.

It has thus been claimed [16] that the availability of *qualitative information* might ease the problem: at the cost of decreasing accuracy, the user can obtain a better understanding of the data, being free to focus on the overall organization at a larger scale; once a first insight is obtained, the process can be repeated at a smaller scale, considering only a subset of the original dataset, or a projection with lower dimensionality.

In this paper, we claim that qualitative information can provide very useful and compact guidelines to designers, in the preliminary set-up of systems for automatic data analysis. Also, recent findings [17] show that neural processes activated by human comprehension hint toward a grammar-based inner construction of knowledge representation; hence, modeling data in the form of grammars might help users to figure out the main structure behind relevant information. Grammar representations have been devised for *syntactic pattern recognition* [18]; in this work, we use some ideas pertaining to this research area, adapting and updating them with recent advances in data analysis.

3 Inferring the structure behind data through formal grammars

Assuming as a working hypothesis that the environment observed by the agent is computable, our goal is to exploit the

available data in order to infer a model that closely matches the unknown model for the environment. In other words, we assume that a (yet unknown) language describing our data exists; admittedly, this language may be extremely complex and data may be corrupted by noise, so that reconstructing the original language from raw data is likely to be a very challenging task. However, relying on formal languages to represent, organize and process knowledge is advantageous as they naturally provide a description of the relations between their elements, which may be regarded as their *hidden structure*.

A formal language is a (finite or infinite) set of sentences, each finite in length and made up of a finite set of symbols [19]. In real-life problems, however, data is often represented by a projection in a geometric space, whose dimensions are the chosen features, so a preliminary step requires translating the original representation of the data into a symbolic one. This is the first step shown in Figure 1, which depicts a high-level representation of our approach.

By encoding data as symbolic strings, we in fact move from a representation in a classical Euclidean space to a hierarchical organization; we rely on an ultra-metric space organized as a tree, where each node is associated to a string representing its path from the root, as will be detailed in the next sections.

The core of our approach is to use the symbolic data to infer the underlying target language through one of its possible representations. Generally speaking, two different descriptions can be associated to a language, namely a *generative* description, and a *recognition-based* one. In this paper, we focus on regular languages, so the corresponding representations are *regular grammars* and *Deterministic Finite Automata* (DFAs), respectively.

The *generative* description corresponds to a grammar, that is a formal system able to produce exactly the set of strings of the given language by applying predefined rewriting rules, expressed in the form of productions [20], [21]. A taxonomy of grammars has been proposed based on the complexity of such transformation rules, with regular grammars at its lowest level [22]. Generative descriptions are appealing to humans because they are intuitive, but their straightforward implementation is inefficient.

In the *recognition-based* description, a language is considered as the set of strings accepted by an automaton, that is a formal system that accepts all the set of strings belonging to the given language and rejects the others. Automata are appealing to machines, because they are formal, compact, low-level machines and can be implemented easily and efficiently; on the other hand, they are hardly understandable by a user.

Inferring a language through a grammar is by all means a learning process which may be characterized by its capability of *generalizing*. Unlike other learning approaches, where generalization is obtained by optimization, GI belongs to the category of algorithms that generalize through a search in a hypothesis space, so it may be regarded as an instance of the general framework known as *Version Space* strategy [23]. The key insight of this strategy is the assumption that hypotheses in the search space are organized through a “general-to-specific” ordering; a learning algorithm can explore the infinite hypothesis space by exploiting its structure, without explicitly visiting every element of it. In GI, the general-to-specific ordering is defined in terms of relations between automata, and the order is thus induced on languages.

3.1 Grammatical inference

As stated in [4], identifying a language is the main concern of *Grammatical Inference* (GI), which may be defined as the process of searching for a hidden grammar by exploiting the scarce available information, often consisting of just a set of strings; as such, GI belongs to the broader framework of *Algorithmic Learning Theory* (ALT), whose central concept is that of a *language learnability model*. Its main components are a *definition of learnability*, a *method of information presentation*, and a *naming relation*.

In this context, learnability is expressed by the principle of *identification in the limit* formulated by Gold [24]: the learning algorithm should identify the correct hypothesis on every possible data sequence consistent with the problem space. This idea is a non-probabilistic equivalent of statistical consistency, where the learner can fail on data sequences whose probability measure is 0; in this case, a learner (an algorithm) will identify a language in the limit if, after a number of presented strings, *its hypothesis no longer changes*.

The way in which input data are provided to the learner is called a *presentation*; let L indicate a language defined over an alphabet Σ , this is a function $\phi : \mathbb{N} \rightarrow X$, defined over the set of natural numbers, with codomain some set of samples $X \subset L$. As regards the methods of information presentation, two main procedures are available:

- presentation from *text*: a sequence of strings (x_1, x_2, \dots) belonging to L is provided; every string in L must appear at least once in the sequence. This presentation, denoted by $T(L)$, is also known as *positive* presentation:

$$T(L) = \{\phi : \mathbb{N} \rightarrow \Sigma^* : \phi(\mathbb{N}) = L\};$$

- presentation from *informant*: the learner is supplied with strings marked as *positive* (i.e. belonging to the language L) or *negative* (not in L). This kind of presentation, denoted by $I(L)$, is known as *complete*:

$$I(L) = \{\phi : \mathbb{N} \rightarrow \Sigma^* \times \{0, 1\} : \phi(\mathbb{N}) = L \times \{1\} \cup \bar{L} \times \{0\}\},$$

where \bar{L} indicates the complement of L with respect to Σ^* .

Finally, the naming function is some surjective function $\mathbb{L} : \mathcal{G} \rightarrow \mathcal{L}$, with the set \mathcal{G} of grammars as the domain, and the set of languages \mathcal{L} as the codomain.

The language learnability paradigm has some theoretical limitations. As Gold showed in [24], a class of *super-finite languages*² cannot be identified in the limit from a text presentation. This class includes regular languages, hence they cannot be inferred from positive examples only; in other words, a set of strings belonging to the target regular language is not sufficient to learn it.

Even if we turn to a presentation from informant, we incur some limitations, as also pointed out in [24]. In particular, the following holds:

Theorem 1 *The whole class of recursive languages can not be identified in the limit from a complete presentation.*

However in the same work Gold showed that, when we restrict the class of languages, it may be proven that:

² A super-finite language class is a class that contains all finite languages and at least one infinite language.

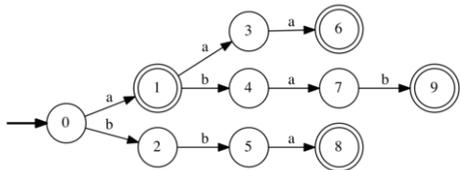


Figure 2. $PTA(I_+)$ for $I_+ = \{a, aaa, abab, bba\}$.

Theorem 2 *The class of primitive recursive languages³ can be identified in the limit by a complete presentation.*

Regular languages are primitive recursive languages, so a complete presentation of examples guarantees their theoretical learnability. We can thus turn our attention to how the inference process can be practically carried out. Motivated by the generalization principle, we are interested in identifying the most general DFA consistent with the given samples, i.e. the minimum canonical automaton.

Even though, given a complete presentation of positive and negative examples $I = I_+ \cup I_-$, an automaton consistent with I exists and is unique [20], Gold also showed that finding the minimum consistent automaton with a set of samples is an NP-hard problem; therefore, some heuristic is needed to carry out this search in an efficient way.

3.2 Generalization as search

We will characterize the search space for our problem through the following basic elements:

- *initial node*: an “acceptable” DFA;
- *successor function*: a set of successors of an automaton generated by pairwise state merging;
- *target*: minimum automaton that is consistent with the samples I .

This search space may thus be described as a Boolean lattice [26], whose initial node is a tree automaton – the so-called *Prefix Tree Acceptor* (PTA) – accepting precisely the positive examples I_+ , such as the one shown in Figure 2.

The complexity of the search can be eased by exploiting some general-to-specific ordering of the nodes; intuitively, in grammatical induction, this ordering is based on constraints characterizing the hypotheses, with fewer constraints entailing more general hypotheses, and vice versa. By construction, the $PTA(I_+)$ is the most specific DFA for the positive examples, and we want to explore the space moving toward the minimum consistent automaton, with the negative examples as our bounds.

The set of successors of an automaton is generated by pairwise merging operations: two states of the original automaton are chosen for merging, resulting into a new automaton with one fewer state with respect to the original, as shown in Figure 3 which depicts an excerpt of a lattice. Even though merging two states might result into a non-deterministic automaton, it is possible to carry out the generalization process

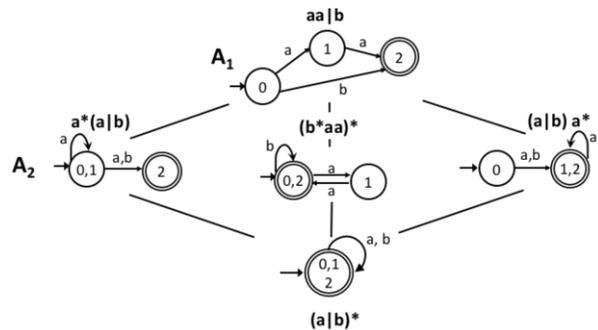


Figure 3. Excerpt of a lattice: automata in the middle row are obtained from the initial one by merging different pairs of its states.

avoiding non-determinism, by making use of the so-called *folding* operation, as described in [4].

Pairwise merging may be formally defined as a partition of the set of states of the original automaton A , and is a *derivation operation*, which defines a partial order relation over the set $\Pi(A)$ of all the possible partitions of the set of states in A . Notably, it preserves the property of *language inclusion*, as shown in [26], which means that the application of the merging operator:

- either causes the number of states to decrease, but the recognized language is preserved;
- or it also implies a change in the language recognized by the resulting automaton, but such language is more general, and properly includes the original one.

The Boolean lattice $Lat(PTA(I_+))$ is thus completely defined by its initial node, i.e. $PTA(I_+)$, and the nodes obtained by repeatedly applying merging operations included in $\Pi(PTA(I_+))$; the deepest node in $Lat(PTA(I_+))$ is the *Universal Automaton* (UA), that accepts all the strings defined over an alphabet Σ . The inference of regular languages, provided a presentation from an informant, can be turned into the search for an automaton $A' \in Lat(PTA(I_+))$, given the additional hypothesis of structural completeness of I_+ ⁴.

It may be proven [26] that if I_+ is a structurally complete sample with respect to the minimal automaton A accepting a regular language L , then A belongs to $Lat(PTA(I_+))$, so the inference of a regular language by presentation from an informant can be turned into the search for an automaton in the space defined by that Boolean lattice.

The definition of minimal DFA consistent with the sample set I can also be visualized in terms of the elements of the lattice, through the so-called *Border Set*, which establishes the limit of generalization in the search process under the control of negative samples I_- , as graphically shown by the dotted line in Figure 4. The border set parts the lattice into two main subsets: *admissible* automata, between the root and the border, and *inadmissible* ones, falling beyond the border. The minimum DFA consistent with I is the deepest (i.e. smallest) automaton falling right on the border set, hence still admissible.

³ A language is primitive recursive if its characteristic function is primitive recursive. The formal definition may be found for instance in [25].

⁴ A I_+ sample set is said to be structurally complete with respect to an automaton A , if every transition of A is used by at least a string in I_+ , and every final state in A corresponds to at least one string in I_+ .

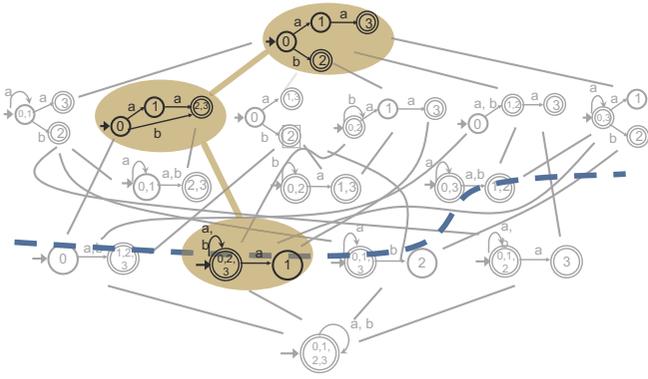


Figure 4. Sketch of a search in the Boolean lattice.

Since the number of automata in the lattice generated by an initial *PTA* with n states is given by the *Bell number*:

$$\omega(n) = \sum_{p=0}^{n-1} \binom{n-1}{p} \omega(n-1),$$

with $\omega(0) = 1$, then the space defined by such lattice is clearly too large to be searched exhaustively; therefore, some approaches have been proposed to carry out the search more efficiently.

Evidence-Driven State Merging (EDSM) represents a state-of-the-art iterative algorithm to perform such search, whose detailed description can be found in [6]. It was introduced to reduce the number of comparisons necessary during merging, and makes use of a heuristic that computes a score for all possible merges by counting the number of strings that would end in the same state; the function returns $-\infty$ if the merge makes the automaton inadmissible (i.e. an element of I_- would be accepted or an element of I_+ rejected); the pair with the highest score is chosen for merging.

The results of applying this algorithm in order to perform structural knowledge extraction in a practical scenario will be discussed in the next section.

4 A proof of concept: mobility data

In order to provide a proof of concept for our approach to structural knowledge extraction, we consider a case study aimed to infer and represent user mobility models via regular languages.

A mobility model is a concise and meaningful representation of past and future mobility behaviors of users. Nowadays, location data are easy to collect, thanks to the availability of a wide set of common devices, such as smartphones or tablets, that easily provide large amounts of measurements [27]. Extracting meaningful information from this wealth of data, however, is still an open issue. The main questions, for instance, regard the selection of the most significant features, the proper granularity necessary to perform effective analysis, and the metric to use to compare the mobility habits of various users.



Figure 5. An example of the first two bits of a geohash string.

4.1 Positions as symbols

Following the approach presented in the previous sections, the first step of the process requires translating paths into a symbolic representation; to this aim, we selected an encoding system for geographical coordinates known as *geohash* encoding.

Geohash assigns a hash string to each (*latitude, longitude*) pair; originally, it was developed to provide a smart and easy representation of URLs, but it has been since widely used to store spatial coordinates in databases [28]. The encoding is based on a hierarchical spatial data structure that recursively subdivides the whole globe into “buckets” according to a grid; unlike traditional coordinate systems, it does not actually represent a point, but rather a bounding area to which the point is restricted. The space is partitioned according to a 4×8 grid; each cell can be recursively divided into 32 smaller cells, and so on, thus providing a hierarchical structure that resembles that of a recursive quadtree; at each iteration, each cell is identified by an alphanumeric character from an alphabet of 32 symbols.

In the geohash string, even bits encode information about longitude, while odd ones encode latitude; an example of encoding at the first 2 levels is reported in Figure 5, which shows two rectangles partitioning the entire globe longitudinally (left), and the four rectangles that may be obtained with a successive latitudinal partition (right). This process can be iterated until the desired spatial accuracy is obtained: the longer the geohash string, the smaller the area. The length of the binary string must always be a multiple of 5 to allow its conversion to a sequence of symbols from geohash alphabet.

The following table shows the size of the area identified by a geohash code with respect to its length.

Table 1. Area covered by a cell with respect to the length of its geohash encoding string.

Geohash length	\approx Covered Area km^2
1	16,000,000
2	500,000
3	15,000
4	500
5	15
6	0.5
7	0.02

Geohash encoding possesses two notable properties, namely:

- *inclusion*: it is always possible to add a character to a geohash string, obtaining a new string that identifies a cell contained in the original one. For example, the coordinates (38.120281, 13.357278) identify a point included inside the `sqc2zg` cell, but also inside `sqc2zgw` or `sqc2zgwk`;

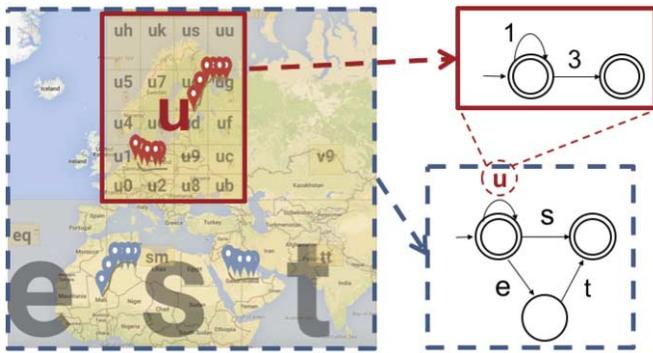


Figure 6. From trajectories to DFA hierarchy: given the DFA for a larger cell (dashed-line box), a more detailed model can be built by inferring the DFA for transition *u* (solid-line box).

- *locality*: strings with common prefix mark contiguous cells. Thus, it is very simple to check if two cells are neighbors. The converse is not always true: two cells could be next to each other even if they do not share a common prefix.

In the following, we exploit both properties to achieve an effective implementation of our GI process.

4.2 Mobility models as automata

The source data we will consider consists of *movement tracks* [29]:

Definition 1 (Movement track) *This is a temporally ordered sequence of spatial-temporal position records captured by a device during the whole lifespan of the user observation. Each record contains a position and the instant of the capture, with no two records having the same instant value.*

Movement tracks have to be turned into *trajectories* [30] in order to be able to filter out noise, and to estimate other movement features, such as speed and direction. The true aim of the analysis may however be identified in the *paths*:

Definition 2 (Path) *A path is the portion of a trajectory between two relevant points in time or space dimensions.*

Paths reveal user behavior and highlight relevant places where users spend most of their time. Being aware of these places is crucial in many applications, and they are fundamental in comparing habits of several users or in recognizing anomalies or changes in their routines.

In our approach, trajectories are transformed into symbolic sequences by turning each pair of coordinates into the corresponding geohash string; through this encoding, they can easily be analyzed at different spatial scales: once the required precision is set, it is sufficient to truncate every geohash string of each trajectory at the corresponding length. The user mobility model is finally decomposed by following the trajectories with respect to every cell of geohash encoding: a regular language is thus learned for each cell of the geographical area crossed by user movements, starting at the highest level of granularity, as shown in Figure 6. At any level, a more complex and detailed automaton may be obtained by substituting



Figure 7. Subsequences with the same prefix originate mini trajectories: the third element of each string (in bold face) may be concatenated to obtain a mini trajectory.

to each symbol the recognizer for the corresponding cell (see Figure 6); this is equivalent to concatenating a new symbol to the geohash string, and inspecting the movements at a finer detail. The process stops at the cell granularity representing the required accuracy.

At smaller scales, *mini trajectories* can be obtained for each cell by considering all the contiguous subsequences of strings within each trajectory that share the prefix corresponding to the cell. For each element of the subsequence, only the symbol of the sub-cell is considered, thus the subsequence is turned into a string (see Figure 7); after recovering all the strings related to the cell, the needed information to infer a regular language is obtained.

As discussed earlier, a presentation from an informant is required to infer a regular language; so, in order to obtain the mobility models for a user, a set of examples of their paths is not enough. Selecting a proper negative sample set in case only positive samples are available is an open issue in GI. As a practical solution, we consider the symmetric difference between the trajectories of other users and those of the current one, as they intuitively provide valid trajectories that were not actually traversed by the specific user. These samples can be considered as the negative sample set for the language representing the mobility habits of the current user. We thus use the EDMS algorithm to infer the corresponding regular language, given the mini-trajectory sets of negative and positive route samples.

4.3 The language of paths

In order to assess our approach, we examined data provided by the *Geolife* dataset [31], which is a collection of time-stamped triples of the form (*latitude, longitude, altitude*), representing the spatial behavior of 182 users monitored for 5 years, collected by Microsoft Research Asia. Most trajectories took place in China, near Beijing, but routes crossing USA and Europe are also present. More than 17,000 trajectories are contained in the dataset, for a total of approximately 50,000 hours of tracked movements. GPS loggers and smartphones acted as acquisition devices, providing a high sampling rate (1 ~ 5 seconds in time, and 5 ~ 10 meters in space) for more than 90% of the data.

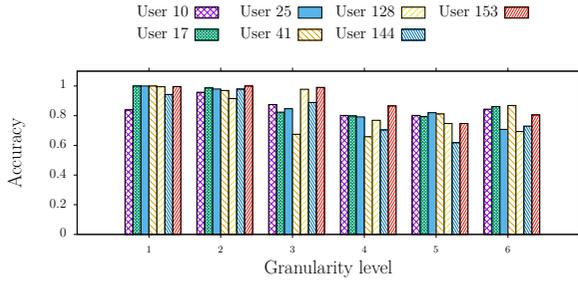


Figure 8. Accuracy with respect to varying granularity for 7 users (80% training, 20% test).

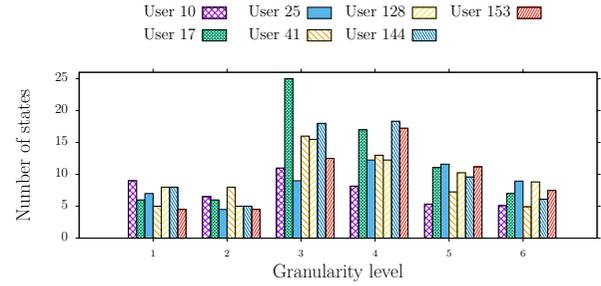


Figure 9. Number of states with respect to the granularity of considered paths.

The goal of our analysis is to address the main issues in the analysis of mobility data, as mentioned at the beginning of this section, and in particular: to assess the correlation between the complexity of spatial behaviors and the respective complexity of the model; to figure out the most representative granularity level that should be considered; and finally to provide a guideline for the definition of a mobility similarity measure between users.

The first issue was addressed by estimating the accuracy of our structural models at varying granularity degrees; in this context, we refer to the classical definition of accuracy as the ratio between the number of corrected classified samples (paths that were actually travelled by the current user, or are correctly disregarded) and the number of the examined samples (all the paths to be classified). We chose to ignore measurements about altitude, as they did not provide any significant information, and to consider only longitude and latitude; no further prior knowledge was assumed. Our analysis was based on the selection of the subset of the most representative users, i.e. the ones that have at least 300 paths at every granularity. For each user, the available data were partitioned into training (80%) and testing (the remaining 20%). The maximum string length for geohash encoding was set to 7, corresponding to a precision of 153 meters (an area of about 0.02 km²). Results for seven representative users are reported in Figure 8, which shows that high accuracy is obtained at all spatial scales; moreover, it is evident that performances are satisfactory even with higher resolution trajectories. We are thus supported in our claim that the complexity of the users’ spatial behavior may be captured by models as simple as regular languages. This is confirmed by several other works in literature [32], [33], [34] that, based on statistical approaches, revealed that human spatial trajectories are highly predictable by simple models: meaningful patterns can be described by a sequence of locations, and are characterized by particular shapes. Thus, we can reasonably conclude that a preliminary insight on data is able to hint the complexity of the model needed for a deeper analysis.

As a second step, we moved on to assess how choosing different granularities affects the complexity of the resulting models, measured in terms of number of states; as is clear from Figure 9, which reports values averaged over all automata for each granularity, medium granularities (encoding lengths 3 and 4) require more complex recognizers with respect to both higher and lower granularities. Arguably, user mobility shows

the highest variability at intermediate spatial resolutions (e.g. city-wide), where more features are needed to separate different behaviors, whereas most users typically remain within the same nation, thus exhibiting a simpler behavior that can be explained through a less complex model.

Finally, we tackled the challenge of providing insight about how to identify similarities among users; here, we refer to the definition of similarity used in [35], i.e. a measure for capturing the affinity between two users according to their trajectory patterns, encoded in the respective mobility profiles. Due to the intrinsic recursive nature of users’ paths, it is very common that pronounced similarities emerge naturally both at a sufficiently low, and at a very high resolution. In fact, most users share the same behavior at nation-wide scale, since they spend most of their time without leaving their own country; at the other end of the scale, short paths are typically very basic, due to the physical constraints of the urban landscape, so most users will likely show similarities when they traverse small areas, e.g. within a few blocks.

Those considerations appear in all evidence from our experiments, and two representative cases are reported in Figure 10. The first row shows the automata produced for users 128 and 153 at the highest possible granularity, i.e. considering only the first symbol of the geohash encoding. By referring to Table 1 we see that such granularity corresponds to an area of 16 million km², such as an entire country or larger; the automaton for user 128 tells us that its strings contain just one symbol (*w*, which is the geohash code roughly covering China), whereas for user 153 two symbols are allowed (*w* and *9*, which roughly encode China and the USA/Mexico region, respectively).

It is informative to look at the alternative representation of the two automata in the form of the corresponding regular grammars:

$$\mathcal{G}_{128} : \begin{cases} S \rightarrow WS \mid W \\ W \rightarrow \mathbf{w} \end{cases} \quad \mathcal{G}_{153} : \begin{cases} S \rightarrow WS \mid W \\ \quad \quad \quad \mid YS \mid Y \\ W \rightarrow \mathbf{w} \\ Y \rightarrow \mathbf{9} \end{cases}$$

where the dissimilarities between the two users are evident: despite the fact that there exist trajectories for both users that are confined within China, as coded by the *W* productions, only user 153 moves to a different area altogether, as represented by the *Y* productions in the rightmost grammar.

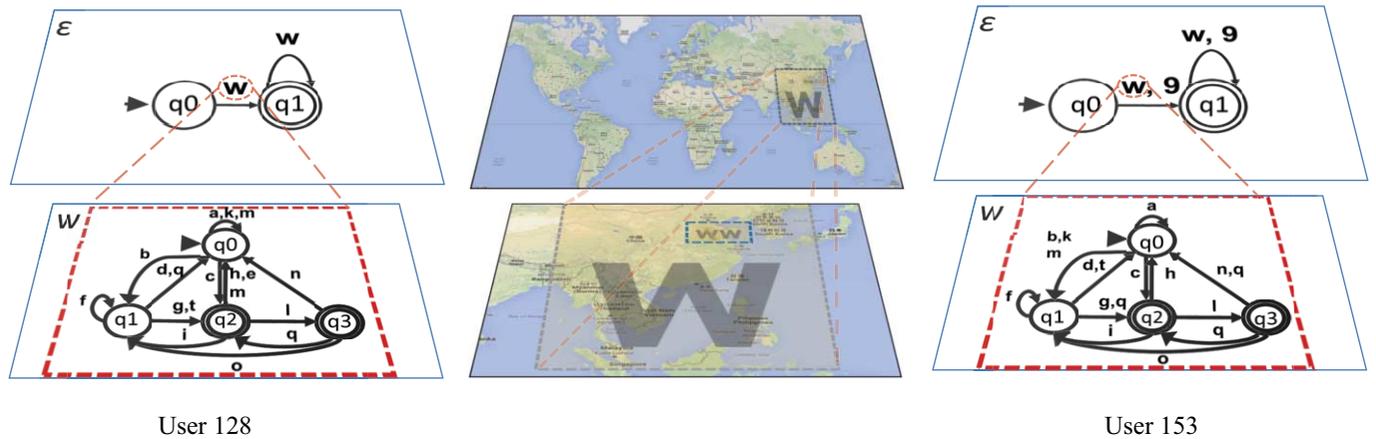


Figure 10. Comparison of DFAs representing the movements of two users, at different levels of granularity.

We can now exploit one of the key peculiarities of our structural analysis, namely the possibility of a simple navigation across different spatial scales of granularity or, equivalently, a “hierarchical” navigation through the pool of automata obtained by specifying a symbol in a transition through a more specific automaton representing a finer detail (see Figure 10). The automata depicted in the bottom row “expand” the w -transitions of the corresponding upper-level automata; in other words, they specify the behavior for each of the two users when they move within the region encoded as w ⁵. Visually inspecting such automata is sufficient to recognize their similarities; this qualitatively shows that paths have a *multi-scale* nature: significant information can be extracted by observing data at different granularity degrees, and a similarity metric should take this characteristic into account. Our structural models are able to highlight the most appropriate representation level for the problem, hence to provide useful insight to the system designer.

5 Conclusion

This paper described a proposal for a structural approach to coping with the complexity represented by big collections of data. Our claim is that often knowledge can be represented by means of the structure inferred from the wealth of collected samples, limiting the amount of a-priori information needed.

By using a syntactically driven inference algorithm, we showed that it is possible to build generative models able to suggest the relevant relations between different subsets of the samples, and to perform multi-scale analysis suitable to identify the most important features emerging at different granularities.

