

ECAI 2016

Frontiers in Artificial Intelligence and Applications

The book series Frontiers in Artificial Intelligence and Applications (FAIA) covers all aspects of theoretical and applied Artificial Intelligence research in the form of monographs, doctoral dissertations, textbooks, handbooks and proceedings volumes.

The FAIA series contains several sub-series, including ‘Information Modelling and Knowledge Bases’ and ‘Knowledge-Based Intelligent Engineering Systems’. It also includes the biennial European Conference on Artificial Intelligence (ECAI) proceedings volumes, and other EurAI (European Association for Artificial Intelligence, formerly ECCAI) sponsored publications. An editorial panel of internationally well-known scholars is appointed to provide a high quality selection.

Series Editors:

J. Breuker, N. Guarino, J.N. Kok, J. Liu, R. López de Mántaras,
R. Mizoguchi, M. Musen, S.K. Pal and N. Zhong

Volume 285

Recently published in this series

- Vol. 284. D. Pearce and H.S. Pinto (Eds.), STAIRS 2016 – Proceedings of the Eighth European Starting AI Researcher Symposium
- Vol. 283. R. Ferrario and W. Kuhn (Eds.), Formal Ontology in Information Systems – Proceedings of the 9th International Conference (FOIS 2016)
- Vol. 282. J. Mizera-Pietraszko, Y.-L. Chung and P. Pichappan (Eds.), Advances in Digital Technologies – Proceedings of the 7th International Conference on Applications of Digital Information and Web Technologies 2016
- Vol. 281. G. Chen, F. Liu and M. Shojafar (Eds.), Fuzzy System and Data Mining – Proceedings of FSDM 2015
- Vol. 280. T. Welzer, H. Jaakkola, B. Thalheim, Y. Kiyoki and N. Yoshida (Eds.), Information Modelling and Knowledge Bases XXVII
- Vol. 279. A. Rotolo (Ed.), Legal Knowledge and Information Systems – JURIX 2015: The Twenty-Eighth Annual Conference
- Vol. 278. S. Nowaczyk (Ed.), Thirteenth Scandinavian Conference on Artificial Intelligence – SCAI 2015
- Vol. 277. E. Armengol, D. Boixader and F. Grimaldo (Eds.), Artificial Intelligence Research and Development, Proceedings of the 18th International Conference of the Catalan Association for Artificial Intelligence
- Vol. 276. H. Fujita and S.-F. Su (Eds.), New Trends on System Sciences and Engineering – Proceedings of ICSSE 2015
- Vol. 275. J. Mizera-Pietraszko and S. Fong (Eds.), Advances in Digital Technologies – Proceedings of the 6th International Conference on Applications of Digital Information and Web Technologies 2015
- Vol. 274. W.C.-C. Chu, H.-C. Chao and S.J.-H. Yang (Eds.), Intelligent Systems and Applications – Proceedings of the International Computer Symposium (ICS) held at Taichung, Taiwan, December 12–14, 2014
- Vol. 273. J. Seibt, R. Hakli and M. Nørskov (Eds.), Sociable Robots and the Future of Social Relations – Proceedings of Robo-Philosophy 2014
- Vol. 272. B. Thalheim, H. Jaakkola, Y. Kiyoki and N. Yoshida (Eds.), Information Modelling and Knowledge Bases XXVI
- Vol. 271. R. Hoekstra (Ed.), Legal Knowledge and Information Systems – JURIX 2014: The Twenty-Seventh Annual Conference
- Vol. 270. H.-M. Haav, A. Kalja and T. Robal (Eds.), Databases and Information Systems VIII – Selected Papers from the Eleventh International Baltic Conference, DB&IS 2014
- Vol. 269. L. Museros, O. Pujol and N. Agell (Eds.), Artificial Intelligence Research and Development – Recent Advances and Applications
- Vol. 268. A. Utkā, G. Grigonytė, J. Kapočiūtė-Dzikienė and J. Vaičėnonienė (Eds.), Human Language Technologies – The Baltic Perspective: Proceedings of the Sixth International Conference Baltic HLT 2014
- Vol. 267. P. Garbacz and O. Kutz (Eds.), Formal Ontology in Information Systems – Proceedings of the Eighth International Conference (FOIS 2014)
- Vol. 266. S. Parsons, N. Oren, C. Reed and F. Cerutti (Eds.), Computational Models of Argument – Proceedings of COMMA 2014
- Vol. 265. H. Fujita, A. Selamat and H. Haron (Eds.), New Trends in Software Methodologies, Tools and Techniques – Proceedings of the Thirteenth SoMeT_14
- Vol. 264. U. Endriss and J. Leite (Eds.), STAIRS 2014 – Proceedings of the 7th European Starting AI Researcher Symposium
- Vol. 263. T. Schaub, G. Friedrich and B. O’Sullivan (Eds.), ECAI 2014 – 21st European Conference on Artificial Intelligence
- Vol. 262. R. Neves-Silva, G.A. Tshirintzis, V. Uskov, R.J. Howlett and L.C. Jain (Eds.), Smart Digital Futures 2014
- Vol. 261. G. Phillips-Wren, S. Carlsson, A. Respicio and P. Brézillon (Eds.), DSS 2.0 – Supporting Decision Making with New Technologies
- Vol. 260. T. Tokuda, Y. Kiyoki, H. Jaakkola and N. Yoshida (Eds.), Information Modelling and Knowledge Bases XXV
- Vol. 259. K.D. Ashley (Ed.), Legal Knowledge and Information Systems – JURIX 2013: The Twenty-Sixth Annual Conference
- Vol. 258. K. Gerdes, E. Hajičová and L. Wanner (Eds.), Computational Dependency Theory
- Vol. 257. M. Jaeger, T.D. Nielsen and P. Viappiani (Eds.), Twelfth Scandinavian Conference on Artificial Intelligence – SCAI 2013

ISSN 0922-6389 (print)
ISSN 1879-8314 (online)

ECAI 2016

22nd European Conference on Artificial Intelligence
29 August–2 September 2016, The Hague, The Netherlands

Including
Prestigious Applications of Artificial Intelligence (PAIS 2016)

Proceedings

Edited by

Gal A. Kaminka

Bar Ilan University, Israel

Maria Fox

King's College London, United Kingdom

Paolo Bouquet

University of Trento & OKKAM srl, Italy

Eyke Hüllermeier

Paderborn University, Germany

Virginia Dignum

Delft University of Technology, The Netherlands

Frank Dignum

Utrecht University, The Netherlands

and

Frank van Harmelen

Vrije Universiteit Amsterdam, The Netherlands

Organized by the European Association for Artificial Intelligence (EurAI)
and the Benelux Association for Artificial Intelligence (BNVKI).

Part 1/Part 2

IOS
Press

Amsterdam • Berlin • Washington, DC

© 2016 The Authors and IOS Press.

This book is published online with Open Access and distributed under the terms of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).

ISBN 978-1-61499-671-2 (print)

ISBN 978-1-61499-672-9 (online)

Library of Congress Control Number: 2016949341

Publisher

IOS Press BV

Nieuwe Hemweg 6B

1013 BG Amsterdam

Netherlands

fax: +31 20 687 0019

e-mail: order@iospress.nl

Distributor in the USA and Canada

IOS Press, Inc.

4502 Rachael Manor Drive

Fairfax, VA 22032

USA

fax: +1 703 323 3668

e-mail: iosbooks@iospress.com

LEGAL NOTICE

The publisher is not responsible for the use which might be made of the following information.

PRINTED IN THE NETHERLANDS

Introduction

This volume contains the proceedings of the Twenty-second European Conference on Artificial Intelligence (ECAI 2016), held from August 29th to September 2nd, 2016, in The Hague, The Netherlands. Since 1974, the biennial European Conference on Artificial Intelligence, organized by the European Association for Artificial Intelligence (EurAI, formerly named ECCAI), has been the premier venue for presenting AI research in Europe. ECAI is the place for researchers and practitioners of Artificial Intelligence (AI) to gather and to discuss the latest trends and challenges in all subfields of AI as well as to demonstrate innovative applications and uses of advanced AI technology.

ECAI 2016 was co-located with Collective Intentionality X, the interdisciplinary conference on collective intentionality, and the IEEE Symposium on Ethics of Autonomous Systems (SEAS Europe). As in the past, ECAI 2016 incorporates the Conference on Prestigious Applications of Intelligent Systems (PAIS 2016) and the Starting AI Researcher Symposium (STAIRS). The papers from PAIS are included in this volume, while the papers from STAIRS are published in a separate volume. ECAI 2016 also featured a special topic on Artificial Intelligence for Human Values, with a dedicated track and a public event in the Peace Palace in The Hague.

The program of ECAI 2016 included five invited plenary talks, including two, by Michael Bratman and Johanna Seibt, shared with Collective Intentionality X, and an extensive workshop and tutorial program.

In total, 656 papers were submitted to ECAI 2016. Of these, 177 (27%) were accepted as long papers and 123 (19%) were accepted as short papers, of which 108 were presented at the conference. This makes ECAI 2016 the largest edition in the 40-year history of ECAI conferences, reflecting the growth and vitality of AI as a research field. This year we pioneered the idea of supporting continuity of the peer reviewing process, by allowing authors to submit resubmissions of papers rejected from IJCAI 2016, alongside their reviews from that conference. We also introduced a Summary Reject process, enabling Senior Program Committee (SPC) members to use their knowledge and experience to help to reduce the load on reviewers. We also tried a new process of allocating papers to Program Committee (PC) members, in which SPCs allocated papers to a team of PC members that they had recruited themselves, in order to encourage team-working. Thanks to the dedicated support of SPC and PC members, these innovations worked well. Out of the short paper acceptances, 12 (9.8%) and out of the accepted long papers, 31 (17.4%) were former IJCAI submissions. Clearly people took advantage of the opportunity, and a large proportion of their revised submissions were rewarded by success. During the paper discussion period, four papers were given the option of transferring to PAIS. The PAIS programme consisted of 10 papers presenting substantial applications of AI research.

We were lucky to be part of a dedicated team. We would like to thank Meir Kalech for putting in place an extensive workshop program of 18 workshops; Sophia Ananiadou and Leon van der Torre for attracting 13 exciting tutorials, including 5 “Spotlight” tutorials; Helena Sofia Pinto and David Pearce for developing an exciting STAIRS 2016 program; Jeroen van den Hoven and Henry Prakken for managing the AI and Human Values Track; Robert-Jan Sips for organising the AIckathon, and last but not least Eyke Hullermeijer and Paolo Bouquet for chairing PAIS 2016.

We would also like to thank the local organizers, Frank Dignum and Virginia Dignum, and the General Chair, Frank van Harmelen. We are indebted to our predecessor, Torsten Schaub, whose materials we found helpful, and Thomas Preuss for help and advice in using the Confmaster system. Finally, heartfelt thanks to all PC and SPC members, reviewers, sponsors, and all authors who submitted to ECAI 2016.

Maria Fox and Gal A. Kaminka
Program Chairs
ECAI 2016

This page intentionally left blank

Conference Organization

General chair: Frank van Harmelen, The Netherlands

ECAI program chairs: Gal A. Kaminka, Israel and Maria Fox, UK

Local organization chairs: Frank Dignum and Virginia Dignum, The Netherlands

Workshop chair: Meir Kalech, Israel

Tutorials chairs: Sophia Ananiadou, UK and Leon van der Torre, Luxembourg

PAIS program chairs: Eyke Hüllermeier, Germany, and Paolo Bouquet, Italy

STAIRS program chairs: H. Sofia Pinto, Portugal, and David Pearce, Spain

AI and Human Values Track chairs: Jeroen van den Hoven and Henry Prakken

Local Organization Committee

Sponsorship: Martijn Warnier, The Netherlands

Publicity: Suzan Verberne

Website: Huib Aldewereld

Logistics: Amineh Ghorbani and Helle Hvid Hansen, The Netherlands

Student Office: Tibor Bosse, The Netherlands

AIckathon: Robert-Jan Sips, The Netherlands

Finance and Administration: MCI Netherlands

Sponsors

PricewaterhouseCoopers

City of The Hague

Taylor & Francis Group

Essence ITN Network

Vrije Universiteit Amsterdam

Supporting Organisations

Artificial Intelligence Journal

SIKS Graduate School

Utrecht University

Delft University of Technology

Organising Body

EurAI – European Association for Artificial Intelligence

BNVKI – Benelux Association for Artificial Intelligence

Invited Plenary Speakers

Michael Bratman (USA)

Jeff Rosenschein (Israel)

Marie-Christine Rousset (France)

Johanna Seibt (Denmark)

Michael Zillich (Austria)

Senior Program Committee

Noa Agmon, Sven Behnke, Sara Bernardini, Hendrik Blockeel, Ronen Brafman, Frances Brazier, Pompeu Casanovas, Amedeo Cesta, Amanda Coles, Adnan Darwiche, Amal El Fallah Seghrouchni, Ulle Endriss, Esra Erdem, Alessandro Farinelli, Helene Fargier, Linda van der Gaag, Hector Geffner, Lluís Godo, Malte Helmert, Koen Hindriks, Anthony Hunter, Catholijn Jonker, Meir Kalech, Antonin (Tony) Kucera, Gerhard Lakemeyer, Jérôme Lang, Pedro Larrañaga, Alessio Lomuscio, Derek Long, Daniele Magazzeni, Pedro Meseguer, Michela Milano, Daniele Nardi, Ann Nowé, Barry O’Sullivan, Michal Pechoucek, Eric Postma, Luc De Raedt, Francesca Rossi, Marie-Christine Rousset, David Sarne, Torsten Schaub, Michele Sebag, Katia Sycara, Leon van der Torre, Harko Verhagen, Cees Witteveen, Pinar Yolum, Filip Zelezny

Program Committee

Rui Abreu, Carole Adam, Stéphane Airiau, Charilaos Akasiadis, Marco Alberti, Alex Albore, Huib Aldewereld, Alexandre Allauzen, Teresa Alsinet, Claudia d’Amato, Leila Amgoud, Francesco Amigoni, Francisco Andrade, Angelo Angelo Fanelli, Nino Antulov-Fantulin, Yashar Araghi, Alexander Artikis, Kevin Ashley, Manuel Atencia, Katie Atkinson, Reyhan Aydogan, Amos Azaria, Christer Bäckström, Antonio Bahamonde, Josef Bajada, Marcello Balduccini, Stefania Bandini, Jacopo Banfi, Ariel Bar, Elias Bareinboim, Pietro Baroni, Roman Bartak, Roberto Basili, Nicola Basilico, Christian Bauckhage, Kim Bauters, Francesco Belardinelli, Vaishak Belle, Nahla Ben Amor, Trevor Bench-Capon, Philippe Besnard, Floris Bex, Aurélie Beynier, Concha Bielza, Xavier Binefa, Gilles Bisson, Filippo Bistaffa, Stefano Bistarelli, Susanne Biundo, Elizabeth Black, Bernhard Bliem, Domenico Bloisi, Christian Blum, Fernando Bobillo, Blai Bonet, Elise Bonzon, Richard Booth, Rafael Bordini, Daniel Borrajo, Branislav Božanský, Sylvain Bouveret, Ronen Brafman, Tomas Brazdil, Robert Bredebeck, Davide Bresolin, Marco Bressan, Thomas Bridi, Joost Broekens, Jan Broersen, Ken Brown, Tim Brys, Joanna Bryson, Katarzyna Budzynska, Nils Bulling, Andreas Bunte, Andreas Bunte, Miroslav Bureš, Pedro Cabalar, Philippe Caillou, Olivier Cailloux, Michal Čáp, Clément Carbonnel, Michael Cashmore, Giovanni Casini, Claudette Cayrol, Tristan Cazenave, Jesus Cerquides, Federico Cerutti, Georgios Chalkiadakis, François Charpillet, Tristan Charrier, Jiehua Chen, Yann Chevaleyre, Sylvain Chevallier, Angelos Chlioutakis, Yoonsuck Choe, Arthur Choi, Amit Chopra, Lukas Chrapa, Davide Ciucci, Jens Claßen, Luca Console, Giorgio Corani, Cristina Cornelio, Juan David Correa Granada, Vincent Corruble, Gabriella Cortellessa, Fabrizio Costa, Stefania Costantini, Giuseppe Cota, Marcos Cramer, Natalia Criado, Danilo Croce, Matthew Crosby, Scott Cunningham, James Cussens, Andreas Darmann, Mehdi Dastani, Jérôme David, Bernard De Baets, Riccardo De Benedictis, Cassio de Campos, Martine De Cock, Giuseppe De Giacomo, Joachim De Greeff, Marina De Vos, Mathijs De Weerd, Aurelien Decelle, Juan Jose del Coz, Maria J. del Jesus, Dario Della Monica, Guissepe Della Penna, Pilar Dellunde, Louise Dennis, Sebastien Destercke, Sam Devlin, Virginia Dignum, Yannis Dimopoulos, Juergen Dix, Carmine Dodaro, Patrick Doherty, Carmel Domshlak, Klaus Dorer, Sylvie Doutre, Agostino Dovier, Jan Drchal, Nicolas Drougard, Didier Dubois, Stefano Duca, Barbara Dunin Keplicz, Soma Dutta, Saso Dzeroski, Stefan Edelkamp, Kyriakos Efthymiadis, Christian Eichhorn, Edith Elkind, Avshalom Elmalech, Zied Elouedi, Ozan Erdem, Salome Eriksson, Jerome Euzenat, Patricia Everaere, Piotr Faliszewski, Jorge Fandinno, Alexander Feldman, Stefano Ferilli, Alexander Ferrein, Marcelo Finger, Alberto Finzi, Ferdinando Fioretto, Pierre Flener, Federico Fogolari, Vojtech Forejt, Nicoletta Fornara, Alexander Förster, Guillem Frances, Enrico Francesconi, Simone Fratini, Alfredo Gabaldon, Christian Gagné, Lucie Galand, Joao Gama, Aldo Gangemi, Guglielmo Gemignani, Katie Genter, Alexander Gepperth, Malik Ghallab, Amineh Ghorbani, Massimiliano Giacomin, Maria Gini, Jesús Giráldez-Cru, Enrico Giunchiglia, François Goasdoué, Vibhav Gogate, Vicenç Gómez, Jorge González-Conejero, Laurent Gourvès, Guido Governatori, Umberto Grandi, Alban Grastien, Gregory Grefenstette, Andreas Griesmayer, Diarmuid Grimes, Dagmar Gromann, Stefano Gualandi, Javier Guerrero, Akin Gunay, Tias Guns, Thomas Guyet, Galit Haim, Barbara Hammer, Christian Hammerschmidt, Christopher Hampson, Blaise Hanczar, Marc Hanheide, Helle Hansen, Amelia Harrison, Anna Harutyunyan, Salima Hassas, Patrick Healy, Fredrik Heintz, Dirk Helbing, Ulrich Hillenbrand, Rinke Hoekstra, Jörg Hoffmann, Han Hoogeveen, Enda Howley, Joris Hulstijn, Giovambattista Ianni, Rasmus Ibsen-Jensen, Ayumi Igarashi, Felix Ingrand, Iñaki Inza, Luca Iocchi, Quiliano Isaac Moro, Rezarta Islamaj, Amin Jaber, François Jacquenet, Wojtek Jamroga, Baptiste Jeudy, Ke Jiang, Sergio Jiménez, Anders Jonsson, Fabrice Jouanot, Souhila Kaci, Magdalena Kacprzak, Serdar Kadioglu, Ozgur Kafali, Lars Karlsson, Erez Karpas, George Katsirelos, Hilal Kazan, Thomas Keller, Gabriele Kern-Isberner, Kristian Kersting, Piyush Khandelwal, Angelika Kim-