The presented results, regarding the issue of understanding mobility data, show how, in this context, the availability of generative and multi-scale models allows to get a useful insight of the whole dataset.

⁵ For instance, a valid string for the automaton at the lower left is **akc**; this means that the user is moving across cells whose geohash codes are **a**, **k**, and **c**, all of which are subcells of macro-region **w**.

Acknowledgments

The authors would like to thank Gabriele Pergola, student at our lab, for his enthusiastic help during the implementation of parts of the system, and for proofreading the article.

This work was partially supported by the Italian Ministry of Education, University and Research on the “StartUp” call funding the “BIGGER DATA” project, ref. PAC02L1_0086 – CUP: B78F13000700008.

References

- [1] U. M. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, “From data mining to knowledge discovery in databases,” *AI magazine*, vol. 17, no. 3, pp. 37–54, 1996.
- [2] Z. C. Lipton, “The mythos of model interpretability,” in *Proceedings of the workshop on human interpretability in machine learning*, ser. WHI 2016, New York, NY, USA, 2016.
- [3] D. H. Jonassen, K. Beissner, and M. Yacci, *Structural knowledge: Techniques for representing, conveying, and acquiring structural knowledge*. Psychology Press, 1993.
- [4] C. de la Higuera, *Grammatical inference: Learning automata and grammars*. New York, NY, USA: Cambridge University Press, 2010.
- [5] D. Angluin, “Queries and concept learning,” *Machine learning*, vol. 2, no. 4, pp. 319–342, Apr. 1988.
- [6] K. J. Lang, B. A. Pearlmutter, and R. A. Price, “Results of the Abbadingo One DFA learning competition and a new Evidence-Driven State Merging Algorithm,” in *Proceedings of the 4th international colloquium on grammatical inference*, ser. ICGI ’98, London, UK: Springer-Verlag, 1998, pp. 1–12.
- [7] Y. Bengio, “Learning deep architectures for AI,” *Foundations and trends in machine learning*, vol. 2, no. 1, pp. 1–127, Jan. 2009.

- [8] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on machine learning*, ser. ICML '08, Helsinki, Finland: ACM, 2008, pp. 1096–1103.
- [9] D. H. Wolpert and W. G. Macready, "Coevolutionary free lunches," *IEEE transactions on evolutionary computation*, vol. 9, no. 6, pp. 721–735, Dec. 2005.
- [10] D. Whitley and J. P. Watson, "Complexity theory and the no free lunch theorem," in *Search methodologies: Introductory tutorials in optimization and decision support techniques*, E. K. Burke and G. Kendall, Eds. Boston, MA: Springer US, 2005, pp. 317–339.
- [11] A. R. Ganguly, J. Gama, O. A. Omitaomu, M. M. Gaber, and R. R. Vatsavai, *Knowledge discovery from sensor data*, 1st. Boca Raton, FL, USA: CRC Press, Inc., 2008.
- [12] P. Cottone, S. Gaglio, G. Lo Re, and M. Ortolani, "User activity recognition for energy saving in smart homes," *Pervasive and mobile computing*, vol. 16, Part A, pp. 156–170, 2015.
- [13] S. Gaglio, G. Lo Re, and M. Morana, "Human activity recognition process using 3-D posture data," *Ieee transactions on human-machine systems*, vol. 45, no. 5, pp. 586–597, Oct. 2015.
- [14] A. De Paola, A. Farruggia, S. Gaglio, G. Lo Re, and M. Ortolani, "Exploiting the human factor in a wsn-based system for ambient intelligence," in *International conference on complex, intelligent and software intensive systems (CISIS '09)*, Mar. 2009, pp. 748–753.
- [15] A. Rajaraman and J. D. Ullman, *Mining of massive datasets*. New York, NY, USA: Cambridge University Press, 2011.
- [16] G. Carlsson, "Topology and data," *Bulletin of the american mathematical society*, vol. 46, no. 2, pp. 255–308, 2009.
- [17] N. Ding, L. Melloni, H. Zhang, X. Tian, and D. Poeppel, "Cortical tracking of hierarchical linguistic structures in connected speech," *Nature neuroscience*, vol. 19, no. 1, pp. 158–164, Jan. 2016.
- [18] K. S. Fu, *Syntactic methods in pattern recognition*, ser. Mathematics in science and engineering. New York: Academic, 1974, vol. 112.
- [19] N. Chomsky, *Syntactic structures*. Walter de Gruyter, 2002.
- [20] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to automata theory, languages, and computation*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2006.
- [21] W. Levelt, *An introduction to the theory of formal languages and automata*. John Benjamins Pub., 2008.
- [22] N. Chomsky, "Three models for the description of language," *IRE transactions on information theory*, vol. 2, no. 3, pp. 113–124, 1956.
- [23] T. M. Mitchell, "Generalization as search," *Artificial Intelligence*, vol. 18, no. 2, pp. 203–226, 1982.
- [24] E. M. Gold, "Language identification in the limit," *Information and control*, vol. 10, no. 5, pp. 447–474, 1967.
- [25] M. L. Minsky, *Computation: Finite and infinite machines*. Prentice-Hall, Inc., 1967.
- [26] P. Dupont, L. Miclet, and E. Vidal, "What is the search space of the regular inference?" In *In proceedings of the second international colloquium on grammatical inference (ICGI'94)*, Springer Verlag, 1994, pp. 25–37.
- [27] P. Cottone, S. Gaglio, G. Lo Re, and M. Ortolani, "A machine learning approach for user localization exploiting connectivity data," *Engineering applications of artificial intelligence*, vol. 50, pp. 125–134, 2016.
- [28] Z. Balkić, D. Šoštarić, and G. Horvat, "Geohash and uuid identifier for multi-agent systems," in *Proceedings of the 6th KES international conference on agent and multi-agent systems: Technologies and applications*, ser. KES-AMSTA'12, Dubrovnik, Croatia: Springer-Verlag, 2012, pp. 290–298.
- [29] D. C. Renso, D. S. Spaccapietra, and D. E. Zimnyi, *Mobility data: Modeling, management, and understanding*. New York, NY, USA: Cambridge University Press, 2013.
- [30] Y. Zheng and X. Zhou, *Computing with spatial trajectories*. Springer Publishing Company, Incorporated, 2011.
- [31] Y. Zheng, L. Liu, L. Wang, and X. Xie, "Learning transportation mode from raw GPS data for geographic applications on the web," in *Proceedings of the 17th international conference on world wide web*, ser. WWW '08, Beijing, China: ACM, 2008, pp. 247–256.
- [32] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, 2008.
- [33] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.
- [34] Y. Chon, H. Shin, E. Talipov, and H. Cha, "Evaluating mobility models for temporal prediction with high-granularity mobility data," in *Proceedings of the 2012 IEEE international conference on pervasive computing and communications (PerCom)*, IEEE, 2012, pp. 206–212.
- [35] X. Chen, J. Pang, and R. Xue, "Constructing and comparing user mobility profiles," *ACM Transactions on the Web*, vol. 8, no. 4, 21:1–21:25, Nov. 2014.

Complexity of Threshold Query Answering in Probabilistic Ontological Data Exchange

Thomas Lukasiewicz¹

Livia Predoiu²

Abstract. We study the complexity of threshold query answering in the logical framework for probabilistic ontological data exchange, which is an extension of the classical probabilistic data exchange framework with (1) probabilistic databases compactly encoded with several different annotations according to three different probability models used and (2) existential rules of different expressiveness. The ontological data exchange framework provides a logical formalization of exchanging probabilistic data and knowledge from one ontology to another via either deterministic or probabilistic mappings. We define the threshold query answering task in this framework and provide a thorough analysis of its computational complexity for different classes of existential rules and types of complexity. We also delineate several classes of existential rules and a probability model along with a compact encoding in which the threshold query answering problem can be solved in polynomial time in the data complexity.

1 INTRODUCTION

Being able to process uncertainty attached to data is becoming increasingly important in many areas such as information extraction, data cleaning, and Web data integration. Applications in these areas produce large volumes of uncertain data. At the moment, the best way to model, store, and process uncertain data can be considered to be in probabilistic databases [27]. At the same time, the field of databases enriched with ontological knowledge has gained importance through ontology-based data access (OBDA) [26]. Crucial challenges of such ontologically enhanced databases are the integration and the exchange of data and knowledge. There is currently also a huge need for combining the latter two areas in ontology-based probabilistic databases, especially due to important applications in the Semantic Web and in ontology-based access to Big Data.

In this paper, we tackle these challenges by studying an extension of the well-known framework of data exchange [14], which is an important and powerful theoretical framework used for studying data-interoperability tasks that require data to be transferred from source databases to a target database that comes with its own (independently created) ontological schema (and schema constraints). The data is translated from one database to the other via schema mappings, which are declarative specifications that describe the relationship between two database schemas. In [13], a probabilistic extension of the classical deterministic framework of data exchange has been proposed. Recently, the works [23, 24] have extended this probabilistic data exchange framework towards probabilistic ontological data exchange where source and target ontology-based data access systems

have been considered for data exchange. While research in (deterministic and probabilistic) data exchange has only considered weakly acyclic existential rules (see, e.g., [6]), in [23, 24], several other classes of existential rules from Datalog+/- have been considered as ontology and source-to-target mapping languages. In addition, compared to [13], where only the elementary-event-independent probabilistic model has been considered together with full Boolean formulas as annotations, in [23, 24], an additional probabilistic model based on Bayesian networks, as well as annotations consisting of positive Boolean formulas without negation, and annotations consisting of a single positive literal have been considered.

This paper continues this line of research. We consider answering threshold queries in probabilistic ontological data exchange with different classes of existential rules, representing both ontological rules and mapping rules, as well as different probabilistic models and compact encodings for the probabilistic ontological data and mappings. Probabilities of annotations with Boolean events are specified via either pairwise independent random variables or with Bayesian networks. We study the data and combined complexity of threshold query answering, obtaining a detailed picture of the data complexity and the general, bounded-arity, and fixed-program combined complexity for the main classes of existential rules from Datalog+/- along with the considered probability models.

Note that annotations with Boolean events are widely used for encoding probabilities in probabilistic logical knowledge representation [15, 27] and are also known as data provenance and lineage [19, 15, 18, 27]. Note also that closely related to ontological data as studied in [23, 24] is exchanging incomplete databases as proposed in [3], which considers incomplete deterministic source and target databases in the data exchange problem and deterministic mappings. Also related is the approach to knowledge base exchange between deterministic *DL-Lite_{RDFS}* and *DL-Lite_R* ontologies in [2, 1].

The main contributions of this paper are briefly as follows.

- We define the problem of threshold query answering in the (probabilistic) ontological data exchange framework and study its data complexity, fixed-program-combined complexity, bounded-arity-combined complexity, and general combined complexity. For the complexity analysis, we consider the following main Datalog+/- languages: acyclic (full), weakly acyclic, linear (full), full, guarded (full), weakly guarded, sticky (full), and weakly sticky existential rules together with negative constraints.
- Besides considering different Datalog+/- languages, the complexity analysis for threshold query answering also investigates the impact of different probabilistic models on the complexity. More specifically, probabilities of annotations with Boolean events are specified via either pairwise independent random variables or

¹ University of Oxford, UK, email: thomas.lukasiewicz@cs.ox.ac.uk

² University of Oxford, UK, email: livia.predoiu@cs.ox.ac.uk; Otto-von-Guericke University, Magdeburg, Germany

with Bayesian networks. Furthermore, we investigate the impact of compact encodings for pairwise independent random variables: compact encodings via fully expressive Boolean formulas (elementary-event-independent) and via positive Boolean formulas without negation (PosBool and tuple-independence).

- We obtain a complete picture of the complexity of threshold query answering for the elementary-event-independent and the Bayesian-network encoding of probabilistic models. In particular, even in the data complexity, all considered Datalog+/- languages but weakly guarded existential rules (which have EXP-complete data complexity) for ontologies and mappings are PP-complete. In the fixed-program- and the bounded-arity-combined complexity, we obtain a complexity of PP^{NP} for Datalog+/- languages for which Boolean conjunctive query answering is NP-complete.
- For tuple-independent probabilistic databases and databases annotated with positive Boolean formulas, we obtain the same upper bounds as for the elementary-event-independent and the Bayesian-network encoding, and in many cases also the same lower bounds. In addition, we delineate the first-order rewritable classes of existential rules as an interesting case, where we obtain special-case tractability in the tuple-independent case in the data complexity, and where we conjecture a dichotomy of threshold query answering of either polynomial time or PP-hard queries.

The rest of this paper is organized as follows. In Section 2, we provide the preliminaries on Datalog+/- and the main terminology. In Section 3, we describe the ontological data exchange framework with probabilistic databases and with both deterministic mappings (Section 3.1) and probabilistic mappings (Section 3.2); we also present the probabilistic models and the compact encodings that we consider (Section 3.3). In Section 4, we define the threshold query entailment problem and provide its complexity analysis. Finally, in Section 5, we conclude with a summary and an outlook to future work.

2 PRELIMINARIES

We now recall the basics of Datalog+/- [8, 9], including especially relational databases, tuple-generating dependencies (TGDs, or existential rules), and (Boolean) conjunctive queries (BCQs).

We assume infinite sets of *constants* \mathbf{C} , (*labeled*) *nulls* \mathbf{N} , and *variables* \mathbf{V} . A *term* t is a constant, null, or variable. An *atom* has the form $p(t_1, \dots, t_n)$, where p is an n -ary predicate, and t_1, \dots, t_n are terms. Conjunctions of atoms are often identified with the sets of their atoms. An *instance* I is a (possibly infinite) set of atoms $p(\mathbf{t})$, where \mathbf{t} is a tuple of constants and nulls. A *database* D is a finite instance that contains only constants. A *homomorphism* is a mapping $h : \mathbf{C} \cup \mathbf{N} \cup \mathbf{V} \rightarrow \mathbf{C} \cup \mathbf{N} \cup \mathbf{V}$ that is the identity on \mathbf{C} . We assume familiarity with *conjunctive queries* (CQs). The answer to a CQ q over an instance I is denoted $q(I)$. A Boolean CQ (BCQ) q evaluates to *true* over I , denoted $I \models q$, if $q(I) \neq \emptyset$.

A *tuple-generating dependency* (TGD, or *existential rule*) σ is a first-order formula $\forall \mathbf{X} \forall \mathbf{Y} \varphi(\mathbf{X}, \mathbf{Y}) \rightarrow \exists \mathbf{Z} p(\mathbf{X}, \mathbf{Z})$, where $\mathbf{X} \cup \mathbf{Y} \cup \mathbf{Z} \subseteq \mathbf{V}$, $\varphi(\mathbf{X}, \mathbf{Y})$ is a conjunction of atoms, and $p(\mathbf{X}, \mathbf{Z})$ is an atom. We call $\varphi(\mathbf{X}, \mathbf{Y})$ the *body* of σ , denoted $body(\sigma)$, and $p(\mathbf{X}, \mathbf{Z})$ the *head* of σ , denoted $head(\sigma)$. A *copy TGD* is of the form $\forall \mathbf{X} p(\mathbf{X}) \rightarrow p(\mathbf{X})$, where $p(\mathbf{X})$ is an atom with the variables $\mathbf{X} \subseteq \mathbf{V}$ as pairwise different arguments. We consider only TGDs with a single atom in the head, but our results can be extended to TGDs with a conjunction of atoms in the head. An instance I *satisfies* σ , written $I \models \sigma$, if whenever there exists a homomorphism h

such that $h(\varphi(\mathbf{X}, \mathbf{Y})) \subseteq I$, then there exists $h' \supseteq h|_{\mathbf{X} \cup \mathbf{Y}}$, where $h|_{\mathbf{X} \cup \mathbf{Y}}$ is the restriction of h to $\mathbf{X} \cup \mathbf{Y}$, such that $h'(p(\mathbf{X}, \mathbf{Z})) \in I$. A *negative constraint* (NC) ν is a first-order formula $\forall \mathbf{X} \varphi(\mathbf{X}) \rightarrow \perp$, where $\mathbf{X} \subseteq \mathbf{V}$, $\varphi(\mathbf{X})$ is a conjunction of atoms, called the *body* of ν , denoted $body(\nu)$, and \perp denotes the truth constant *false*. An instance I *satisfies* ν , denoted $I \models \nu$, if there is no homomorphism h such that $h(\varphi(\mathbf{X})) \subseteq I$. Given a set Σ of TGDs and NCs, I *satisfies* Σ , denoted $I \models \Sigma$, if I satisfies each TGD and NC of Σ . For brevity, we omit the universal quantifiers in front of TGDs and NCs.

Given a database D and a set Σ of TGDs and NCs, the answers that we consider are those that are true in *all* models of D and Σ . Formally, the *models* of D and Σ , denoted $mods(D, \Sigma)$, is the set of instances $\{I \mid I \supseteq D \text{ and } I \models \Sigma\}$. The *answer* to a CQ q relative to D and Σ is defined as the set of tuples $ans(q, D, \Sigma) = \bigcap_{I \in mods(D, \Sigma)} \{t \mid t \in q(I)\}$. The answer to a BCQ q is *true*, denoted $D \cup \Sigma \models q$, if $ans(q, D, \Sigma) \neq \emptyset$. The problem of *CQ answering* is defined as follows: given a database D , a set Σ of TGDs and NCs, a CQ q , and a tuple of constants \mathbf{t} , decide whether $\mathbf{t} \in ans(q, D, \Sigma)$. It is well-known that such CQ answering can be reduced in LOGSPACE to BCQ answering, and we thus focus on BCQ answering only. Following Vardi's taxonomy [29], the *combined complexity* of BCQ answering is calculated by considering all the components, i.e., the database, the set of dependencies, and the query, as part of the input. The *bounded-arity combined complexity* (*ba-combined complexity*) is calculated by assuming that the arity of the underlying schema is bounded by an integer constant. Notice that in the context of description logics (DLs), whenever we refer to the combined complexity in fact we refer to the *ba-combined complexity*, since, by definition, the arity of the underlying schema is at most two. In the *data complexity*, only the database is part of the input; the *fixed-program combined complexity* (*fp-combined complexity*) is calculated by considering the set of TGDs and NCs as fixed.

3 ONTOLOGICAL DATA EXCHANGE

The source (resp., target) of the ontological data exchange problem that we consider here in this paper is a probabilistic database (resp., probabilistic instance), each relative to a deterministic ontology.

A *probabilistic database* (resp., *probabilistic instance*) over a schema \mathbf{S} is a probability space $Pr = (\mathcal{I}, \mu)$ such that \mathcal{I} is the set of all (possibly infinitely many) databases (resp., instances) over \mathbf{S} , and $\mu : \mathcal{I} \rightarrow [0, 1]$ is a function that satisfies (i) $\mu(I) > 0$ for only finitely many $I \in \mathcal{I}$ and (ii) $\sum_{I \in \mathcal{I}} \mu(I) = 1$.

We next provide the definitions of *deterministic* and *probabilistic ontological data exchange* (as proposed in [23, 24]).

3.1 Deterministic Ontological Data Exchange

Ontological data exchange formalizes data exchange from a probabilistic database relative to a source ontology Σ_s (consisting of TGDs and NCs) over a schema \mathbf{S} to a probabilistic target instance Pr_t relative to a target ontology Σ_t (consisting of TGDs and NCs) over a schema \mathbf{T} via a (source-to-target) mapping (also TGDs and NCs).

More specifically, an *ontological data exchange (ODE) problem* $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$ consists of (i) a source schema \mathbf{S} , (ii) a target schema \mathbf{T} disjoint from \mathbf{S} , (iii) a finite set Σ_s of TGDs and NCs over \mathbf{S} (called *source ontology*), (iv) a finite set Σ_t of TGDs and NCs over \mathbf{T} (called *target ontology*), and (v) a finite set Σ_{st} of TGDs and NCs σ over $\mathbf{S} \cup \mathbf{T}$ (called *source-to-target mapping*) such that $body(\sigma)$ and $head(\sigma)$ are defined over $\mathbf{S} \cup \mathbf{T}$ and \mathbf{T} , respectively.

Ontological data exchange with deterministic databases is based on defining a target instance J over \mathbf{T} as being a *solution* for a deterministic source database I over \mathbf{S} relative to the ODE problem $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$ iff $(I \cup J) \models \Sigma_s \cup \Sigma_t \cup \Sigma_{st}$. We denote by $Sol_{\mathcal{M}}$ the set of all such (I, J) .

Among the possible deterministic solutions J to a deterministic source database I relative to \mathcal{M} in $Sol_{\mathcal{M}}$, we prefer *universal* solutions, which are the most general ones carrying only the necessary information for data exchange, i.e., those that transfer only the source database along with the relevant implicit derivations via Σ_s to the target ontology. A universal solution can be homomorphically mapped to all other solutions leaving the constants unchanged. A deterministic target instance J over \mathbf{S} is a *universal solution* for a deterministic source database I over \mathbf{T} relative to a schema mapping \mathcal{M} iff (i) J is a solution, and (ii) for each solution J' for I relative to \mathcal{M} , there is a homomorphism $h: J \rightarrow J'$. We denote by $USol_{\mathcal{M}} (\subseteq Sol_{\mathcal{M}})$ the set of all pairs (I, J) of deterministic source databases I and target instances J such that J is a universal solution for I relative to \mathcal{M} .

When considering probabilistic databases and instances, a joint probability space Pr over the solution relation $Sol_{\mathcal{M}}$ and the universal solution relation $USol_{\mathcal{M}}$ must exist.

A probabilistic target instance $Pr_t = (\mathcal{J}, \mu_t)$ is a *probabilistic solution* (resp., *probabilistic universal solution*) for a probabilistic source database $Pr_s = (\mathcal{I}, \mu_s)$ relative to an ODE problem $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$ iff there exists a probability space $Pr = (\mathcal{I} \times \mathcal{J}, \mu)$ such that (i) the left and right marginals of Pr are Pr_s and Pr_t , respectively, i.e., (i.a) $\sum_{J \in \mathcal{J}} \mu(I, J) = \mu_s(I)$ for all $I \in \mathcal{I}$ and (i.b) $\sum_{I \in \mathcal{I}} \mu(I, J) = \mu_t(J)$ for all $J \in \mathcal{J}$, and (ii) $\mu(I, J) = 0$ for all $(I, J) \notin Sol_{\mathcal{M}}$ (resp., $(I, J) \notin USol_{\mathcal{M}}$). Intuitively, this says that all non-solutions (I, J) have probability zero, and that even if a solution exists, there still may be some zero-probability source database(s) without corresponding target instance(s).

Example 1 An ontological data exchange (ODE) problem $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$ is given by the source schema $\mathbf{S} = \{Researcher/2, ResearchArea/2, Publication/3\}$ (the number after the relation name denotes its arity), the target schema $\mathbf{T} = \{UResearchArea/3, Lecture/2\}$, the source ontology $\Sigma_s = \{\sigma_s, \nu_s\}$, the target ontology $\Sigma_t = \{\sigma_t, \nu_t\}$, and the mapping $\Sigma_{st} = \{\sigma_{st}, \nu_m\}$, where:

$$\begin{aligned} \sigma_s &: Publication(X, Y, Z) \rightarrow ResearchArea(X, Y), \\ \nu_s &: Researcher(X, Y) \wedge ResearchArea(X, Y) \rightarrow \perp, \\ \sigma_t &: UResearchArea(U, D, T) \rightarrow \exists Z Lecture(T, Z), \\ \nu_t &: Lecture(X, Y) \wedge Lecture(Y, X) \rightarrow \perp, \\ \sigma_{st} &: ResearchArea(N, T) \wedge \\ & \quad Researcher(N, U) \rightarrow \exists D UResearchArea(U, D, T), \\ \nu_m &: ResearchArea(N, T) \wedge UResearchArea(U, T, N) \rightarrow \perp. \end{aligned}$$

Given the probabilistic source database in Table 1, two possible probabilistic solution instances are shown in Table 1: $Pr_{t_1} = (\mathcal{J}_1, \mu_{t_1})$ and $Pr_{t_2} = (\mathcal{J}_2, \mu_{t_2})$. While both Pr_{t_1} and Pr_{t_2} are probabilistic solutions, only Pr_{t_1} is also a probabilistic universal solution. ■

For a deterministic source database D relative to an ODE problem $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$ and a CQ $q(\mathbf{X}) = \exists \mathbf{Y} \phi(\mathbf{X}, \mathbf{Y})$ over \mathbf{T} , the set of *answers* for q to D relative to \mathcal{M} is defined as $ans(q, D, \Sigma_s \cup \Sigma_t \cup \Sigma_{st})$. We now generalize this to probabilistic source databases relative to ODE problems and unions of CQs (UCQs).

A *union of CQs* (or *UCQ*) has the form $q(\mathbf{X}) = \bigvee_{i=1}^k \exists \mathbf{Y}_i \phi_i(\mathbf{X}, \mathbf{Y}_i)$, where each $\exists \mathbf{Y}_i \phi_i(\mathbf{X}, \mathbf{Y}_i)$ with $i \in \{1, \dots, k\}$ is a CQ with exactly the variables \mathbf{X} and \mathbf{Y}_i . Given an ODE problem $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$, probabilistic source database

$Pr_s = (\mathcal{I}, \mu_s)$, UCQ $q(\mathbf{X}) = \bigvee_{i=1}^k \exists \mathbf{Y}_i \phi_i(\mathbf{X}, \mathbf{Y}_i)$, and tuple \mathbf{t} (a ground instance of \mathbf{X} in q) over \mathbf{C} , the *confidence* of \mathbf{t} relative to q , denoted $conf_q(\mathbf{t})$, in Pr_s relative to \mathcal{M} is the infimum of $Pr_t(q(\mathbf{t}))$ subject to all probabilistic solutions Pr_t for Pr_s relative to \mathcal{M} . Here, $Pr_t(q(\mathbf{t}))$ for $Pr_t = (\mathcal{J}, \mu_t)$ is the sum of all $\mu_t(J)$ such that $q(\mathbf{t})$ evaluates to true in the instance $J \in \mathcal{J}$ (i.e., some BCQ $\exists \mathbf{Y}_i \phi_i(\mathbf{t}, \mathbf{Y}_i)$ with $i \in \{1, \dots, k\}$ evaluates to true in J).

3.2 Probabilistic Ontological Data Exchange

Probabilistic ontological data exchange extends deterministic ontological data exchange by turning the deterministic source-to-target mapping into a probabilistic source-to-target mapping, i.e., we now have a probability distribution over the set of all subsets of Σ_{st} .

A *probabilistic ontological data exchange (PODE) problem* $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st}, \mu_{st})$ consists of (i) a source schema \mathbf{S} , (ii) a target schema \mathbf{T} disjoint from \mathbf{S} , (iii) a finite set Σ_s of TGDs and NCs over \mathbf{S} (*source ontology*), (iv) a finite set Σ_t of TGDs and NCs over \mathbf{T} (*target ontology*), (v) a finite set Σ_{st} of TGDs and NCs σ over $\mathbf{S} \cup \mathbf{T}$, and (vi) a function $\mu_{st}: 2^{\Sigma_{st}} \rightarrow [0, 1]$ such that $\sum_{\Sigma' \subseteq \Sigma_{st}} \mu_{st}(\Sigma') = 1$ (*probabilistic (source-to-target) mapping*).

The notion of a probabilistic (universal) solution is defined as follows. A probabilistic target instance $Pr_t = (\mathcal{J}, \mu_t)$ is a *probabilistic solution* (resp., *probabilistic universal solution*) for a probabilistic source database $Pr_s = (\mathcal{I}, \mu_s)$ relative to a PODE problem $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st}, \mu_{st})$ iff there exists a probability space $Pr = (\mathcal{I} \times \mathcal{J} \times 2^{\Sigma_{st}}, \mu)$ such that: (i) the three marginals of μ are μ_s , μ_t , and μ_{st} , such that (i.a) $\sum_{J \in \mathcal{J}, \Sigma' \subseteq \Sigma_{st}} \mu(I, J, \Sigma') = \mu_s(I)$ for all $I \in \mathcal{I}$, (i.b) $\sum_{I \in \mathcal{I}, \Sigma' \subseteq \Sigma_{st}} \mu(I, J, \Sigma') = \mu_t(J)$ for all $J \in \mathcal{J}$, and (i.c) $\sum_{I \in \mathcal{I}, J \in \mathcal{J}} \mu(I, J, \Sigma') = \mu_{st}(\Sigma')$ for all $\Sigma' \subseteq \Sigma_{st}$; and (ii) $\mu(I, J, \Sigma') = 0$ for all $(I, J) \notin Sol_{(\mathbf{S}, \mathbf{T}, \Sigma')}$ (resp., $(I, J) \notin USol_{(\mathbf{S}, \mathbf{T}, \Sigma')}$).