mig, Zeynep Kiziltan, Jiri Klema, Marius Kloft, Tomas Klos, Michal Knapik, Max Knobbout, Jens Kober, Antonín Komenda, Sebastien Konieczny, Pavel Kordik, Frederic Koriche, Lars Kotthoff, Panagiotis Kouvaros, Petr Kremen, Jan Kretinsky, Martin Kronegger, Daniel Kudenko, Ondrej Kuzelka, Bruno Lacerda, Arnaud Lallouet, Javier Larrosa, Aron Larsson, Paweł Laskoś-Grabowski, Jimmy Lee, Joao Leite, Julien Lesca, Yves Lesperance, Mario Leung, Omer Lev, Jordi Levy, Beishui Liao, Churn-Jung Liao, Martin Liebenberg, Carlos Linares López, Marius Lindauer, Tony Lindgren, Thomas Linsbichler, Nir Lipovetzky, Marco Lippi, Francesca Alessandra Lisi, Weiru Liu, Yongmei Liu, Brian Logan, Katrin Lohan, Ramon Lopez de Mantaras, Andrea Loreggia, Emiliano Lorini, Peter Lucas, Stephan Lukosch, Jianbing Ma, Yue Ma, Brian Mac Namee, Samhar Mahmoud, Alexander Maier, Jean-Guy Mailly, Julien Mairal, Kleanthis Malialis, Yuri Malitsky, Michailis Mamakos, Bernard Manderick, René Mandiau, Lawrence Mandow, Marco Manna, Filip Manyá, Marco Maratea, Gaetan Marceau-Caron, Fabio Marfia, Joao Marques-Silva, Pierre Marquis, Mercedes Martínez-gonzález, Joao Martins, Viviana Mascardi, Nicholas Mattei, Robert Mattmüller, Nicolas Maudet, Deepak Mehta, Reshef Meir, Antonella Meneghetti, Jerome Mengin, Jiří Menšík, Fabio Mercorio, Rijk Mercur, Tekin Mericli, John-Jules Meyer, Roberto Micalizio, Jakub Michalyszyn, Andrea Micheli, Steffen Michels, Pedro Miraldo, Reuth Mirsky, Sanjay Modgil, Karthika Mohan, Jérôme Monnot, Marco Montali, Angelo Montanari, Michael Morak, Serafin Moral, Paul Morris, Sergio Mover, Marie-Laure Mugnier, Enrique Munoz De Cote, Aniello Murano, Bernhard Nebel, Trung Thanh Nguyen, Alexandre Niveau, Pablo Noriega, Paulo Novais, Peter Novak, Petr Novotny, Philipp Obermeier, Svetlana Obrasztova, Angelo Oddi, Dimitri Ognibene, Andrei Olaru, Nir Oren, Emmanouil Orfanoudakis, Petter Orgen, Andrea Orlandini, Nardine Osman, Max Ostrowski, Arzucan Ozturk, Umut Oztok, Erhan Oztok, Ugo Pagallo, Maurice Pagnucco, Hector Palacios, Ravi Palla, Papapetrou Panagiotis, Pere Pardo, Xavier Parent, Fabio Patrizi, MariaTeresa Paziienza, Justin Pearson, Federico Pecora, Marieke Peeters, Bernhard Peischl, Jose Peña, Rafael Penalzoza Nyssen, Wojciech Penczek, Nathalie Pernelle, Patrice Perny, Laurent Perrussel, Ginevra Peruginelli, Dominik Peters, Ron Petrick, Tomas Pevny, Guillermo A. Pérez, Rens Philipsen, Chiara Piacentini, Gauthier Picard, Gabriella Pigozzi, Ingo Pill, Maria Silvia Pini, Matija Piskorec, Jeremy Pitt, Enric Plaza, Sylwia Polberg, Florian Pommerening, Matei Popovici, Daniele Porello, Bary Pradelski, Luca Pretto, Jörg Pührer, Luis Quesada, Rahim Rahmani, Franco Raimondi, David Rajaratnam, Subramanian Ramamoorthy, Sarvapali Ramchurn, Miquel Ramirez, Karinne Ramirez Amaro, Jan Ramon, Riccardo Rasconi, Irma Ravkic, Jesse Read, Shulamit Reches, Ioannis Refanidis, Vojtech Rehak, Alessandro Ricci, Francesco Riccio, Ariela Richadson, Bram Ridder, Fabrizio Riguzzi, Jussi Rintanen, Livio Robaldo, Odinaldo Rodrigues, Juan Rodriguez, Juan Rodriguez-Aguilar, Victor Rodríguez-Doncel, Emma Rollon, Nico Roos, Jörg Rothe, Ovi Rouly, Marco Roveri, Sasha Rubin, Sebastian Rudolph, Michael Ruhnke, Jordi Sabater-Mir, Regis Sabbadin, Orkunt Sabuncu, Alessandro Saetti, Brigitte Safar, Abdallah Saffidine, Alessandro Saffiotti, Fatiha Sais, Pietro Sala, Antonio Salmeron, Francesco Santini, Vitor Santos Costa, Ario Santoso, Sebastian Sardina, Giovanni Sartor, Enrico Scala, Enrico Scala, Burkhard Schafer, Leander Schietgat, Stefan Schiffer, Steven Schockaert, Marc Schoenauer, Christoph Schommer, Marco Schorlemmer, Claudia Schulz, Francois Schwarzentruher, Christoph Schwering, Guido Sciavicco, Luciano Serafini, Ivan Serina, Antonio Sgorbissa, Yang Yang Shi, Mohamed Siala, Carles Sierra, Gerardo Simari, Gilles Simonin, Piotr Skowron, Marija Slavkovik, Christine Solnon, Petr Somol, Tran Cao Son, Ulrich Sorger, Matthijs Spaan, Olivier Spanjard, Siddharth Srivastava, Michal Sroka, Michele Stawowy, Gerald Steinbauer, Roni Stern, Michiel Stock, Michal Stolba, Umberto Straccia, Alina Strachocka, Jörg Stückler, Jose Such, David Sundgren, Vojtech Svatek, Anton Talantsev, Matthew Taylor, Cem Tekin, Isabelle Tellier, Annette ten Teije, Michael Thielscher, Matthias Thimm, Michael Thomazo, Jin Tian, Paolo Turrini, Federico Ulliana, Marian van de Akker, Jan Van Haaren, Kristof Van Laerhoven, Pieter van Langen, Martijn van Otterlo, Joaquin Vanschoren, Tiago Veiga, Kristen Brent Venable, Celine Vens, Carmine Ventre, Sebastian Ventura, Bart Verheij, Sicco Verwer, Srdjan Vesic, Paolo Viappiani, Serena Villata, Arnoud Visser, Jonas Vlasselaer, Sven Wachsmuth, Martijn Warnier, Erwin Walraven, Philipp Wanko, Martin Wehrle, Paul Weng, Emil Weydert, Nic Wilson, Stefan Woltran, Guohui Xiao, Liping Xiong, Fangkai Yang, Roi Yehoshua, Reyyan Yeniterzi, William Yeoh, Neil Yorke-Smith, Changhe Yuan, Anna Zamansky, Tom Zamir, Benjamin Zarriess, John Zeleznikow, Riccardo Zese, Yingqian Zhang, Yair Zick, Danielle Ziebelin, Michael Zillich, Albrecht Zimmermann, Inon Zuckerman.

Prestigious Applications of Intelligent Systems

PAIS Program Committee

Bert Brede weg, Ken Brown, Simon Colton, Alfredo Cuzzocrea, Simon De Givry, Marina De Vos, Agostino Dovier, Martin Dow, Alexander Felfernig, Johannes Fürnkranz, Joao Gama, Matjaz Gams, Fuyuki Ishikawa, Michal Jakob, Dietmar Jannach, Mustafa Jarrar, Ernesto Jimenez-Ruiz, Nicola Leone, Ramon Lopez De Mantaras, Peter Lucas, Michael Madden, Wolfgang Mayer, Andrea Molinari, Barry Smyth, Daniel Sonntag, Erich Teppan, Marc Troemel, Giovanni Tummarello, Pascal Van Hentenryck, Franz Wotawa

Additional Reviewers

Stefano Bortoli, Stefano Burigat, Giovanni Grasso, Francesco Ricca

Starting Artificial Intelligence Research Symposium

STAIRS Program Committee

Harith Alani, Natasha Alechina, José Júlio Alferes, Ronen Brafman, Gerhard Brewka, Pedro Cabalar, Philipp Cimiano, Stefania Costantini, Ulle Endriss, Joerg Hoffmann, João Leite, Paolo Liberatore, Weiru Liu, Ramon López de Mántaras, Emiliano Lorini, Enrico Motta, Igor Mozetič, Pablo Noriega, Manuel Ojeda Aciego, Julian Padget, David Pearce, H. Sofia Pinto, Steven Schockaert, Levan Uridia, Agustin Valverde, Harko Verhagen, Stefan Woltran

Additional Reviewers

Pedro Barahona, Martin Kronegger, Evgeny Kuznetsov, Martin Lackner, Brian Logan, Jumana Nassour, Andreas Pfandler, Jasmina Smailović

Contents

Introduction	v
<i>Maria Fox and Gal A. Kaminka</i>	
Conference Organization	vii
Part 1	
ECAI Long Papers	
When Do Rule Changes Count-As Legal Rule Changes?	3
<i>Thomas C. King, Virginia Dignum and Catholijn M. Jonker</i>	
Constant Time EXPECTed Similarity Estimation for Large-Scale Anomaly Detection	12
<i>Markus Schneider, Wolfgang Ertel and Günther Palm</i>	
AUC Maximization in Bayesian Hierarchical Models	21
<i>Mehmet Gönen</i>	
Validating Cross-Perspective Topic Modeling for Extracting Political Parties' Positions from Parliamentary Proceedings	28
<i>Janneke M. van der Zwaan, Maarten Marx and Jaap Kamps</i>	
Finite Unary Relations and Qualitative Constraint Satisfaction	37
<i>Peter Jonsson</i>	
Dynamic Choice of State Abstraction in Q-Learning	46
<i>Marco Tamassia, Fabio Zambetta, William L. Raffe, Florian 'Floyd' Mueller and Xiaodong Li</i>	
A Dialectical Proof Theory for Universal Acceptance in Coherent Logic-Based Argumentation Frameworks	55
<i>Abdallah Arioua and Madalina Croitoru</i>	
Adaptive Binary Quantization for Fast Nearest Neighbor Search	64
<i>Zhujin Li, Xianglong Liu, Junjie Wu and Hao Su</i>	
Exploring Parallel Tractability of Ontology Materialization	73
<i>Zhangquan Zhou, Guilin Qi and Birte Glimm</i>	
Student- t Process Regression with Dependent Student- t Noise	82
<i>Qingtao Tang, Yisen Wang and Shu-Tao Xia</i>	
An Extension of the Owen-Value Interaction Index and Its Application to Inter-Links Prediction	90
<i>Piotr L. Szczepański, Tomasz P. Michalak, Talal Rahwan and Michael Wooldridge</i>	
Cluster-Driven Model for Improved Word and Text Embedding	99
<i>Zhe Zhao, Tao Liu, Bofang Li and Xiaoyong Du</i>	
Learning Temporal Context for Activity Recognition	107
<i>Claudio Coppola, Tomáš Krajník, Tom Duckett and Nicola Bellotto</i>	
Leader-Follower MDP Models with Factored State Space and Many Followers – Followers Abstraction, Structured Dynamics and State Aggregation	116
<i>Régis Sabbadin and Anne-France Viet</i>	

CUBE: A CUDA Approach for Bucket Elimination on GPUs <i>Filippo Bistaffa, Nicola Bombieri and Alessandro Farinelli</i>	125
Managing Energy Markets in Future Smart Grids Using Bilateral Contracts <i>Caillière Romain, Aknine Samir, Nongaillard Antoine and Sarvapal D. Ramchurn</i>	133
A Rational Account of Classical Logic Argumentation for Real-World Agents <i>M. D'Agostino and S. Modgil</i>	141
Two Dimensional Uncertainty in Persuadee Modelling in Argumentation <i>Anthony Hunter</i>	150
A Data Driven Similarity Measure and Example Mapping Function for General, Unlabelled Data Sets <i>Damien Lejeune and Kurt Driessens</i>	158
On Stochastic Primal-Dual Hybrid Gradient Approach for Compositely Regularized Minimization <i>Linbo Qiao, Tianyi Lin, Yu-Gang Jiang, Fan Yang, Wei Liu and Xicheng Lu</i>	167
Decentralized Large-Scale Electricity Consumption Shifting by Prosumer Cooperatives <i>Charilaos Akasiadis and Georgios Chalkiadakis</i>	175
Analysing Approximability and Heuristics in Planning Using the Exponential-Time Hypothesis <i>Meysam Aghighi, Christer Bäckström, Peter Jonsson and Simon Ståhlberg</i>	184
A Simple Account of Multi-Agent Epistemic Planning <i>Martin C. Cooper, Andreas Herzig, Faustine Maffre, Frédéric Maris and Pierre Régnier</i>	193
Lexicographic Refinements in Possibilistic Decision Trees <i>Nahla Ben Amor, Zeineb El Khalfi, Hélène Fargier and Régis Sabbadin</i>	202
Even Angels Need the Rules: AI, Roboethics, and the Law <i>Ugo Pagallo</i>	209
One-Class to Multi-Class Model Update Using the Class-Incremental Optimum-Path Forest Classifier <i>Mateus Riva, Moacir Ponti and Teofilo de Campos</i>	216
More than a Name? On Implications of Preconditions and Effects of Compound HTN Planning Tasks <i>Pascal Bercher, Daniel Höller, Gregor Behnke and Susanne Biundo</i>	225
A Probabilistic Logic Programming Approach to Automatic Video Montage <i>Bram Aerts, Toon Goedemé and Joost Vennekens</i>	234
Checking the Conformance of Requirements in Agent Designs Using ATL <i>Nitin Yadav and John Thangarajah</i>	243
A Uniform Account of Realizability in Abstract Argumentation <i>Thomas Linsbichler, Jörg Pührer and Hannes Strass</i>	252
A Scalable Clustering-Based Local Multi-Label Classification Method <i>Lu Sun, Mineichi Kudo and Keigo Kimura</i>	261
Multiscale Triangular Centroid Distance for Shape-Based Plant Leaf Recognition <i>Chengzhan Yang, Hui Wei and Qian Yu</i>	269
Complexity of Control by Partitioning Veto and Maximin Elections and of Control by Adding Candidates to Plurality Elections <i>Cynthia Maushagen and Jörg Rothe</i>	277
Agent-Based Refinement for Predicate Abstraction of Multi-Agent Systems <i>Francesco Belardinelli, Alessio Lomuscio and Jakub Michaliszyn</i>	286
Combining Deterministic and Nondeterministic Search for Optimal Journey Planning Under Uncertainty <i>Akihiro Kishimoto, Adi Botea and Elizabeth Daly</i>	295