Using probabilistic (universal) solutions for probabilistic source databases relative to PODE problems, the semantics of UCQs is lifted to PODE problems as follows. Given a PODE problem $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st}, \mu_{st})$, a probabilistic source database $Pr_s = (\mathcal{I}, \mu_s)$, a UCQ $q(\mathbf{X}) = \bigvee_{i=1}^k \exists \mathbf{Y}_i \phi_i(\mathbf{X}, \mathbf{Y}_i)$, and a tuple \mathbf{t} (a ground instance of \mathbf{X} in q) over \mathbf{C} , the *confidence* of \mathbf{t} relative to q , denoted $conf_q(\mathbf{t})$, in Pr_s relative to \mathcal{M} is the infimum of $Pr_t(q(\mathbf{t}))$ subject to all probabilistic solutions Pr_t for Pr_s relative to \mathcal{M} . Here, $Pr_t(q(\mathbf{t}))$ for $Pr_t = (\mathcal{J}, \mu_t)$ is the sum of all $\mu_t(J)$ such that $q(\mathbf{t})$ evaluates to true in the instance $J \in \mathcal{J}$.

3.3 Compact Encoding

We use a compact encoding of both probabilistic databases and probabilistic mappings, which is based on annotating database atoms, TGDs, and NCs by probabilistic Boolean events rather than explicitly specifying the whole probability space. That is, database atoms, TGDs, and NCs are annotated with Boolean combinations of elementary events, where every annotation describes when the annotated item is true and is associated with a probability. We first define general annotations and general annotated atoms.

Let e_1, \dots, e_n be $n \geq 1$ *elementary events*. A *world* w is a conjunction $\ell_1 \wedge \dots \wedge \ell_n$, where each $\ell_i, i \in \{1, \dots, n\}$, is either the elementary event e_i or its negation $\neg e_i$. An *annotation* λ is any Boolean combination of elementary events (i.e., all elementary events are annotations, and if λ_1 and λ_2 are annotations, then also $\neg \lambda_1$ and $\lambda_1 \wedge \lambda_2$); as usual, $\lambda_1 \vee \lambda_2$ abbreviates $\neg(\neg \lambda_1 \wedge \neg \lambda_2)$. An *annotated atom* has the form $a: \lambda$, where a is an atom, and λ is an annotation.

Possible source database facts		Derived source database facts		Probabilistic source database $Pr_s = (\mathcal{I}, \mu_s)$	
r_a	<i>Researcher</i> (Alice, UnivOfOxford)	a_{aml}	<i>ResearchArea</i> (Alice, ML)	$I_1 = \{r_a, r_p, p_{aml}, p_{pdb}, a_{aml}, a_{pai}\}$	0.3
r_p	<i>Researcher</i> (Paul, UnivOfOxford)	a_{adb}	<i>ResearchArea</i> (Alice, DB)	$I_2 = \{r_a, r_p, p_{aml}, p_{pai}, a_{aml}, a_{pai}\}$	0.3
p_{aml}	<i>Publication</i> (Alice, ML, JMLR)	a_{pdb}	<i>ResearchArea</i> (Paul, DB)	$I_3 = \{r_a, r_p, p_{adb}, p_{pai}, a_{adb}, a_{pai}\}$	0.2
p_{adb}	<i>Publication</i> (Alice, DB, TODS)	a_{pai}	<i>ResearchArea</i> (Paul, AI)	$I_4 = \{r_a, r_p, p_{adb}, p_{pdb}, a_{adb}, a_{pdb}\}$	0.1
p_{pdb}	<i>Publication</i> (Paul, DB, TODS)			$I_5 = \{r_a, p_{adb}, a_{adb}\}$	0.1
p_{pai}	<i>Publication</i> (Paul, AI, AIJ)				

Possible target instance facts		Prob. target instance $Pr_{t_1} = (\mathcal{J}_1, \mu_{t_1})$		Prob. target instance $Pr_{t_2} = (\mathcal{J}_2, \mu_{t_2})$	
u_{ml}	<i>UResearchArea</i> (UnivOfOxford, N_1 , ML)	$J_1 = \{u_{ml}, u_{db}, l_{ml}, l_{db}\}$	0.3	$J_5 = \{u_{ml}, u_{db}, l_{ml}, l_{db}\}$	0.35
u_{ai}	<i>UResearchArea</i> (UnivOfOxford, N_2 , AI)	$J_2 = \{u_{ml}, u_{ai}, l_{ml}, l_{ai}\}$	0.3	$J_6 = \{u_{ml}, u_{ai}, l_{ml}, l_{ai}\}$	0.2
u_{db}	<i>UResearchArea</i> (UnivOfOxford, N_3 , DB)	$J_3 = \{u_{ai}, u_{db}, l_{ai}, l_{db}\}$	0.2	$J_7 = \{u_{ml}, u_{ai}, u_{db}, l_{ml}, l_{ai}, l_{db}\}$	0.45
l_{ml}	<i>Lecture</i> (ML, N_4)	$J_4 = \{u_{db}, l_{db}\}$	0.2		
l_{ai}	<i>Lecture</i> (AI, N_5)				
l_{db}	<i>Lecture</i> (DB, N_6)				

Table 1. Probabilistic source database Pr_s and two probabilistic target instances Pr_{t_1} and Pr_{t_2} (with nulls N_i) for Example 1.

The compact encoding of probabilistic databases is then defined as follows. This encoding is also underlying our complexity analysis below. A set \mathbf{A} of annotated atoms along with a probability $\mu(w) \in [0, 1]$ for every world w *compactly encodes a probabilistic database* $Pr = (\mathcal{I}, \mu)$ whenever:

1. the probability μ of every annotation λ is the sum of the probabilities of all worlds in which λ is true, and
2. the probability μ of every subset-maximal database $\{a_1, \dots, a_m\} \in \mathcal{I}$ such that $\{a_1: \lambda_1, \dots, a_m: \lambda_m\} \subseteq \mathbf{A}$ for some annotations $\lambda_1, \dots, \lambda_m$ is the probability μ of $\lambda_1 \wedge \dots \wedge \lambda_m$ (and the probability μ of every other database in \mathcal{I} is 0).

We assume that all annotations are in disjunctive normal form (DNF), i.e., disjunctions of conjunctions of literals, and we consider the following four cases:

Elementary-event-independence: elementary events and their negations are pairwise probabilistically independent (i.e., the probability of worlds $\ell_1 \wedge \dots \wedge \ell_n$ of elementary events ($\ell_i = e_i$) and their negations ($\ell_i = \neg e_i$) is defined as $\prod_{i=1}^n \nu(\ell_i)$, where $\nu(\ell_i) = \mu(e_i)$ for $\ell_i = e_i$, and $\nu(\ell_i) = 1 - \mu(e_i)$ for $\ell_i = \neg e_i$);

PosBool: a special case of elementary-event-independence where all annotations are arbitrary many disjunctions of arbitrary many conjunctions of positive elementary events. Again, elementary events are pairwise probabilistically independent (i.e., the probability of worlds $\ell_1 \wedge \dots \wedge \ell_n$ of elementary events ($\ell_i = e_i$) is defined as $\prod_{i=1}^n \nu(\ell_i)$, where $\nu(\ell_i) = \mu(e_i)$);

Tuple-independence: special case of PosBool where annotations are elementary and worlds have positive probability.

Elementary-event-dependence encoded by a Bayesian network:

Here, we assume that the probability distributions for the underlying elementary events are given by a Bayesian network.

Note that in the tuple-independent case, annotations consist of as many elementary events as database atoms, and each database atom is annotated with a different single elementary event. The following example illustrates the encoding of a probabilistic database.

Example 2 In Table 2, an annotation encoding of a probabilistic source database is shown. It has four elementary events $e_1, e_2, e_3,$

and e_4 along with their probabilities $p(e_1) = 3/10, p(e_2) = 3/7, p(e_3) = 1/2,$ and $p(e_4) = 1/2,$ respectively. The encoding compactly represents the probabilistic source database in Table 1. ■

If also the mapping is probabilistic, then we use two disjoint sets of elementary events, one for encoding the probabilistic source database and the other one for the mapping. In this way, the probabilistic source database is independent from the probabilistic mapping. We now define the compact encoding of probabilistic mappings.

An *annotated* TGD (resp., NC) has the form $\sigma: \lambda$, where σ is a TGD (resp., NC), and λ is an annotation. A set Σ of annotated TGDs and NCs $\sigma: \lambda$ with $\sigma \in \Sigma_{st}$ along with a probability $\mu(w) \in [0, 1]$ for every world w *compactly encodes a probabilistic mapping* $\mu_{st}: 2^{\Sigma_{st}} \rightarrow [0, 1]$ whenever:

1. the probability μ of every annotation λ is the sum of the probabilities of all worlds in which λ is true, and
2. the probability μ_{st} of every subset-maximal $\{\sigma_1, \dots, \sigma_k\} \subseteq \Sigma_{st}$ such that $\{\sigma_1: \lambda_1, \dots, \sigma_k: \lambda_k\} \subseteq \Sigma$ for some annotations $\lambda_1, \dots, \lambda_k$ is the probability μ of $\lambda_1 \wedge \dots \wedge \lambda_k$ (and the probability μ_{st} of every other subset of Σ_{st} is 0).

4 COMPUTATIONAL COMPLEXITY

We now analyze the computational complexity of deciding threshold query answering for deterministic and probabilistic ontological data exchange problems. We also delineate some tractable special cases.

More precisely, we consider the following decision problem. We query the target ontology and ask whether a Boolean UCQ (BUCQ) is entailed with a probability of at least a given threshold $\tau \in [0, 1]$.

Definition 3 (Threshold Query Answering) Given a (P)ODE problem \mathcal{M} , a probabilistic source database Pr_s , a UCQ $q(\mathbf{X})$, a ground instance \mathbf{a} of \mathbf{X} over \mathbf{C} , and a threshold $\tau \in (0, 1]$, decide whether $\text{conf}_q(\mathbf{a}) \geq \tau$ in Pr_s relative to \mathcal{M} ; we then say that \mathbf{a} is a τ -*threshold answer* to $q(\mathbf{X})$. ■

W.l.o.g., the underlying ontologies and the mapping are in the same language, as the more expressive one always defines the complexity class of the ontological data exchange problem as a whole.

	Possible source database facts	Annotation
r_a	<i>Researcher</i> (Alice, UnivOfOxford)	true
r_p	<i>Researcher</i> (Paul, UnivOfOxford)	$e_1 \vee e_2 \vee e_3 \vee e_4$
p_{aml}	<i>Publication</i> (Alice, ML, JMLR)	$e_1 \vee e_2$
p_{adb}	<i>Publication</i> (Alice, DB, TODS)	$\neg e_1 \wedge \neg e_2$
p_{pdb}	<i>Publication</i> (Paul, DB, TODS)	$e_1 \vee (\neg e_2 \wedge \neg e_3 \wedge e_4)$
p_{pai}	<i>Publication</i> (Paul, AI, AIJ)	$(\neg e_1 \wedge e_2) \vee (\neg e_1 \wedge e_3)$

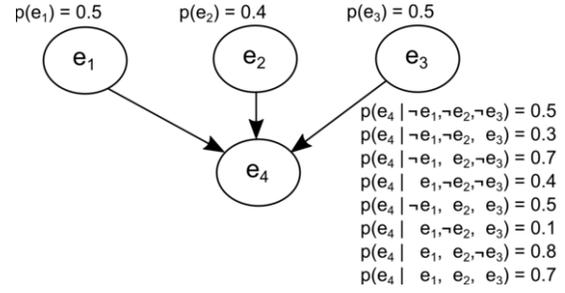


Table 2. Left: Annotation encoding of the probabilistic source database in Table 1. A possible elementary-event-independent interpretation is presented in Example 2. **Right:** An elementary-event-dependent interpretation represented by a polytree Bayesian network.

4.1 Complexity Classes

We assume some elementary background in complexity theory; see [20, 25]. We now briefly recall the complexity classes that we encounter in our complexity results below. The complexity class PSPACE (resp., P, EXP, 2EXP) contains all decision problems that can be solved in polynomial space (resp., polynomial, exponential, double exponential time) on a deterministic Turing machine, while the complexity classes NP and NEXP contain all decision problems that can be solved in polynomial and exponential time on a nondeterministic Turing machine, respectively, and coNP and coNEXP are their complementary classes, where “Yes” and “No” instances are interchanged. The complexity class AC^0 is the class of all languages that are decidable by uniform families of Boolean circuits of polynomial size and constant depth. Finally, the complexity class PP (resp., PP^{NP}) contains the problems decidable by a polynomial-time Turing machine (resp., polynomial-time Turing machine with an oracle for NP) that accepts an input iff the majority of its runs halt in an accepting state. The functional analog of PP is the well-known class #P, which is the class of all functions f (from strings to the nonnegative integers) for which there exists a nondeterministic polynomial-time Turing machine T such that for every input string w , it holds that $f(w)$ is the number of accepting runs of T on w . The above (decision) complexity classes and their inclusion relationships (which are all currently believed to be strict) are shown below:

$$AC^0 \subseteq P \subseteq NP, \text{coNP} \subseteq PP \subseteq PP^{NP}$$

$$\subseteq PSPACE \subseteq EXP \subseteq NEXP, \text{coNEXP} \subseteq 2EXP.$$

4.2 Decidability Paradigms

The main (syntactic) conditions on TGDs that guarantee the decidability of CQ answering are guardedness [7], stickiness [10], and acyclicity. Each of them has its “weak” counterpart: weak guardedness [7], weak stickiness [10], and weak acyclicity [14], respectively.

A TGD σ is *guarded*, if there exists an atom in its body that contains (or “guards”) all the body variables of σ . The class of guarded TGDs, denoted G, is defined as the family of all possible sets of guarded TGDs. A key subclass of guarded TGDs are the so-called *linear* TGDs with just one body atom (which is automatically a guard), and the corresponding class is denoted L. *Weakly guarded* TGDs extend guarded TGDs by requiring only “harmful” body variables to appear in the guard, and the associated class is denoted WG. More specifically, weakly guardedness requires guards to cover all variables occurring in affected positions only, where affected positions are positions in predicates that may contain some fresh labeled nulls that are generated during the construction of the chase [7].

It is easy to verify that $L \subseteq G \subseteq WG$. Note that guardedness and weak guardedness are generalized by the notions of frontier-guardedness and weak frontier-guardedness, respectively [4]. More precisely, *frontier-guardedness* relaxes the guardedness condition of TGDs by requiring the guard atom to contain only the frontier of a TGD, which is the set of all variables that appear in both the body and the head of the TGD. Generalizing frontier-guardedness, a set of TGDs is *weakly frontier-guarded*, if each TGD has an atom in its body that contains all affected variables from the frontier of the TGD.

Stickiness is inherently different from guardedness, and its central property can be described as follows: variables that appear more than once in a body (i.e., join variables) are always propagated (or “stick”) to the inferred atoms. A set of TGDs that enjoys the above property is called *sticky*, and its class is denoted S. Weak stickiness is a relaxation of stickiness where only “harmful” variables are taken into account. A set of TGDs that enjoys weak stickiness is *weakly sticky*, and the associated class is denoted WS. Observe that $S \subseteq WS$.

A set Σ of TGDs is *acyclic* if its predicate graph is acyclic, and the underlying class is denoted A. In fact, an acyclic set of TGDs can be seen as a nonrecursive set of TGDs. We say Σ is *weakly acyclic* if its dependency graph enjoys a certain acyclicity condition, which actually guarantees the existence of a finite canonical model; the associated class is denoted WA. Note that $A \subseteq WA \subseteq WS$.

Another key fragment of TGDs are *full* TGDs, i.e., TGDs without existentially quantified variables, and the corresponding class is denoted F. If we further assume that full TGDs enjoy linearity, guardedness, stickiness, or acyclicity, then we obtain the classes LF, GF, SF, and AF, respectively. Note that $F \subseteq WA$ and $F \subseteq WG$.

4.3 Overview of Complexity Results

Our complexity results for deciding threshold query answering in the elementary-event-independent and the Bayesian-network case for both ODE and PODE problems are summarized in Table 4, while our complexity results for deciding threshold query answering in the tuple-independent and the PosBool case are summarized in Table 5.

More precisely, in the elementary-event-independent and the Bayesian-network case (see Table 4), threshold query answering is complete for PP (resp., PP^{NP}) in the data (resp., *fp*-combined) complexity for all fragments of existential rules, except for WG_{\perp} , where it is complete for EXP. The *ba*-combined complexity in the elementary-event-independent and the Bayesian-network case is among PP^{NP} (for L, LF, AF, S, SF, F, and GF), EXP (for G and WG), NEXP (for A), and 2EXP (for WS and WA), while the combined complexity is among PSPACE (for L, LF, and AF), EXP (for S, SF, F, and GF), NEXP (for A), and 2EXP (for G, WG, WS, and WA).

Note that the same complexity results hold for other fragments of TGDs where standard BCQ answering has the same complexity, e.g., since standard BCQ answering in the frontier-guarded fragment is complete for P, 2EXP, and 2EXP (EXP, 2EXP, and 2EXP in the weakly frontier-guarded case) in the data, *ba*-combined, and combined complexity, respectively [4], in the elementary-event-independent and the Bayesian-network case, threshold BUCQ answering in the frontier-guarded fragment is complete for PP, 2EXP, and 2EXP (EXP, 2EXP, and 2EXP in the weakly frontier-guarded case) in the data, *ba*-combined, and combined complexity, respectively.

In the tuple-independent and the PosBool case, we obtain the same complexity classes, except that the matching lower bounds for the PP and PP^{NP} cases are still open; though, we were able to provide a #P completeness result for the function problem of computing the exact probability of a BUCQ for the PP cases in the data complexity.

Thus, as for the complexity cases above PSPACE, the complexity of threshold BUCQ answering coincides with the complexity of standard BCQ answering (see Table 3).

	Data	<i>fp</i> -comb.	<i>ba</i> -comb.	Comb.
L, LF, AF	in AC ⁰	NP	NP	PSPACE
G	P	NP	EXP	2EXP
WG	EXP	EXP	EXP	2EXP
S, SF	in AC ⁰	NP	NP	EXP
F, GF	P	NP	NP	EXP
A	in AC ⁰	NP	NEXP	NEXP
WS, WA	P	NP	2EXP	2EXP

Table 3. Complexity of BCQ answering [22]. All entries except for the “in” ones are completeness results; hardness for all entries but the *fp*-combined ones holds even for ground atomic BCQs.

	Data	<i>fp</i> -comb.	<i>ba</i> -comb.	Comb.
L, LF, AF	PP	PP ^{NP}	PP ^{NP}	PSPACE
G	PP	PP ^{NP}	EXP	2EXP
WG	EXP	EXP	EXP	2EXP
S, SF	PP	PP ^{NP}	PP ^{NP}	EXP
F, GF	PP	PP ^{NP}	PP ^{NP}	EXP
A	PP	PP ^{NP}	NEXP	NEXP
WS, WA	PP	PP ^{NP}	2EXP	2EXP

Table 4. Complexity of threshold query entailment (for both ODE and PODE problems) in the elementary-event-independent and the Bayesian-network case. All entries are completeness results.

	Data	<i>fp</i> -comb.	<i>ba</i> -comb.	Comb.
L, LF, AF	in PP	in PP ^{NP}	in PP ^{NP}	PSPACE
G	in PP	in PP ^{NP}	EXP	2EXP
WG	EXP	EXP	EXP	2EXP
S, SF	in PP	in PP ^{NP}	in PP ^{NP}	EXP
F, GF	in PP	in PP ^{NP}	in PP ^{NP}	EXP
A	in PP	in PP ^{NP}	NEXP	NEXP
WS, WA	in PP	in PP ^{NP}	2EXP	2EXP

Table 5. Complexity of threshold query entailment (for both ODE and PODE problems) for tuple-independent and PosBool annotated probabilistic databases. All entries except for the “in” ones are completeness results.

4.4 Deterministic Ontological Data Exchange

We first consider threshold query answering on target ontologies relative to an ODE problem in Datalog+/- languages where BCQ

answering is complete for $\mathcal{C} \supseteq$ PSPACE (see also Table 3 [22]). The following result shows that in these cases, threshold query answering is complete for \mathcal{C} in all four annotation cases. This proves all completeness entries in Tables 4 and 5 above PSPACE.

Theorem 4 Given (i) an ODE problem $\mathcal{M} = (\mathcal{S}, \mathcal{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$ such that $\Sigma_s \cup \Sigma_{st} \cup \Sigma_t$ belongs to a class of TGDs and NCs for which BCQ answering is complete for a deterministic complexity class $\mathcal{C} \supseteq$ PSPACE or nondeterministic complexity class $\mathcal{C} =$ NEXP, (ii) a probabilistic source database Pr_s w.r.t. Σ_s , (iii) a UCQ $q(\mathbf{X})$ over \mathcal{T} , (iv) a ground instance \mathbf{a} of \mathbf{X} over \mathcal{C} , and (v) $\tau \in (0, 1]$, deciding whether \mathbf{a} is a τ -threshold answer to $q(\mathbf{X})$ is complete for \mathcal{C} .

Proof (sketch). As for membership, with annotations consisting of n variables, we create a full valuation of an annotation at a time and compute its probability, which is in PSPACE. Then, we check whether $q(\mathbf{a})$ is true in the corresponding world, which is in \mathcal{C} . As we examine one valuation after another, we also add up its probability, if $q(\mathbf{a})$ is true in the corresponding world, until we reach the threshold τ , or we have examined all valuations. Hence, if standard BCQ answering belongs to the deterministic complexity class $\mathcal{C} \supseteq$ PSPACE, then the upper bound is \mathcal{C} . If standard BCQ answering belongs to the nondeterministic complexity class $\mathcal{C} =$ NEXP, we guess a set of worlds where $q(\mathbf{a})$ evaluates to true, and verify this guess. Since both steps are in NEXP, the computation is overall also in NEXP.

Hardness is shown by a reduction from BCQ answering in Datalog+/- ontologies to threshold query answering. Consider a source schema \mathcal{S} and a target schema \mathcal{T} , as well as a set of n source relations $R_{S,i}$, $1 \leq i \leq n$, and a set of n target relations $R_{T,i}$, $1 \leq i \leq n$, where each $R_{S,i}$ has the same arity as $R_{T,i}$. We also assume a source database with each tuple having the probability 1. The target database is empty. By employing a simple copy mapping from the source database to the target database $R_{S,i}(x_1, \dots, x_{n_i}) \rightarrow R_{T,i}(y_1, \dots, y_{n_i})$, and Σ_s being empty, while Σ_t contains a set of TGDs and NCs in the language we consider. Then, a ground instance \mathbf{a} is a τ -threshold answer to $q(\mathbf{X})$ with $\tau = 1$ in the ODE problem iff the BCQ $q(\mathbf{a})$ is true for the ontology $\Sigma_s \cup \Sigma_{st} \cup \Sigma_t$ with the source database. \square

The next result shows that in the elementary-event-independent and in the Bayesian-network case, if the language of the ontologies and the mappings belongs to a class of TGDs and NCs for which standard BCQ answering is complete for NP in the *fp*-combined and the *ba*-combined complexity, then threshold query answering is complete for PP^{NP} in the *fp*-combined and the *ba*-combined complexity. This proves all PP^{NP} completeness entries in Table 4.

Theorem 5 Given (i) an ODE problem $\mathcal{M} = (\mathcal{S}, \mathcal{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$ such that $\Sigma_s \cup \Sigma_{st} \cup \Sigma_t$ belongs to a class of TGDs and NCs for which BCQ answering is NP-complete in the *fp*-combined and the *ba*-combined complexity, and which includes copy TGDs, (ii) an elementary-event-independent or a Bayesian-network-annotated probabilistic source database Pr_s relative to Σ_s , (iii) a UCQ $q(\mathbf{X})$ over \mathcal{T} , (iv) a ground instance \mathbf{a} of \mathbf{X} over \mathcal{C} , and (v) $\tau \in (0, 1]$, deciding whether \mathbf{a} is a τ -threshold answer to $q(\mathbf{X})$ is PP^{NP}-complete in the *fp*-combined and the *ba*-combined complexity.

Proof (sketch). As for membership in PP^{NP}, intuitively, we first create multiples of each world (which then correspond to the nondeterministic branches of a Turing machine), so that the probability distribution over all thus generated worlds is the uniform distribution. Then,

for thresholds properly below (resp., above) 0.5, we introduce artificial success (resp., failure) worlds (which correspond to other non-deterministic success (resp., failure) branches of a Turing machine), so that satisfying the resulting threshold corresponds to having a majority of success worlds. We thus only have to verify whether for the majority of the worlds, the query evaluates to true. As query evaluation is in NP, the computation is overall in PP^{NP} .

Hardness for PP^{NP} holds by a reduction from the PP^{NP} -complete problem of deciding, given $n \geq 0$ and a quantified Boolean formula (QBF) $\Phi = \forall x_1 \dots x_l \exists y_1 \dots y_m \phi_1 \wedge \phi_2 \wedge \dots \wedge \phi_k$, where $l, m, k \geq 1$, and every ϕ_i is a disjunction of literals over $x_1, \dots, x_l, y_1, \dots, y_m$, whether there are at least n truth assignments τ to x_1, \dots, x_l such that $\exists y_1 \dots y_m \tau(\phi_1) \wedge \tau(\phi_2) \wedge \dots \wedge \tau(\phi_k)$ is satisfiable [30]. W.l.o.g., every ϕ_i is a disjunction of three literals over $x_1, \dots, x_l, y_1, \dots, y_m$. For every ϕ_i , let u_i, v_i, w_i denote its variables, and let the source database contain the probabilistic facts $r(i, \nu(u_i), \nu(v_i), \nu(w_i))$: ψ such that (i) ν is a truth assignment to u_i, v_i, w_i that satisfies ϕ_i , and (ii) ψ is a conjunction of literals over $\{x_1, \dots, x_l\} \cap \{u_i, v_i, w_i\}$ that exactly represents the restriction of ν to x_1, \dots, x_l . Here, every variable x_i has the probability $\mu(x_i) = 0.5$, and thus every world over x_1, \dots, x_l has the probability 2^{-l} . Let Σ_s and Σ_t be empty, and let Σ_{st} contain the copy mapping rule $r(I, U, V, W) \rightarrow r'(I, U, V, W)$. Then, $\exists x_1 \dots x_l \exists y_1 \dots y_m r'(1, u_1, v_1, w_1) \wedge r'(2, u_2, v_2, w_2) \wedge \dots \wedge r'(k, u_k, v_k, w_k)$ holds with the probability of at least $n \cdot 2^{-l}$ iff there are at least n truth assignments τ to x_1, \dots, x_l such that $\exists y_1 \dots y_m \tau(\phi_1) \wedge \tau(\phi_2) \wedge \dots \wedge \tau(\phi_k)$ is satisfiable. Observe that the above reduction can clearly be done in polynomial time in the size of Φ , and that the set of TGDs and NCs is fixed (*fp*-combined case), and that the arity of all predicates is 4 (*ba*-combined case). \square

The next theorem shows that in the elementary-event-independent and in the Bayesian-network case, if the language of the ontologies and the mappings belongs to a class of TGDs and NCs for which standard BCQ answering is in P in the data complexity, then threshold query answering is complete for PP in the data complexity. This proves all PP completeness entries in Table 4.

Theorem 6 *Given (i) an ODE problem $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$ such that $\Sigma_s \cup \Sigma_{st} \cup \Sigma_t$ belongs to a class of TGDs and NCs for which BCQ answering is in P in the data complexity, and which includes copy TGDs, (ii) an elementary-event-independent or a Bayesian-network-annotated probabilistic source database Pr_s relative to Σ_s , (iii) a UCQ $q(\mathbf{X})$ over \mathbf{T} , (iv) a ground instance \mathbf{a} of \mathbf{X} over \mathbf{C} , and (v) $\tau \in (0, 1]$, deciding whether \mathbf{a} is a τ -threshold answer to $q(\mathbf{X})$ is PP-complete in the data complexity.*

Proof (sketch). The PP-membership proof is similar to the PP^{NP} -membership proof for Theorem 5, except that the standard BCQ query answering oracle is now in P and not in NP.

Hardness for PP holds by a reduction from the PP-complete #3SAT ($\geq 2^{n/2}$) decision problem [5]: given a 3CNF formula ϕ with n variables, does ϕ have at least $2^{n/2}$ satisfying truth assignments? Given an instance ϕ of #3SAT ($\geq 2^{n/2}$), we construct an ODE problem as follows. We assume a source database with a source schema \mathbf{S} consisting of a binary relation symbol R_S and to contain a single tuple with ϕ as annotation in 3CNF. Each of the n variables of the annotation has the probability 0.5. The schema \mathbf{T} of the target database consists of a binary relation symbol R_T ; the target database is empty. The sets Σ_s and Σ_t are empty as well. The set Σ_{st} contains the following mapping rule $R_S(x, y) \rightarrow R_T(x, y)$, which copies the tuple

of the source relation to the target relation. Then, the transferred tuple has a probability higher than $2^{-n/2}$ iff the 3CNF formula has at least $2^{n/2}$ satisfying truth assignments, which proves PP-hardness. \square

The following theorem says that in the tuple-independent and the PosBool case, if the language of the ontologies and the mappings belongs to a class of TGDs and NCs for which standard BCQ answering is complete for NP in the *fp*-combined and the *ba*-combined complexity, then threshold query answering is in PP^{NP} in the *fp*-combined and the *ba*-combined complexity. The theorem follows from the membership in PP^{NP} of the problem in the more general elementary-event-independent case (see Theorem 5). This result proves all PP^{NP} membership entries in Table 5.

Theorem 7 *Given (i) an ODE problem $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$ such that $\Sigma_s \cup \Sigma_{st} \cup \Sigma_t$ belongs to a class of TGDs and NCs for which BCQ answering is in NP in the *fp*-combined and the *ba*-combined complexity, (ii) a tuple-independent or PosBool probabilistic source database Pr_s relative to Σ_s , (iii) a UCQ $q(\mathbf{X})$ over \mathbf{T} , (iv) a ground instance \mathbf{a} of \mathbf{X} over \mathbf{C} , and (v) $\tau \in (0, 1]$, deciding whether \mathbf{a} is a τ -threshold answer to $q(\mathbf{X})$ is in PP^{NP} in the *fp*-combined and the *ba*-combined complexity.*

The next theorem shows that in the tuple-independent and the PosBool case, if the language of the ontologies and the mappings belongs to a class of TGDs and NCs for which standard BCQ answering is in P in the data complexity, then threshold query answering is in PP in the data complexity, proving all PP membership entries in Table 5. The theorem also shows that the function problem of computing the exact probability is #P-complete in the data complexity for ontologies and mappings encoded as NCs and full and guarded TGDs (i.e., this #P-completeness holds for the classes G, F, GF, WS, and WA).

Theorem 8 *Given (i) an ODE problem $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$ such that $\Sigma_s \cup \Sigma_{st} \cup \Sigma_t$ belongs to a class of TGDs and NCs for which BCQ answering is in P in the data complexity, (ii) a tuple-independent or PosBool probabilistic source database Pr_s relative to Σ_s , (iii) a UCQ $q(\mathbf{X})$ over \mathbf{T} , (iv) a ground instance \mathbf{a} of \mathbf{X} over \mathbf{C} , and (v) $\tau \in (0, 1]$, deciding whether \mathbf{a} is a τ -threshold answer to $q(\mathbf{X})$ is in PP in the data complexity. Given (i) to (iv), where $\Sigma_s \cup \Sigma_{st} \cup \Sigma_t$ is full and guarded, computing $\text{conf}_q(\mathbf{a})$ in Pr_s relative to \mathcal{M} is #P-complete in the data complexity.*

Proof (sketch). Membership in PP is immediate by the membership in PP of the problem in the more general elementary-event-independent case (see Theorem 6). Membership in #P of the function problem follows by a similar line of argumentation.

Hardness for #P follows from a reduction from the #P-complete monotone 2SAT problem [28]: given a Boolean formula $\phi = \phi_1 \wedge \phi_2 \wedge \dots \wedge \phi_k$ over the variables x_1, x_2, \dots, x_l , where each ϕ_i is a disjunction of two variables, compute the number of truth assignments to x_1, x_2, \dots, x_l that satisfy ϕ . For every variable x_i , let the source database contain the probabilistic facts $s(x_i)$: x_i , along with the probability $\mu(x_i) = 0.5$. Furthermore, for every ϕ_i , let u_i and v_i denote its variables, and let the source database contain the probabilistic facts $r(i, u_i)$: $e_{r(i, u_i)}$ and $r(i, v_i)$: $e_{r(i, v_i)}$, along with the probabilities $\mu(e_{r(i, u_i)}) = 1$ and $\mu(e_{r(i, v_i)}) = 1$. For every $i, j \in \{1, \dots, k\}$ with $i < j$, let the source database contain the probabilistic facts $\text{succ}(i, j)$: $e_{\text{succ}(i, j)}$, along with the probability $\mu(e_{\text{succ}(i, j)}) = 1$. Furthermore, let the source database contain the probabilistic fact $\text{max}(k)$: $e_{\text{max}(k)}$, along with the probability $\mu(e_{\text{max}(k)}) = 1$. Let Σ_t be empty, and let Σ_s contain the full and

guarded rules $r(I, X) \wedge s(X) \rightarrow t(I)$, $max(X) \wedge t(X) \rightarrow t'(X)$, and $t'(I) \wedge succ(J, I) \wedge r(J) \rightarrow t'(J)$. Let Σ_{st} contain the mapping rule $t'(X) \rightarrow t''(X)$. Observe that the query and the set of TGDs and NCs are both fixed, and that the TGDs are full and guarded. Furthermore, $t''(1)$ holds with the probability m iff ϕ has $m \cdot 2^l$ satisfying truth assignments τ to x_1, \dots, x_l . \square

The following theorem shows that in the tuple-independent and the PosBool case, if the language of the ontologies and the mappings is in a class of TGDs and NCs where standard BCQ answering is in AC^0 in the data complexity, then the function problem of computing the exact probability is also $\#P$ -complete in the data complexity (i.e., this $\#P$ -completeness holds for the classes L, LF, AF, S, SF, and A).

Theorem 9 *Given (i) an ODE problem $\mathcal{M} = (\mathcal{S}, \mathcal{T}, \Sigma_s, \Sigma_t, \Sigma_{st})$ such that $\Sigma_s \cup \Sigma_{st} \cup \Sigma_t$ is in a class of TGDs and NCs for which BCQ answering is in AC^0 , (ii) a tuple-independent or PosBool probabilistic source database Pr_s relative to Σ_s , (iii) a UCQ $q(\mathbf{X})$ over \mathcal{T} , and (iv) a ground instance \mathbf{a} of \mathbf{X} over \mathcal{C} , computing $conf_q(\mathbf{a})$ in Pr_s relative to \mathcal{M} is $\#P$ -complete in the data complexity.*

Proof (sketch). Membership in $\#P$ of the function problem follows by a similar line of argumentation as in the proof of Theorem 8.

Hardness for $\#P$ follows from the observation that the $\#P$ -complete evaluation of unsafe UCQs in [12] is actually a special case of our function problem of computing the probability of a query. \square

4.5 Tractable Cases

In the tuple-independent case, if the language of the ontologies and the mappings is in a class of TGDs and NCs where standard BCQ answering is in AC^0 in the data complexity, we arrive at a tuple-independent source database with a rewritten first-order UCQ q_Σ (e.g., by applying the algorithm XRewrite from [17] to the initial query $q(\mathbf{X})$ on the target database and $\Sigma = \Sigma_t \cup \Sigma_{st} \cup \Sigma_s$). This is now exactly the problem handled in [27, 12], which has important tractable cases (called *safe queries*): they are those queries that can be computed by extensional query evaluation, i.e., solely by the query syntax or its annotation. Extensional query evaluation consists of extending the relational operators with a probabilistic semantics. An example of tractable queries are hierarchical queries. A query is hierarchical, if for each existential sub-query expression $\exists x q_{sub}$, the quantified variable x occurs in all atoms of q_{sub} . Every hierarchical query has a read-once Boolean formula as query annotation. Further examples for tractable cases can be found in [27, 12]. In particular, [27] delineates six syntactical rules to check whether UCQs are tractable, and calls a UCQ q R_6 -safe, if it adheres to one of them; if a UCQ is not R_6 -safe, it is $\#P$ -hard. In [27, 12], a polynomial algorithm for R_6 -safe UCQs is given involving Möbius' inversion function, which ensures that it also covers UCQs, and not just CQs. If the query is not R_6 -safe, it may be approximated.

4.6 Probabilistic Ontological Data Exchange

All the results in Theorems 4, 5, 6, 7, 8 and 9 carry over to PODE problems. Clearly, the hardness results carry over immediately, as deterministic ontological data exchange is a special case of probabilistic ontological data exchange. As for the membership results, we also have to consider and iterate through the worlds for the probabilistic mapping. However, iterating through these worlds in addition to the worlds for the probabilistic source database does not increase the overall complexity of threshold query answering.

5 SUMMARY AND OUTLOOK

We have studied the impact of different probabilistic models and compact encodings on the computational complexity of the problem of threshold query answering in ontological data exchange with the main Datalog+/- languages for representing the source and target ontologies as well as the mappings. We have considered the data complexity, the *fp*-combined complexity, the *ba*-combined complexity, and the combined complexity. We have provided a complete picture for the elementary-event-independent and the Bayesian network case with a compact encoding with Boolean formulas. For tuple-independent and PosBool-annotated probabilistic databases, we have provided either completeness results or upper bounds.

While ontology and mapping languages with BCQ answering complexity above PSPACE dominate the complexity of threshold query answering, for ontology languages with BCQ answering complexity below PSPACE, we obtain interesting results, one of them being a potential dichotomy of either an upper bound of P or PP-hardness in the data complexity for threshold query answering in the first-order-rewritable cases. A similar dichotomy of either an upper bound of P or $\#P$ -hardness in the data complexity exists for query answering in probabilistic databases (see [27, 12]) and has been lifted to OWL QL in [21], which is also first-order-rewritable. Note that OWL QL corresponds to *DL-Lite_R*, which is strictly less expressive than most of the Datalog+/- languages considered here.

Another interesting result is PP-completeness for threshold query answering in the data complexity for most of the Datalog+/- languages that we considered for the elementary-event-independent and the Bayesian network case. Furthermore, we have obtained PP^{NP} -completeness in the *fp*-combined and *ba*-combined complexity for the elementary-event-independent and the Bayesian-network case for threshold query answering for several Datalog+/- languages. This complexity class is mostly known from a blow-up in succinct representations of problems (see [30]).

There is no related work on threshold query entailment in probabilistic data exchange or probabilistic ontological data exchange. Most other works on probabilistic databases and ontologies do not consider threshold query answering, but consider the function problem of query answering, mostly in the data complexity, such as the works [27, 12, 21]. Perhaps closest to our work is [11], where threshold query answering has been studied under the name “probabilistic query entailment” for the ontology language \mathcal{EL} , annotated with annotations that are related to our Bayesian-network annotations. Our work goes beyond that, as most of the ontology languages that we are considering here are strictly more expressive than \mathcal{EL} . In addition, we also consider a probabilistic model where the events are elementary-event-independent annotations and PosBool annotations, as well as tuple-independent and mapping-independent annotations. In addition, our complexity analysis for threshold query entailment with Bayesian networks as probabilistic model contains also completeness results for the languages we considered, while this is not the case in [11]. Threshold query answering with probabilistic ontologies has also been studied in [16], but the probabilistic uncertainty models used there are Markov logic networks.

An interesting topic for future research is a more detailed analysis of the tractable case for threshold query answering, especially also in combination with ontological query rewriting. Based on the complexity results and membership proofs of this paper, another topic for future research is to explore whether there are other (special-case or approximation) tractable cases for threshold query answering.

ACKNOWLEDGEMENTS

This work was supported by the UK EPSRC grants EP/J008346/1, EP/L012138/1, and EP/M025268/1, and the EU FP7 Marie-Curie IE Fellowship PRODIMA.

REFERENCES

- [1] Marcelo Arenas, Elena Botoeva, Diego Calvanese, and Vladislav Ryzhikov, ‘Exchanging OWL 2 QL knowledge bases’, in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013)*, (2013).
- [2] Marcelo Arenas, Elena Botoeva, Diego Calvanese, Vladislav Ryzhikov, and Evgeny Sherkhonov, ‘Exchanging description logic knowledge bases’, in *Proceedings of 13th International Conference on the Principles of Knowledge Representation and Reasoning (KR 2012)*, pp. 563–567, (2012).
- [3] Marcelo Arenas, Jorge Pérez, and Juan Reutter, ‘Data exchange beyond complete data’, *Journal of the ACM*, **60**(4), (2013).
- [4] Jean-François Baget, Marie-Laure Mugnier, Sebastian Rudolph, and Michaël Thomazo, ‘Walking the complexity lines for generalized guarded existential rules’, in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pp. 712–717. IJCAI/AAAI Press, (2011).
- [5] Delbert D. Bailey, Victor Dalmau, and Phokion G. Kolaitis, ‘Phase transitions of PP-complete satisfiability problems’, *Discrete Applied Mathematics*, **155**(12), (2007).
- [6] Pablo Barcelo, ‘Logical foundations of relational data exchange’, *SIGMOD Record*, **38**(1), 49–58, (2009).
- [7] Andrea Cali, Georg Gottlob, and Michael Kifer, ‘Taming the infinite chase: Query answering under expressive relational constraints’, *Journal of Artificial Intelligence Research*, **48**, (2013).
- [8] Andrea Cali, Georg Gottlob, and Thomas Lukasiewicz, ‘A general Datalog-based framework for tractable query answering over ontologies’, *Journal of Web Semantics*, **14**, 57–83, (2012).
- [9] Andrea Cali, Georg Gottlob, Thomas Lukasiewicz, Bruno Marnette, and Andreas Pieris, ‘Datalog+/-: A family of logical knowledge representation and query languages for new applications’, in *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science (LICS 2010)*, (2010).
- [10] Andrea Cali, Georg Gottlob, and Andreas Pieris, ‘Towards more expressive ontology languages: The query answering problem’, *Artificial Intelligence*, **139**, (2012).
- [11] Ismail Ilkan Ceylan and Rafael Penaloza, ‘Probabilistic query answering in the Bayesian description logic \mathcal{BEL} ’, in *Proceedings of the 9th International Conference on Scalable Uncertainty Management (SUM 2015)*, (2015).
- [12] Nilesh Dalvi and Dan Suciu, ‘The dichotomy of probabilistic inference for union of conjunctive queries’, *Journal of the ACM*, **59**(6), (2012).
- [13] Ronald Fagin, Benny Kimelfeld, and Phokion G. Kolaitis, ‘Probabilistic data exchange’, *Journal of the ACM*, **58**(4), (2011).
- [14] Ronald Fagin, Phokion G. Kolaitis, Renee J. Miller, and Lucian Popa, ‘Data exchange: Semantics and query answering’, *Theoretical Computer Science*, **336**(1), 89–124, (2005).
- [15] Norbert Fuhr and Thomas Rölleke, ‘A probabilistic relational algebra for the integration of information retrieval and database systems’, *ACM Transactions on Information Systems*, **15**(1), 32–66, (1997).
- [16] Georg Gottlob, Thomas Lukasiewicz, Maria Vanina Martinez, and Gerardo I. Simari, ‘Query answering under probabilistic uncertainty in Datalog+/- ontologies’, *Annals of Mathematics and Artificial Intelligence*, **69**(1), (2013).
- [17] Georg Gottlob, Giorgio Orsi, and Andreas Pieris, ‘Query rewriting and optimization for ontological databases’, *ACM Transactions on Database Systems*, **39**(3), (2014).
- [18] Todd J. Green, Grigoris Karvounarakis, and Val Tannen, ‘Provenance semirings’, in *Proceedings of the 26th Symposium on Principles of Database Systems (PODS 2007)*, (2007).
- [19] Tomasz Imielinski and Witold Lipski, ‘Incomplete information in relational databases’, *Journal of the ACM*, **31**(4), 761–791, (1984).
- [20] David S. Johnson, ‘A catalog of complexity classes’, in *Handbook of Theoretical Computer Science*, 67–161, MIT Press, (1990).
- [21] Jean Christoph Jung and Carsten Lutz, ‘Ontology-based access to probabilistic data with OWL QL’, in *Proceedings of the 11th International Semantic Web Conference (ISWC 2012)*, (2012).
- [22] Thomas Lukasiewicz, Maria Vanina Martinez, Andreas Pieris, and Gerardo I. Simari, ‘From classical to consistent query answering under existential rules’, in *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI 2015)*, (2015).
- [23] Thomas Lukasiewicz, Maria Vanina Martinez, Livia Predoiu, and Gerardo I. Simari, ‘Existential rules and Bayesian networks for probabilistic ontological data exchange’, in *Rule Technologies: Foundations, Tools, and Applications*, pp. 294–310, (2015).
- [24] Thomas Lukasiewicz, Maria Vanina Martinez, Livia Predoiu, and Gerardo I. Simari, ‘Basic probabilistic ontological data exchange with existential rules’, in *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI 2016)*, (2016).
- [25] Christos H. Papadimitriou, *Computational Complexity*, Addison Wesley Longman, 1994.
- [26] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati, ‘Linking data to ontologies’, *Journal on Data Semantics*, **10**, 133–173, (2008).
- [27] Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch, *Probabilistic Databases*, Morgan & Claypool, 2011.
- [28] Leslie G. Valiant, ‘The complexity of enumeration and reliability problems’, *SIAM Journal on Computing*, **8**(3), (1979).
- [29] Moshe Y. Vardi, ‘The complexity of relational query languages (extended abstract)’, in *Proceedings of the 14th Annual ACM Symposium on Theory of Computing (STOC 1982)*, pp. 137–146, (1982).
- [30] Klaus W. Wagner, ‘The complexity of combinatorial problems with succinct input representation’, *Acta Informatica*, **23**(3), 325–356, (1986).

Markov Logic Networks with Numerical Constraints

Melisachew Wudage Chekol and Jakob Huber and Christian Meilicke and Heiner Stuckenschmidt¹

Abstract. Markov logic networks (MLNs) have proven to be useful tools for reasoning about uncertainty in complex knowledge bases. In this paper, we extend MLNs with numerical constraints and present an efficient implementation in terms of a cutting plane method. This extension is useful for reasoning over uncertain temporal data. To show the applicability of this extension, we enrich log-linear description logics (DLs) with concrete domains (datatypes). Thereby, allowing to reason over weighted DLs with datatypes. Moreover, we use the resulting formalism to reason about temporal assertions in DBpedia, thus illustrating its practical use.

1 Motivation

Recent advances in data mining and information extraction have paved the way for the automatic construction of knowledge bases (KBs) from different sources, for instance, the NELL KB [26]. Often, the extraction tools used to construct such KBs produce weighted (or probabilistic) facts, due to the awareness that the implemented techniques cannot guarantee the completeness, correctness or consistency of the generated facts. Moreover, a large fraction of these facts can be temporal and may contain concrete data values, for example dates, times, latitudes/longitudes, numerical values measured in different units, and so on. Besides, the set of rules used to consolidate the KB will also be a set of weighted and unweighted rules with datatypes (numerical constraints). In this work, we propose an extension of Markov logic networks (MLNs) [32] that supports reasoning on numerical constraints.

We extend an MLNs inference approach (cutting plane [33]) to handle numerical constraints. This is difficult because numerical constraints may be infinite or continuous and reasoning about comparisons of two constraints is difficult to handle natively in MLNs. As an example, consider the following hard constraints which express: (1) a person whose age is between 13 and 19 is a teenager, and (2) a valid life span of a person is between 0 and 150 years:

$$\begin{aligned} 1 \forall x, y: & Person(x) \wedge age(x, y) \wedge y \geq 13 \wedge y \leq 19 \Rightarrow Teen(x) \\ 2 \forall p, bd, dd: & bdate(p, bd) \wedge ddate(p, dd) \Rightarrow lifeSpan(bd, dd), \\ & lifeSpan(bd, dd) = ((dd - bd) > 0) \wedge ((dd - bd) \leq 150) \end{aligned}$$

To the best of our knowledge, MLNs do not support numerical constraints (like the rules in the above example). Therefore, we propose an extended cutting plane approach to MLNs inference to introduce just the necessary constraint violations for the current MAP hypothesis taking also violated numerical constraints into account.

The proposed extension is useful for reasoning over uncertain temporal KBs as well as weighted Horn DLs with concrete domains. In this paper, we show that this extension is powerful enough to reason

over log-linear description logics (DLs) with concrete domains. Log-linear DLs [27] have been proposed as a way to combine tractable DLs with uncertainty reasoning while preserving the semantics of the DL part. They have also proven to be useful in situations where conflicting information from different sources has to be integrated into a coherent model. Successful applications include ontology matching [31] and information fusion [27]. So far the applicability of the combined model has been limited by the fact that it only supported reasoning on the schema level. In our work, we extend log-linear DLs to overcome its restriction to schema-level reasoning by introducing instances and numerical concrete domains (datatypes).

In DL, a concrete domain (CD) is a construct that can be used to define new classes by specifying constraints on attributes that have literal values (as opposed to relationships to other abstract entities). For instance, the axiom $Teenager \equiv Person \sqcap \geq_{13}(age) \sqcap \leq_{19}(age)$ defines a teenager as a person whose age is at least 13 and at most 19. While CDs, also referred to as *datatypes*, are a well-studied construct in classical DL (see for instance [22, 24]), this is barely the case in the probabilistic or log-linear extensions of DLs that have been proposed more recently (e.g. [7, 16, 11, 27, 21, 6, 35]). The contribution of this paper is the following:

- an extension of MLN with numerical constraints (MLN_{NC}),
- an extension of log-linear \mathcal{EL} with nominals and CDs by making use of MLN_{NC},
- an application of MLN_{NC} for reasoning about a specific sub-domain in the DBpedia knowledge base that contains numerical datatypes.
- an application of MLN_{NC} for debugging weighted temporal KBs.

Outline: In the next section we present the preliminaries: log-linear models, the lightweight ontology language \mathcal{EL}^{++} , and log-linear \mathcal{EL} obtained by combining log-linear models with \mathcal{EL}^{++} . In order to reason over uncertain temporal KBs, in Section 3, we extend MLNs with numerical constraints. In order to demonstrate additional benefits of this framework, we (1) extend log-linear \mathcal{EL} with CDs (in Section 4 and Section 5), and (2) present experimental results on debugging temporal KBs and computing the MAP state of an uncertain KB which contains datatypes (Section 6). Finally, we summarize the related work in Section 6, before making concluding remarks in Section 7.

2 Preliminaries

In this section, we present a brief summary of: Log-linear models, \mathcal{EL}^{++} , and log-linear \mathcal{EL} . For a detailed discussion on these subjects, we refer the reader to [2, 32, 34, 27] and the references therein.

¹ Data and Web Science Group, University of Mannheim, 68161 Mannheim, Germany, email: mel|jakob|christian|heiner@informatik.uni-mannheim.de

2.1 Log-Linear Models

A *log-linear model* is a compact representation of a probability distribution over assignments to a set of discrete-valued random variables $\mathcal{X} = \{X_1, \dots, X_n\}$ [17, 9]. The log-linear model is defined in terms of a set of feature functions $F = \{f_1(\mathbf{X}_1), \dots, f_k(\mathbf{X}_k)\}$, each of which is a function that defines a numerical value for each assignment \mathbf{x}_k to some subset $\mathbf{X}_k \subseteq \mathcal{X}$. Given a set of feature functions F , the parameters of the log-linear model are weights $W = \{w_k : f_k \in F\}$. The overall distribution is then defined as: $P(\mathbf{x}) = Z^{-1} \exp(\sum_{f_k \in F} w_k f_k(\mathbf{x}_k))$, where \mathbf{x}_k is the assignment to \mathbf{X}_k within \mathbf{x} , and Z is the normalization constant. A log-linear model induces a *Markov network* over \mathcal{X} , where there is an edge between every pair of variables $X_i, X_j \in \mathbf{X}_k$ that appear together in some feature $f_k(\mathbf{X}_k)$ and \mathbf{X}_k is a clique. Markov networks can also be encoded as a log-linear models by defining a feature function for every assignment of variables \mathbf{x}_c to a clique \mathbf{X}_c .

Markov Logic Networks (MLNs) can be seen as a first-order template language for log-linear models with binary variables. MLNs combine Markov networks and first-order logic (FOL) by attaching weights to first-order formulas and viewing these as templates for features of Markov networks [32]. An MNL L is a set of pairs (F_i, w_i) where F_i is a formula in FOL and w_i is a real number representing a weight. Together with a finite set of constants C , it defines a Markov Network $M_{L,C}$, where $M_{L,C}$ contains one node for each possible grounding of each predicate appearing in L . The value of the node is 1 if the ground predicate is true, and 0 otherwise. The probability distribution over possible worlds x specified by the ground Markov network $M_{L,C}$ is given by:

$$P(X = x) = \frac{1}{Z} \exp\left(\sum_{i=1}^F w_i n_i(x)\right)$$

where F is the number of formulas in the MLN and $n_i(x)$ is the number of true groundings of F_i in x . The groundings of a formula are formed simply by replacing its variables with constants in all possible ways. The *Herbrand base* HB is the set of all ground predicates (atoms) that can be constructed using the predicates in L and the constants in C . Each subset of the Herbrand base is a *Herbrand interpretation* specifying which ground atoms are true. A Herbrand interpretation H is a Herbrand model of L , written $\models_H L$, iff it satisfies all groundings of formulas in L .

There are two principal reasoning tasks in MLN, namely, MAP inference and Marginal inference. MAP inference is the task of finding the most probable world given some observations also referred to as evidence. Given the observed variables $E = e$, the MAP problem aims to find an assignment of all non-evidence (hidden) variables $X = x$ such that $\mathbf{I} = \operatorname{argmax}_x P(X = x \mid E = e)$. We denote by \mathbf{I} , the assignment x which leads P to be maximal, i.e., a MAP state. In order to compute a MAP state of an MLN, the problem can be formulated as an integer linear program (ILP) using the cutting plane inference algorithm.

2.2 \mathcal{EL}^{++}

\mathcal{EL}^{++} is the DL underlying the OWL 2 profile OWL-EL².

Syntax: Given a set of concept names \mathbf{N}_C , role names \mathbf{N}_R , individuals \mathbf{N}_I , and feature names \mathbf{N}_F , \mathcal{EL}^{++} concepts and roles are formed

according to the syntax given in Table 1. A concept in \mathcal{EL}^{++} is either a top, bottom concept, an atomic concept, a concrete concept or a complex concept (formed by conjunction and existential restriction). A *concrete domain* \mathbf{D} is a pair $(\Delta^{\mathbf{D}}, \mathcal{P}^{\mathbf{D}})$ with $\Delta^{\mathbf{D}}$ a set and $\mathcal{P}^{\mathbf{D}}$ a set of predicate names. Each $p \in \mathcal{P}$ is associated with an arity $n > 0$ and an extension $p^{\mathbf{D}} \subseteq (\Delta^{\mathbf{D}})^n$. The abstract and concrete domains are linked via a set of *feature names* \mathbf{N}_F . In Table 1, p denotes a predicate of some concrete domain \mathbf{D}_j and f_1, \dots, f_n are feature names. In this work, we consider only numerical CDs. However, our approach can easily be extended to handle other CDs. An \mathcal{EL}^{++} TBox contains a set of GCI (General Concept Inclusion) axioms, i.e., $C \sqsubseteq D$, as well as (RI) role inclusion axioms, i.e., $r_1 \circ r_2 \sqsubseteq r$.