Robust Real-Time Human Perception with Depth Camera <i>Guyue Zhang, Luchao Tian, Ye Liu, Jun Liu, Xiang An Liu, Yang Liu and Yan Qiu Chen</i>	304
A New Kernelized Associative Memory and Some of Its Applications <i>Matthew Saltz and Lluís A. Belanche</i>	311
Strategical Argumentative Agent for Human Persuasion <i>Ariel Rosenfeld and Sarit Kraus</i>	320
Formalizing Commitment-Based Deals in Boolean Games <i>Sofie De Clercq, Steven Schockaert, Ann Nowé and Martine De Cock</i>	329
A Joint Model for Sentiment-Aware Topic Detection on Social Media <i>Kang Xu, Guilin Qi, Junheng Huang and Tianxing Wu</i>	338
A Reinforcement Learning Framework for Trajectory Prediction Under Uncertainty and Budget Constraint <i>Truc Viet Le, Siyuan Liu and Hoong Chuin Lau</i>	347
Person Re-Identification via Multiple Coarse-to-Fine Deep Metrics <i>Mingfu Xiong, Jun Chen, Zheng Wang, Zhongyuan Wang, Ruimin Hu, Chao Liang and Daming Shi</i>	355
How Hard Is Bribery with Distance Restrictions? <i>Yongjie Yang, Yash Raj Shrestha and Jiong Guo</i>	363
Beyond IC Postulates: Classification Criteria for Merging Operators <i>Adrian Haret, Andreas Pfandler and Stefan Woltran</i>	372
Schema-Based Debugging of Federated Data Sources <i>Andreas Nolle, Christian Meilicke, Melisachew Wudage Chekol, German Nemirovski and Heiner Stuckenschmidt</i>	381
Belief Contraction Within Fragments of Propositional Logic <i>Nadia Creignou, Raïda Ktari and Odile Papini</i>	390
A Novel Cross-Modal Topic Correlation Model for Cross-Media Retrieval <i>Yong Cheng, Fei Huang, Cheng Jin, Yuejie Zhang and Tao Zhang</i>	399
Situation Calculus Game Structures and GDL <i>Giuseppe De Giacomo, Yves Lespérance and Adrian R. Pearce</i>	408
The Game of Reciprocation Habits <i>Gleb Polevoy, Mathijs de Weerd and Catholijn Jonker</i>	417
Randomized Distribution Feature for Image Classification <i>Hongming Shan and Junping Zhang</i>	426
ShapeLearner: Towards Shape-Based Visual Knowledge Harvesting <i>Huayong Xu, Yafang Wang, Kang Feng, Gerard de Melo, Wei Wu, Andrei Sharf and Baoquan Chen</i>	435
Observation-Based Multi-Agent Planning with Communication <i>Luca Gasparini, Timothy J. Norman and Martin J. Kollingbaum</i>	444
A Framework for Actionable Clustering Using Constraint Programming <i>Thi-Bich-Hanh Dao, Christel Vrain, Khanh-Chuong Duong and Ian Davidson</i>	453
Repetitive Branch-and-Bound Using Constraint Programming for Constrained Minimum Sum-of-Squares Clustering <i>Tias Guns, Thi-Bich-Hanh Dao, Christel Vrain and Khanh-Chuong Duong</i>	462
Is Spearman's Law of Diminishing Returns (SLODR) Meaningful for Artificial Agents? <i>José Hernández-Orallo</i>	471

Efficient Computation of Exact IRV Margins <i>Michelle Blom, Vanessa Teague, Peter J. Stuckey and Ron Tidhar</i>	480
On Labelling Statements in Multi-Labeling Argumentation <i>Pietro Baroni, Guido Governatori and Régis Riveret</i>	489
Annotate-Sample-Average (ASA): A New Distant Supervision Approach for Twitter Sentiment Analysis <i>Felipe Bravo-Marquez, Eibe Frank and Bernhard Pfahringer</i>	498
Semi-Supervised Group Sparse Representation: Model, Algorithm and Applications <i>Longwen Gao, Yeqing Li, Junzhou Huang and Shuigeng Zhou</i>	507
Modeling Bounded Rationality for Sponsored Search Auctions <i>Jiang Rong, Tao Qin, Bo An and Tie-Yan Liu</i>	515
An Improved State Filter Algorithm for SIR Epidemic Forecasting <i>Weipeng Huang and Gregory Provan</i>	524
Socially-Aware Multiagent Learning: Towards Socially Optimal Outcomes <i>Xiaohong Li, Chengwei Zhang, Jianye Hao, Karl Tuyls, Siqi Chen and Zhiyong Feng</i>	533
Factors of Collective Intelligence: How Smart Are Agent Collectives? <i>Nader Chmait, David L. Dowe, Yuan-Fang Li, David G. Green and Javier Insa-Cabrera</i>	542
Synthesizing Argumentation Frameworks from Examples <i>Andreas Niskanen, Johannes P. Wallner and Matti Järvisalo</i>	551
Budgeted Multi-Armed Bandit in Continuous Action Space <i>Francesco Trovò, Stefano Paladino, Marcello Restelli and Nicola Gatti</i>	560
A Framework for Automatic Debugging of Functional and Degradation Failures <i>Nuno Cardoso, Rui Abreu, Alexander Feldman and Johan de Kleer</i>	569
Online Adaptation of Deep Architectures with Reinforcement Learning <i>Thushan Ganegedara, Lionel Ott and Fabio Ramos</i>	577
Multi-Class Probabilistic Active Learning <i>Daniel Kottke, Georg Kreml, Dominik Lang, Johannes Teschner and Myra Spiliopoulou</i>	586
Online Prediction of Exponential Decay Time Series with Human-Agent Application <i>Ariel Rosenfeld, Joseph Keshet, Claudia V. Goldman and Sarit Kraus</i>	595
Set-Valued Conditioning in a Possibility Theory Setting <i>Salem Benferhat, Amélie Levray, Karim Tabia and Vladik Kreinovich</i>	604
An Improved CNF Encoding Scheme for Probabilistic Inference <i>Anicet Bart, Frédéric Koriche, Jean-Marie Lagniez and Pierre Marquis</i>	613
A Bayesian Approach to Norm Identification <i>Stephen Cranefield, Felipe Meneguzzi, Nir Oren and Bastin Tony Roy Savarimuthu</i>	622
Subsumed Label Elimination for Maximum Satisfiability <i>Jeremias Berg, Paul Saikko and Matti Järvisalo</i>	630
Vertical Optimization of Resource Dependent Flight Paths <i>Anders N. Knudsen, Marco Chiarandini and Kim S. Larsen</i>	639
Exploiting Bayesian Network Sensitivity Functions for Inference in Credal Networks <i>Janneke H. Bolt, Jasper De Bock and Silja Renooij</i>	646
Interval-Based Relaxation for General Numeric Planning <i>Enrico Scala, Patrik Haslum, Sylvie Thiebaux and Miquel Ramirez</i>	655

Randomized Canonical Correlation Discriminant Analysis for Face Recognition <i>Bo Ma, Hui He, Hongwei Hu and Meili Wei</i>	664
Reconsidering AGM-Style Belief Revision in the Context of Logic Programs <i>Zhiqiang Zhuang, James Delgrande, Abhaya Nayak and Abdul Sattar</i>	671
Value Based Reasoning and the Actions of Others <i>Katie Atkinson and Trevor Bench-Capon</i>	680
Analogical Classifiers: A Theoretical Perspective <i>Nicolas Hug, Henri Prade, Gilles Richard and Mathieu Serrurier</i>	689
Uncertainty-Sensitive Reasoning for Inferring sameAs Facts in Linked Data <i>Mustafa Al-Bakri, Manuel Atencia, Jérôme David, Steffen Lalande and Marie-Christine Rousset</i>	698
Can a Condorcet Rule Have a Low Coalitional Manipulability? <i>François Durand, Fabien Mathieu and Ludovic Noirie</i>	707
Upper and Lower Time and Space Bounds for Planning <i>Christer Bäckström and Peter Jonsson</i>	716
Abstraction-Based Verification of Infinite-State Reactive Modules <i>Francesco Belardinelli and Alessio Lomuscio</i>	725
Translation-Based Revision and Merging for Minimal Horn Reasoning <i>Gerhard Brewka, Jean-Guy Mailly and Stefan Woltran</i>	734
Parallel Filter-Based Feature Selection Based on Balanced Incomplete Block Designs <i>Antonio Salmerón, Anders L. Madsen, Frank Jensen, Helge Langseth, Thomas D. Nielsen, Dario Ramos-López, Ana M. Martínez and Andrés R. Masegosa</i>	743
Distributed Controllers for Norm Enforcement <i>Bas Testerink, Mehdi Dastani and Nils Bulling</i>	751
Automatic Verification of Golog Programs via Predicate Abstraction <i>Peiming Mo, Naiqi Li and Yongmei Liu</i>	760
An Assessment Study of Features and Meta-Level Features in Twitter Sentiment Analysis <i>Jonnathan Carvalho and Alexandre Plastino</i>	769
Crowdfunding Public Projects with Provision Point: A Prediction Market Approach <i>Praphul Chandra, Sujit Gujar and Y. Narahari</i>	778
Welfare of Sequential Allocation Mechanisms for Indivisible Goods <i>Haris Aziz, Thomas Kalinowski, Toby Walsh and Lirong Xia</i>	787
A Computational Approach to Consensus-Finding <i>Eric Grégoire and Jean-Marie Lagniez</i>	795
Complexity and Tractability Islands for Combinatorial Auctions on Discrete Intervals with Gaps <i>Janosch Döcker, Britta Dorn, Ulle Endriss and Dominikus Krüger</i>	802
Efficient SAT Approach to Multi-Agent Path Finding Under the Sum of Costs Objective <i>Pavel Surynek, Ariel Felner, Roni Stern and Eli Boyarski</i>	810
Fixed-Domain Reasoning for Description Logics <i>Sarah Gaggl, Sebastian Rudolph and Lukas Schweizer</i>	819
On Redundancy in Simple Temporal Networks <i>Jae Hee Lee, Sanjiang Li, Zhiguo Long and Michael Sioutis</i>	828
On Metric Temporal Description Logics <i>Victor Gutiérrez-Basulto, Jean Christoph Jung and Ana Ozaki</i>	837

Plan-Based Narrative Generation with Coordinated Subplots <i>Julie Porteous, Fred Charles and Marc Cavazza</i>	846
Hybrid Gaussian and von Mises Model-Based Clustering <i>Sergio Luengo-Sanchez, Concha Bielza and Pedro Larrañaga</i>	855
Adaptive Symbiotic Collaboration for Targeted Complex Manipulation Tasks <i>Rui Silva, Francisco S. Melo and Manuela Veloso</i>	863
Attuning Ontology Alignments to Semantically Heterogeneous Multi-Agent Interactions <i>Paula Chocron and Marco Schorlemmer</i>	871
On the Construction of High-Dimensional Simple Games <i>Martin Olsen, Sascha Kurz and Xavier Molinero</i>	880
A Dynamic Logic of Norm Change <i>Max Knobbout, Mehdi Dastani and John-Jules Meyer</i>	886
h-Index Manipulation by Undoing Merges <i>René van Bevern, Christian Komusiewicz, Hendrik Molter, Rolf Niedermeier, Manuel Sorge and Toby Walsh</i>	895
Planning Under Uncertainty for Aggregated Electric Vehicle Charging with Renewable Energy Supply <i>Erwin Walraven and Matthijs T.J. Spaan</i>	904

Part 2

On the Computation of Top-k Extensions in Abstract Argumentation Frameworks <i>Said Jabbour, Badran Raddaoui, Lakhdar Sais and Yakoub Salhi</i>	913
On Revision of Partially Specified Convex Probabilistic Belief Bases <i>Gavin Rens, Thomas Meyer and Giovanni Casini</i>	921
Solving Dynamic Controllability Problem of Multi-Agent Plans with Uncertainty Using Mixed Integer Linear Programming <i>Guillaume Casanova, Cédric Pralet, Charles Lesire and Thierry Vidal</i>	930
Combining Efficient Preprocessing and Incremental MaxSAT Reasoning for MaxClique in Large Graphs <i>Hua Jiang, Chu-Min Li and Felip Manyà</i>	939
An Efficient Approach for the Generation of Allen Relations <i>Kleanthi Georgala, Mohamed Ahmed Sherif and Axel-Cyrille Ngonga Ngomo</i>	948
You Can't Always Forget What You Want: On the Limits of Forgetting in Answer Set Programming <i>Ricardo Gonçalves, Matthias Knorr and João Leite</i>	957
Solving Set Optimization Problems by Cardinality Optimization with an Application to Argumentation <i>Wolfgang Faber, Mauro Vallati, Federico Cerutti and Massimiliano Giacomin</i>	966
The Need for Knowledge Extraction: Understanding Harmful Gambling Behavior with Neural Networks <i>Chris Percy, Artur S. d'Avila Garcez, Simo Dragičević, Manoel V.M. França, Greg Slabaugh and Tillman Weyde</i>	974
Strategy Representation and Reasoning in the Situation Calculus <i>Liping Xiong and Yongmei Liu</i>	982
Exploiting MUS Structure to Measure Inconsistency of Knowledge Bases <i>Said Jabbour and Lakhdar Sais</i>	991

Gaining Insight by Structural Knowledge Extraction <i>Pietro Cottone, Salvatore Gaglio, Giuseppe Lo Re and Marco Ortolani</i>	999
Complexity of Threshold Query Answering in Probabilistic Ontological Data Exchange <i>Thomas Lukasiewicz and Livia Predoiu</i>	1008
Markov Logic Networks with Numerical Constraints <i>Melisachew Wudage Chekol, Jakob Huber, Christian Meilicke and Heiner Stuckenschmidt</i>	1017
Revisiting the Cross Entropy Method with Applications in Stochastic Global Optimization and Reinforcement Learning <i>Ajin George Joseph and Shalabh Bhatnagar</i>	1026
Online Auctions for Dynamic Assignment: Theory and Empirical Evaluation <i>Sujit Gujar and Boi Faltings</i>	1035
Mathematical Programming Models for Optimizing Partial-Order Plan Flexibility <i>Buser Say, Andre A. Cire and J. Christopher Beck</i>	1044
On Distances Between KD45n Kripke Models and Their Use for Belief Revision <i>Thomas Caridroit, Sébastien Konieczny, Tiago de Lima and Pierre Marquis</i>	1053
Unsupervised Activity Recognition Using Latent Semantic Analysis on a Mobile Robot <i>Paul Duckworth, Muhannad Alomari, Yiannis Gatsoulis, David C. Hogg and Anthony G. Cohn</i>	1062
Dichotomy for Pure Scoring Rules Under Manipulative Electoral Actions <i>Edith Hemaspaandra and Henning Schnoor</i>	1071
Higher-Order Correlation Coefficient Analysis for EEG-Based Brain-Computer Interface <i>Ye Liu, Qibin Zhao, Min Chen and Liqing Zhang</i>	1080
Classtering: Joint Classification and Clustering with Mixture of Factor Analysers <i>Emanuele Sansone, Andrea Passerini and Francesco G.B. De Natale</i>	1089
Extending the Description Logic $\tau\mathcal{EL}(deg)$ with Acyclic TBoxes <i>Franz Baader and Oliver Fernández Gil</i>	1096
Clique-Width and Directed Width Measures for Answer-Set Programming <i>Bernhard Bliem, Sebastian Ordyniak and Stefan Woltran</i>	1105
Emotion Analysis as a Regression Problem – Dimensional Models and Their Implications on Emotion Representation and Metrical Evaluation <i>Sven Buechel and Udo Hahn</i>	1114
Aspect-Based Relational Sentiment Analysis Using a Stacked Neural Network Architecture <i>Soufian Jebbara and Philipp Cimiano</i>	1123
Learning Invariant Representation for Malicious Network Traffic Detection <i>Karel Bartos, Michal Sofka and Vojtech Franc</i>	1132
Making Sense of Item Response Theory in Machine Learning <i>Fernando Martínez-Plumed, Ricardo B.C. Prudêncio, Adolfo Martínez-Usó and José Hernández-Orallo</i>	1140
Skeptical, Weakly Skeptical, and Credulous Inference Based on Preferred Ranking Functions <i>Christoph Beierle, Christian Eichhorn, Gabriele Kern-Isberner and Steven Kutsch</i>	1149
Real-Time Timeline Summarisation for High-Impact Events in Twitter <i>Yiwei Zhou, Nattiya Kanhabua and Alexandra I. Cristea</i>	1158
Towards Better Models of Externalities in Sponsored Search Auctions <i>Nicola Gatti, Marco Rocco, Paolo Serafino and Carmine Ventre</i>	1167