Semantics: The semantics of \mathcal{EL}^{++} concepts and roles, shown in Table 1, is given by an interpretation function $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ which consists of a non-empty (abstract) domain $\Delta^{\mathcal{I}}$ and a mapping $\cdot^{\mathcal{I}}$ that assigns to each atomic concept $A \in \mathbf{N}_C$ a subset of $\Delta^{\mathcal{I}}$, to each abstract role $R \in \mathbf{N}_R$ a subset of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, to each concrete relation $f \in \mathbf{N}_F$ a subset of $\Delta^{\mathcal{I}} \times \mathbf{D}$, and to each individual $a \in \mathbf{N}_I$ an element of $\Delta^{\mathcal{I}}$. The mapping $\cdot^{\mathcal{I}}$ is extended to all concepts and roles as shown in Table 1.

Table 1. The \mathcal{EL}^{++} with concrete domains.

Name	Syntax	Semantics
top	\top	$\Delta^{\mathcal{I}}$
bottom	\perp	\emptyset
nominal	$\{a\}$	$\{a^{\mathcal{I}}\}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
existential restriction	$\exists r.C$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y \in \Delta^{\mathcal{I}} : (x, y) \in r^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
concrete domain	$p(f_1, \dots, f_n)$ for $p \in \mathcal{P}^{\mathbf{D}_j}$	$\{x \in \Delta^{\mathcal{I}} \mid \exists y_1, \dots, y_k \in \Delta^{\mathbf{D}_j} : f_i^{\mathcal{I}}(x) = y_i \text{ for } 1 \leq i \leq k \wedge (y_1, \dots, y_k) \in p^{\mathbf{D}_j}\}$
GCI	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
RI	$r_1 \circ r_2 \sqsubseteq r$	$r_1^{\mathcal{I}} \circ r_2^{\mathcal{I}} \subseteq r^{\mathcal{I}}$
concept assertion	$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
role assertion	$r(a, b)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I}}$

Knowledge about specific objects can be expressed using concept and role assertions of the form $C(a)$ and $R(a, b)$. The axioms and assertions are contained in the TBox and ABox, respectively, which together form a knowledge base (KB). An \mathcal{EL}^{++} knowledge base $KB = (\mathcal{T}, \mathcal{A})$ consists of a set \mathcal{T} of general concept inclusion axioms (TBox) and role inclusion axioms, and possibly a set \mathcal{A} of assertional axioms (ABox). Despoina et al [8] have extended the inference/completion rules of \mathcal{EL}^{++} for the concrete domains by exploiting the notion of safety which keeps tractability of reasoning while enhancing expressivity. We will make use of these rules to provide datatype reasoning in log-linear \mathcal{EL} .

To simplify the translation of weighted \mathcal{EL}^{++} KBs into FOL, we first obtain the *normal* form of the KB in such a way that satisfiability is preserved [2, 18]. An \mathcal{EL}^{++} KB is in *normal* form $\tau(KB)$ if its axioms are in the following form:

$$C(a) \quad r(a, b) \quad A \sqsubseteq C \quad A \sqcap B \sqsubseteq C$$

² <http://www.w3.org/TR/owl2-profiles/>

$$\begin{aligned} \exists r.A \sqsubseteq C \quad A \sqsubseteq \exists r.B \quad r_1 \sqsubseteq r_2 \quad r_1 \circ r_2 \sqsubseteq r \\ A \sqsubseteq p(f_1, \dots, f_n) \quad p(f_1, \dots, f_n) \sqsubseteq B \end{aligned}$$

where $A, B, C \in \mathcal{N}_C \cup \{\top\}$ and $C \in \{\perp\}$; $r, r_1, r_2 \in \mathcal{N}_R$; $f_1, \dots, f_n \in \mathcal{N}_F$; and $a, b, c \in \mathcal{N}_I$. For a finite set $\mathcal{N} \subseteq \mathcal{N}_C \cup \mathcal{N}_R$ of concept and role names the set of all normalized axioms constructible from \mathcal{N} is the union of (a) all normalized GCIs constructible from concept and role names in \mathcal{N} and the top and bottom concepts; and (b) all normalized RIs constructible from role names in \mathcal{N} .

Axioms of the form $C \sqsubseteq \{o\}$ are shown to be problematic for practical reasoning in the \mathcal{EL} family with nominals. However, experiments have shown that such axioms are rarely found in real world ontologies [14, 15]. In addition, by restricting their usage efficient practical reasoning can be achieved. Furthermore, for safe \mathcal{EL}^{++} , the inference rules given in [2] and proved in [15] are sound and complete. Therefore, for this study, we consider safe nominals as defined below.

Definition 1 (Kazakov et al [15]) An \mathcal{EL}^{++} concept C is safe if C has only occurrences of nominals in subconcepts of the form $\exists R.\{o\}$; C is negatively safe if C is either safe or a nominal. A GCI $C \sqsubseteq D$ is safe if C is negatively safe and D is safe. An \mathcal{EL}^{++} ontology is safe if all its concept inclusions are safe.

For instance, the axioms $\{a\} \sqsubseteq C$ and $\{a\} \sqsubseteq \exists r.\{b\}$ are safe. In this study, we consider safe \mathcal{EL}^{++} . It is possible to provide a probabilistic extension of safe \mathcal{EL}^{++} using MLNs. In which, a safe \mathcal{EL}^{++} KB can be seen as a set of hard constraints on the set of possible interpretations: if an interpretation violates even one axiom or assertion, it has zero probability. The basic idea in MLNs is to soften these constraints, i.e., when an interpretation violates one axiom or assertion in the KB it is less probable, but not impossible. The fewer axioms an interpretation violates, the more probable it becomes. Each axiom and assertion has an associated weight that reflects how strong a constraint is: the higher the weight, the greater the difference in log probability between an interpretation that satisfies the axiom and one that does not, other things being equal [32].

2.3 Log-Linear \mathcal{EL}

Log-linear \mathcal{EL} (LogEL) is an extension of \mathcal{EL}^{++} with log-linear models that enables reasoning over uncertain \mathcal{EL}^{++} TBoxes [27].

Syntax: A LogEL knowledge base $\text{KB} = (\text{KB}^D, \text{KB}^U)$ consists of a deterministic knowledge base KB^D and an uncertain knowledge base KB^U with $\text{KB}^D \not\models_{\mathcal{L}} \perp_{\mathcal{L}}$ and $\text{KB}^D \cap \text{KB}^U = \emptyset$ for some logic \mathcal{L} that supports a valid entailment relation $\models_{\mathcal{L}}$ and notion of contradiction $\perp_{\mathcal{L}}$. The uncertain KB is defined as $\text{KB}^U = \{\langle c_i, w_{c_i} \rangle\}$ where c_i is an axiom or assertion in \mathcal{L} and w_{c_i} is a real-valued weight assigned to c_i . The syntax of an axiom (resp. assertion) is similar to the underlying logic where an uncertain axiom (resp. assertion) has an associated weight as $\{\langle c_i, w_{c_i} \rangle\}$.

Semantics: The semantics of a LogEL knowledge base is based on joint probability distributions over the uncertain KB. Formally, for a given log-linear $\text{KB} = (\text{KB}^D, \text{KB}^U)$ and some KB' over the same signature, the probability of KB' is defined as:

$$P(\text{KB}') = \begin{cases} \frac{1}{Z} \exp \left(\sum_{\{(c_i, w_{c_i}) \in \text{KB}^U: \text{KB}' \models_{\mathcal{L}} c_i\}} w_{c_i} \right) & \text{if } \text{KB}' \not\models_{\mathcal{L}} \perp_{\mathcal{L}} \wedge \text{KB}' \models_{\mathcal{L}} \text{KB}^D, \\ 0 & \text{otherwise} \end{cases}$$

where Z is the normalization constant of the log-linear probability distribution P . Note that in MAP inference (i.e., obtaining the most probable KB) Z is not computed.

Example 1 Consider the following uncertain LogEL axioms: (1) a researcher is (probably) someone who published something, and (2) a famous researcher influenced (probably) someone who in turn has influenced someone else.

- (1) $\langle \text{Researcher} \sqsubseteq \exists \text{published}.\top, 0.8 \rangle$
- (2) $\langle \text{FamousResearcher} \sqsubseteq \exists \text{influenced}.\langle \exists \text{influenced}.\top \rangle \sqcap \text{Researcher}, 0.6 \rangle$

A LogEL KB can be normalized into an equivalent KB, thus, it can be mapped into first order predicates using a function φ as follows:

$$\begin{aligned} A \sqsubseteq C &\mapsto \text{sub}(A, C) & \exists r.A \sqsubseteq C &\mapsto \text{rsub}(A, r, C) \\ A \sqcap B \sqsubseteq C &\mapsto \text{int}(A, B, C) & r_1 \sqsubseteq r_2 &\mapsto \text{psub}(r_1, r_2) \\ A \sqsubseteq \exists r.B &\mapsto \text{rsup}(A, r, B) & r_1 \circ r_2 \sqsubseteq r &\mapsto \text{pcom}(r_1, r_2, r) \end{aligned}$$

The predicates in this listing are typed, r, r_1, r_2 are role names, A and B are concept names, and C is a concept name or the bottom concept (\perp). Note that φ is incomplete given an LogEL as defined above. In this work we will extend φ in order to deal with both ABox assertions and CDs (i.e., $p(f_1, \dots, f_n)$).

The translation of KB^D and KB^U results in unweighted (hard) and weighted first order formulas respectively. The hard formulas are used, together with the TBox completion rules $(F_1) - (F_9)$ to enforce $\text{KB}' \not\models_{\text{EL}} \perp_{\text{EL}}$ and $\text{KB}' \models_{\text{EL}} \text{KB}^D$ for any possible TBox KB' . Computing the MAP state of a LogEL knowledge base will always result in the most probable non contradictory subset of KB^U that entails the previously known axioms KB^D .

- (F_1) $\text{sub}(c, c)$
- (F_2) $\text{sub}(c, \top)$
- (F_3) $\text{sub}(c, c') \wedge \text{sub}(c', d) \Rightarrow \text{sub}(c, d)$
- (F_4) $\text{sub}(c, c_1) \wedge \text{sub}(c, c_2) \wedge \text{int}(c_1, c_2, d) \Rightarrow \text{sub}(c, d)$
- (F_5) $\text{sub}(c, c') \wedge \text{rsup}(c', r, d) \Rightarrow \text{rsup}(c, r, d)$
- (F_6) $\text{rsup}(c, r, d) \wedge \text{sub}(d, d') \wedge \text{rsub}(d', r, e) \Rightarrow \text{sub}(c, e)$
- (F_7) $\text{rsup}(c, r, d) \wedge \text{psub}(r, s) \Rightarrow \text{rsup}(c, s, d)$
- (F_8) $\text{rsup}(c, r_1, d) \wedge \text{rsup}(d, r_2, e) \wedge \text{pcom}(r_1, r_2, r_3) \Rightarrow \text{rsup}(c, r_3, e)$
- (F_9) $\neg \text{sub}(c, \perp)$

The above formulas (collectively denoted by \mathbf{F}) are universally quantified over all variables. They are partially derived from the completion rules of \mathcal{EL}^{++} [2]. \top and \perp are constant symbols representing the top and bottom concepts. Note that rule F_9 does not belong to the completion rules for \mathcal{EL}^{++} . This rule takes the notion of incoherence into account. An incoherent ontology is an ontology that contains an unsatisfiable concept, i.e., a concept that is subsumed by \perp . Usually, an unsatisfiable concept indicates that the ontology contains a contradictory set of axioms. An incoherent ontology is not necessarily inconsistent. Thus, we added rule F_9 which allows us to extend the notion of contradiction \perp_{EL} from inconsistency to incoherence. For more technical details on applying the principle of log-linear logic to OWL-EL, we refer the reader to [27]. We will extend these completion rules in order to deal with ABox assertions and CDs.

3 MLN with Numerical Constraints

We extend MLN with numerical constraints resulting in a formalism denoted MLN_{NC} . The constraints are predicates of the form $\theta \bowtie \psi$, where θ and ψ denote variables, numerical constants or algebraic expressions (that might contain elementary operators), and \bowtie is a binary operator which returns a truth value under some grounding.

Definition 2 (MLN with Numerical Constraints (MLN_{NC})) A numerical constraint NC is composed of numerical constants (such as elements of \mathbb{N} , \mathbb{I} , and so on), variables, elementary operators or functions (such as, $+$, $*$, $-$, \div , $\%$, $\sqrt{\quad}$), standard relations ($>$, $<$, $=$, \neq , \geq , \leq), and boolean operators (\wedge , \vee , \neg). An MLN_{NC} is a set of pairs (FC_i, w_i) where FC_i is a formula in FOL that may contain a NC and w_i is a real number representing the weight of formula FC_i .

Example 2 Using MLN_{NC} it is possible to represent the hard constraint: people born before 1850 are probably not alive: $\{\forall a, y : \text{person}(a) \wedge \text{born}(a, y) \wedge \text{NC}(y) \Rightarrow \text{dead}(y), \text{NC}(y) = y < 1850\}$.

MAP Inference in MLN_{NC} . A common inference task over MLNs is finding the most probable state of the world, i.e., finding a complete assignment to all ground atoms which maximizes the probability. A MAP query corresponds to an optimization problem with linear constraints and a linear objective function. Hence, it can be formulated and solved as an instance of an integer linear program (ILP) using the cutting plane approach, proposed in [34] and extended in [29]. Consequently, we extend the approach in [30] to compute a MAP state for a MLN_{NC} knowledge base. Cutting plane inference (CPI) operates between the grounding algorithm and the ILP solver. Instead of immediately adding one linear constraint for each ground clause to the ILP formulation, the ILP is initially formulated so as to enforce the evidence to hold in any solution. Based on the solution of this more compact ILP, one determines the violated constraints, adds these to the ILP, and solves the ILP again. This process is repeated until no constraints are violated by an intermediate solution. To elaborate, at the beginning of each CPI iteration it is necessary to determine the violated ground clauses G that are specified by the MLN and are in conflict with the intermediate solution. A binary ILP variable $x_\ell \in \{0, 1\}$ gets assigned to each grounded predicate occurring in a violated clause $g \in G$. The value of the variable x_ℓ is 1 if the respective literal ℓ is true and 0 if it is false. These variables are used to generate ILP constraints that are added to the ILP for each violated ground clause. For each clause $g \in G$, we define $L^+(g)$ as the set of ground atoms that occur unnegated in g and $L^-(g)$ as the set of ground atoms that occur negated in g . The transformation scheme depends on the weight $w_g \in \mathbb{R}$ of the violated clause g . It is also necessary to create a binary variable z_g for every g with $w_g \neq \infty$ that is used in the objective of the ILP. For every ground clause g with $w_g > 0$, the following constraint has to be added to the ILP.

$$\sum_{\ell \in L^+(g)} x_\ell + \sum_{\ell \in L^-(g)} (1 - x_\ell) \geq z_g$$

A ground atom ℓ that is set to false (or true if it appears negated) by evidence will not be included in the ILP as it cannot fulfill the respective constraint. For every g with weight $w_g < 0$, we add the following constraint to the ILP:

$$\sum_{\ell \in L^+(g)} x_\ell + \sum_{\ell \in L^-(g)} (1 - x_\ell) \leq (|L^+(g)| + |L^-(g)|)z_g$$

Algorithm 1 Extended cutting planes algorithm

Input: G : ground clauses; E : evidence clauses
Input: G_{NC} : ground clauses with NC
Input: $L_{\text{NC}}^+ \cup L_{\text{NC}}^-$: ground (positive or negated) numerical literals
Output: $H^{(t)}$: MAP state

- 1: **procedure** COMPUTEMAP(G, E)
- 2: $G_{\text{ILP}} \leftarrow E \cup \{\text{literal}(g) \in G \text{ and } w_g > 0\}$
- 3: Initial solution $H^{(0)} \leftarrow$ all atoms in G_{ILP}
- 4: $\text{ILP} \leftarrow \text{intoILP}(\forall g \in G_{\text{ILP}})$
- 5: $t \leftarrow 0$
- 6: **repeat**
- 7: $G_{\text{new}} \leftarrow \emptyset$
- 8: **for** $g \in G \setminus G_{\text{ILP}}$ **do**
- 9: **if** $((w_g > 0 \text{ or } w_g = \infty) \text{ and } H^{(t)} \not\models g)$
- 10: **or** $(w_g < 0 \text{ and } H^{(t)} \models g)$ **then**
- 11: **if** $g \in G_{\text{NC}}$ and g contains g_p and
- 12: $((\text{comp}(g_p) = 1 \text{ and } g_p \in L_{\text{NC}}^-) \text{ or}$
- 13: $(\text{comp}(g_p) = 0 \text{ and } g_p \in L_{\text{NC}}^+))$ **then**
- 14: **add** $g \setminus g_p$ to G_{new}
- 15: **else** **add** g to G_{new}
- 16: **end if**
- 17: **end for**
- 18: **end for**
- 19: **if** $G_{\text{new}} \neq \emptyset$ **then**
- 20: $t \leftarrow t + 1$
- 21: $\text{ILP} \leftarrow \text{intoILP}(\forall g \in G_{\text{new}})$
- 22: **add** $g \in G_{\text{new}}$ to G_{ILP}
- 23: $H^{(t)} \leftarrow$ solution of ILP
- 24: **end if**
- 25: **until** $G_{\text{new}} = \emptyset$ **return** $H^{(t)}$
- 26: **end procedure**

The variable z_g expresses if a ground formula g is true considering the optimal solution of the ILP. However, for every g with weight $w_g = \infty$ this variable can be replaced with 1 as the respective formula cannot be violated in any solution:

$$\sum_{\ell \in L^+(g)} x_\ell + \sum_{\ell \in L^-(g)} (1 - x_\ell) \geq 1$$

Finally, the objective of the ILP sums up the weights of the (satisfied) ground formulas:

$$\max \sum_{g \in \mathcal{G}} w_g z_g.$$

The MAP problem can be solved as an ILP problem. Thus, the MAP state corresponds to the solution of the ILP in the last CPI iteration. It can be directly obtained from the solution as the assignment of the variables x_ℓ can be directly mapped to the optimal truth values for the ground predicates, i.e., $x_i = \text{true}$ if the corresponding ILP variable is 1 and $x_i = \text{false}$ otherwise.

We have extended the cutting planes algorithm proposed by Riedel [33] and optimized in [34] to the truth value of numerical predicates on-demand during each CPI iteration. This extension is sketched in Algorithm 1. The first initial MAP hypothesis is the set $H^{(0)}$. It contains all evidence clauses and ground clauses (containing only a single literal) with positive weights. These clauses have been added to the ILP and are denoted by G_{ILP} . At each CPI iteration, the violated constraints are identified and added to G_{new} . Thus, depending on the numerical predicates and the weight of the ground clauses, the violated constraints are determined (Lines 9–13). Consequently, for

every ground clause $g \in \mathbf{G} \setminus \mathbf{G}_{\text{ILP}}$, if g is not satisfied by the current hypothesis $H^{(t)}$ and $w_g > 0$ or $w_g = \infty$, or g is satisfied and $w_g < 0$, we test if g contains a NC. If g contains a NC (i.e., $g \in \mathbf{G}_{\text{NC}}$) and the numerical predicate g_p is in g , then we compute the numerical constraints corresponding to the predicate g_p with the current grounding, i.e., $\text{comp}(g_p)$. If $\text{comp}(g_p)$ is *true* and g_p is a negated literal $g_p \in L_{\text{NC}}^-$ or if $\text{comp}(g_p)$ is *false* and g_p is a positive literal $g_p \in L_{\text{NC}}^+$, we remove the numerical predicate from g and add g to \mathbf{G}_{new} . Note that we do not introduce ILP variables for numerical predicates as they will not be added to the \mathbf{G}_{new} , hence, to the ILP. Note also that our approach computes the truth value of numerical predicates on-demand, i.e., only when g is not satisfied by the current hypothesis. If g does not contain a numerical predicate, we add g to \mathbf{G}_{new} . If \mathbf{G}_{new} is empty, then the current hypothesis is optimal and we return it as MAP state. Otherwise, it iterates until an optimal solution is found.

Theorem 1 *Given a ground MLN_{NC} network, a set of observed variables E and a set of hidden variables X , an interpretation \mathbf{I} is a MAP state iff*

$$\mathbf{I} = \underset{X}{\text{argmax}} P(X | E) = \underset{X}{\text{argmax}} \sum_i w_i n_i(E, X).$$

Proof 1 (sketch) *It is proved that the MAP problem in MLN can be reduced to ILP using the CPI approach [34]. Likewise, we reduce MLN_{NC} 's MAP inference into a maximization problem in ILP using cutting planes. In Algorithm 1, if \mathbf{G} contains no ground numerical predicates, then the MAP inference in MLN_{NC} coincides with MAP inference in MLN. Otherwise, it is possible to compute a MAP state by transforming \mathbf{G} into ILP and integrating the truth value of the numerical predicates to the ILP by computing it outside the CPI setting as shown in Algorithm 1.*

4 Extending LogEL with Concrete Domains

We extend LogEL with uncertain knowledge expressed through numerical CDs.

Definition 3 *A numerical concrete domain \mathbf{D} is a pair $(\Delta^{\mathbf{D}}, \mathbf{P}^{\mathbf{D}})$ with $\Delta^{\mathbf{D}} \in \{\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}\}$ and $\mathbf{P}^{\mathbf{D}}$ a set of predicate names. Each $p \in \mathbf{P}$ is associated with an arity n and an extension $p^n \subseteq (\Delta^{\mathbf{D}})^n$.*

4.1 Unary concrete domains

We call a CD \mathbf{D} *unary* if all predicates $p \in \mathbf{P}^{\mathbf{D}}$ are unary, i.e., $n = 1$. In this section, we consider the unary predicates $\{<_v, \leq_v, >_v, \geq_v, =_v\}$ where $v \in \mathbb{R}$. The DL underlying LogEL admits safeness of CDs to retain tractability of reasoning. We present a unary CD \mathbf{R} inspired by safe numerical CDs introduced in [8]. *Safeness* is defined in terms of the position in which a predicate appears with respect to the subsumption (\sqsubseteq) relation in an axiom. If a predicate p appears on the right side of \sqsubseteq , then we call it a *positive predicate* ($p \in \mathbf{P}_+^{\mathbf{D}}$) and if on the left side, then it is a *negative predicate* ($p \in \mathbf{P}_-^{\mathbf{D}}$).

Definition 4 *A safe unary numerical CD \mathbf{R} is a triple $(\Delta^{\mathbf{R}}, \mathbf{P}_+^{\mathbf{R}}, \mathbf{P}_-^{\mathbf{R}})$ with $\Delta^{\mathbf{R}} \subseteq \mathbb{R}$ a set and $\mathbf{P}_+^{\mathbf{R}}, \mathbf{P}_-^{\mathbf{R}} \subseteq \{<_v, \leq_v, >_v, \geq_v, =_v\}$ are sets of positive and negative predicate names respectively.*

In the following, we consider an extension of LogEL with a safe unary numerical CD \mathbf{R} and refer to it as LogEL(\mathbf{R}).

Example 3 *We extend Example 1 with axioms that use safe numerical CDs where $<_{1850} \in \mathbf{P}_-^{\mathbf{D}}$ and $<_{1800} \in \mathbf{P}_+^{\mathbf{D}}$.*

$$\begin{aligned} & \langle \text{University} \sqcap <_{1850}(\text{foundedIn}) \\ & \quad \sqsubseteq \text{DistinguishedUniversity}, 0.4 \rangle \\ & \langle \text{OldUniversity} \sqsubseteq <_{1800}(\text{foundedIn}) \sqcap \text{University}, 0.6 \rangle \end{aligned}$$

Inference involving axioms that contain the CD \mathbf{R} can be done according to the following deduction/completion rules. In Definition 6, we will transform them into FOL formulas for reasoning over LogEL(\mathbf{R}) KB.

$$\frac{A \sqsubseteq B \quad B \sqsubseteq p_v(f)}{A \sqsubseteq p_v(f)} \quad \frac{A \sqsubseteq p_v(f) \quad p_{v'}(f) \sqsubseteq B \quad \text{eval}(p, v, v', v')}{A \sqsubseteq B}$$

where $p_v \in \mathbf{P}_+^{\mathbf{D}}, p_{v'} \in \mathbf{P}_-^{\mathbf{D}}$ and $v, v' \in \mathbb{R}$. In addition, the *eval* function checks if all possible values of the first *operator-value* pair (p, v) are covered by the possible values of the second *operator-value* pair (p', v') , when so, it evaluates to true otherwise false. The function *eval* is defined based on the domain \mathbb{R} and algebraic operators in \mathbf{R} . Some of the inequalities that are computed using the *eval* function are listed below:

$$\begin{aligned} \text{eval}(=, v, =, v') & := v = v' \\ \text{eval}(=, v, >, v') & := v > v' \end{aligned}$$

Example 4 *From the axioms of Example 3, it is possible to infer that $\text{OldUniversity} \sqsubseteq \text{DistinguishedUniversity}$ because $\text{eval}(<, 1800, <, 1850) = 1800 \leq 1850 = \text{true}$.*

In Section 5, we will see that the computation of the *eval* function integrates well with the proposed extension of the cutting planes algorithm.

4.2 Binary concrete domains

Unary CDs are not expressive enough to enable to describe the *relationship* between concrete values [22]. For instance, to describe a person that passed away at birth, we can use a binary CD as: $\text{DeadAtBirth} \sqsubseteq =(\text{birthDate}, \text{deathDate})$. Next, we introduce such expressive binary CDs. The main advantage of binary CDs over unary ones is that the former provides for a richer set of predicates [23]. Consequently, we introduce a CD \mathbf{T} which is based on real numbers and a set of unary and binary predicates. \mathbf{T} is defined as follows: $\mathbf{T} = (\Delta^{\mathbf{T}}, \mathbf{P}^{\mathbf{T}})$ where $\Delta^{\mathbf{T}} = \mathbb{R}$ and $\mathbf{P}^{\mathbf{T}}$ is defined as the smallest set containing the following predicates:

- unary predicates $\top_{\mathbf{T}}$ with $(\top_{\mathbf{T}})^{\mathbf{T}} = \mathbb{R}$ and $\perp_{\mathbf{T}}$ with $(\perp_{\mathbf{T}})^{\mathbf{T}} = \emptyset$,
- a unary predicate p_r for each $p \in \{<, \leq, =, \neq, \geq, >\}$ and each $r \in \mathbb{R}$ with $(p_r)^{\mathbf{T}} = \{r' \in \mathbb{R} \mid r' p r\}$, and
- binary predicates $p \in \{<, \leq, =, \neq, \geq, >\}$ with $(p)^{\mathbf{T}} = \{(r, r') \in \mathbb{R}^2 \mid r p r'\}$.

We will use the c \mathbf{T} for temporal reasoning over weighted LogEL(\mathbf{T}) KBs as shown in Section 6.2. In order to reason over such KBs, we design a set of domain specific rules which are used either for inferring a new knowledge or checking conflicts (aka. debugging). This is partly due to most of the current KBs do not contain TBox axioms that involve binary CDs. Note that it is possible to define a temporal CD based solely on Allen's interval relations which are often used in temporal reasoning [1]. Allen's relations describe the relationship between any two intervals over some temporal structure (for instance, a set of time points under a strict total ordering relation \prec) and can

be defined in terms of the interval endpoints. And thus, they can be expressed via the binary predicates of the CD \mathcal{T} . Hence, the CD \mathcal{T} is more expressive than a CD obtained by using Allen relations.

For the binary CDs we do not provide TBox completion rules, instead for the purpose of experimentation, we design a set of KB specific constraints. Examples of such rules are the following:

- (1) $\forall x, t1, t2 : Person(x) \wedge birthDate(x, t1) \wedge graduationDate(x, t2) \Rightarrow \langle t1, t2 \rangle$ 0.9
- (2) $\forall x, y, t1 : YoungAuthor(x) \wedge age(x, t1) \Rightarrow \langle t1, 20 \rangle$ 0.8

These rules are used in the experiments as discussed in Section 6.2.