A Computational Method for Extracting, Representing, and Predicting Social Closeness <i>Katherine Metcalf and David Leake</i>	1176
Planning Using Actions with Control Parameters <i>Emre Savaş, Maria Fox, Derek Long and Daniele Magazzeni</i>	1185
Fixed-Parameter Tractable Optimization Under DNNF Constraints <i>Frédéric Koriche, Daniel Le Berre, Emmanuel Lonca and Pierre Marquis</i>	1194
Preference Modeling with Possibilistic Networks and Symbolic Weights: A Theoretical Study <i>Nahla Ben Amor, Didier Dubois, H�ela Gouider and Henri Prade</i>	1203
Leveraging Stratification in Twitter Sampling <i>Vikas Joshi, Deepak P. and L.V. Subramaniam</i>	1212
A Minimization-Based Approach to Iterated Multi-Agent Belief Change <i>Paul Vicol, James Delgrande and Torsten Schaub</i>	1221
Parameterised Model Checking for Alternating-Time Temporal Logic <i>Panagiotis Kouvaros and Alessio Lomuscio</i>	1230
Interpretable Encoding of Densities Using Possibilistic Logic <i>Ondr�ej Ku�zelka, Jesse Davis and Steven Schockaert</i>	1239
Unsupervised Ranking of Knowledge Bases for Named Entity Recognition <i>Yassine Mrabet, Halil Kilicoglu and Dina Demner-Fushman</i>	1248
Skeleton-Based Orienteering for Level Set Estimation <i>Lorenzo Bottarelli, Manuele Bicego, Jason Blum and Alessandro Farinelli</i>	1256
Interruptible Task Execution with Resumption in Golog <i>Gesche Gierse, Tim Niemueller, Jens Cla�en and Gerhard Lakemeyer</i>	1265
Constructing Hierarchical Task Models Using Invariance Analysis <i>Damir Lotinac and Anders Jonsson</i>	1274
Learning the Structure of Dynamic Hybrid Relational Models <i>Davide Nitti, Irma Ravkic, Jesse Davis and Luc De Raedt</i>	1283
A Distributed Asynchronous Solver for Nash Equilibria in Hypergraphical Games <i>Mohamed Wahbi and Kenneth N. Brown</i>	1291
Summary Information for Reasoning About Hierarchical Plans <i>Lavindra de Silva, Sebastian Sardina and Lin Padgham</i>	1300
Knowledge-Based Programs with Defaults in a Modal Situation Calculus <i>Jens Cla�en and Malte Neuss</i>	1309
Solving Multi-Agent Knapsack Problems Using Incremental Approval Voting <i>Nawal Benabbou and Patrice Perny</i>	1318
Propositional Abduction with Implicit Hitting Sets <i>Alexey Ignatiev, Antonio Morgado and Joao Marques-Silva</i>	1327
Improved Multi-Label Classification Using Inter-Dependence Structure via a Generative Mixture Model <i>Ramanuja Simha and Hagit Shatkay</i>	1336
Accelerating Norm Emergence Through Hierarchical Heuristic Learning <i>Tianpei Yang, Zhaopeng Meng, Jianye Hao, Sandip Sen and Chao Yu</i>	1344
Entity Embeddings with Conceptual Subspaces as a Basis for Plausible Reasoning <i>Shoaib Jameel and Steven Schockaert</i>	1353

Automatic Bridge Bidding Using Deep Reinforcement Learning <i>Chih-Kuan Yeh and Hsuan-Tien Lin</i>	1362
Hierarchical Strategy Synthesis for Pursuit-Evasion Problems <i>Rattanachai Ramaithitima, Siddharth Srivastava, Subhrajit Bhattacharya, Alberto Speranzon and Vijay Kumar</i>	1370
Decidable Reasoning in a First-Order Logic of Limited Conditional Belief <i>Christoph Schwering and Gerhard Lakemeyer</i>	1379
A Temporal-Causal Modelling Approach to Integrated Contagion and Network Change in Social Networks <i>Romy Blankendaal, Sarah Parinussa and Jan Treur</i>	1388
Exact Particle Filter Modularization Improves Runtime Performance <i>Padraic D. Edgington and Anthony S. Maida</i>	1397
Using the Sugeno Integral in Optimal Assignment Problems with Qualitative Utilities <i>Soufiane Drissi Oudghiri, Patrice Perny, Olivier Spanjaard and Mohamed Hachimi</i>	1406
Complexity Results for Probabilistic Datalog [±] <i>İsmail İlkan Ceylan, Thomas Lukasiewicz and Rafael Peñaloza</i>	1414
Strategic Voting in a Social Context: Considerate Equilibria <i>Laurent Gourvès, Julien Lesca and Anaëlle Wilczynski</i>	1423
The Complexity of Deciding Legality of a Single Step of Magic: The Gathering <i>Krishnendu Chatterjee and Rasmus Ibsen-Jensen</i>	1432
Description Logics Reasoning w.r.t. General TBoxes Is Decidable for Concrete Domains with the EHD-Property <i>Claudia Carapelle and Anni-Yasmin Turhan</i>	1440
Realisability of Production Recipes <i>Lavindra de Silva, Paolo Felli, Jack C. Chaplin, Brian Logan, David Sanderson and Svetan Ratchev</i>	1449
Towards Lifelong Object Learning by Integrating Situated Robot Perception and Semantic Web Mining <i>Jay Young, Valerio Basile, Lars Kunze, Elena Cabrio and Nick Hawes</i>	1458
Declaratively Capturing Local Label Correlations with Multi-Label Trees <i>Reem Al-Otaibi, Meelis Kull and Peter Flach</i>	1467
Get Me to My GATE on Time: Efficiently Solving General-Sum Bayesian Threat Screening Games <i>Aaron Schlenker, Matthew Brown, Arunesh Sinha, Milind Tambe and Ruta Mehta</i>	1476
False-Name-Proof Mechanisms for Path Auctions in Social Networks <i>Lei Zhang, Haibin Chen, Jun Wu, Chong-Jun Wang and Junyuan Xie</i>	1485
Multi-Robot Adversarial Coverage <i>Roi Yehoshua and Noa Agmon</i>	1493
Parameterized Complexity Results for the Kemeny Rule in Judgment Aggregation <i>Ronald de Haan</i>	1502
AGM-Style Revision of Beliefs and Intentions <i>Marc van Zee and Dragan Doder</i>	1511
Facility Location Games with Optional Preference <i>Hongning Yuan, Kai Wang, Ken C.K. Fong, Yong Zhang and Minming Li</i>	1520
Iterative Judgment Aggregation <i>Marija Slavkovic and Wojciech Jamroga</i>	1528

Partial Order Temporal Plan Merging for Mobile Robot Tasks <i>Lenka Mudrova, Bruno Lacerda and Nick Hawes</i>	1537
ECAI Short Papers	
Towards Online Concept Drift Detection with Feature Selection for Data Stream Classification <i>Mahmood Hammoodi, Frederic Stahl and Mark Tennant</i>	1549
Towards SPARQL-Based Induction for Large-Scale RDF Data Sets <i>Simon Bin, Lorenz Bühmann, Jens Lehmann and Axel-Cyrille Ngonga Ngomo</i>	1551
Burg Matrix Divergence Based Multi-Metric Learning <i>Yan Wang and Han-Xiong Li</i>	1553
A Novel Approach of Applying the Differential Evolution to Spatial Discrete Data <i>Vojtěch Uher, Petr Gajdoš, Michal Radecký and Václav Snášel</i>	1555
A Stochastic Belief Change Framework with a Stream of Expiring Observations (Short Paper) <i>Gavin Rens</i>	1557
Topic-Level Influencers Identification in the Microblog Sphere <i>Yakun Wang, Zhongbao Zhang, Sen Su, Cheng Chang and Muhammad Azam Zia</i>	1559
Multilevel Agent-Based Modelling for Assignment or Matching Problems <i>Antoine Nongillard and Sébastien Picault</i>	1561
Simple Epistemic Planning: Generalised Gossiping <i>Martin C. Cooper, Andreas Herzig, Faustine Maffre, Frédéric Maris and Pierre Régnier</i>	1563
A General Characterization of Model-Based Diagnosis <i>Gregory Provan</i>	1565
On the Impact of Subproblem Orderings on Anytime AND/OR Best-First Search for Lower Bounds <i>William Lam, Kalev Kask, Rina Dechter and Javier Larrosa</i>	1567
Salient Region Detection Based on the Global Contrast Combining Background Measure for Indoor Robots <i>Na Li, Zhenhua Wang, Lining Sun and Guodong Chen</i>	1569
Semi-Supervised Learning on an Augmented Graph with Class Labels <i>Nan Li and Longin Jan Latecki</i>	1571
Identifying and Rewarding Subcrowds in Crowdsourcing <i>Siyuan Liu, Xiuyi Fan and Chunyan Miao</i>	1573
Data Set Operations to Hide Decision Tree Rules <i>Dimitris Kalles, Vassilios S. Verykios, Georgios Feretzakis and Athanasios Papagelis</i>	1575
Evolutionary Agent-Based Modeling of Past Societies' Organization Structure <i>Angelos Chliaoutakis and Georgios Chalkiadakis</i>	1577
Strategic Path Planning Allowing On-the-Fly Updates <i>Ofri Keidar and Noa Agmon</i>	1579
Finding Diverse High-Quality Plans for Hypothesis Generation <i>Shirin Sohrabi, Anton V. Riabov, Octavian Udrea and Oktie Hassanzadeh</i>	1581
Using a Deep Understanding of Network Activities for Network Vulnerability Assessment <i>Mona Lange, Felix Kuhr and Ralf Möller</i>	1583
All-Transfer Learning for Deep Neural Networks and Its Application to Sepsis Classification <i>Yoshihide Sawada, Yoshikuni Sato, Toru Nakada, Kei Ujimoto and Nobuhiro Hayashi</i>	1586

Computing Extensions' Probabilities in Probabilistic Abstract Argumentation: Beyond Independence <i>Bettina Fazzinga, Sergio Flesca and Filippo Furfaro</i>	1588
Explained Activity Recognition with Computational Assumption-Based Argumentation <i>Xiuyi Fan, Siyuan Liu, Huiguo Zhang, Cyril Leung and Chunyan Miao</i>	1590
Execution Errors Enable the Evolution of Fairness in the Ultimatum Game <i>Fernando P. Santos, Francisco C. Santos, Ana Paiva and Jorge M. Pacheco</i>	1592
Encoding Cryptographic Functions to SAT Using TRANSALG System <i>Ilya Otpuschennikov, Alexander Semenov, Irina Gribanova, Oleg Zaikin and Stepan Kochemazov</i>	1594
Explanatory Diagnosis of an Ontology Stream via Reasoning About Actions <i>Quan Yu, Hai Wan, Jiangtao Xu, Freddy Lécué and Liang Chang</i>	1596
DARDIS: Distributed And Randomized DIspatching and Scheduling <i>Thomas Bridi, Michele Lombardi, Andrea Bartolini, Luca Benini and Michela Milano</i>	1598
Reasoning About Belief and Evidence with Extended Justification Logic <i>Tuan-Fang Fan and Churn-Jung Liau</i>	1600
Scaling Structure Learning of Probabilistic Logic Programs by MapReduce <i>Fabrizio Riguzzi, Elena Bellodi, Riccardo Zese, Giuseppe Cota and Evelina Lamma</i>	1602
Employing Hypergraphs for Efficient Coalition Formation with Application to the V2G Problem <i>Filippos Christianos and Georgios Chalkiadakis</i>	1604
Increasing Coalition Stability in Large-Scale Coalition Formation with Self-Interested Agents <i>Pavel Janovsky and Scott A. DeLoach</i>	1606
Strategies for Privacy Negotiation in Online Social Networks <i>Dilara Kekülliöğlu, Nadin Kökciyan and Pınar Yolum</i>	1608
DA-BSP: Towards Data Association Aware Belief Space Planning for Robust Active Perception <i>Shashank Pathak, Antony Thomas, Asaf Feniger and Vadim Indelman</i>	1610
Bagged Boosted Trees for Classification of Ecological Momentary Assessment Data <i>Gerasimos Spanakis, Gerhard Weiss and Anne Roefs</i>	1612
Reputation in the Academic World <i>Nardine Osman and Carles Sierra</i>	1614
Collective Future Orientation and Stock Markets <i>Mohammed Hasanuzzaman, Wai Leung Sze, Mahammad Parvez Salim and Gaël Dias</i>	1616
Learning of Classification Models from Noisy Soft-Labels <i>Yanbing Xue and Milos Hauskrecht</i>	1618
Distributed Learning in Expert Referral Networks <i>Ashiqur R. KhudaBukhsh, Peter J. Jansen and Jaime G. Carbonell</i>	1620
Transfer Learning for Automatic Short Answer Grading <i>Shourya Roy, Himanshu S. Bhatt and Y. Narahari</i>	1622
Long-Time Sensor Data Analysis for Estimation of Physical Capacity <i>Yoshikuni Sato, Yoshihide Sawada, Toru Nakada, Tadayoshi Nonoyama, Masafumi Kubota, Yusuke Koie, Masaki Yasutake and Osamu Yamamura</i>	1624
Enhancing Sketch-Based Image Retrieval via Deep Discriminative Representation <i>Fei Huang, Yong Cheng, Cheng Jin, Yuejie Zhang and Tao Zhang</i>	1626
Structure in the Value Function of Two-Player Zero-Sum Games of Incomplete Information <i>Auke J. Wiggers, Frans A. Oliehoek and Diederik M. Roijers</i>	1628

GDL-III: A Proposal to Extend the Game Description Language to General Epistemic Games <i>Michael Thielscher</i>	1630
Impact of Automated Action Labeling in Classification of Human Actions in RGB-D Videos <i>David Jardim, Luis Nunes and Miguel Dias</i>	1632
Transductive Learning for the Identification of Word Sense Temporal Orientation <i>Mohammed Hasanuzzaman, Gaël Dias and Stéphane Ferrari</i>	1634
Towards a Framework for Detecting Opportunism in Multi-Agent Systems <i>JiETING Luo, John-Jules Meyer and Max Knobbout</i>	1636
Learning a Bayesian Network Classifier by Jointly Maximizing Accuracy and Information <i>Dan Halbersberg and Boaz Lerner</i>	1638
Transfer of Reinforcement Learning Negotiation Policies: From Bilateral to Multilateral Scenarios <i>Ioannis Efstathiou and Oliver Lemon</i>	1640
Substantive Irrationality in Cognitive Systems <i>Pierre Bisquert, Madalina Croitoru, Florence Dupin de Saint-Cyr and Abdelraouf Hecham</i>	1642
A History Tree Heuristic to Generate Better Initiation Sets for Options in Reinforcement Learning <i>Alper Demir, Erkin Çilden and Faruk Polat</i>	1644
On Truthful Auction Mechanisms for Electricity Allocation <i>Heng Song, Junwu Zhu, Bin Li and Yi Jiang</i>	1646
Towards a BDI Player Model for Interactive Narratives <i>Jessica Rivera-Villicana, Fabio Zambetta, James Harland and Marsha Berry</i>	1648
A Typicality-Based Revision to Handle Exceptions in Description Logics <i>Roberto Micalizio and Gian Luca Pozzato</i>	1650
A New Stochastic Local Search Approach for Computing Preferred Extensions of Abstract Argumentation <i>Dangdang Niu, Lei Liu and Shuai Lü</i>	1652
Crowdsourced Referral Auctions <i>Praphul Chandra, Sujit Gujar and Y. Narahari</i>	1654
Minisum and Minimax Committee Election Rules for General Preference Types <i>Dorothea Baumeister, Toni Böhnlein, Lisa Rey, Oliver Schaudt and Ann-Kathrin Selker</i>	1656
Space Debris Removal: A Game Theoretic Analysis <i>Richard Klima, Daan Bloembergen, Rahul Savani, Karl Tuyls, Daniel Hennes and Dario Izzo</i>	1658
Cuilt: A Scalable, Mix-and-Match Framework for Local Iterative Approximate Best-Response Algorithms <i>Mihaela Verman, Philip Stutz, Robin Hafen and Abraham Bernstein</i>	1660
Not Being at Odds with a Class: A New Way of Exploiting Neighbors for Classification <i>Myriam Bounhas, Henri Prade and Gilles Richard</i>	1662
Value-Based Reasoning and Norms <i>Trevor Bench-Capon</i>	1664
Using Recursive Neural Networks to Detect and Classify Drug-Drug Interactions from Biomedical Texts <i>Victor Suárez-Paniagua and Isabel Segura-Bedmar</i>	1666
Efficient Computation of Deterministic Extensions for Dynamic Abstract Argumentation Frameworks <i>Sergio Greco and Francesco Parisi</i>	1668
The Post-Modern Homunculus <i>Eric Neufeld and Sonje Finnestad</i>	1670