5 Translating LogEL(R) and LogEL(T) into MLN_{NC}

As shown in previous work, the MAP state for a LogEL KB can be computed using a MLN that represents the first order transformation of the KB and entailment rules.

We have already presented the function φ for the translation of LogEL TBox axioms into FOL predicates. We now extend this function to map the CDs and ABox assertions of a LogEL KB into ground FOL predicates. Since the DL, \mathcal{EL}^{++} , underlying LogEL is equipped with nominals, ABox assertions can be converted into TBox axioms. Thus, with nominals, the ABox becomes syntactic sugar as shown below:

$$C(a) \Leftrightarrow \{a\} \sqsubseteq C, \quad r(a, b) \Leftrightarrow \{a\} \sqsubseteq \exists r.\{b\}$$

Instance checking in turn is directly reducible to subsumption checking in the presence of nominals. Thereby reducing ABox reasoning into TBox reasoning. Thus, in LogEL(R) and LogEL(T), we transform the ABox assertions into TBox axioms using nominals. We also extend the definition of normal forms for LogEL KBs with CDs as follows: for a LogEL KB = (KB^D, KB^U) , its normal form is given by: $\tau'(KB) = \tau(KB^D) \cup \bigcup_{(c_i, w_{c_i}) \in KB^U} \tau(c_i)$.

Definition 5 φ translates a normalized LogEL(R) and LogEL(T) KB's ABox and CDs into FOL formulas in MLN_{NC} as follows:

$$\begin{aligned} C(a) &\mapsto \text{sub}(\{a\}, C) \\ r(a, b) &\mapsto \text{rsup}(\{a\}, r, \{b\}) \\ A \sqsubseteq p_v(f) &\mapsto \text{sub}_+(A, p, v, f) \\ p_v(f) \sqsubseteq B &\mapsto \text{sub}_-(p, v, f, C) \end{aligned}$$

Definition 6 The translation of LogEL(R) TBox completion rules into FOL formulas \mathbf{F} in MLN_{NC} is given below:

$$\begin{aligned} (F_{10}) \quad &\text{sub}(a, b) \wedge \text{sub}_+(b, p, v, f) \Rightarrow \text{sub}_+(a, p, v, f) \\ (F_{11}) \quad &\text{sub}_+(a, p, v, f) \wedge \text{sub}_-(p', v', f, b) \wedge \\ &\text{eval}(p, v, p', v') \Rightarrow \text{sub}(a, b) \end{aligned}$$

Note that the FOL formulas \mathbf{F} corresponding to the translation of the completion rules are hard constraints. As the translation function φ is bijective, a possible world of \mathbf{F} and a given KB can be turned into a classified and consistent \mathcal{EL}^{++} (R) KB and vice versa as shown in the following lemma.

Lemma 1 Let KB be a normalized LogEL(R) (resp. LogEL(T)) knowledge base over a finite set \mathbf{N} of concept, role, and feature names and let HB be the Herbrand base of \mathbf{F} with respect to \mathbf{N} . If $KB' \subseteq KB$ is a classified and coherent knowledge base, then $\varphi(KB')$ is a Herbrand model of \mathbf{F} . If $HB' \subseteq HB$ is a Herbrand model of \mathbf{F} then $\varphi^{-1}(HB')$ is a classified and coherent knowledge base.

For a finite set of concept, role, individual and feature names \mathbf{F} , every normalized KB over the signature \mathbf{N} which is classified and coherent, corresponds to exactly one Herbrand model of \mathbf{F} .

MAP Inference Computing the MAP state of a given LogEL(R) (resp. or LogEL(T)) KB requires to translate the KB with the function φ (see Definition 5) to the equivalent Markov logic formalization. Then the completion rules of Definition 6 are added to this translation. The MAP state is computed with the help of Algorithm 1 applied to this input data. To do so, the evidence clauses $\varphi(KB)$ and the grounding of \mathbf{F} with respect to $\varphi(KB)$ are given as input to the algorithm. Besides, Algorithm 1 uses the external *comp* function whenever one of the grounding of the completion rules (F_{10}) and (F_{11}) requires to check a concrete grounding of the *eval* predicate. Note that the *comp* function can be easily implemented for the specified set of unary and binary numerical predicates (corresponding to a particular CD). Moreover, further extension of the proposed approach will also benefit from the fact that the externally defined *comp* function can be easily extended to deal with other predicates, for example, to support string comparisons. Applying the inverse translation function φ^{-1} to the MAP state, yields the most probable, classified, and consistent \mathcal{EL}^{++} (R) (resp. \mathcal{EL}^{++} (T)) KB.

Theorem 2 Given the following:

- a LogEL(R) (resp. LogEL(T)) normalized KB = (KB^D, KB^U) over a finite set \mathbf{N} of concept, role, individual and feature names,
- the Herbrand base HB of the formulas \mathbf{F} with respect to \mathbf{N} ,
- the set of ground formulas G_1 constructed from KB^D , and
- the set of ground formulas G_2 constructed from KB^U .

The most probable coherent and classified ontology is obtained with:

$$\varphi^{-1}(I) = \arg \max_{HB \supseteq I = G_1 \cup F} \left(\sum_{(g_j, w_j) \in G_2: I \neq g_j} w_j \right)$$

From Theorem 2 and the results in [27], the problem of computing the most probable, classified and coherent LogEL(R) (resp. LogEL(T)) KB is NP-hard.

6 Experiments

In the following we report about two types of experiments: temporal reasoning in LogEL(R) and LogEL(T) KBs.

6.1 Reasoning in LogEL(R)

In this experiment, we first illustrate in how far our formalism can be applied to a scenario where we reason with a knowledge base $KB = (KB^D, KB^U)$ with $KB^U = \emptyset$. In the second set of experiments we add weights to all assertions in KB^D , while adding at the same time a set of weighted erroneous assertions. We show that we achieve similar reasoning results in this noisy setting, while automatically detecting and removing erroneous numerical assertions.

6.1.1 Deterministic Temporal Reasoning

We apply our method to a specific subdomain of DBpedia that deals with researchers, their alma mater, their birth and death date, their publications together with the publishing date, and the influence relationships between researchers. For our experiments we worked with

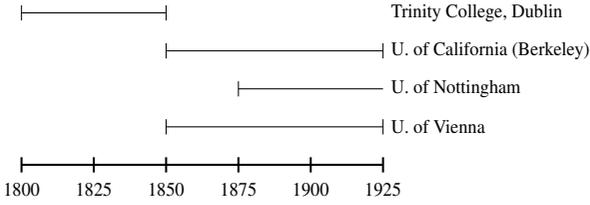


Figure 1. A part of the elite universities from 1800 to 1925.

a slightly simplified signature. For example, we used the object property *influenced* for all assertions using one of the DBpedia properties *influenced*, *academic advisor*, and *notable student*. We extracted a dataset from DBpedia containing >11K persons, >1K universities, >10K books, and \approx 10K birth and death dates, >19K influenced assertions, and >700 publishing dates.

Then we added the axiom *FamousResearcher* $\equiv \exists \textit{influenced} . (\exists \textit{influenced} . \top)$ as well as a set of axioms for which the following axiom is an example.

$$\begin{aligned} & \exists \textit{hasStudent} . (\textit{FamousResearcher} \sqcap \\ & >_{1830}(\textit{born}) \sqcap \exists \textit{authorOf} . <_{1910}(\textit{published})) \\ & \sqsubseteq \textit{EliteUniversity}_{1850-1900} \end{aligned}$$

The first axiom defines a famous researcher as someone has influenced someone who in turn has influenced someone else. The second axiom states that an elite or distinguished university is one which had a student who was a famous researcher born after 1830 and published some work before 1910. With the help of these two axioms we can entail which academic institutions had a significant impact on the research community during the period 1850–1900. We created the same type of axioms for all 50 year periods starting in 1700 using a step size of 25 years. Then we computed the MAP state and retrieved the instances for all *EliteUniversity* concepts. A subset of the results of our query, with respect to the periods in the range from 1800 to 1925, is shown in Figure 1. Note that the number of *EliteUniversity* increases over time, which might also be caused by the fact that DBpedia is more complete with respect to more recent publications. Nevertheless, all of the universities from the results obtained are well known and distinguished universities.

6.1.2 Uncertain Temporal Reasoning

In this set of experiments we generated erroneous assertions specifying the publishing date of a book and the birth and death dates of persons. We injected a fraction of 20%, 40%, 60%, 80%, and finally 100% incorrect statements to the dataset. For instance, injecting 20% erroneous facts means that we added 20% additional wrong assertions for each of the three datatype properties. We randomly assigned weights in the range [0.5, 1.0] to the injected assertions and [0.8, 1.0] to each of the originally stated assertions.

We added a set of axioms which allow to detect that some combinations of birth, death and publishing dates are inconsistent. For instance, the following axiom expresses the fact that a person who is younger than 15 years of age cannot publish something.

$$\begin{aligned} & \textit{FamousResearcher} \equiv \exists \textit{influenced} . (\exists \textit{influenced} . \top) & 0.6 \\ & >_{1850}(\textit{born}) \sqcap \exists \textit{authorOf} . <_{1865}(\textit{published}) \sqsubseteq \perp & 0.82 \\ & \exists \textit{hasStudent} . (\textit{FamousResearcher} \sqcap \\ & >_{1830}(\textit{born}) \sqcap \exists \textit{authorOf} . <_{1910}(\textit{published})) \\ & \sqsubseteq \textit{EliteUniversity}_{1850-1900} & 0.75 \end{aligned}$$

We added similar axioms to express, e.g., that nobody can be born after he died. All these axioms use only unary predicates. Thus we had to generate the axioms for different time points. We did this with a step size of 25, i.e., for each type of axiom we generated a set of axioms to cover the time span from 1700 to 2000.

The MAP state for this input data will be a subset of the weighted input assertions, which might contain incorrect and correct assertions. The MAP state will also contain the instantiations of all *EliteUniversity* concepts. We compare the outcome of our approach against a gold standard, by assuming that each originally stated fact in DBpedia is correct, and each added fact is incorrect. Furthermore, we treat the *EliteUniversity* query result of the first experiment as a gold standard to measure in how far the results are negatively affected by the noisy setting.

Table 2. Precision (P) and recall (R) scores for computing the MAP state with an increasing number of injected erroneous assertions.

Injection		0%	20%	40%	60%	80%	100%
Query	P	1.00	0.92	0.81	0.76	0.66	0.61
	R	1.00	0.93	0.86	0.79	0.76	0.70
Repair	P	-	0.83	0.83	0.83	0.83	0.83
	R	-	0.62	0.62	0.61	0.61	0.60
Time (s)		390	631	1313	2351	3336	4599

The results of our experiments are shown in Table 2. Since we randomly assigned weights, we repeated each experiment 10 times and present average scores. We were able to compute meaningful results in highly inconsistent settings. Even in a setting where we added 100% incorrect assertions (as much correct assertions as incorrect assertions), we are still able to achieve a query precision of 61% and a recall of 70%. The good query results are caused by the relatively good results for the repair, where we computed the fraction of incorrect statements that have been removed (recall of repair) and the fraction of correct removal decisions (precision of repair). Both values are stable for different injection rates. The measured runtimes do not increase linearly with respect to the size of the input data. This is in line with our expectation, since the original data (0% injection) set will only require a materialization, while each added incorrect assertion might be involved in a conflict resulting finally in a non trivial optimization problem.

6.2 Debugging LogEL(T) KBs

The objective of this experiment is debugging temporal LogEL(T) KBs. We define a set of domain specific temporal rules (similar to those in the example of Section 4.2) based on a common sense understanding of this dataset. Typical examples of such rules are the following ones.

$$\begin{aligned} (3) \quad & \forall x, t1, t2 : \textit{person}(x) \wedge \textit{birthDate}(x, t1) \wedge \\ & \textit{deathDate}(x, t2) \Rightarrow <(t1, t2) & 0.7 \\ (4) \quad & \forall x, y, t1, t2 : \textit{almaMater}(x, y) \wedge \textit{established}(y, t1) \wedge \\ & \textit{deathDate}(x, t2) \Rightarrow <(t1, t2) & 0.5 \end{aligned}$$

Besides, we injected several types of erroneous facts to the extracted dataset. The results of our experiments are depicted in Table 3. The first column informs about the precision of the generated dataset. We have injected a fraction of 1%, 10%, 25%, 50%, 75%, and finally 100% incorrect statements to the dataset. Thus, the precision of the generated dataset varies from 0.99 (260K facts) to 0.5

(520K facts). For example, the last row of the table refers to a dataset where every second fact is incorrect. The recall of this dataset is always 1.0, because we never removed a fact of the extracted dataset. For all datasets, we have computed the MAP state, which corresponds to the repaired dataset. In doing so we computed precision, recall and F-measure for the debugging process and for the final outcome. The precision of the debugging process refers to the fraction of removed axioms or assertions that were indeed incorrect; recall refers in this context to the fraction of incorrect axioms or assertions that have been removed. The precision and recall of the repaired dataset is computed by comparing it to the originally extracted dataset. The rightmost column (ΔF) informs about the gain in F-measure that we computed by comparing the F-measure of the input dataset with the F-measure of the repaired dataset.

Table 3. Precision (P), Recall (R) and F-measure (F) for debugging process and repaired dataset.

Input	Debugging			Repaired Dataset			ΔF
	P	R	F	P	R	F	
0.99	0.80	0.63	0.70	1.00	1.00	1.00	0.002
0.91	0.80	0.64	0.71	0.97	0.98	0.97	0.022
0.80	0.81	0.65	0.72	0.92	0.96	0.94	0.050
0.67	0.82	0.65	0.73	0.84	0.93	0.88	0.084
0.57	0.83	0.65	0.73	0.78	0.90	0.83	0.106
0.50	0.84	0.65	0.73	0.72	0.87	0.79	0.119

The results show that we are able to improve the data quality of an erroneous dataset. This is depicted in the increase of F-measure in the ΔF column for all test cases. With respect to the test case where every second fact is incorrect, we are able to increase the precision by 0.22 (from 0.5 up to 0.72), while recall is only reduced to 0.87 (from 1.0). Most of the removed facts are indeed incorrect (a repair precision of > 0.8 means that at least for 4 of 5 removals are proper removals), while we are able to detect more than half of the incorrect facts (debugging recall of the repair is > 0.5).

7 Related Work

Since in MLNs all variables and features are discrete, Hybrid MLNs that allow for both discrete and continuous variables in addition to limited numerical constraints have been introduced [36]. While our approach does not support continuous variables, the numerical constraints that we consider here are more expressive and the ability to compute them outside the cutting planes algorithm, enables us for far more richer constraints and functions. In the context of probabilistic programming, supporting numerical constraints in probabilistic reasoning is also explored [5, 4, 3].

The study of extending DLs to handle uncertainty and vagueness has gained momentum recently. There have been several proposals to add probabilities to various DLs. Probabilistic DLs can be classified in several dimensions. One possible classification is on the reasoning mechanism used: Markov logic networks (MLNs) and Bayesian networks. There exist some studies that employ MLNs to extend various DLs. The study in [21] extends \mathcal{EL}^{++} with probabilistic uncertainty based on the annotation of axioms using MLNs. The main focus of this work is ranking queries in descending order of probability of atomic inferences which is different from the objective of this paper. Another study in [27], presents a probabilistic extension of the DL \mathcal{EL}^{++} without nominals and CDs in MLN in order to find the most probable coherent ontology. In doing so, they have developed a

reasoner for probabilistic OWL-EL called ELOG [29]. In this study, we extend this work in order to deal with CDs in addition to nominals and instances. In databases, MLNs have been used to create a probabilistic datalog called Datalog+/. It is an extension of datalog that allows to express ontological axioms by using rule-based constraints [10]. The probabilistic extension of Datalog+/- uses MLNs as the underlying probabilistic semantics. The focus of this work is on scalable threshold query answering which is different from that of this work.

Other literatures extend DLs with Bayesian networks. Some notable works include: an extension of \mathcal{EL} with Bayesian networks called \mathcal{BEL} is presented in [6]. They study the complexity of reasoning under \mathcal{BEL} to show that reasoning is intractable. However, their work does not discuss probabilities in the ABox and concrete domains are excluded. On the other hand, in [7], they added uncertainty to DL-Lite based on Bayesian networks. Additionally, they have shown that satisfiability test and query answering in probabilistic DL-Lite can be reduced to satisfiability test and query answering in the DL-Lite family. Further, it is proved that satisfiability checking and union of conjunctive query answering can be done in LogSpace in the data complexity. Query answering in probabilistic OWL QL, where ABox assertions are probabilistic and TBox axioms are deterministic, has been studied in [13]. They prove that only very simple conjunctive queries can be answered in PTime, while most queries are #P-hard.

It is possible to extend DLs with uncertainty by using Halpern's probabilistic first order logic [12]. These studies include Prob- \mathcal{ALC} [25], P- $\mathcal{SHOIN}(\mathbf{D})$ [35]. On the other hand, P- $\mathcal{SHIQ}(\mathbf{D})$ uses probabilistic lexicographic entailment from probabilistic default reasoning [19]. P- $\mathcal{SHOIN}(\mathbf{D})$ and P- $\mathcal{SHIQ}(\mathbf{D})$ support datatype reasoning in a probabilistic setting. While these logics are different from the one studied in this paper, they also do not support expressive CDs. In addition, while LogEL is based on log-linear models, P- $\mathcal{SHIQ}(\mathbf{D})$ is based on Nilsson's probabilistic logic [28] and P- $\mathcal{SHOIN}(\mathbf{D})$ is based on Halpern's probabilistic FOL; both of these formalisms are different from log-linear models. A survey of probabilistic extensions of DLs can be found in [20].

Consequently, as discussed above, most of the studies that involve extending DLs to deal with uncertainty by using either Bayesian or MLNs often excluded CDs. This is partly due to either the lack of supporting features or the difficulty in dealing with them. In this paper, we studied a novel way of dealing with uncertainty involving CDs by log-linear models extended with numerical constraints.

8 Conclusion

In this paper, we extended MLN with numerical constraints. To show its usefulness, we have applied this framework to extend LogEL with instances and CDs. This work can be seen as a first step towards more practical uses of log-linear models with numerical constraints. We have illustrated the practical merits of the approach using an example application. What needs to be done is a systematic evaluation of the approach. Furthermore, there are two directions for improving the formalism. On the one hand, we can further extend the modeling capabilities by including other kinds of datatypes and probability distributions over attribute values. On the other hand, we will investigate tractable subsets of extended MLNs to enable polynomial time reasoning.

REFERENCES

- [1] James F Allen, ‘Maintaining knowledge about temporal intervals’, *Communications of the ACM*, **26**(11), 832–843, (1983).
- [2] Franz Baader, Sebastian Brandt, and Carsten Lutz, ‘Pushing the envelope’, in *IJCAI*, volume 5, pp. 364–369, (2005).
- [3] Vaishak Belle, Andrea Passerini, and Guy Van den Broeck, ‘Probabilistic inference in hybrid domains by weighted model integration’, in *Proceedings of 24th International Joint Conference on Artificial Intelligence (IJCAI)*, (2015).
- [4] Vaishak Belle, Guy Van den Broeck, and Andrea Passerini, ‘Hashing-based approximate probabilistic inference in hybrid domains’, in *Proceedings of the 31st Conference on Uncertainty in Artificial Intelligence (UAI)*, (2015).
- [5] Rodrigo de Salvo Braz, Ciaran O’Reilly, Vibhav Gogate, and Rina Dechter, ‘Probabilistic inference modulo theories’, (2016). to appear.
- [6] Ismail Ilkan Ceylan and Rafael Penaloza, ‘Bayesian description logics’, in *Proceedings of the 27th International Workshop on Description Logics (DL 2014). CEUR Workshop Proceedings*, volume 1193, pp. 447–458, (2014).
- [7] Claudia d’Amato, Nicola Fanizzi, and Thomas Lukasiewicz, ‘Tractable reasoning with bayesian description logics’, in *Scalable Uncertainty Management*, 146–159, Springer, (2008).
- [8] M Despoina, Y Kazakov, and I Horrocks, ‘Tractable extensions of the description logic el with numerical datatypes’, *Journal of Automated Reasoning*, (2011).
- [9] Lise Getoor, *Introduction to statistical relational learning*, MIT press, 2007.
- [10] Georg Gottlob, Thomas Lukasiewicz, Maria Vanina Martinez, and Gerardo I Simari, ‘Query answering under probabilistic uncertainty in datalog+/- ontologies’, *Annals of Mathematics and Artificial Intelligence*, **69**(1), 37–72, (2013).
- [11] Víctor Gutiérrez-Basulto, Jean Christoph Jung, Carsten Lutz, and Lutz Schröder, ‘A closer look at the probabilistic description logic prob-el’, in *AAAI*, (2011).
- [12] Joseph Y Halpern, ‘An analysis of first-order logics of probability’, *Artificial intelligence*, **46**(3), 311–350, (1990).
- [13] Jean Christoph Jung and Carsten Lutz, ‘Ontology-based access to probabilistic data with owl ql’, in *The Semantic Web–ISWC 2012*, 182–197, Springer, (2012).
- [14] Yevgeny Kazakov, Markus Krötzsch, and František Simančík, ‘Practical reasoning with nominals in the \mathcal{EL} family of description logics’, in *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR’12)*, eds., Gerhard Brewka, Thomas Eiter, and Sheila A. McIlraith, pp. 264–274. AAAI Press, (2012).
- [15] Yevgeny Kazakov, Markus Krötzsch, and František Simančík, ‘The incredible ELK: From polynomial procedures to efficient reasoning with \mathcal{EL} ontologies’, *Journal of Automated Reasoning*, **53**, 1–61, (2013).
- [16] Pavel Klinov, ‘Pronto: a non-monotonic probabilistic description logic reasoner’, in *Proceedings of the 5th European semantic web conference on The semantic web: research and applications*, pp. 822–826. Springer-Verlag, (2008).
- [17] Daphne Koller and Nir Friedman, *Probabilistic graphical models: principles and techniques*, MIT press, 2009.
- [18] Markus Krötzsch, ‘Efficient rule-based inferencing for owl el’, in *IJCAI*, volume 11, pp. 2668–2673, (2011).
- [19] Thomas Lukasiewicz, ‘Expressive probabilistic description logics’, *Artificial Intelligence*, **172**(6), 852–883, (2008).
- [20] Thomas Lukasiewicz, ‘Expressive probabilistic description logics’, *Artificial Intelligence*, **172**(6), 852–883, (2008).
- [21] Thomas Lukasiewicz, Maria Vanina Martinez, Giorgio Orsi, and Gerardo I. Simari, ‘Heuristic ranking in tightly coupled probabilistic description logics’, in *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, August 14-18, 2012*, pp. 554–563, (2012).
- [22] Carsten Lutz, ‘Description logics with concrete domains-a survey’, in *Advances in Modal Logic 4, papers from the fourth conference on "Advances in Modal logic," held in Toulouse (France) in October 2002*, pp. 265–296, (2002).
- [23] Carsten Lutz, ‘Combining interval-based temporal reasoning with general TBoxes’, *Artificial Intelligence*, **152**(2), 235–274, (2004).
- [24] Carsten Lutz and Maja Miličić, ‘A tableau algorithm for description logics with concrete domains and general tboxes’, *Journal of Automated Reasoning*, **38**(1-3), 227–259, (2007).
- [25] Carsten Lutz and Lutz Schröder, ‘Probabilistic description logics for subjective uncertainty’, in *KR*, (2010).
- [26] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling, ‘Never-ending learning’, in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*, (2015).
- [27] Mathias Niepert, Jan Noessner, and Heiner Stuckenschmidt, ‘Log-linear description logics’, in *IJCAI*, pp. 2153–2158, (2011).
- [28] Nils J Nilsson, ‘Probabilistic logic’, *Artificial intelligence*, **28**(1), 71–87, (1986).
- [29] Jan Noessner and Mathias Niepert, ‘Elog: a probabilistic reasoner for owl el’, in *Web Reasoning and Rule Systems*, 281–286, Springer, (2011).
- [30] Jan Noessner, Mathias Niepert, and Heiner Stuckenschmidt, ‘Rockit: Exploiting parallelism and symmetry for map inference in statistical relational models’, in *Twenty-Seventh AAAI Conference on Artificial Intelligence*, (2013).
- [31] Jan Noessner, Heiner Stuckenschmidt, Christian Meilicke, and Mathias Niepert, ‘Completeness and optimality in ontology alignment debugging’, in *Proceedings of the 9th International Workshop on Ontology Matching collocated with the 13th International Semantic Web Conference (ISWC 2014)*, pp. 25–36, Riva del Garda, Trentino, Italy, (2014).
- [32] Matthew Richardson and Pedro Domingos, ‘Markov logic networks’, *Machine learning*, **62**(1-2), 107–136, (2006).
- [33] S Riedel, ‘Improving the accuracy and efficiency of map inference for markov logic’, in *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence, UAI 2008*, pp. 468–475, (2008).
- [34] Sebastian Riedel, ‘Improving the accuracy and efficiency of map inference for markov logic’, *arXiv preprint arXiv:1206.3282*, (2012).
- [35] Fabrizio Riguzzi, Elena Bellodi, Evelina Lamma, and Riccardo Zese, ‘Reasoning with probabilistic ontologies’, in *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pp. 4310–4316. AAAI Press, (2015).
- [36] Jue Wang and Pedro M Domingos, ‘Hybrid markov logic networks.’, in *AAAI*, volume 8, pp. 1106–1111, (2008).

Revisiting the Cross Entropy Method with Applications in Stochastic Global Optimization and Reinforcement Learning

Ajin George Joseph¹ and Shalabh Bhatnagar²

Abstract. In this paper, we provide a new algorithm for the problem of stochastic global optimization where only noisy versions of the objective function are available. The algorithm is inspired by the well known cross entropy (CE) method. The algorithm takes the shape of a multi-timescale stochastic approximation algorithm, where we reuse the previous samples based on discounted averaging, and hence it saves the overall computational and storage cost. We provide proof of the stability and the global optimization property of our algorithm. The algorithm shows good performance on the noisy versions of global optimization benchmarks and outperforms a state-of-the-art algorithm for non-linear function approximation in reinforcement learning.

1 Introduction and Preliminaries

In this paper, we solve the following problem: For a latent probability measure \mathbb{P}_y over $\mathcal{Y} \subset \mathbb{R}^a$,

$$\text{Find } x^* \in \arg \max_{x \in \mathcal{X} \subset \mathbb{R}^m} L(\mathbb{E}_y [J(y)], x), \quad (1)$$

where $L : \mathbb{R}^b \times \mathbb{R}^m \rightarrow \mathbb{R}$ is a bounded continuous function and $J : \mathcal{Y} \rightarrow \mathbb{R}^b$ is a bounded function (hence $\mathbb{E}_y [||J||] < \infty$). Here $\mathbb{E}_y [\cdot]$ is the expectation w.r.t. \mathbb{P}_y . The region \mathcal{X} referred as the *solution space* is a compact subset of \mathbb{R}^m . For brevity, we let $\mathcal{H}(x) \triangleq L(\mathbb{E}_y [J(y)], x)$. In this paper, we assume x^* is unique and $x^* \in \text{interior}(\mathcal{X})$.

In this paper, we consider a “black-box” setting, where neither a closed form expression nor structural properties of the objective function \mathcal{H} are available. However, for a given $x \in \mathcal{X}$, a noisy measurement $\tilde{\mathcal{H}}(x)$ of the objective function \mathcal{H} is available, where $\tilde{\mathcal{H}}(x) = \mathcal{H}(x) + \epsilon(x)$. Here $\epsilon(x)$ is the noise incurred during the measurement of $\mathcal{H}(x)$ which is primarily attributed to the inability to measure accurately the quantity $\mathbb{E}_y [\cdot]$. The intractability of $\mathbb{E}_y [\cdot]$ is due to the hidden probability measure \mathbb{P}_y . Problems of the above kind are found in areas of engineering and in discrete-event system simulation [11, 13].

Intuitively, it follows directly from the definition of the problem that any algorithm which solves the problem of the above kind has to be a zero-order method. Since the higher order values of the objective function are unavailable, the algorithm relies only on the function values. These methods are generally called *gradient free* methods. A few prominent algorithms of this kind are simultaneous perturbation stochastic approximation method (SPSA) [26], model reference adaptive search method (MRAS) [8], cross entropy method (CE)

[25], estimation of distribution algorithms (EDA) [29] and gradient based adaptive search (GASSO) [30]. The advantage of the above methods is the non-dependency on the structural properties of the objective function and hence can be applied in a generalized setting.

In this paper, we consider the well known cross entropy method. The cross entropy method was motivated by the rare event probability estimation method proposed in [22], which is a variance reduction technique. Later, an iterative procedure based on the above method was applied in various combinatorial optimization problems [23]. Cross entropy method also found successful applications in continuous multi-extremal optimization [21]. A few other applications of the CE method include buffer allocation [1], queuing models [3], DNA sequence alignment [14], control and navigation [6], reinforcement learning [17, 19] and several NP-hard problems [24, 21].

In this paper, we apply a stochastic approximation variant of the CE method to the continuous multi-extremal stochastic optimization problem defined in (1). This particular version of the algorithm is designed to overcome the drawbacks arising due to the inordinate computational and storage requirements of the original CE method. The stochastic approximation nature of our algorithm also helps to adapt naturally to the stochastic optimization setting.

2 The Cross Entropy Method

Notation: We use \mathbf{x} for random variable and x for deterministic variable. Let $\lceil a \rceil$ denote the smallest integer greater than a . Let $\text{supp}(f) \triangleq \{x | f(x) \neq 0\}$ and $\text{interior}(A)$ be the *interior* of set A . Let $f_\theta(\cdot)$ denote the *probability density function* parametrized by θ and $\mathbb{E}_\theta[\cdot]$ be the *expectation* w.r.t. f_θ . Let $\gamma_\rho(\mathcal{H}(\cdot), \theta)$ denote the $(1 - \rho)$ -*quantile* of $\mathcal{H}(\mathbf{x})$ w.r.t. f_θ , i.e.,

$$\gamma_\rho(\mathcal{H}(\cdot), \theta) \triangleq \sup\{l : P_\theta(\mathcal{H}(\mathbf{x}) \geq l) \geq \rho\}. \quad (2)$$

$$\text{For } x, y \in \mathbb{R}, \text{ define } \chi(x, y) = \begin{cases} 1 & \text{if } x \geq y \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

The *Cross Entropy (CE) method* [25, 9, 4] is a zero-order optimization method to solve optimization problems where the objective function does not possess smooth differentiable structural properties. CE method belongs to a broader class of methods called the *model based search methods*. The goal of the CE method is to find an optimal “*model*” or probability distribution over the solution space \mathcal{X} which concentrates on the global maxima of the objective function. The CE method is an iterative procedure where at each iteration k , a search is conducted on a space of parametrized probability distributions $\{f_\theta | \theta \in \Theta\}$ on \mathcal{X} , where Θ is the parameter space, to find a

¹ Indian Institute of Science, Bangalore, email: ajin@csa.iisc.ernet.in

² Indian Institute of Science, Bangalore, email: shalabh@csa.iisc.ernet.in

distribution parameter θ_k which reduces the *Kullback-Leibler (KL)*³ distance from the optimal model. The most commonly used distribution family here is the *exponential family of distributions*.

Exponential Family of Distributions: These distributions are represented by $\mathcal{C} \triangleq \{f_\theta(x) = h(x)e^{\theta^\top \Gamma(x) - K(\theta)} \mid \theta \in \Theta \subset \mathbb{R}^d\}$, where $h : \mathbb{R}^m \rightarrow \mathbb{R}$, $\Gamma : \mathbb{R}^m \rightarrow \mathbb{R}^d$ and $K : \mathbb{R}^d \rightarrow \mathbb{R}$ and $K(\theta) = \log \int h(x)e^{\theta^\top \Gamma(x)} dx$. The Gaussian distribution with mean vector $\mu \in \mathbb{R}^m$ and the covariance matrix $\Sigma \in \mathbb{R}^{m \times m}$ belongs to \mathcal{C} . In this case,

$$f_\theta(x) = \frac{1}{\sqrt{(2\pi)^m |\Sigma|}} e^{-\frac{(x-\mu)^\top \Sigma^{-1} (x-\mu)}{2}}, \quad (4)$$

and so one may let $h(x) = 1/(2\pi)^{m/2}$, $\Gamma(x) = (x, xx^\top)^\top$ and $\theta = (\Sigma^{-1}\mu, -\frac{1}{2}\Sigma^{-1})^\top$.

Assumption (A1): The parameter space Θ is a compact set.

2.1 CE Method (Ideal Version)

The CE method aims to find a sequence of model parameters $\{\theta_k \in \Theta\}_{k \in \mathbb{Z}^+}$ and an increasing sequence of thresholds $\{\gamma_k \in \mathbb{R}\}_{k \in \mathbb{Z}^+}$ with the property that the support of f_{θ_k} satisfies $\text{supp}(f_{\theta_k}) \subseteq \{x \mid \mathcal{H}(x) \geq \gamma_k\}$. By assigning greater weight to the higher values of \mathcal{H} at each iteration, the expected behaviour of the probability distribution sequence should improve. This is achieved by solving at each instant $k+1$, the following optimization problem: $\theta_{k+1} = \arg \min_{\theta \in \Theta} KL(\hat{f}_{\theta_k}, f_\theta)$, where $\hat{f}_{\theta_k}(x) = \varphi(\mathcal{H}(x))\chi(\mathcal{H}(x), \gamma_{k+1})f_{\theta_k}(x)$. Further simplification yields,

$$\theta_{k+1} = \arg \max_{\theta \in \Theta} \mathbb{E}_{\theta_k} [\varphi(\mathcal{H}(\mathbf{x}))\chi(\mathcal{H}(\mathbf{x}), \gamma_{k+1}) \log f_\theta(\mathbf{x})], \quad (5)$$

where $\varphi : \mathbb{R} \rightarrow \mathbb{R}_+$ is positive and strictly monotonically increasing. The most common choice for γ_{k+1} is $\gamma_\rho(\mathcal{H}(\cdot), \theta_k)$: the $(1-\rho)$ -quantile of $\mathcal{H}(\mathbf{x})$ w.r.t. f_{θ_k} , where $\rho \in (0, 1)$ is set *a priori*. Also, the parameter space Θ is large enough so that the solution of (5) is contained in *interior*(Θ).

We take Gaussian distribution as the preferred choice for $f_\theta(\cdot)$ in this paper. In this case, the model parameter is $\theta = (\mu, \Sigma)^\top$, where $\mu \in \mathbb{R}^m$ is the mean vector and $\Sigma \in \mathbb{R}^{m \times m}$ is the covariance matrix. We obtain a closed-form expression for μ_{k+1} and Σ_{k+1} by equating to 0 the gradient w.r.t. $(\Sigma^{-1}\mu, -\frac{1}{2}\Sigma^{-1})^\top$ of the objective function in (5) and using (4) for $f_\theta(\cdot)$. We obtain

$$\mu_{k+1} = \frac{\mathbb{E}_{\theta_k} [\mathbf{g}_1(\mathcal{H}(\mathbf{x}), \mathbf{x}, \gamma_{k+1})]}{\mathbb{E}_{\theta_k} [\mathbf{g}_0(\mathcal{H}(\mathbf{x}), \gamma_{k+1})]}, \quad (6)$$

$$\Sigma_{k+1} = \frac{\mathbb{E}_{\theta_k} [\mathbf{g}_2(\mathcal{H}(\mathbf{x}), \mathbf{x}, \gamma_{k+1}, \mu_{k+1})]}{\mathbb{E}_{\theta_k} [\mathbf{g}_0(\mathcal{H}(\mathbf{x}), \gamma_{k+1})]}, \quad (7)$$

where $\mathbf{g}_0(\mathcal{H}(x), \gamma) \triangleq \varphi(\mathcal{H}(x))\chi(\mathcal{H}(x), \gamma)$,

$$\mathbf{g}_1(\mathcal{H}(x), x, \gamma) \triangleq \varphi(\mathcal{H}(x))\chi(\mathcal{H}(x), \gamma)x \quad \text{and}$$

$$\mathbf{g}_2(\mathcal{H}(x), x, \gamma, \mu) \triangleq \varphi(\mathcal{H}(x))\chi(\mathcal{H}(x), \gamma)(x - \mu)(x - \mu)^\top.$$

Remark 1: Note that in the expression of μ_{k+1} in (6), x is being weighted with $\varphi(\mathcal{H}(x))$ in the region $\{x \mid \mathcal{H}(x) \geq \gamma_{k+1}\}$. Since the function φ is positive and strictly monotonically increasing, the region where $\mathcal{H}(x)$ is higher (hence $\varphi(\mathcal{H}(x))$ is also higher) is given more weight and hence μ_{k+1} concentrates in the region where $\mathcal{H}(x)$ takes higher values. In case where $\mathcal{H}(\cdot)$ is positive, we can choose

³ The Kullback-Leibler distance between two probability distributions g_1 and g_2 is $KL(g_1, g_2) \triangleq \mathbb{E}_{g_1} \left[\log \frac{g_1}{g_2} \right]$

$\varphi(x) = x$. However, in general scenarios, where $\mathcal{H}(\cdot)$ takes positive and negative values, the identity function is not an appropriate choice since the effect of the positive weights is reduced by the negative ones. In such cases, we take $\varphi = \exp(rx)$, $r \in \mathbb{R}_+$.

2.2 CE Method (Monte-Carlo Version)

It is hard in general to evaluate $\mathbb{E}_{\mathbf{y}} [\cdot]$. Hence the objective function values $\mathcal{H}(x)$ might not be available for every value of $x \in \mathcal{X}$. To overcome this, estimates obtained using sample averages are used. The algorithm utilizes a user configured observation allocation rule $\{M_k \in \mathbb{Z}_+\}_{k \in \mathbb{Z}_+}$ to decide the sample size, where $M_k \uparrow \infty$. This means at each iteration k , for a given $x \in \mathcal{X}$, the estimate $\bar{\mathcal{H}}(x)$ is obtained as follows:

$$\bar{\mathcal{H}}(x) = L\left(\frac{1}{M_k} \sum_{i=1}^{M_k} J(\mathbf{y}_i), x\right), \quad \text{where } \mathbf{y}_i \sim \mathbb{P}_{\mathbf{y}}. \quad (8)$$

Also hard to compute are the terms $\mathbb{E}_{\theta_k} [\cdot]$ and γ_{k+1} ($= \gamma_\rho(\mathcal{H}(\cdot), \theta_k)$) in equations (6) and (7). To overcome this, their corresponding stochastic counterparts are employed. Here also we use a user configured observation allocation rule $\{N_k \in \mathbb{Z}_+\}_{k \in \mathbb{Z}_+}$ to decide the sample size, where $N_k \uparrow \infty$. In this Monte-Carlo version, the algorithm generates sequences $\{\bar{\theta}_k = (\bar{\mu}_k, \bar{\Sigma}_k)^\top\}_{k \in \mathbb{Z}_+}$ and $\{\bar{\gamma}_k\}_{k \in \mathbb{Z}_+}$ as follows: At each iteration k , N_k samples $\Lambda_k = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{N_k}\}$ are chosen using $f_{\bar{\theta}_k}$ and the threshold $\bar{\gamma}_{k+1}$ is obtained as follows:

$$\bar{\gamma}_{k+1} = \bar{\mathcal{H}}_{(\lceil(1-\rho)N_k\rceil)}, \quad (9)$$

where $\bar{\mathcal{H}}_{(i)}$ is the i th-order statistic of $\{\bar{\mathcal{H}}(\mathbf{x}_i)\}_{i=1}^{N_k}$. The model parameter update also uses sample averages. The model parameter $\bar{\theta}_{k+1} = (\bar{\mu}_{k+1}, \bar{\Sigma}_{k+1})^\top$ is updated as follows:

$$\bar{\mu}_{k+1} = \frac{\frac{1}{N_k} \sum_{i=1}^{N_k} \mathbf{g}_1(\mathcal{H}(\mathbf{x}_i), \mathbf{x}_i, \bar{\gamma}_{k+1})}{\frac{1}{N_k} \sum_{i=1}^{N_k} \mathbf{g}_0(\mathcal{H}(\mathbf{x}_i), \bar{\gamma}_{k+1})}, \quad (10)$$

$$\bar{\Sigma}_{k+1} = \frac{\frac{1}{N_k} \sum_{i=1}^{N_k} \mathbf{g}_2(\mathcal{H}(\mathbf{x}_i), \mathbf{x}_i, \bar{\gamma}_{k+1}, \bar{\mu}_{k+1})}{\frac{1}{N_k} \sum_{i=1}^{N_k} \mathbf{g}_0(\mathcal{H}(\mathbf{x}_i), \bar{\gamma}_{k+1})}. \quad (11)$$

The Monte-Carlo version is given in Algorithm 1.

Algorithm 1: The Monte-Carlo CE Algorithm

Step 0: Choose an initial *p.d.f.* $f_{\bar{\theta}_0}(\cdot)$ on \mathcal{X} , where $\bar{\theta}_0 = (\bar{\mu}_0, \bar{\Sigma}_0)^\top$ and fix an $\epsilon > 0$ (dependent on \mathcal{H}).

1. [Sampling Candidate Solutions]: Sample N_k *i.i.d.* solutions $\Lambda_k = \{\mathbf{x}_1, \dots, \mathbf{x}_{N_k}\}$ using $f_{\bar{\theta}_k}$.

2. [Performance Evaluation]: For each \mathbf{x} in Λ_k , take M_k observations $\{\mathbf{y}_i\}_{i=1}^{M_k}$, where $\mathbf{y}_i \sim \mathbb{P}_{\mathbf{y}}$ and calculate the sample average $\bar{\mathcal{H}}(\mathbf{x}) = L\left(\frac{1}{M_k} \sum_{i=1}^{M_k} J(\mathbf{y}_i), \mathbf{x}\right)$.

3. [Threshold Evaluation]: Calculate the sample $(1-\rho)$ -quantile $\bar{\gamma}_{k+1} = \bar{\mathcal{H}}_{(\lceil(1-\rho)N_k\rceil)}$, $\bar{\mathcal{H}}_{(i)}$ is the i th-order statistic of the sequence $\{\bar{\mathcal{H}}(\mathbf{x}_i)\}_{i=1}^{N_k}$.

4. [Threshold Comparison]

if $\bar{\gamma}_{k+1} \geq \bar{\gamma}_k^* + \epsilon$ **then**

$\bar{\gamma}_{k+1} = \bar{\gamma}_{k+1}$,

else

$\bar{\gamma}_{k+1} = \bar{\gamma}_k^*$.

5. [Model parameter update]: Update $\bar{\theta}_{k+1} = (\bar{\mu}_{k+1}, \bar{\Sigma}_{k+1})^\top$ using (10) and (11) with $\bar{\gamma}_{k+1}^*$ instead of $\bar{\gamma}_{k+1}$.

6: If the stopping rule is satisfied, then return $\bar{\theta}_{k+1}$ and terminate, else set $k := k+1$ and go Step 1.

2.3 Drawbacks of the CE Method

• *Inefficient use of prior information:* The naive approach of the Monte-Carlo CE does not utilize prior information efficiently. Note that Monte-Carlo CE possesses a stateless behaviour. At each iteration k , a completely new collection of samples are drawn using the distribution f_{θ_k} . The samples are used to derive the estimates $\bar{\gamma}_{k+1}$, $\bar{\mu}_{k+1}$ and $\bar{\Sigma}_{k+1}$. The algorithm does not utilize the estimates generated prior to k .

• *Computational limitations:* These arise due to the dependence on the sample size N_k . One does not know a priori *the best value for the sample size N_k* . Higher values of N_k while resulting in higher accuracy also require more computational resources. One often needs to apply brute force in order to obtain a good choice of N_k . Also as m , the dimension of the solution space \mathcal{X} , takes large values, more samples are required for better accuracy, making N_k large as well. This makes *finding the i th-order statistic $\bar{\mathcal{H}}_{(i)}$ in Step 3 harder*. Note that the order statistic $\bar{\mathcal{H}}_{(i)}$ is obtained by sorting the list $\{\bar{\mathcal{H}}(\mathbf{x}_1), \bar{\mathcal{H}}(\mathbf{x}_2), \dots, \bar{\mathcal{H}}(\mathbf{x}_{N_k})\}$. The computational effort required in that case is $O(N_k \log N_k)$ which is expensive. Also note that N_k diverges to infinity and hence this super linear relationship is computationally very costly.

• *Storage limitations:* The storage requirement for storing sample Λ_k is $N_k * \text{size}(\mathbf{x}_k)$. In situations when m and N_k are large, the storage requirement is a major concern.

Note that similar issues are expected to arise with the sample size M_k also.

An illustration in Figure 1 demonstrates the dependency of the performance of Monte-Carlo CE on the sample size schedules $\{N_k\}$ and $\{M_k\}$. We consider here, the Griewank function on \mathbb{R}^{100} -cf. (36). We take $M_{k+1} = \lceil \alpha_1 M_k \rceil$, $\alpha_1 > 1$ and $N_{k+1} = \lceil \alpha_2 N_k \rceil$, $\alpha_2 > 1$. So a particular schedule can be identified by M_0, α_1 and N_0, α_2 . Here we take $\alpha_1 = 1.01$ and $\alpha_2 = 1.005$ for all the schedules, however they differ in their initial values M_0 and N_0 . From the Figure 1, note that for $(M_0, N_0) \in \{(100, 100), (150, 150), (200, 200)\}$, the convergence behaviour is close, however for $(M_0, N_0) \in \{(500, 500), (1000, 1000), (2000, 2000)\}$, Monte-Carlo CE converges to a better value. Also when observed carefully, we can notice a significant difference in the limit point of each individual trajectory.

Different variants of the naive Monte-Carlo CE have been proposed in the literature, such as the gradient based Monte-Carlo cross entropy method (GMCCCE) [10] and the modified Monte-Carlo cross entropy method [28]. All the variants differ only in the model updating step, the other steps remain the same. Hence they also suffer from the above drawbacks.

3 Proposed Algorithm

In this paper, we resolve the shortcomings of the Monte-Carlo CE method with regards to the concerns mentioned in the previous section, by remodelling the same in the stochastic approximation framework. We follow the same sequence of steps as in Algorithm 1, but differ in their implementation. The stochastic approximation approach streamlines the batch processing of the Monte-Carlo CE. We provide a multi-timescale stochastic approximation algorithm which is efficient, stable, incremental in nature and imposes minimal restriction on the objective function. We avail ourselves of the continuity relationship that holds between the $(1 - \rho)$ -quantile $\gamma_\rho(\mathcal{H}(\cdot), \theta)$ and the model parameter θ . We also exploit the continuity that holds between μ_{k+1} and Σ_{k+1} w.r.t. θ_k . The bootstrapping property inherent in the stochastic approximation algorithms helps to utilize these

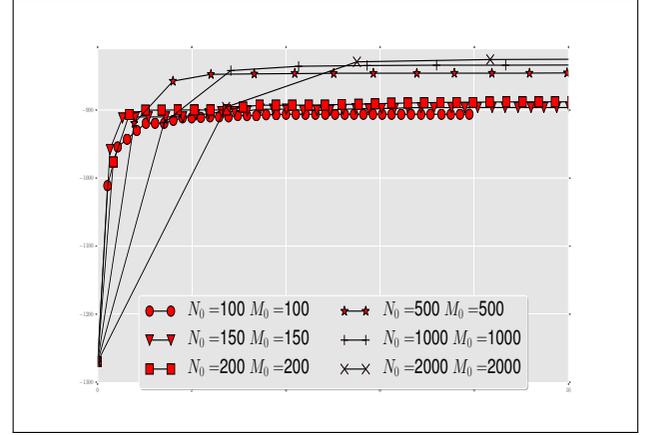


Figure 1: Plot of $\mathcal{H}(\bar{\mu}_k)$, where \mathcal{H} is the Griewank function. The plot shows the dependency of Monte-Carlo CE on the schedules $\{N_k\}$ and $\{M_k\}$.

relationships efficiently.

Stochastic approximation algorithms [2, 15, 20] are a natural way of utilizing prior information. It does so by discounted averaging of the prior information and are usually expressed as recursive equations of the following form:

$$\mathbf{z}_{k+1} = \mathbf{z}_k + \alpha_{k+1} \Delta \mathbf{z}(\mathbf{z}_k, b_k, D_{k+1}), \quad (12)$$

where $\Delta \mathbf{z}(z, b, D) = h(z) + b + D$ is the increment term, b_k is the bias term with $b_k \rightarrow 0$, D_k is a random noise with zero-mean and $h(\cdot)$ a Lipschitz continuous function. The learning rate α_k satisfies $\sum \alpha_k = \infty, \sum \alpha_k^2 < \infty$.

In our algorithm, we do not apply the naive order statistic to estimate the $(1 - \rho)$ -quantile. Rather we employ a stochastic recursion which serves the same purpose, but efficiently. To achieve this, we make use of the following lemma from [7]. This lemma provides a characterization of the quantile in terms of an optimization problem.

Lemma 1 *The $(1 - \rho)$ -quantile of a bounded real-valued function $H(\cdot)$ (with $H(x) \in [H_1, H_2]$) w.r.t the probability distribution $f_\theta(\cdot)$ is reformulated as the optimization problem*

$$\text{Find } \gamma_\rho(H(\cdot), \theta) = \min_{u \in [H_1, H_2]} \mathbb{E}_\theta [\psi(H(\mathbf{x}), u)], \quad (13)$$

where $\psi(H(x), u) = (1 - \rho)(H(x) - u)\chi(H(x), u) + \rho(u - H(x))\chi(u, H(x))$.

We develop a stochastic gradient recursion which solves the optimization problem in (13). The increment term for the recursion is the sub-differential of ψ w.r.t. u , and is given by

$$\nabla_u \psi(H(x), u) = -(1 - \rho)\chi(H(x), u) + \rho\chi(u, H(x)). \quad (14)$$

The model parameter update step involves three stochastic recursions. For this we introduce two new variables, $\xi^{(0)}$ and $\xi^{(1)}$, which estimate the RHS of equations (6) and (7) respectively. We also require two increment functions which are defined as follows:

$$\Delta \xi^{(0)}(x, \omega, \gamma) = \mathbf{g}_1(L(\omega, x), x, \gamma) - \xi^{(0)} \mathbf{g}_0(L(\omega, x), \gamma), \quad (15)$$

$$\Delta \xi^{(1)}(x, \omega, \mu, \gamma) = \mathbf{g}_2(L(\omega, x), x, \gamma, \mu) - \xi^{(1)} \mathbf{g}_0(L(\omega, x), \gamma). \quad (16)$$

The algorithm is formally presented in Algorithm 2.

Algorithm 2:

Data: $\alpha_k, \beta_k, \lambda_k, c \in (0, 1), \epsilon_1 \in (0, 1)$;
Init: $k = 0, \gamma_0 = 0, \gamma_0^* = -\infty, T_0 = 0, \lambda = \lambda_0, \theta_{old} = NULL,$
 $\theta_0 = (\mu_0, \Sigma_0)^\top, \xi_k^{(0)} = 0_{m \times 1}, \xi_k^{(1)} = 0_{m \times m}, \omega_0 = 0_{b \times 1}$;

while stopping criteria is not satisfied do

$\mathbf{y}_{k+1} \sim \mathbb{P}_{\mathbf{y}}$;

- [Estimating the objective function]

$$\omega_{k+1} = \omega_k + \alpha_{k+1} (J(\mathbf{y}_{k+1}) - \omega_k); \quad (17)$$

- [Generating the mixture distribution]

$$\tilde{f}_{\theta_k} = (1 - \lambda)f_{\theta_k} + \lambda f_{\theta_0};$$

$$\mathbf{x}_{k+1} \sim \tilde{f}_{\theta_k};$$

- [Estimating the $(1 - \rho)$ -quantile]

$$\gamma_{k+1} = \gamma_k - \beta_{k+1} \nabla_u \psi(L(\omega_k, \mathbf{x}_{k+1}), \gamma_k); \quad (18)$$

- [Estimating the RHS of equations (6) and (7)]

$$\xi_{k+1}^{(0)} = \xi_k^{(0)} + \beta_{k+1} \Delta \xi^{(0)}(\mathbf{x}_{k+1}, \omega_k, \gamma_k); \quad (19)$$

$$\xi_{k+1}^{(1)} = \xi_k^{(1)} + \beta_{k+1} \Delta \xi^{(1)}(\mathbf{x}_{k+1}, \omega_k, \xi_k^{(0)}, \gamma_k); \quad (20)$$

if $\theta_{old} \neq NULL$ **then**

$$\tilde{f}_{\theta_{old}} = (1 - \lambda)f_{\theta_{old}} + \lambda f_{\theta_0}; \quad \mathbf{x}_{k+1}^{old} \sim \tilde{f}_{\theta_{old}};$$

$$\gamma_{k+1}^* = \gamma_k^* - \beta_{k+1} \nabla_u \psi(L(\omega_k, \mathbf{x}_{k+1}^{old}), \gamma_k^*); \quad (21)$$

- [Threshold comparison]

$$T_{k+1} = T_k + c(\chi(\gamma_{k+1}, \gamma_{k+1}^*) - \chi(\gamma_{k+1}^*, \gamma_{k+1}) - T_k); \quad (22)$$

- [Updating the model parameters]

if $T_{k+1} > \epsilon_1$ **then**

$$\theta_{old} = \theta_k;$$

$$\theta_{k+1} = \theta_k + \alpha_{k+1} \left((\xi_k^{(0)}, \xi_k^{(1)})^\top - \theta_k \right); \quad (23)$$

$$\gamma_{k+1}^* = \gamma_k; \quad T_k = 0; \quad \lambda = \lambda_k; \quad (24)$$

else

$$\gamma_{k+1} = \gamma_k^*; \quad \theta_{k+1} = \theta_k;$$

$$k := k + 1;$$

Note that the algorithm uses only 3 samples per iteration: $\mathbf{y}_{k+1}, \mathbf{x}_{k+1}$ and \mathbf{x}_{k+1}^{old} . This is a big improvement both computational and storage wise, compared to the original CE method.

Mixture Distribution: In the algorithm, we use a mixture distribution \tilde{f}_{θ_k} to generate the sample \mathbf{x}_{k+1} , where $\tilde{f}_{\theta_k} = (1 - \lambda)f_{\theta_k} + \lambda f_{\theta_0}$ with λ the mixing weight. λ takes its values from a pre-defined decaying sequence $\{\lambda_k\}_{k \in \mathbb{Z}_+}$, with assignment happening in (24) during the model parameter update step. The initial distribution parameter θ_0 is chosen *s.t.* the density function f_{θ_0} is strictly positive on every point in the solution space \mathcal{X} , *i.e.*, $f_{\theta_0}(x) > 0, \forall x \in \mathcal{X}$. The mixture approach facilitates exploration of the solution space and prevents the iterates from getting stranded in suboptimal solutions.

Learning Rates: The learning rates α_k, β_k and the mixing weight

λ_k are deterministic, non-increasing and satisfy the following:

$$\begin{aligned} \lambda_k > 0, \alpha_k > 0, \beta_k > 0, \quad \lim_{k \rightarrow \infty} \lambda_k = 0, \\ \sum_{k=1}^{\infty} \alpha_k = \sum_{k=1}^{\infty} \beta_k = \infty, \quad \sum_{k=1}^{\infty} (\alpha_k^2 + \beta_k^2) < \infty, \quad \lim_{k \rightarrow \infty} \frac{\alpha_k}{\beta_k} = 0. \end{aligned} \quad (25)$$

Since $\alpha_k \rightarrow 0$ faster than β_k , the timescale obtained from $\beta_k, k \geq 0$ is faster as compared to the other.