Symmetric Multi-Aspect Evaluation of Comments – Extended Abstract <i>Theodore Patkos, Giorgos Flouris and Antonis Bikakis</i>	1672
An Efficient and Expressive Similarity Measure for Relational Clustering Using Neighbourhood Trees <i>Sebastijan Dumančić and Hendrik Blockeel</i>	1674
On Inconsistency Measuring and Resolving <i>Said Jabbour</i>	1676
A Fuzzy Semantic CEP Model for Situation Identification in Smart Homes <i>Amina Jarraya, Nathan Ramoly, Amel Bouzeghoub, Khedija Arour, Amel Borgi and Béatrice Finance</i>	1678
Multi-Context Systems in Time <i>Stefania Costantini and Andrea Formisano</i>	1680
Speech Emotion Recognition Using Voiced Segment Selection Algorithm <i>Yu Gu, Eric Postma, Hai-Xiang Lin and Jaap van den Herik</i>	1682
Scalable Exact MAP Inference in Graphical Models <i>Radu Marinescu, Akihiro Kishimoto and Adi Botea</i>	1684
Hiding Actions in Concurrent Games <i>Vadim Malvone, Aniello Murano and Loredana Sorrentino</i>	1686
Shape Invariant Formulation for Change Point Models in Multiple Dimensions <i>Jesse Glass and Zoran Obradovic</i>	1688
Shaping Proto-Value Functions Using Rewards <i>Raj Kumar Maity, Chandrashekar Lakshminarayanan, Sindhu Padakandla and Shalabh Bhatnagar</i>	1690
Heuristic Constraint Answer Set Programming <i>Erich C. Teppan and Gerhard Friedrich</i>	1692
Non-Deterministic Planning with Numeric Uncertainty <i>Liana Marinescu and Andrew Coles</i>	1694
How Good Is Predictive Routing in the Online Version of the Braess Paradox? <i>László Z. Varga</i>	1696
Delete-Free Reachability Analysis for Temporal and Hierarchical Planning <i>Arthur Bit-Monnot, David E. Smith and Minh Do</i>	1698
Intention Selection with Deadlines <i>Yuan Yao, Brian Logan and John Thangarajah</i>	1700
Cost-Optimal Algorithms for Planning with Procedural Control Knowledge <i>Vikas Shivashankar, Ron Alford, Mark Roberts and David W. Aha</i>	1702
Adaptive Condorcet-Based Stopping Rules Can Be Efficient <i>Omer Reingold and Nina Narodytska</i>	1704
Landmark-Based Plan Recognition <i>Ramon Fraga Pereira and Felipe Meneguzzi</i>	1706
Multi-Level Semantics with Vertical Integrity Constraints <i>Alison R. Panisson, Rafael H. Bordini and Antônio Carlos da Rocha Costa</i>	1708
Supervised Graph-Based Term Weighting Scheme for Effective Text Classification <i>Nilofer Shanavas, Hui Wang, Zhiwei Lin and Glenn Hawe</i>	1710
Temporal Planning with Constants in Context <i>Josef Bajada, Maria Fox and Derek Long</i>	1712

Secure Multi-Agent Planning Algorithms <i>Michal Štolba, Jan Tožička and Antonín Komenda</i>	1714
Analysis of Swarm Communication Models <i>Musad Haque, Christopher Ren, Electa Baker, Douglas Kirkpatrick and Julie A. Adams</i>	1716
Fault Manifestability Verification for Discrete Event Systems <i>Lina Ye, Philippe Dague, Delphine Longuet, Laura Brandán Briones and Agnes Madalinski</i>	1718
Using Petri Net Plans for Modeling UAV-UGV Cooperative Landing <i>Andrea Bertolaso, Masoume M. Raeissi, Alessandro Farinelli and Riccardo Muradore</i>	1720
Applications of Argumentation: The <i>SoDA</i> Methodology <i>Nikolaos I. Spanoudakis, Antonis C. Kakas and Pavlos Moraitis</i>	1722
Decoupling a Resource Constraint Through Fictitious Play in Multi-Agent Sequential Decision Making <i>Frits de Nijs, Matthijs T.J. Spaan and Mathijs M. de Weerd</i>	1724
GPU-Accelerated Value Iteration for the Computation of Reachability Probabilities in MDPs <i>Zhimin Wu, Ernst Moritz Hahn, Akin Günay, Lijun Zhang and Yang Liu</i>	1726
Sets of Contrasting Rules to Identify Trigger Factors <i>Marharyta Aleksandrova, Armelle Brun, Oleg Chertov and Anne Boyer</i>	1728
Link Prediction by Incidence Matrix Factorization <i>Sho Yokoi, Hiroshi Kajino and Hisashi Kashima</i>	1730
Dialogues as Social Practices for Serious Games <i>Agnese Augello, Manuel Gentile, Lucas Weideveld and Frank Dignum</i>	1732
Abducting Workflow Traces: A General Framework to Manage Incompleteness in Business Processes <i>Federico Chesani, Riccardo De Masellis, Chiara Di Francescomarino, Chiara Ghidini, Paola Mello, Marco Montali and Sergio Tessaris</i>	1734
Optimal Simple Strategies for Persuasion <i>Elizabeth Black, Amanda Coles and Christopher Hampson</i>	1736
Non-Utilitarian Coalition Structure Generation <i>Oskar Skibski, Henryk Michalewski, Andrzej Nagórko, Tomasz P. Michalak, Andrew Dowell, Talal Rahwan and Michael Wooldridge</i>	1738
PA*: Optimal Path Planning for Perception Tasks <i>Tiago Pereira, Manuela Veloso and António Moreira</i>	1740
Cross-Domain Error Correction in Personality Prediction <i>Işıl Doğa Yakut Kılıç and Shimei Pan</i>	1742
Classical Planning with Communicative Actions <i>Tânia Marques and Michael Rovatsos</i>	1744
Mixed Strategy Extraction from UCT Tree in Security Games <i>Jan Karwowski and Jacek Mańdziuk</i>	1746
Boolean Negotiation Games <i>Nils Bulling and Koen V. Hindriks</i>	1748
Toward Addressing Collusion Among Human Adversaries in Security Games <i>Shahzad Gholami, Bryan Wilder, Matthew Brown, Dana Thomas, Nicole Sintov and Milind Tambe</i>	1750
Detecting Communities Using Coordination Games: A Short Paper <i>Radhika Arava and Pradeep Varakantham</i>	1752

Pricing Options with Portfolio-Holding Trading Agents in Direct Double Auction <i>Sarvar Abdullaev, Peter McBurney and Katarzyna Musial</i>	1754
Efficient Semantic Tableau Generation for Abduction in Propositional Logic <i>Yifan Yang, Ricardo De Aldama, Jamal Atif and Isabelle Bloch</i>	1756
Case-Based Classification on Hierarchical Structure of Formal Concept Analysis <i>Qi Zhang, Chongyang Shi, Ping Sun and Zhengdong Niu</i>	1758
Stochastic Area Pooling for Generic Convolutional Neural Network <i>Zhidong Deng, Zhenyang Wang and Shiyao Wang</i>	1760
Hole in One: Using Qualitative Reasoning for Solving Hard Physical Puzzle Problems <i>Xiaoyu Ge, Jae Hee Lee, Jochen Renz and Peng Zhang</i>	1762
Relational Grounded Language Learning <i>Leonor Becerra-Bonache, Hendrik Blockeel, María Galván and François Jacquenet</i>	1764
PAIS Papers	
Learning the Repair Urgency for a Decision Support System for Tunnel Maintenance <i>Y. Gatsoulis, M.O. Mehmood, V.G. Dimitrova, D.R. Magee, B. Sage-Vallier, P. Thiaudiere, J. Valdes and A.G. Cohn</i>	1769
ONE – A Personalized Wellness System <i>Ajay Chander and Ramya Srinivasan</i>	1775
Planning Search and Rescue Missions for UAV Teams <i>Chris A.B. Baker, Sarvapali Ramchurn, W.T. Luke Teacy and Nicholas R. Jennings</i>	1777
Integrating ARIMA and Spatiotemporal Bayesian Networks for High Resolution Malaria Prediction <i>A.H.M. Imrul Hasan and Peter Haddawy</i>	1783
Rapid Adaptation of Air Combat Behaviour <i>Armon Toubman, Jan Joris Roessingh, Pieter Spronck, Aske Plaat and Jaap van den Herik</i>	1791
An Intelligent System for Personalized Conference Event Recommendation and Scheduling <i>Aldy Gunawan, Hoong Chuin Lau, Pradeep Varakantham and Wenjie Wang</i>	1797
Continuous Live Stress Monitoring with a Wristband <i>Martin Gjoreski, Hristijan Gjoreski, Mitja Luštrek and Matjaž Gams</i>	1803
An Intelligent System for Aggression De-Escalation Training <i>Tibor Bosse, Charlotte Gerritsen and Jeroen de Man</i>	1805
A Practical Approach to Fuse Shape and Appearance Information in a Gaussian Facial Action Estimation Framework <i>Teena Hassan, Dominik Seuss, Johannes Wollenberg, Jens Garbas and Ute Schmid</i>	1812
Planning Tourist Agendas for Different Travel Styles <i>Jesus Ibañez, Laura Sebastia and Eva Onaindia</i>	1818
Author Index	1825

This page intentionally left blank

ECAI Long Papers

This page intentionally left blank

When Do Rule Changes Count-As Legal Rule Changes?

Thomas C. King and Virginia Dignum and Catholijn M. Jonker¹

Abstract. Institutions regulate societies. Comprising Searle’s constitutive counts-as rules, “A counts-as B in context C”, an institution ascribes from brute and institutional facts (As), a social reality comprising institutional facts (Bs) conditional on the social reality (contexts Cs). When brute facts change an institution evolves from one social reality to the next. Rule changes are also regulated by *rule-modifying* counts-as rules ascribing rule change in the past/present/future (e.g. a majority rule change vote *counts-as* a rule change). Determining rule change legality is difficult, since changing counts-as rules both alters and is conditional on the social reality, and in some cases hypothetical rule-change effects (e.g. not retroactively criminalising people). However, without a rigorous account of rule change ascriptions, AI agents cannot support humans in understanding the laws imposed on them. Moreover, advances in automated governance design for socio-technical systems, are limited by agents’ ability to understand how and when to enact institutional changes. Consequently, we answer “when do rule changes count-as legal rule changes?” in a temporal setting with a novel formal framework.

1 Introduction

Institutions regulate and govern society and have been widely formalised (see [Andrighetto et al. \[2013\]](#)). Institutions construct a descriptive and prescriptive social reality from brute facts with Searle’s ([Searle \[1969, 2005\]](#)) *counts-as* rules, “A counts-as B in context C”. When the brute facts change an institution’s social reality evolves according to counts-as rules. Counts-as rules can also be modified over time. In legal systems, secondary counts-as rules ascribe rule change ([Biagioli \[1997\]](#)). We view these rules as rule-modifying counts-as rules - “A counts-as *modifying a rule* in context C”.

Yet, it is difficult to determine which rule changes can be made according to rule-modifying counts-as rules. Rules build the social reality, ascribe rule changes conditional on the social reality, and are also subject to being changed. This affects which rule changes are possible in the first place, for example:

- A group of people voting to change a rule counts-as a legal rule change if they constitute the government. A rule change can affect the social reality by redefining it (e.g. who counts-as being in government); rule changes are conditional on the built social reality.
- The UK government voted to *retroactively* require UK residents in a business partnership abroad to pay tax ([\[Fin, 2008, Sec. 58\]](#)), criminalising people in the past. Criminalising retroactive modifications are impossible according to the European Convention of Human Rights ([\[Council of Europe, 1953, Art. 7\]](#)). Rule change affects the social reality (e.g. criminalising people in the past); rule change is conditional on its hypothetical effects (e.g. being impossible if it would criminalise people in the past).

- A monarch or parliament can change laws. The monarch enacts a law obliging all fences are painted white. The parliament retroactively repeals the power for the monarch to enact laws, reversing the fence-painting law enactment. Retroactive rule change affects past rule-modifying counts-as rules; past rule modifications can be unravelled due to retroactive modifications.

An interdependency exists between the counts-as rules that construct a social reality and rule-modifying counts-as rules. Changing counts-as rules affects the past/present/future social reality and can change the modifications which happened in the past up until the present; rule modifications are conditional on the past/present/future social reality and the hypothetical rule change effects. Whether a rule change counts-as a *legal* rule change requires assessing the social reality in which the change takes place and the potential rule change effects, thus affecting whether a rule change is legal in the first place.

A defeasible logic for rule change over time has been proposed ([Governatori et al. \[2005\]](#); [Governatori and Rotolo \[2010\]](#)). But, crucially, not for rule change ascribed by counts-as rules accounting for the interdependency between changing rules and building a social reality. In ([Boella and van der Torre \[2004\]](#)) counts-as rules that regulate rule modifications are formalised, but not in a temporal setting. Yet, there has been little attention paid to formalising rule change regulated by counts-as rules in a temporal setting. This limits endeavours in AI to assist human agents in understanding the laws that govern them. Moreover, whilst AI agents are increasingly used to synthesise normative systems ([Morales et al. \[2014, 2015\]](#)), they are held back in enacting institutional rule changes by not understanding how and when laws can be changed.

This raises the question, in a temporal setting - *when do rule changes count-as legal rule changes?* We address this question with a novel formalisation for past/present/future institution rule change ascribed by counts-as rules. Our desiderata being that if a rule change is legal then it occurs, and otherwise it does not and the institution continues to operate ‘as usual’. In particular, taking into account the interdependency between ascribing rule change and changing rules. We posit that the most recent rule modifications take precedent and potentially change past modifications. We extend rules commonly found in the literature from being conditional on the present, to the past, different institution versions and hypothetical rule change effects.

We continue with our approach (2). Then we introduce the formalism, comprising a representation (3) and semantics (4). The framework is applied to five case studies (5). Finally, we discuss related work (6) and conclusions (7).

2 Approach

This paper formalises institutional rule change in a temporal setting. Foundational reasoning is required for institutions in a temporal setting, on which our framework is built. We require reasoning about

¹ Delft University of Technology, Netherlands, email addresses: {t.c.king-1, m.v.dignum, c.m.jonker}@tudelft.nl

counts-as rules, “A counts-as B in context C”, of two types. Firstly, rules that ascribe institutional events from other institutional events or observable events (brute facts) conditional on the social reality. For example, “the event we call a person paying tax (brute) counts-as paying tax (institutional)”, and “paying tax (institutional) counts-as fulfilling your duties (institutional)”. Secondly, rules which ascribe institutional fluents (institutional facts describing the state of affairs) from institutional events and cause the social reality (institutional state) to change. For example, “signing a business declaration counts-as initiating you are a business partner”. We require reasoning for counts-as rules that cause events to occur and the institutional state to change, which the InstAL (Institution Action Language) framework (Cliffe et al. [2007]; Cliffe [2007]) provides. Crucially, InstAL lacks representation and reasoning for rule change ascription.

We extend InstAL’s institutions with counts-as rules that ascriptively regulate rule changes and counts-as rules that themselves are modifiable. Rule modifications activate/deactivate rules in the past/present/future, analogous to enacting regulatory changes. Rule-modifying counts-as rules, “an event A counts-as a *rule-modifying* event B in context C”, ascribe past/present/future rule modifications.

Unlike in InstAL, in this paper institutions evolve dually: 1. when a rule change is ascribed by counts-as rules, the institution evolves to the next version potentially comprising different (active) counts-as rules at different time points, and 2. when observable events occur, each institution version evolves from one state to the next.

For example, an institution starts at version one, only comprising rules which enable rules to be added. A rule is added to the institution on Monday, stating that the tax year’s start causes an obligation to pay tax. Thus, on the Monday the institution evolves to version two, where the tax rule becomes active on the Monday, at which point version two becomes the *current version*. On Tuesday it is the first tax year month, both versions evolve to a new state, in version two the new state contains an obligation to pay tax, but not in version one since the tax rule was activated in version two. Each institution version evolves from state to state, and the institution evolves from one current version to the next when rule change events occur.

Contexts in counts-as rules are extended from being conditional on the present state to also past institution versions and states. This supports representing rule change conditional on its potential retroactive effects. For example, a condition on rule change not criminalising people in a version’s past *compared to the previous version’s past*. To summarise, we extend institutions to evolve along rule version and state timelines according to counts-as rules conditional on past versions and states, and potential rule change effects.

3 Representation

We begin with representing institutions which regulate their own temporal rule modifications.

Definition 1. Institution An institution is a tuple $\mathcal{I} = \langle \mathcal{E}, \mathcal{F}, \mathcal{C}, \mathcal{G}, \Delta \rangle$. Institutions are distinguished with a superscript (e.g. $\mathcal{I}^{uk} = \langle \mathcal{E}^{uk}, \mathcal{F}^{uk}, \mathcal{C}^{uk}, \mathcal{G}^{uk}, \Delta^{uk} \rangle$). $\Sigma = 2^{\mathcal{F}}$ denotes all states for \mathcal{I} .

Where:

1. $\mathcal{E} = \mathcal{E}_{obs} \cup \mathcal{E}_{inst} \cup \mathcal{E}_{mod}$ is a finite set of *events* comprising:
 - Observable events \mathcal{E}_{obs} and institutional events \mathcal{E}_{inst} .
 - Rule modification events $\mathcal{E}_{mod} = \{mod(op, id, t) \mid op \in \{act, deact\}, id \in \mathbb{ID}, t \in \mathbb{N}\}$ - a rule with the identifier id (the identifier set being \mathbb{ID}) is activated/deactivated (op) at a time t .
2. $\mathcal{F} = \mathcal{F}_{dom} \cup \mathcal{F}_{ract}$ is a finite set of *fluents* describing the:

- Domain \mathcal{F}_{dom} .
 - Active rules $\mathcal{F}_{ract} = \{active(id) \mid id \in \mathbb{ID}\}$ identified as id .
3. \mathcal{X} is the set of all *contexts* φ expressible in the following grammar for fluents $f \in \mathcal{F}$:

$$\begin{aligned} \varphi ::= & \top \mid f \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi \mid P \mid \\ & PrS(\varphi) \mid PaS(\varphi) \mid PrV(\varphi) \mid PaV(\varphi) \\ \phi ::= & \varphi \mid NS(\varphi) \end{aligned}$$

- Each expression’s informal meaning is the usual for propositional logic symbols. The operators bear truth in the following cases: (a) P if the context is retroactive (i.e. the state in which P operates on is at a time before the version to which it belongs becomes the current version), and (b) if φ is true in: the previous state ($PrS(\varphi)$), all past states ($PaS(\varphi)$), the same state in the previous version ($PrV(\varphi)$), the same state in all past versions ($PaV(\varphi)$), and the next state ($NS(\varphi)$). The next state operator is restricted to past versions, meaning rules are never conditional on the actual future.
4. $\mathcal{G} : \mathcal{X} \times 2^{\mathcal{E}} \rightarrow 2^{\mathcal{E}_{inst}}$ - is the *event generation function* where $\mathcal{G}(X, E)$ is an event set caused by the events that occur (E) when the context X holds.
 5. $\mathcal{C} : \mathcal{X} \times \mathcal{E} \rightarrow 2^{\mathcal{F}_{dom}} \times 2^{\mathcal{F}_{dom}}$ is the *state consequence function* where for a context $X \in \mathcal{X}$ and an event $e \in \mathcal{E}$ the consequence function’s result is notated $\mathcal{C}(X, e) = \langle \mathcal{C}^\uparrow(X, e), \mathcal{C}^\downarrow(X, e) \rangle$ s.t. the initiated fluent set is $\mathcal{C}^\uparrow(X, e)$ and the terminated fluent set is $\mathcal{C}^\downarrow(X, e)$
 6. $\Delta \subseteq \mathcal{F}$ is the *initial institution state*

For example, the following rule states that if Ada is found guilty ($g(ada)$) then she becomes a criminal ($crim(ada)$). That is, the fluent $crim(ada)$ is initiated by the event of being found guilty according to the consequence function (\mathcal{C}^\uparrow).

$$\mathcal{C}^\uparrow(\top, g(ada)) \ni crim(ada)$$

A government rule change ($gmod(act, id, t)$) that does not retroactively criminalise people counts-as a legal rule change. The condition is in all past retroactive states someone is not a criminal ($crim(ada)$) if in the previous version (prior to rule change) they were not.

$$\begin{aligned} \mathcal{G}(PaS(P \rightarrow PrV(\neg crim(ada)) \rightarrow \neg crim(ada))), \\ \{gmod(act, id, t)\} \ni act(id, t) \end{aligned}$$

In order to reason about modifying specific institutional rules, we tie rule identifiers to the institutional rules they represent. Specifically we map the inputs and single outputs of \mathcal{G} and \mathcal{C} to identifiers (i.e. not the whole set of events or initiated/terminated fluents).

Definition 2. Rule Identifier Function A rule identifier function for an event generation function $\mathcal{G} : \mathcal{X} \times 2^{\mathcal{E}} \rightarrow 2^{\mathcal{E}_{inst}}$ is $rid^{\mathcal{G}} : \mathcal{X} \times 2^{\mathcal{E}} \times \mathcal{E}_{inst} \rightarrow \mathbb{ID}$. The rule identifier functions for a consequence function $\mathcal{C} : \mathcal{X} \times \mathcal{E} \rightarrow 2^{\mathcal{F}_{dom}} \times 2^{\mathcal{F}_{dom}}$ are $rid^{\mathcal{C}^\uparrow} : \mathcal{X} \times \mathcal{E} \times \mathcal{F}_{dom} \rightarrow \mathbb{ID}$ and $rid^{\mathcal{C}^\downarrow} : \mathcal{X} \times \mathcal{E} \times \mathcal{F}_{dom} \rightarrow \mathbb{ID}$.

So, the previous rule criminalising Ada has the ID $crim0 = rid^\uparrow(\top, g(ada), crim(ada))$. Examples/case studies omit this function.

4 Semantics

This section defines institution semantics, following InstAL’s method using just sets and functions, with the following considerations.

Observable events cause an institution rule version to transition from state to state by generating transitioning events according to

the event generation function \mathcal{G} and initiating and terminating fluents according to the consequence function \mathcal{C} . An institution transitions from one version of rules to another when rule modifying events are generated by the event generation function \mathcal{G} .

An institutional interpretation represents this dual evolution as a tuple $M = \langle R, V \rangle$ where: 1. $V = \langle V_0, \dots, V_j \rangle$ is a tuple of versions each comprising a state and event set sequence up to length k with typical element $V_v = \langle S_v, E_v \rangle$. The state sequence for v is $S_v = \langle S_{v:0}, \dots, S_{v:k+1} \rangle$ with typical element $S_{v:i} \in \Sigma$ and the event set sequence (the events transitioning between states) is $E_v = \langle E_{v:0}, \dots, E_{v:k} \rangle$ with typical element $E_{v:i} \subseteq \mathcal{E}$. States denoted $S_{v:i}$ and event sets $E_{v:t}$ are denoted with the version v to which they belong and their time instant i . 2. $R : [0, k] \rightarrow [0, j]$ is a function stating which institution version is the *current* version for a given time.

R also represents when rule change events occurring in a version can change that version's rules. Rule modification events only change version rules if the institution has not already evolved to a later version. For example, if on Monday a rule is added, then the institution evolves to a new *current* version where that rule is actually added on Monday. When the version evolves, previous versions become *obsolete* from then onwards (e.g. Monday) meaning their rules are not changeable. If $R(i) \leq v$ then an event occurring in version v at time i can modify rules in v since the version is not yet obsolete.

The semantics are defined with respect to the interpretation $M = \langle R, V \rangle$, an institution $\mathcal{I} = \langle \mathcal{E}, \mathcal{F}, \mathcal{C}, \mathcal{G}, \Delta \rangle$, the set of all institutional interpretations \mathbb{I} , and an observable event trace $et = \langle O_0, \dots, O_k \rangle$.

4.1 Institutional Change

Counts-as rules, causing institution state and version change, are conditional on a context being *modelled* by the state in an *interpretation*.

Definition 3. Modelling Context For all $X \in \mathcal{X}$ and $f \in \mathcal{F}$, context models $\langle M, S_{v:t} \rangle \models X$ is defined for \top , \vee and \rightarrow w.r.t. \neg and \wedge as usual and for the other symbols as:

$$\langle M, S_{v:t} \rangle \models f \quad \Leftrightarrow \quad f \in S_{v:t} \quad (3.1)$$

$$\langle M, S_{v:t} \rangle \models \neg \psi \quad \Leftrightarrow \quad \langle M, S_{v:t} \rangle \not\models \psi \quad (3.2)$$

$$\langle M, S_{v:t} \rangle \models \psi \wedge \phi \quad \Leftrightarrow \quad \langle M, S_{v:t} \rangle \models \psi \text{ and } \langle M, S_{v:t} \rangle \models \phi \quad (3.3)$$

$$\langle M, S_{v:t} \rangle \models P \quad \Leftrightarrow \quad R(t) < v \quad (3.4)$$

$$\langle M, S_{v:t} \rangle \models PrS(\psi) \quad \Leftrightarrow \quad \langle M, S_{v:t-1} \rangle \models \psi \quad (3.6)$$

$$\langle M, S_{v:t} \rangle \models PaS(\psi) \quad \Leftrightarrow \quad \forall t' \in [0, t-1] : \langle M, S_{v:t-1} \rangle \models \psi \quad (3.7)$$

$$\langle M, S_{v:t} \rangle \models PrV(\psi) \quad \Leftrightarrow \quad \langle M, S_{v-1:t} \rangle \models \psi \quad (3.8)$$

$$\langle M, S_{v:t} \rangle \models PaV(\psi) \quad \Leftrightarrow \quad \forall v' \in [0, v-1] : \langle M, S_{v'-1:t} \rangle \models \psi \quad (3.9)$$

$$\langle M, S_{v:t} \rangle \models NS(\psi) \quad \Leftrightarrow \quad \langle M, S_{v:t+1} \rangle \models \psi \quad (3.10)$$

Semantics are as usual for modelling a fluent (3.1), weak negation (3.2) and conjunction (3.3). A state is retroactive if at that time the version is not the current version but it will be in the future (3.4) - for example, if on a Wednesday the institution evolves to a new version, then anything occurring on the Monday is retroactive to the new version (i.e. occurring in the version's past). States model formula as expected for a previous state (3.6), all previous states (3.7), the previous version (3.8), all past versions (3.9) and the next state (3.10).

An event 'B' occurs when transitioning to a new state in a version according to a rule - "A counts-as B in context C" (\mathcal{G}) - if an event 'A' occurs, the context 'C' is modelled by the state and the counts-as rule itself is active in the version's state. Events occurring in response to observable events E are formalised as an event generation operation.

Definition 4. Event Generation Operation The event generation operation $GR : \Sigma \times 2^{\mathcal{E}} \times \mathbb{I} \rightarrow 2^{\mathcal{E}}$ is defined such that $GR(S_{v:t}, E, M) = E'$ iff E' only satisfies the following conditions:

$$E \subseteq E' \quad (4.1)$$

$$\begin{aligned} \exists X \in \mathcal{X}, e \subseteq E, e' \in \mathcal{G}(X, e) \cap \mathcal{E}_{inst} : id = rid^{\mathcal{G}}(X, e, e'), \\ \langle M, S_{v:t} \rangle \models X \wedge active(id) \Rightarrow e' \in E' \end{aligned} \quad (4.2)$$

$$\begin{aligned} \exists X \in \mathcal{X}, e \subseteq E, e' \in \mathcal{G}(X, e) \cap \mathcal{E}_{mod} : id = rid^{\mathcal{G}}(X, e, e'), \\ \langle M, S_{v:t} \rangle \models X \wedge active(id), R(t) \neq v \Rightarrow e' \in E' \end{aligned} \quad (4.3)$$

$$\begin{aligned} \exists X \in \mathcal{X}, e \subseteq E, e' \in \mathcal{G}(X, e) \cap \mathcal{E}_{mod} : id = rid^{\mathcal{G}}(X, e, e'), \\ \langle M, S_{v:t} \rangle \models X \wedge active(id), R(t) = v \Rightarrow (e' \in E' \text{ or } e' \notin E') \end{aligned} \quad (4.4)$$

Any fixed point reached after iterative applications of GR is denoted as $GR^{\omega}(S_{v:t}, E, M)$.

Events that have occurred still occur (4.1). If an active rule states an event e causes an event e' in a context modelled by the state, then e can cause e' to occur depending on e' 's type. Specifically, whether e' is a type that could cause an inconsistency (e.g. removing rules that ascribe rule modifications, for more on the paradox of rule change see Suber [1990]). An event e' always occurs if it is a non-rule-modifying institutional event (4.2) or occurs when the version is obsolete and it cannot modify rules (4.3). Rule modifying events in non-obsolete versions *can* cause rule changes and a potential paradox. So they *optionally* occur in a non-obsolete version where they can cause rule change and/or a paradox (4.4). Hence, GR is *multi-valued*.

Iterating the event generation operation until a *fixed point* is reached obtains all events which occur. At least one fixed point is guaranteed.

Lemma 1. For any set of events $E \subseteq \mathcal{E}$, interpretation M and state $S_{v:t} \in \Sigma$ there exists a fixed point $GR^{\omega}(S_{v:t}, E, M)$.

Proof sketch. GR always has a monotonically increasing value (w.r.t. set inclusion) and a finite domain. \square

An institution version transitions between states, driven by event occurrences, according to a state transition operation.

Definition 5. State Transition Operation The state transition operation $TR : \Sigma \times 2^{\mathcal{E}} \times \mathbb{I} \rightarrow 2^{\mathcal{E}}$ is defined for a state $S_{v:t}$, a set of events $E_{v:t}$ and an interpretation M as:

$$TR(S_{v:t}, E_{v:t}, M) = \{f \mid f \in S_{v:t} \cap TERM(S_{v:t}, E_{v:t}, M) \text{ or } f \in INIT(S_{v:t}, E_{v:t}, M)\} \quad (5.1)$$

$$f \in INIT(S_{v:t}, E_{v:t}, M) \quad (5.2)$$

where:

$$INIT(S_{v:t}, E_{v:t}, M) = \{f \mid \exists e \in E_{v:t}, X \in \mathcal{X} : id = rid^{\mathcal{C}^\dagger}(X, e, f), f \in \mathcal{C}^\dagger(X, e) \cap \mathcal{F}_{dom}, \langle M, S_{v:t} \rangle \models X \wedge active(id) \text{ or } \exists t' \in [0, k], \ddagger t'' \in [t', k] : id = rid^{\mathcal{C}^\dagger}(X, e, f), R(t') \leq v, R(t'') \leq v, mod(act, id, t) \in E_{v:t'}, mod(deact, id, t) \in E_{v:t''}, f = active(id)\} \quad (5.3)$$

$$\exists t' \in [0, k], \ddagger t'' \in [t', k] : id = rid^{\mathcal{C}^\dagger}(X, e, f), R(t') \leq v, R(t'') \leq v, mod(act, id, t) \in E_{v:t'}, mod(deact, id, t) \in E_{v:t''}, f = active(id) \quad (5.4)$$

$$TERM(S_{v:t}, E_{v:t}, M) = \{f \mid \exists e \in E_{v:t}, X \in \mathcal{X} : id = rid^{\mathcal{C}^\dagger}(X, e, f), f \in \mathcal{C}^\dagger(X, e) \cap \mathcal{F}_{dom}, \langle M, S_{v:t} \rangle \models X \wedge active(id) \text{ or } \exists t' \in [0, k], \ddagger t'' \in [t', k] : id = rid^{\mathcal{C}^\dagger}(X, e, f), R(t') \leq v, R(t'') \leq v, mod(deact, id, t) \in E_{v:t'}, mod(act, id, t) \in E_{v:t''}, f = active(id)\} \quad (5.5)$$

$$\exists t' \in [0, k], \ddagger t'' \in [t', k] : id = rid^{\mathcal{C}^\dagger}(X, e, f), R(t') \leq v, R(t'') \leq v, mod(deact, id, t) \in E_{v:t'}, mod(act, id, t) \in E_{v:t''}, f = active(id) \quad (5.6)$$

Transitioning from one state to the next follows common-sense inertia - a fluent holds in a new state if it held in the previous state and was not terminated (5.1) or it was initiated in the previous state (5.2). A domain fluent is initiated/terminated if an event causes it to be according to a rule defined by the state consequence function \mathcal{C} that is active in the current state with a condition (context) that is modelled in the state (5.3 for initiation and 5.5 for termination). A fluent denoting an active rule is initiated/terminated in a state if a rule activating/deactivating event occurs at a time when the version is not obsolete and no contradictory deactivation/activation event occurs at a later time when the version is not obsolete (5.4 for activating rules and 5.6 for deactivating rules). The most recent modifications in a version take precedent if they occur when the version is a non-obsolete version and simultaneous contradictory rule modifications are cancelled.

4.2 Models

Now we define when an interpretation is an institutional model for an observable event set trace. An institutional interpretation is, broadly speaking, an institutional model for an observable event set trace iff: 1. each version evolves according to the event generation and state transition operations, and 2. the institution evolves from one version to another when rules are modified. However, the event generation operation is multi-valued since rule modifications are *optional*. Thus, there are potentially multiple candidate event sets for transitioning between states and therefore multiple interpretations to select as models.

We want to maximise the rule modification events that are not self-contradicting (e.g. not applying modifications that retroactively remove a rule making retroactive rule removal possible). Interpretations are prioritised, denoted as $<$, based on maximising rule modifications. An interpretation has higher priority over another if at the earliest time in the earliest version in which the interpretation differ it contains a superset of rule modifying events compared to the ‘same’ set for the lower priority interpretation.