Threshold Comparison Step: The threshold comparison is achieved using the recursion (22) of the random variable T_k . The model parameter θ_k is not updated at each k . Rather it is updated whenever T_k arches over ϵ_1 , where $\epsilon_1 \in (0, 1)$ is a constant. So the update of θ_k only happens along a subsequence $\{k_{(n)}\}_{n \in \mathbb{Z}_+}$ of $\{k\}_{k \in \mathbb{Z}_+}$. Between $k = k_{(n)}$ and $k = k_{(n+1)}$, the variable γ_k estimates $(1 - \rho)$ -quantile of $L(\omega_k, \cdot)$ *w.r.t.* $\tilde{f}_{\theta_{k_{(n)}}}$. The threshold γ_k^* is also updated in (24) during the ϵ_1 crossover. Thus $\gamma_{k_{(n)}}^*$ is the estimate of $(1 - \rho)$ -quantile *w.r.t.* $\tilde{f}_{\theta_{k_{(n-1)}}}$.

Notation: We denote by $\gamma_\rho(L(\omega, \cdot), \tilde{\theta})$, the $(1 - \rho)$ -quantile of $L(\omega, \cdot)$ *w.r.t.* the mixture distribution \tilde{f}_θ and let $E_{\tilde{\theta}}[\cdot]$ be the expectation *w.r.t.* \tilde{f}_θ .

Proposition 1: T_k belongs to $(-1, 1), \forall k > 0$.

Proof: By rearranging (22) we get,

$$T_{k+1} = (1 - c)T_k + c(\chi(\gamma_{k+1}, \gamma_{k+1}^*) - \chi(\gamma_{k+1}^*, \gamma_{k+1})),$$

where $c \in (0, 1)$. In the worst case, either $\chi(\gamma_{k+1}, \gamma_{k+1}^*) = 1, \forall k$ or $\chi(\gamma_{k+1}^*, \gamma_{k+1}) = 1, \forall k$. Since the two events are mutually exclusive, we will only consider the former event $\{\chi(\gamma_{k+1}, \gamma_{k+1}^*) = 1, \forall k\}$. In this case

$$\lim_{k \rightarrow \infty} T_k = \lim_{k \rightarrow \infty} (c + c(1 - c) + \dots + c(1 - c)^{k-1}) = 1.$$

Similarly for $\{\chi(\gamma_{k+1}^*, \gamma_{k+1}) = 1, \forall k\}$, we have $T_k \rightarrow -1$. ■

T_k in (22) serves two purposes: first, it is a delay mechanism, whereby it stalls the θ_k -recursion so that $\xi_k^{(0)}$ and $\xi_k^{(1)}$ are sufficiently close to their true values. Second, it ensures that the estimates γ_k eventually become greater than the current threshold $\gamma_{k_{(n)}}^*$, *i.e.*, $\gamma_k > \gamma_{k_{(n)}}^*$ for all but finitely many k .

Remark 2: The recursion (21) is not addressed in the discussion above. The assignment of γ_{k+1}^* in (24) happens along a subsequence $\{k_{(n)}\}_{n \geq 0}$. Hence $\gamma_{k_{(n)}}^*$ is the estimate of $\gamma_\rho(L(\omega_{k_{(n)}}(\cdot), \tilde{\theta}_{k_{(n-1)}}))$. But at time $k_{(n)} < k \leq k_{(n+1)}$, γ_k^* is compared with γ_k in (22). But γ_k is derived from a better estimate of $L(\omega_k, \cdot)$. Equation (21) ensures that γ_k^* is updated using the latest estimate of $L(\omega_k, \cdot)$. The variable θ_{old} holds the model parameter $\theta_{k_{(n-1)}}$ and the update of γ_k^* in (21) is performed using \mathbf{x}_{k+1}^{old} sampled from $\tilde{f}_{\theta_{old}}$.

3.1 Convergence Analysis

Assumption (A2): The sequences $\{\omega_k\}_{k \in \mathbb{Z}_+}$ and $\{\gamma_k\}_{k \in \mathbb{Z}_+}$ in (17) and (18) resp. satisfy $\sup_k |\omega_k| < \infty$ and $\sup_k |\gamma_k| < \infty$ *a.s.*

Remark 3: The assumption (A2) is a technical requirement to prove convergence of the algorithm. A commonly used procedure to ensure almost sure boundedness of iterates in a stochastic iterative scheme is to project these after each update to an a priori chosen (large enough) compact and convex set. In this case, the bound on the compact set can be derived from the bound on $L(\cdot, \cdot)$.

Note that the recursion (17) is independent of other recursions and

hence can be analysed independently. For the recursion (17) we have the following result.

Lemma 2 *Let the step-sizes α_k and β_k , $k \in \mathbb{Z}_+$ satisfy (25). Also let (A2) hold. Then the iterates ω_k in (17) satisfy $\omega_k \rightarrow \omega^* = \mathbb{E}_{\mathcal{Y}} [J(\mathbf{y})]$ as $k \rightarrow \infty$ w.p. 1.*

As noted above, the update of θ_k only happens along a subsequence $\{k_{(n)}\}_{n \in \mathbb{Z}_+}$ of $\{k\}_{k \in \mathbb{Z}_+}$. So between $k = k_{(n)}$ and $k = k_{(n+1)}$, θ_k is constant. The lemma and the theorems that follow in this paper depend on the timescale difference in the step-size schedules $\{\alpha_k\}_{k \in \mathbb{Z}_+}$ and $\{\beta_k\}_{k \in \mathbb{Z}_+}$. The step-size $\{\beta_k\}_{k \in \mathbb{Z}_+}$ decays to 0 at a slower rate than $\{\alpha_k\}_{k \in \mathbb{Z}_+}$ and hence the increments in the recursions (18), (19) and (20) which are controlled by β_k are larger and hence converge faster than the recursions (17) and (23) which are controlled by α_k . So the relative evolution of the variables from the slower timescale α_k i.e. ω_k , θ_k is slow and can be considered constant when viewed from the faster timescale β_k . See Chapter 6, [2] for a description on multi-timescale algorithms.

Hence, when viewed from the timescale of the recursion (18), one may consider ω_k and θ_k to be fixed. For recursion (18) we have the following result:

Lemma 3 *Let $L(\omega_k, \cdot) \equiv L(\omega, \cdot)$, $\theta_k \equiv \theta$, $\forall k$. Also let (A2) hold. Then γ_k in (18) satisfy $\lim_{k \rightarrow \infty} \gamma_k = \gamma_\rho(L(\omega, \cdot), \tilde{\theta})$ w.p. 1.*

Again, when viewed from the timescale of the recursions (19) and (20), one may consider ω_k and θ_k to be fixed as before. For the recursions (19) and (20), we have the following:

Lemma 4 *Assume $L(\omega_k, \cdot) \equiv L(\omega, \cdot)$, $\theta_k \equiv \theta$, $\forall k$. Then a.s.,*

$$(i) \lim_{k \rightarrow \infty} \xi_k^{(0)} = \xi_*^{(0)} = \frac{\mathbb{E}_{\tilde{\theta}} \left[\mathbf{g}_1(L(\omega, \mathbf{x}), \mathbf{x}, \gamma_\rho(L(\omega, \cdot), \tilde{\theta})) \right]}{\mathbb{E}_{\tilde{\theta}} \left[\mathbf{g}_0(L(\omega, \mathbf{x}), \gamma_\rho(L(\omega, \cdot), \tilde{\theta})) \right]}$$

$$(ii) \lim_{k \rightarrow \infty} \xi_k^{(1)} = \xi_*^{(1)} = \frac{\mathbb{E}_{\tilde{\theta}} \left[\mathbf{g}_2(L(\omega, \mathbf{x}), \mathbf{x}, \gamma_\rho(L(\omega, \cdot), \tilde{\theta}), \xi_*^{(0)}) \right]}{\mathbb{E}_{\tilde{\theta}} \left[\mathbf{g}_0(L(\omega, \mathbf{x}), \gamma_\rho(L(\omega, \cdot), \tilde{\theta})) \right]}$$

(iii) *If $\gamma_\rho(L(\omega, \cdot), \tilde{\theta}) > \gamma_\rho(L(\omega, \cdot), \tilde{\theta}_{old})$, then $T_k, k \geq 0$ in (22) satisfy $\lim_{k \rightarrow \infty} T_k = 1$ a.s.*

Notation: For the subsequence $\{k_{(n)}\}_{n > 0}$ of $\{k\}_{k \in \mathbb{Z}_+}$, we denote $k_{(n)}^- \triangleq k_{(n)} - 1$ for $n > 0$.

Along the subsequence $\{k_{(n)}\}_{n \geq 0}$ with $k_0 = 0$ the updating of θ_k can be expressed as follows:

$$\theta_{k_{(n+1)}} = \theta_{k_{(n)}} + \alpha_{k_{(n+1)}} \Delta \theta_{k_{(n+1)}}, \quad (26)$$

where $\Delta \theta_{k_{(n+1)}} = (\xi_{k_{(n+1)}^-}^{(0)}, \xi_{k_{(n+1)}^-}^{(1)})^\top - \theta_{k_{(n)}}$. We will prove now that the increment term $\Delta \theta_{k_{(n+1)}}$ in equation (26) is indeed an estimate of $\nabla_{\vartheta(\theta)} \Psi(\theta, \omega^*)|_{\theta = \theta_{k_{(n)}}$, where

$$\Psi(\theta, \omega) = \log \mathbb{E}_\theta [\mathbf{g}_0(L(\omega, \mathbf{x}), \gamma_\rho(L(\omega, \cdot), \theta))] \quad (27)$$

with $\theta = (\mu, \Sigma)^\top$ and $\vartheta(\theta) = (\Sigma^{-1} \mu, -\frac{1}{2} \Sigma^{-1})^\top$. Before proving this, we state a key lemma about the gradient of Ψ .

Lemma 5 *For the given function $L(\omega, \cdot) \in \mathbb{R}$, $\theta = (\mu, \Sigma)^\top$ and $\vartheta(\theta) = (\vartheta_1, \vartheta_2)^\top = (\Sigma^{-1} \mu, -\frac{1}{2} \Sigma^{-1})^\top$, we have*

$$\nabla_{\vartheta_1} \Psi(\theta, \omega) = \frac{\mathbb{E}_\theta [\mathbf{g}_1(L(\omega, \mathbf{x}), \mathbf{x}, \gamma_\rho(L(\omega, \cdot), \theta))] }{\mathbb{E}_\theta [\mathbf{g}_0(L(\omega, \mathbf{x}), \gamma_\rho(L(\omega, \cdot), \theta))] } - \mu.$$

$$\nabla_{\vartheta_2} \Psi(\theta, \omega) = \frac{\mathbb{E}_\theta [\mathbf{g}_2(L(\omega, \mathbf{x}), \mathbf{x}, \gamma_\rho(L(\omega, \cdot), \theta), \mu)] }{\mathbb{E}_\theta [\mathbf{g}_0(L(\omega, \mathbf{x}), \gamma_\rho(L(\omega, \cdot), \theta))] } - \Sigma.$$

We now state our main theorem. The following theorem guarantees the convergence of the model sequence $\{\theta_k\}_{k \in \mathbb{Z}_+}$ generated by Algorithm 2 and it further provides a characterization of its limit points. It also shows that by imposing additional structural restrictions on the objective function \mathcal{H} , the convergence of the algorithm to the degenerate distribution concentrated on the global maximum x^* is ensured.

Theorem 6 *Let $\varphi(x) = \exp(rx)$, $r \in \mathbb{R}$. Assume that the objective function \mathcal{H} satisfies the following two conditions: (i) $\nabla^2 \mathcal{H}$ exists and (ii) $\frac{\partial^2 \mathcal{H}}{\partial x_i \partial x_j}$ is continuous for $1 \leq \forall i, \forall j \leq m$. Let the learning rates α_k and β_k , $k \in \mathbb{Z}_+$ satisfy (25). Let $\{\theta_k = (\mu_k, \Sigma_k)^\top\}_{k \in \mathbb{Z}_+}$ be the sequence generated by Algorithm 2 and assume $\theta_k \in \text{interior}(\Theta)$, $\forall k \in \mathbb{Z}_+$. Also, let (A1) and (A2) hold. Then*

$$\lim_{k \rightarrow \infty} \theta_k = \lim_{k \rightarrow \infty} (\mu_k, \Sigma_k)^\top = (x^*, 0_{m \times m})^\top \text{ w.p. 1,}$$

where x^* is defined in (1).

Proof: Rewriting the equation (23) along the subsequence $\{k_{(n)}\}_{n \in \mathbb{Z}_+}$, we have for $n \in \mathbb{Z}_+$,

$$\theta_{k_{(n+1)}} = \theta_{k_{(n)}} + \alpha_{k_{(n+1)}} \left((\xi_{k_{(n+1)}^-}^{(0)}, \xi_{k_{(n+1)}^-}^{(1)})^\top - \theta_{k_{(n)}} \right). \quad (28)$$

Also $\sup_n \|\theta_{k_{(n)}}\| < \infty$ a.s. Rearranging the equation (28) we get, for $n \in \mathbb{Z}_+$,

$$\theta_{k_{(n+1)}} = \theta_{k_{(n)}} + \alpha_{k_{(n+1)}} \left(\mathbb{E} \left[\nabla_{\vartheta(\theta)} \Psi(\theta_{k_{(n)}}, \omega^*) \middle| \theta_{k_{(n)}} \right] + o(1) \right), \quad (29)$$

where the $o(1)$ term corresponds to errors in the estimation of $\xi_k^{(0)}$ and $\xi_k^{(1)}$ that decays to zero a.s. from Lemma 4.

Now consider the gradient flow ODE

$$\frac{d\theta(t)}{dt} = \nabla_{\vartheta(\theta)} \Psi(\theta(t), \omega^*), \quad t \in \mathbb{R}_+, \quad (30)$$

where ω^* is defined in Lemma 2.

By appealing to Theorem 2, Chapter 2 of [2], the asymptotic equivalence between the equations (29) and (30) can be easily established. Therefore, the recursion (23) reduces to a stochastic gradient ascent procedure which optimizes the objective function $\Psi(\theta, \omega^*)$. Hence the limiting behaviour of the model sequence $\{\theta_k\}_{k \in \mathbb{Z}_+}$ can be obtained by analysing the same of the above ODE. The fixed points of the ODE (30) can be obtained by equating $\nabla \Psi$ to 0.

Equating $\nabla_{\vartheta_1} \Psi(\theta, \omega^*)$ to $0_{m \times 1}$, we get,

$$\mu = \frac{\mathbb{E}_\theta [\mathbf{g}_1(\mathcal{H}(\mathbf{x}), \mathbf{x}, \gamma_\rho(L(\omega^*, \cdot), \theta))] }{\mathbb{E}_\theta [\mathbf{g}_0(\mathcal{H}(\mathbf{x}), \gamma_\rho(L(\omega^*, \cdot), \theta))]}. \quad (31)$$

Equating $\nabla_{\vartheta_2} \Psi(\theta, \omega^*)$ to $\mathbb{O} (= 0_{m \times m})$, we get,

$$\frac{\mathbb{E}_\theta [\mathbf{g}_2(\mathcal{H}(\mathbf{x}), \mathbf{x}, \gamma_\rho(L(\omega^*, \cdot), \theta), \mu)] }{\mathbb{E}_\theta [\mathbf{g}_0(\mathcal{H}(\mathbf{x}), \gamma_\rho(L(\omega^*, \cdot), \theta))] } - \Sigma = \mathbb{O}. \quad (32)$$

To further simplify the notation, let $\gamma_\rho^*(\theta) \triangleq \gamma_\rho(L(\omega^*, \cdot), \theta)$. Also, for brevity let $S(\theta) \triangleq \mathbb{E}_\theta [\mathbf{g}_0(\mathcal{H}(\mathbf{x}), \gamma_\rho^*(\theta))]$ and $\hat{\mathbf{g}}_0(\mathbf{x}, \theta) \triangleq \mathbf{g}_0(\mathcal{H}(\mathbf{x}), \gamma_\rho^*(\theta))$. Substituting the expression for μ from (31) in (32) and after further simplification we get,

$$(1/S(\theta)) \mathbb{E}_\theta [\hat{\mathbf{g}}_0(\mathbf{x}, \theta) \mathbf{x} \mathbf{x}^\top] - \mu \mu^\top - \Sigma = \mathbb{O}.$$

Since $\Sigma = \mathbb{E}_\theta [\mathbf{x} \mathbf{x}^\top] - \mu \mu^\top$, the above equation implies

$$\begin{aligned} (1/S(\theta)) \mathbb{E}_\theta [\hat{\mathbf{g}}_0(\mathbf{x}, \theta) \mathbf{x} \mathbf{x}^\top] - \mathbb{E}_\theta [\mathbf{x} \mathbf{x}^\top] &= \mathbb{O} \\ \implies_1 (1/S(\theta)) \mathbb{E}_\theta [(\hat{\mathbf{g}}_0(\mathbf{x}, \theta) - S(\theta)) \mathbf{x} \mathbf{x}^\top] &= \mathbb{O} \\ \implies_2 \Sigma \Sigma \mathbb{E}_\theta [\nabla_{\mathbf{x}}^2 \mathbf{g}_0(\mathbf{x}, \theta)] &= \mathbb{O} \end{aligned}$$

$$\implies_3 \quad \Sigma^2 \mathbb{E}_\theta [\varphi(\mathcal{H}(\mathbf{x}))G(\mathbf{x})\chi(\mathcal{H}(\mathbf{x}), \gamma_\rho^*(\theta))] = \mathbb{O}, \quad (33)$$

where $G(x) \triangleq r^2 \nabla \mathcal{H}(x) \nabla \mathcal{H}(x)^\top + r \nabla^2 \mathcal{H}(x)$. Note that \implies_2 follows from ‘‘integration by parts’’ rule for multivariate Gaussian and \implies_3 follows from the assumption $\varphi(x) = \exp(rx)$. Note that for each $x \in \mathcal{X}$, $G(x)$ is a $m \times m$ square matrix. Since $(\nabla_i \mathcal{H})^2 \geq 0$, we can find an $r \in \mathbb{R}$ and $1 \leq i \leq m$ s.t. $G_{ii}(x) > 0, \forall x \in \mathcal{X}$. This further implies that $\mathbb{E}_\theta [\varphi(\mathcal{H}(\mathbf{x}))G(\mathbf{x})\chi(\mathcal{H}(\mathbf{x}), \gamma_\rho^*(\theta))] \neq \mathbb{O}, \forall \theta \in \Theta$. Hence, from (33) we get $\Sigma = \mathbb{O}$. This proves that for any $x \in \mathbb{R}^m$, the degenerate distribution concentrated on x given by $\theta_x = (x, 0_{m \times m})^\top$ is an equilibrium point of the ODE (30). Also note that the ODE (30) is asymptotically stable at all local maxima of $\Psi(\cdot, \omega^*)$. The existence of the Lyapunov function $V_x : U_x \rightarrow \mathbb{R}_+$ on the open neighbourhood U_x of θ_x , defined as $V_x(\theta) \triangleq \Psi(\theta_x, \omega^*) - \Psi(\theta, \omega^*)$ is enough to prove the local asymptotic stability.

To prove that the limit is indeed θ^* , the degenerate distribution concentrated at x^* , we use *proof by contradiction* technique. So assume to the contrary, i.e., $\theta_k \rightarrow \hat{\theta} = (\hat{x}, \mathbb{O})^\top$, where $\hat{x} \neq x^*$. Note that $\mathbb{E}_\theta [\hat{\mathbf{g}}_1(\mathbf{x}, \theta)], \mathbb{E}_{\theta_0} [\hat{\mathbf{g}}_1(\mathbf{x}, \theta)], \mathbb{E}_\theta [\hat{\mathbf{g}}_0(\mathbf{x}, \theta)]$ and $\mathbb{E}_{\theta_0} [\hat{\mathbf{g}}_0(\mathbf{x}, \theta)]$ are all continuous on θ . This implies that we can find scalars $\epsilon_2 > 0, \delta_2 > 0$ and $k \in \mathbb{Z}_+$ s.t.

$$\begin{aligned} \|\theta_k - \hat{\theta}\|_\infty &< \delta_2, \\ \|\mathbb{E}_{\hat{\theta}} [\hat{\mathbf{g}}_1(\mathbf{x}, \hat{\theta})] - \mathbb{E}_{\theta_k} [\hat{\mathbf{g}}_1(\mathbf{x}, \theta_k)]\|_\infty &< \epsilon_2, \\ \|\mathbb{E}_{\theta_0} [\hat{\mathbf{g}}_1(\mathbf{x}, \hat{\theta})] - \mathbb{E}_{\theta_0} [\hat{\mathbf{g}}_1(\mathbf{x}, \theta_k)]\|_\infty &< \epsilon_2, \quad (34) \\ \|\mathbb{E}_{\hat{\theta}} [\hat{\mathbf{g}}_0(\mathbf{x}, \hat{\theta})] - \mathbb{E}_{\theta_k} [\hat{\mathbf{g}}_0(\mathbf{x}, \theta_k)]\|_\infty &< \epsilon_2, \\ \|\mathbb{E}_{\theta_0} [\hat{\mathbf{g}}_0(\mathbf{x}, \hat{\theta})] - \mathbb{E}_{\theta_0} [\hat{\mathbf{g}}_0(\mathbf{x}, \theta_k)]\|_\infty &< \epsilon_2, \end{aligned}$$

where $\|\cdot\|_\infty$ is the sup norm, i.e., $\|x\|_\infty = \max_i |x_i|, x \in \mathbb{R}^m$. Now consider

$$\nabla_{\theta_1} \Psi(\theta, \omega^*)|_{\theta=\hat{\theta}_k} = (1/S(\theta)) \mathbb{E}_\theta [\hat{\mathbf{g}}_1(\mathbf{x}, \theta)] - \mu|_{\theta=\hat{\theta}_k} \quad (35)$$

We denote by $e = (1, \dots, 1)^\top \in \mathbb{R}^m$. Applying sup norm on either side of (35) and using the inequalities in (34) we get,

$$\begin{aligned} \left\| \nabla_{\theta_1} \Psi(\theta, \omega^*)|_{\theta=\hat{\theta}_k} \right\|_\infty &\geq \\ \left\| \frac{(1 - \lambda_k) \mathbb{E}_{\hat{\theta}} [\hat{\mathbf{g}}_1(\mathbf{x}, \hat{\theta})] + \lambda_k \mathbb{E}_{\theta_0} [\hat{\mathbf{g}}_1(\mathbf{x}, \hat{\theta})] - \epsilon_2 e}{(1 - \lambda_k) \mathbb{E}_{\hat{\theta}} [\hat{\mathbf{g}}_0(\mathbf{x}, \hat{\theta})] + \lambda_k \mathbb{E}_{\theta_0} [\hat{\mathbf{g}}_0(\mathbf{x}, \hat{\theta})] + \epsilon_2} - \hat{x} - \delta_2 e \right\|_\infty & \\ \geq \left\| \frac{(1 - \lambda_k) \hat{x} \varphi(\mathcal{H}(\hat{x})) + \lambda_k \mathbb{E}_{\theta_0} [\mathbf{g}_1(\mathcal{H}(\mathbf{x}), \mathbf{x}, \mathcal{H}(\hat{x}))] - \epsilon_2 e}{(1 - \lambda_k) \varphi(\mathcal{H}(x^*)) + \lambda_k \mathbb{E}_{\theta_0} [\mathbf{g}_0(\mathcal{H}(\mathbf{x}), \mathcal{H}(\hat{x}))] + \epsilon_2} - \hat{x} - \delta_2 e \right\|_\infty & \\ \geq \left\| \frac{(K_1(\hat{x}, \epsilon_2) - 1) \hat{x} + K_2(\hat{x}, \epsilon_2) \frac{\mathbb{E}_{\theta_0} [\mathbf{g}_1(\mathcal{H}(\mathbf{x}), \mathbf{x}, \gamma_\rho^*(\hat{\theta}))]}{\mathbb{E}_{\theta_0} [\mathbf{g}_0(\mathcal{H}(\mathbf{x}), \gamma_\rho^*(\hat{\theta}))]} - (\epsilon_2 + \delta_2) e}{K_3} \right\|_\infty &> K_3 > 0, \end{aligned}$$

where $K_2(\cdot, \cdot) > 0$ and $0 < K_1(\cdot, \cdot) < 1$ with $K_1(x_1, x_2) \rightarrow 1$ as $x_1 \rightarrow x^*$ and $x_2 \rightarrow 0$. This is a contradiction since $\Psi(\theta, \omega^*)$ is continuously differentiable (easily verifiable). ■

4 Experimental Results

For empirical evaluation, we use two settings: (1) Global optimization benchmarks and (2) Nonlinear function approximation setting in reinforcement learning. In each setting, the results shown are averages over 10 independent sample trajectories obtained with the same initial distribution θ_0 .

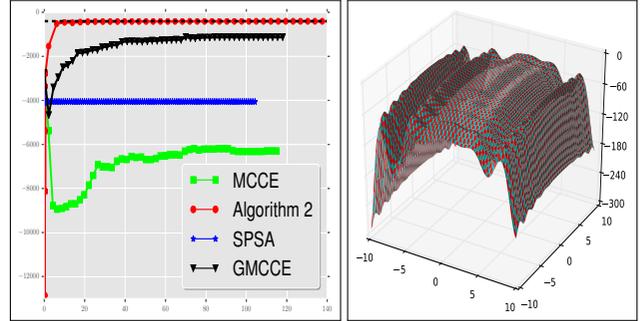
Experiment 1: Global Optimization Benchmarks [12, 5]

We consider here 4 global optimization benchmark functions from [12, 5]. We use their noisy versions to evaluate our algorithm. The function $\varphi(\cdot)$ is chosen as $\varphi(x) = \exp(rx)$, where $r \in \mathbb{R}_+$. We compare the performance of our algorithm with the original Monte-Carlo CE (MCCE), simultaneous perturbation stochastic approximation (SPSA) and the gradient based Monte-Carlo CE (GMCCE) [10] which is a modified version of MCCE. We consider here the noise injected version of SPSA, which is shown to have global optimization properties [18].

(1) Levy function [$m = 50$][Continuous, Differentiable]

$$\begin{aligned} \mathcal{H}_1(x) &= 0.1 * G_1(x) \|\mathbb{E}[Y]\|_2 - \mathbb{E}[Y]^\top \mathbb{E}[Z], \text{ where} \\ G_1(x) &= -1 - \sin^2(\pi y_1) - (y_m - 1)^2 (1 + \sin^2(2\pi y_m)) - \\ &\sum_{i=1}^{m-1} [(y_i - 1)^2 (1 + 10 \sin^2(\pi y_i + 1))] \text{ and } y_i = 1 + \frac{x_i - 1}{4}. \end{aligned}$$

Here, $Y, Z \in \mathbb{R}^{50}$ with $Y \sim \mathcal{N}([2.0, 2.0, \dots, 2.0]^\top, 40 * I_{50 \times 50})$ and $Z \sim \mathcal{N}([4.0, 4.0, \dots, 4.0]^\top, 40 * I_{50 \times 50})$. The function has global maximum at the point $x_i = 1, 1 \leq \forall i \leq m$ with value $\mathcal{H}_1^* = 401.4142135$.



(a) The trajectory of $\mathcal{H}_1(\mu_k)$. The dotted horizontal line is \mathcal{H}_1^* . x -axis is the time in secs relative to the start of the algorithm. y -axis is the function value.

Figure 2: Levy Function

(2) Qing function [$m = 20$][Continuous, Differentiable, Separable, Scalable, Multimodal]

$$\begin{aligned} \mathcal{H}_2(x) &= 10^{-3} * G_2(x) \|\mathbb{E}[Y]\|_2 - \mathbb{E}[Y]^\top \mathbb{E}[Z], \\ \text{where } G_2(x) &= - \sum_{i=1}^m (x_i^2 - i)^2. \end{aligned}$$

Here, $Y, Z \in \mathbb{R}^{20}$ with $Y \sim \mathcal{N}([2.0, 2.0, \dots, 2.0]^\top, 40 * I_{20 \times 20})$ and $Z \sim \mathcal{N}([4.0, 4.0, \dots, 4.0]^\top, 40 * I_{20 \times 20})$.

The function has global maximum at the point $x_i = \sqrt{i}, \forall i, 1 \leq i \leq m$ with value $\mathcal{H}_2^* = -160.0$.