Definition 6. Prioritised Interpretation Let $M^0 = \langle R^0, V^0 \rangle \in \mathbb{I}$ and $M^1 = \langle R^1, V^1 \rangle \in \mathbb{I}$ be two interpretations for institution \mathcal{I} where: $V^0 = \langle V_0^0, \dots, V_i^0 \rangle$ with typical element $V_v^0 = \langle E_v^0, S_v^0 \rangle$ s.t. $E_v^0 = \langle E_{v:0}^0, \dots, E_{v:k}^0 \rangle$, and $V^1 = \langle V_0^1, \dots, V_j^1 \rangle$ with typical element $V_v^1 = \langle E_v^1, S_v^1 \rangle$ s.t. $E_v^1 = \langle E_{v:0}^1, \dots, E_{v:k}^1 \rangle$. The ordering $<$ is a relation between interpretations M^0 and M^1 such that:

$$\begin{aligned} M^0 < M^1 \Leftrightarrow & \exists t \in [0, k], \nexists t' \in [0, t-1]: \\ & v = R^0(t), E_{v:t}^0 \cap \mathcal{E}_{mod} \supset E_{v:t}^1 \cap \mathcal{E}_{mod} \\ & v' = R^0(t'), E_{v':t'}^0 \neq E_{v':t'}^1 \end{aligned}$$

We operationally characterise a model by constructing a ‘correct’ interpretation. That is, constructing versions comprising correct state transitions and generated events. We could construct each institution version by starting at an initial state and proceeding from one state to the next according to the event generation and state transition operations. However, this would require knowing which rule modification events happen in each version’s past, present and future.

To give an example for an observable event set trace $\langle O_0, \dots, O_k \rangle$. An institution starts at an initial state only comprising an active rule enabling a government to make retroactive modifications ($\Delta = S_{0:0} = \{active(gov0)\}$). First, a fence is observably built ($O_0 = \{fb\}$, occurring during the first state transition $fb \in E_{0:0} = GR^{\omega}(S_{0:0}, O_0, M)$). But, there is no active rule that causes the next state to be different ($S_{0:1} = TR(S_{0:0}, E_{0:0}) = S_{0:0} = \{active(gov0)\}$). Then, the government votes to retroactively activate a rule in state zero, stating building

a fence initiates an obligation to paint it. Consequently, the second state which has already been determined, $S_{0:1}$, seems wrong since it lacks the fence painting rule and its effects. In fact, the institution should transition to a new rule version V_1 . This new version should start at the same initial state $S_{1:0} = \Delta$. But, crucially, transition to the next state ($S_{1:0} = TR(S_{1:0}, E_{1:0})$) with the knowledge that in the future of the new version the fence painting rule will be retroactively added at state zero ($S_{1:0}$) and become active in the second state ($S_{1:1}$). State transitions are defined with respect to an interpretation comprising past/present/future rule modification events which might be unknown when each state and transitioning event set is constructed.

We define an interpretation successor operation which addresses the problem of constructing a ‘correct’ interpretation without the knowledge of each version’s past/present/future. The successor operation takes as input a preceding interpretation which supplies versions comprising a past/present/future on which each version in the new succeeding interpretation can be constructed according to TR and GR . That is, a new interpretation is produced using the version timelines of the previous interpretation, taking into account past/present/future rule modifications from the preceding interpretation’s version timelines.

A succeeding interpretation might not be the same as the previous interpretation, since the previous interpretation might have been built without knowledge of its own past/present/future. That is, the new interpretation might differ in its temporal evolution (comparable version timelines in each interpretation being different). Consequently, the succeeding interpretation might have new, previously unknown, rule modification events that also need to be accounted for and thus another succeeding interpretation must be produced.

The idea is to iteratively apply the institution successor operation until a succeeding interpretation is produced that is the same as the previous interpretation. That is, until the operation reaches a fixed point, which is guaranteed according to lemma 3 we give later on. Intuitively, the fixed point characterises an interpretation that is built taking into account its own past/present/future modifications in each version (since it was built with respect to an identical preceding interpretation). Formally, the successor interpretation operation is:

Definition 7. Successor Interpretation Operation Let $et = \langle O_0, \dots, O_k \rangle$ be an observable event trace for \mathcal{I} of length k . Let $M' = \langle R', V' \rangle \in \mathbb{I}$ be an interpretation such that $V' = \langle V'_0, \dots, V'_j \rangle$ is a tuple of institution versions. The interpretation successor operation $SUCC : \mathbb{I} \times ET \rightarrow \mathbb{I}$ is defined for the interpretation M w.r.t. \mathcal{I} and et such that $SUCC(M, et) = M'$ iff M' satisfies the following conditions:

$$\forall v \in [0, j'] : S'_{v:0} = \Delta \quad (7.1)$$

$$\forall v \in [0, j'], t \in [0, k] : E'_{v:t} = GR^{\omega}(S'_{v:t}, O_t, M) \quad (7.2)$$

$$\forall v \in [0, j'], t \in [0, k] : S'_{v:t+1} = TR(S'_{v:t}, E'_{v:t}, M) \quad (7.3)$$

$$R'(t) = \begin{cases} 0, & t = 0, E'_{0:t} \cap \mathcal{E}_{mod} = \emptyset \\ 1, & t = 0, E'_{0:t} \cap \mathcal{E}_{mod} \neq \emptyset \\ R'(t-1), & t > 0, E'_{R'(t-1):t} \cap \mathcal{E}'_{mod} = \emptyset \\ R'(t-1)+1, & t > 0, E'_{R'(t-1):t} \cap \mathcal{E}'_{mod} \neq \emptyset \end{cases} \quad (7.4)$$

$$\text{Given that } V' = \langle V'_0, \dots, V'_j \rangle, R'(k) = j' \quad (7.5)$$

Every institution version starts at the same initial state (7.1). Each state transition (an event set) in a version is produced by the event generation operation applied to the previous state and the observable events occurring at that time (7.2). The next state in a version is the state produced by the state transition operation applied to the previous state and the transitioning events occurring in that version *with respect to the preceding institutional interpretation* (7.3). That is, transitioning

from one state to the next takes into account the rule modification events occurring in the past/present/future of the same version in the preceding interpretation. Rule modifications in the latest version cause the current version to evolve/increment to the next version. If no rule modification takes place the version remains the same or the zeroeth version for the zeroeth time instant (7.4). If a rule modification does take place in the latest version, then the current version at that time incremented by one, or is the first version for the zeroeth time point (7.4). The version sequence only goes up until the current version at the last time instant (7.5).

At least one fixed point for the successor interpretation operation, starting at any initial interpretation, is always guaranteed. A fixed point is denoted as $SUCC^\omega(M, et)$. To see why, the general idea is that there always exists a series of successive interpretations that monotonically increase which versions and states they agree on.

The following lemma is used to prove that there always exists a series of such interpretations and therefore that there always exists a fixed point. Informally, the lemma is conditional on there being two successors M' and M'' to any interpretation that agree with each other up until a particular time (h) in a version (j). The consequence is that the second interpretation M'' has the same events at time h and state transition at time $h+1$ in version j as if the event and state transitions were produced with respect to M'' 's own past/present/future timeline.

Lemma 2. *If \mathcal{I} is an institution, M an interpretation and et an observable event trace of length k for \mathcal{I} and there exists interpretations $M' = SUCC(M, et)$ and $M'' = SUCC(M', et)$ where $\exists h \in [0, k], j \in [0, v'], \forall i \in [0, k]$:*

$$\langle V'_0, \dots, V'_{j-1} \rangle = \langle V''_0, \dots, V''_{j-1} \rangle \quad (\text{A2.1})$$

$$\langle S'_{j:0}, \dots, S'_{j:h} \rangle = \langle S''_{j:0}, \dots, S''_{j:h} \rangle \quad (\text{A2.2})$$

$$\begin{aligned} \text{mod}(op, id, h) \in E'_{v:i}, \quad \text{mod}(op, id, h) \in E''_{j:i}, \\ R^i(t) \leq j \quad \Leftrightarrow \quad R''(t) \leq j \end{aligned} \quad (\text{A2.3})$$

then $E''_{j:h} = GR^\omega(S''_{j:h}, O_h, M'')$ and $S''_{j:h+1} = TR(S''_{j:h}, E''_{j:h}, M'')$

Proof sketch. Follows from the assumptions, and definitions 3-5. \square

The previous lemma's assumptions can always be met starting from any interpretation M . Firstly, since in the worst case, from any interpretation we can obtain a successor starting at the institution's initial state - so both successors agree at least on the initial state. Secondly, by making the non-deterministic choice in the event generation operation to select the same rule modifications for both the successor and the successor to the successor (in the worst case, no rule modifications). We can continue to incrementally produce successive interpretations that monotonically increase the time point they agreed upon. Note that, this may mean backtracking by changing preceding interpretations (e.g. selecting no rule modifications).

Lemma 3. *There exists a fixed point for the interpretation successor operation denoted $SUCC^\omega(M, et)$ for any M and et .*

Proof sketch. A proof can be obtained by structural induction, applying Lemma 2, and ensuring each successive interpretation agrees with the preceding interpretation on rule modifications (potentially removing modifications in previous interpretations). \square

In fact, there can be multiple fixed points, as exemplified:

Example 4.1. An institution \mathcal{I} contains a legislative rule with the id $leg0 \in \mathbb{ID}$ stating that an agent, Ada, voting to activate a rule ($vote_a(act, id, t) \in \mathcal{E}_{obs}$) counts-as activating the rule:

$\mathcal{G}(\top, \{vote_a(act, id, t)\}) \ni \text{mod}(act, id, t)$. In the initial state the legislative rule is active $\Delta = \{active(leg0)\}$. In an observable event trace $et = \langle O_0 \rangle$ Ada votes to activate another rule with the id $leg1 \in \mathbb{ID}$ in the initial state $O_0 = \{vote_a(act, leg1, 0)\}$.

From an initial empty interpretation M we have the following successors and interpretations for example 4.1 (differences are in **bold**):

$$\begin{aligned} M^2 &= SUCC(M, et) = SUCC^\omega(M, et) \text{ s.t. } V^2 = \langle V_0^2 \rangle, R^2(0) = 0, R^2(1) = 0, \\ S_{0:0}^2 &= \{active(leg0)\}, S_{0:1}^2 = \{active(leg0)\}, E_{0:0}^2 = \{vote_a(act, leg1, 0)\} \\ M^1 &= SUCC(M, et) = SUCC^\omega(M, et) \text{ s.t. } V^1 = \langle V_0^1, V_1^1 \rangle, R^1(0) = 1, R^1(1) = 1, \\ S_{0:0}^1 &= \{active(leg0)\}, S_{0:1}^1 = \{active(leg0)\}, \\ E_{0:0}^1 &= \{vote_a(act, leg1, 0), \text{mod}(act, leg1, 0)\} \\ S_{1:0}^1 &= \{active(leg0)\}, S_{1:1}^1 = \{active(leg0)\}, E_{1:0}^1 = \{vote_a(act, leg1, 0)\} \\ M^0 &= SUCC(M, et) = SUCC^\omega(M, et) \text{ s.t. } V^0 = \langle V_0^0, V_1^0 \rangle, R^0(0) = 1, R^0(1) = 1, \\ S_{0:0}^0 &= \{active(leg0)\}, S_{0:1}^0 = \{active(leg0)\}, \\ E_{0:0}^0 &= \{vote_a(act, leg1, 0), \text{mod}(act, leg1, 0)\} \\ S_{1:0}^0 &= \{active(leg0)\}, S_{1:1}^0 = \{active(leg0), active(leg1)\}, \\ E_{1:0}^0 &= \{vote_a(act, leg1, 0), \text{mod}(act, leg1, 0)\} \end{aligned}$$

Each fixed point has different rule modifications. M^2 does not add the rule $leg1$. M^1 contains an attempt to add the rule in the version zero but not in version one. Finally, M^0 adds the rule in the version zero and version one, version one being the current version when the rule is added meaning the rule addition is successful. In fact, the following prioritisation holds $M^0 < M^1 < M^2$ meaning that M^0 maximises successful rule modifications.

Models are interpretations which maximise successful rule modifications. Thus we characterise models by combining the successor interpretation fixed point and interpretation prioritisation. Given an empty interpretation we find a fixed point successor interpretation for a given event set trace (8.1). The fixed point is a model if there is no greater prioritised successor fixed point interpretation (8.2).

Definition 8. Models *Let $M = \langle R, V \rangle$ be an empty interpretation such that $V = \langle V_0 \rangle$, $V_0 = \langle E_0, S_0 \rangle$, $E_0 = \langle \rangle$ and $S_0 = \langle \rangle$. The interpretation $M' = \langle R', V' \rangle$ is a model for \mathcal{I} w.r.t. an observable event set trace $et = \langle O_0, \dots, O_k \rangle$ iff:*

$$M' = SUCC^\omega(M, et) \quad \text{and} \quad (8.1)$$

$$\text{There does not exist an } M'' < M' \text{ meeting 8.1.} \quad (8.2)$$

From lemma 3 and definition 8 we have the following property.

Lemma 4. *There exists at least one model for any institution \mathcal{I} w.r.t. an observable event set trace et .*

These semantics operationalise answering ‘‘when does a rule change count-as a legal rule change?’’. Generally, a rule change counts-as a legal rule change *if and only if* a rule ascribes the change in a context that is consistent with the modification. Models always contain ‘legal’ rule modifications, defined as fixed point interpretations which maximise rule modifications. So, ‘legal’ rule-changes occur in at least one model whilst illegal rule changes do not occur at all (the non-deterministic choice for a rule modification to occur in 4.4) and the institution continues to operate ‘as usual’, meeting our desiderata.

5 Case Studies

Now we apply the framework to concrete case studies. For brevity we use variables to denote: all rule identifiers ($id \in \mathbb{ID}$), all rule change operations ($op \in \{act, deact\}$), and all time instants ($t \in \mathbb{N}$). The first case concerns a simple rule change procedure.

Case 5.1. An institution \mathcal{I}^{sgov} describes a simple government comprising two agents, Ada and Bertrand. Both Ada and Bertrand voting to activate or deactivate a rule in the context that neither are criminals ($crim(ada), crim(ber) \in \mathcal{F}_{dom}^{sgov}$) counts-as activating/deactivating the rule. The rule modifying counts-as rules are identified with $leg0 \in \mathbb{ID}$ and formalised as $\mathcal{G}^{sgov}(\neg crim(ada) \wedge \neg crim(ber), \{vote_a(op, id, t), vote_b(op, id, t)\}) \ni mod(act, id, t)$. At time point one Ada and Bertrand vote to add a rule with id $crim0$, $O_1 = \{vote_a(act, crim0, 1), vote_b(act, crim0, 1)\}$. The rule identified as $crim0$ states that if Ada or Bertrand are found guilty of a crime ($g(ada), g(ber) \in \mathcal{E}_{obs}^{sgov}$) then they become criminals, formally $\mathcal{C}^\dagger(\top, g(ada)) \ni crim(ada)$ and $\mathcal{C}^{sgov\dagger}(\top, g(ber)) \ni crim(ber)$. Next, Bertrand is found guilty of a crime $O_2 = \{g(ber)\}$. Finally, Bertrand and Ada vote to deactivate the criminalising rule, $O_3 = \{vote_a(act, crim0, 3), vote_b(act, crim0, 3)\}$.

For clarity, models are represented graphically. The model for case 5.1 is shown in Figure 1. Lines represent when domain and active rule fluents hold. We distinguish between whether a fluent holds in a state $S_{v:t}$: 1. retroactively in the version's past and not in the previous version, $---$ (i.e. $R(t) < v$ and $\langle M, S_{v-1:t} \rangle \not\models f$), 2. when the version is the current version, $---$ (i.e. $R(t) = v$), and 3. when the version is obsolete, $.....$ (i.e. $R(t) > v$). Time instants are marked if they have successful or non-successful rule modification events in versions where modifications can have an effect (i.e. non-obsolete versions): 1. \blacktriangleright denoting that all the rule modification events occurring in the previous version occur again (i.e. $E_{v:t} \cap \mathcal{E}_{mod} = E_{v-1:t} \cap \mathcal{E}_{mod}$). Meaning, the conditions (contexts) for the rule modifying events to be ascribed are consistent with the version and therefore with applying the rule modifications (the non-deterministic choice to include a rule modification in $E_{v:t}$ according to 4.4 is always made) 2. \blacktriangleleft denoting that at least one rule modification event which occurred in the previous version does not occur again (i.e. $E_{v:t} \cap \mathcal{E}_{mod} \neq E_{v-1:t} \cap \mathcal{E}_{mod}$). Meaning, the conditions (contexts) for rule modifying events to be ascribed are inconsistent with the version they occur in and therefore with applying the rule modifications (a non-deterministic choice according to 4.4 to not include a rule modification is made when building $E_{v:t}$).

Figure 1 shows case 5.1's model. Throughout version zero the legislative rule ($leg0$) is active, stating Ada and Bertrand voting to add a rule counts-as adding a rule. When at time instant one Ada and Bertrand vote to add a new rule ($crim0$), stating people found guilty become criminals, the model succeeds to version one where the new rule is successfully added. At time instant three Bertrand becomes a criminal. When they vote again to modify a rule it is unsuccessful, since rule change is conditional on neither being criminals. Adding a criminalising rule altered the built social reality in version one's future, changing what could be ascribed as a legal rule modification.

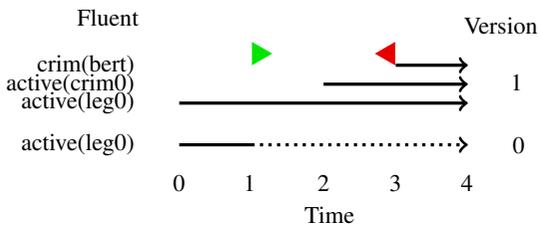


Figure 1. Model for case 5.1 with two institution versions.

The next case presents an institution \mathcal{I}^{uk} representing the UK's legislation rules. The cases are based on past changes to a court decision on UK tax laws (Pad), and past changes to tax laws (Fin [2008]). The UK government can unconditionally enact any rules

effective at any time. Observable events where the government activates/deactivates a rule ($gmod(op, id, t)$) count-as modifying the rule ($mod(op, id, t)$). Legislative rules identified as $leg0 \in \mathbb{ID}$ cause rule activations $\mathcal{G}^{uk}(\top, \{gmod(act, id, t)\}) \ni mod(act, id, t)$ and legislative rules identified as $leg1 \in \mathbb{ID}$ cause rule deactivations $\mathcal{G}^{uk}(\top, \{gmod(deact, id, t)\}) \ni mod(deact, id, t)$. A model $M^{uk} = \langle R^{uk}, V^{uk} \rangle$ is produced for an observable event trace $et = \langle O_0, O_1, O_2, O_3, O_4 \rangle$ for \mathcal{I}^{uk} . The model comprises four versions $V^{uk} = \langle V_0^{uk}, V_1^{uk}, V_2^{uk}, V_3^{uk} \rangle$. We begin the case:

Case 5.2. A rule states that any UK resident (e.g. person a resides in the UK $\neg r(a, uk)$) in a business partnership in the UK ($p(a, uk)$) or elsewhere such as Jersey ($p(a, jers)$) in the first tax year month is obliged to pay tax (obl). We have for all locations $L \in \{uk, jers\}$ a tax rule $\mathcal{C}^{uk\dagger}(r(a, uk) \wedge p(a, L), mon1) \ni obl$ identified as $tax0 \in \mathbb{ID}$. Initially the legislative rules $leg0$ and $leg1$, and the tax rule $tax0$ are active ($\Delta^{uk} = \{active(leg0), active(leg1), active(tax0)\}$). At time point one it is the first tax year month ($O_1 = \{mon1\}$). Following a court challenge (Pad) the government retroactively replaces the tax rule with id $tax0$ with a new rule with id $tax1$ ($O_2 = \{gmod(deact, tax0, 0), gmod(act, tax1, 0)\}$). The new rule, $tax1$, states that only people in a UK business partnership are obliged to pay tax $\neg \mathcal{C}^{uk\dagger}(r(a, uk) \wedge p(a, uk), mon1) \ni obl$.

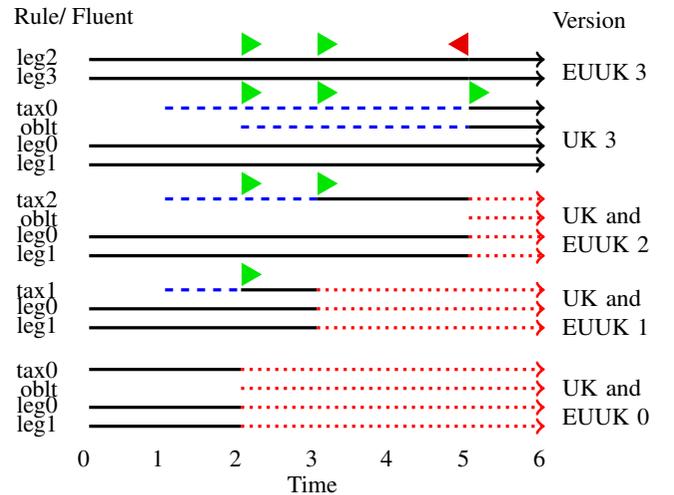


Figure 2. Model for case 5.2 with four institution versions for the institution \mathcal{I}^{uk} (denoted UK) and a model for case 5.3 with four versions for the institution \mathcal{I}^{euk} (denote EUUK). Institutions \mathcal{I}^{uk} and \mathcal{I}^{euk} have identical versions 0 to 2. Not shown, in all states person 'a' is in a Jersey-based business partnership ($p(a, jersey)$) and is a UK resident ($r(a, uk)$).

In Figure 2 version zero (V_0^{uk}) obliges person 'a' to pay tax in state two. In state two ($S_{0:2}^{uk}$) the current institution version changes when the rule obliging UK residents to pay tax ($tax0$) is replaced with the rule obliging UK business partners to pay tax ($tax1$) ($act(t1, 0), deact(t0, 0) \in E_{0:2}^{uk}$) to version one (V_1^{uk} s.t. $R^{uk}(2) = 1$). Due to this change, the version one (V_1^{uk}), does not oblige tax to be paid in its third state ($S_{1:2}^{uk}$) since person a resides in the UK but is in a Jersey-based business partnership.

Case 5.2 (Continued). The government partially reverses the tax change at time point three. This is by retroactively replacing the rule obliging people in a UK business partnership to pay tax ($tax1$) with new rule identified as $tax2$ ($O_3 = \{gmod(deact, tax1, 0), gmod(act, tax2, 0)\}$). The new rule obliges

UK residents in a business partnership to pay tax if it does not criminalise them retroactively (i.e. in a retroactive state an obligation to pay tax is initiated conditional on the obligation holding in the next state of the previous version). For all locations $L \in \{uk, jersey\}$ the rule is $C^{uk\uparrow}((r(a, uk) \wedge p(a, L)) \rightarrow (P) \rightarrow PrV(NS(oblt))), mon1) \ni oblt$. Next, it is the first tax year month again ($O_4 = \{mon1\}$).

In Figure 2 version two (V_2^{uk}), like version zero, does not oblige ‘a’ to pay tax in the past. But, it does oblige them to pay tax after the second time the first tax year month occurs ($mon1 \in E_{2;4}^{uk}$).

Case 5.2 (Continued). The UK government decides to reverse the previous judgements going back to the original rule set ($O_5 = \{gmod(deact, tax2, 0), gmod(act, tax0, 0)\}$).

In Figure 2, version three (V_3^{uk}) reverts to the original legislation. Thus we have the same situation as if the legislation in version zero had not been modified. That is, an obligation to pay tax after the first occurrence of the first tax year month ($mon1 \in E_{3;1}^{uk}$).

The next case is a variation on the previous describing an institution \mathcal{I}^{euk} , incorporating EU human rights law.

Case 5.3. The European Convention on Human Rights [Council of Europe, 1953, Art. 7] (ECHR) blocks retroactive legislative modifications that *criminalise* formerly innocent people. The institution \mathcal{I}^{euk} contains the same rules as \mathcal{I}^{uk} with the same identifiers minus the legislative rules $leg0$ and $leg1$. Instead, legislative rules state that observable rule modifications *count-as* rule modifications *conditional* on the changes not retroactively criminalising people. In all states where rules are being applied retroactively, if there is not an obligation to pay tax in the previous version then there must not be an obligation to pay tax in the current version. We have rules with the identifier $l2$: $\mathcal{G}^{euk}(PaS(P \rightarrow PrV(\neg oblt) \rightarrow \neg oblt)), \{gmod(act, id, t)\} \ni act(id, t)$, and rules with the identifier $l3$: $\mathcal{G}^{euk}(PaS(P \rightarrow PrV(\neg oblt) \rightarrow \neg oblt)), \{gmod(deact, id, t)\} \ni deact(id, t)$. Initially, person ‘a’ is in a Jersey based business partnership ($p(a, jersey)$) and is a UK resident ($r(a, uk)$), and the first tax rule and the legislative rules conditional on being non retroactively criminalising are active such that $\Delta^{euk} = \{p(a, uk), r(a, uk), tax0, l2, l3\}$. The same events occur as in case 5.2, $et = \{\emptyset, \{mon1\}, \{gmod(deact, t0, 0), gmod(act, t1, 0)\}, \{gmod(deact, t1, 0), gmod(act, t2, 0)\}, \{mon1\}, \{gmod(deact, t2, 0), gmod(act, t0, 0)\}$.

Figure 2 shows a model M^{euk} for \mathcal{I}^{euk} . The first three versions are identical to our previous case 5.2 (where the UK’s legislature was not constrained by EU rules blocking retroactively criminalising modifications), since the first two rule modifications do not criminalise people retroactively. Unlike in our previous case 5.2, the version two contains no tax rules. The reason being that tax rule two - “obliging uk residents in a business partnership to pay tax but on the condition that if it is retroactive then those people were obliged to pay tax in the previous version”, is deactivated since its deactivation does not criminalise retroactively. On the other hand, tax rule zero - “any UK resident in a business partnership in the first tax year month is obliged to pay tax” ($C^{uk\uparrow}((r(a, uk) \wedge p(a, L), mon1) \ni oblt)$) is not reactivated, even though it was reactivated in our previous case 5.2. Its reactivation would retroactively criminalise people if activated in version three, meaning its activation does not occur since legislative rule - $l2$: $\mathcal{G}^{euk}(PaS(P \rightarrow PrV(\neg oblt) \rightarrow \neg oblt)), \{gmod(act, id, t)\} \ni act(id, t)$ - has a condition that is not met.

The next cases look at modifying legislative rules themselves.

Case 5.4. An institution \mathcal{I}^p describes a parliament that can retroactively modify rules through a majority vote $pvote(act, id, t) \in \mathcal{E}_{obs}$.

The legislative rules are identified the id $parl0 \in \mathbb{ID}$ for activating rules $\mathcal{G}^p(\top, \{pvote(act, id, t)\} \ni mod(act, id, t))$ and with the id $parl1 \in \mathbb{ID}$ for deactivating rules $\mathcal{G}^p(\top, \{pvote(deact, id, t)\} \ni mod(deact, id, t))$. In the initial state all rules are active such that $active(id) \in \Delta$. In an observable event set trace $tr = \langle O_0, O_1 \rangle$ at time point one the parliament votes to retroactively remove the rule which ascribes retroactive modifications ($O_1 = \{pvote(deact, parl1, 0)\}$).

Depicted in Figure 3 a single model $M^p = \langle R^p, V^p \rangle$ comprises two institution versions $V^p = \langle V_0^p, V_1^p \rangle$. An event occurs in version zero at time instant one, where the parliament votes to retroactively modify a rule and the corresponding rule modification event occurs ($E_{0;1}^p = \{pvote(deact, parl1, 0), mod(deact, parl1, 0)\}$). Consequently the institution transitions to version one ($R^p(1) = 1$). Importantly, in version one, the same rule modifying event *does not occur*. The reason being, if the modification event did occur then the rule $parl1$ ascribing the modification event - $\mathcal{G}^p(\top, \{pvote(deact, id, t)\} \ni mod(deact, id, t))$ - would be inactive in version one state one $S_{1;1}^p$, and the deactivation could not occur in the first place (contradiction). This exemplifies how the formalism always guarantees a model, paradoxical rule modifications do not occur if they make the rule modifying event impossible in the first place.

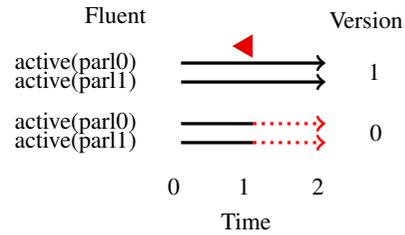


Figure 3. Model for case 5.4

The next case extends the previous case 5.4:

Case 5.5. This case describes an institution \mathcal{I}^{mp} where a monarch and a parliament can retroactively modify rules, including all the rules from the previous case’s institution \mathcal{I}^p . Additionally, a rule identified as $fence0 \in \mathbb{ID}$ states that if a fence is built $fb \in \mathcal{E}_{obs}^{mp}$ it is obliged the fence is painted white $oblpf \in \mathcal{F}_{inst}^{mp} - C^{mp\uparrow}(\top, .fb) \ni oblpf$. A rule identified as $mon0$ states the monarch issuing a rule change decree $mdecree(act, id, t) \in \mathcal{E}_{obs}^{mp}$ to activate a rule counts-as activating the rule - $\mathcal{G}^{mp}(\top, \{mdecree(act, id, t)\} \ni mod(act, id, t))$. A rule identified as $mon1$ state the monarch issuing a decree to deactivate a rule counts-as deactivating the rule $\mathcal{G}^{mp}(\top, \{mdecree(deact, id, t)\} \ni mod(deact, id, t))$. All legislative rules are initially active, but the fence painting rule is not (s.t. $active(fence0) \notin \Delta^{mp}$). At time point one the parliament votes for the fence-painting obligation rule to be activated, ($O_1 = \{pvote(act, fence0, 1)\}$), a fence is built ($O_2 = \{fb\}$), the monarch issues by decree the fence-building rule to be retroactively deactivated at the time it was activated, cancelling its activation ($O_3 = \{mdecree(deact, fence0, 1)\}$). Finally, the parliament votes to retroactively disenable the monarch from deactivating rules ($O_4 = \{pvote(deact, mon1, 0)\}$).

Depicted in Figure 4 the model $M^{mp} = \langle R^{mp}, V^{mp} \rangle$ comprises four versions $V^{mp} = \langle V_0^{mp}, V_1^{mp}, V_2^{mp}, V_3^{mp} \rangle$. At version zero time instant zero the parliament votes to add the rule obliging built fences to be painted white, causing a rule modification event ($E_{0;1}^{mp} = \{pvote(act, fence0, 1), mod(act, fence0, 1)\}$) and the institution to transition to the version one ($R^{mp}(1) = 1$) where the same modification occurs ($E_{1;1}^{mp} = \{pvote(act, fence0, 1), mod(act, fence0, 1)\}$).

In the version one time instant two building a fence ($fb \in E_{1;2}^{mp}$) causes an obligation to paint the fence $oblpf \in S_{1;3}^{mp}$. At time instant three the monarch retroactively deactivates the fence painting rule ($mdecree(deact, fence0, 1) \in E_{1;3}^{mp}$) causing the institution to transition to the version two ($R^{mp}(3) = 2$) where the modification takes effect ($mdecree(deact, fence0, 1) \in E^{mp}$). Consequently, the fence painting obligation rule is deactivated and its effects (an obligation) no longer hold. When the parliament retroactively removes the ability for the monarch to deactivate rules the institution transitions to the final version three ($R^{mp}(4) = 3$) where the parliament’s retroactive rule removal takes effect ($pvote(deact, mon1, 0), mod(deact, mon1, 0) \in E_{3;4}^{mp}$) causing the monarch’s modifications to be unravelled (note that at the final version’s third time instant the monarch’s rule modification is unsuccessful even though it was successful in the previous version). Consequently, the fence painting obligation rule and its effects (an obligation) is reinstated by retroactively removing the ability to deactivate the fence painting rule.

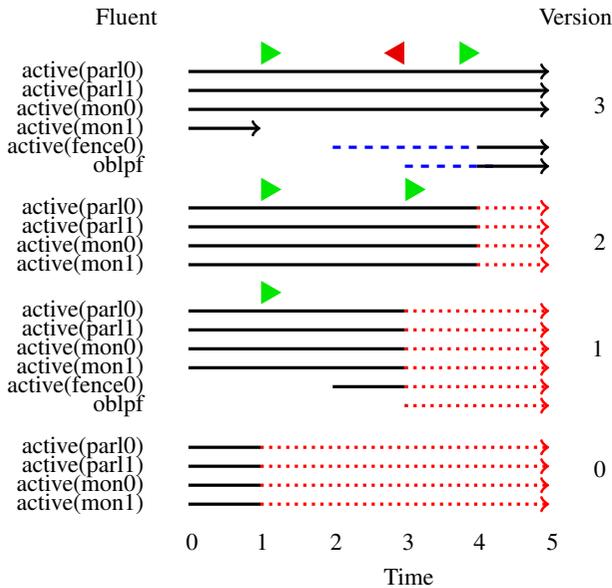


Figure 4. Model for case 5.5 with four institution versions.

6 Related Work

In (Governatori et al. [2005]; Governatori and Rotolo [2010]) a de-feasible logic is proposed for temporal rule modification operations. Operations include, in (Governatori and Rotolo [2010]), complete rule removal (annulment) and removing immediate rule effects (abrogation). Meta-rules are used to introduce rule changes, which bear similarity to our rule-modifying counts-as rules. However, the meta-rules are only conditional on a single state. For comparison, we formalise richer conditions on past versions, states and hypothetical rule changes required to capture a number of important examples we address (such as rule change being non-retroactively criminalising). The focus of these papers is on rule change operations found in the legal domain, rather than the relation between ascribing a social reality and rule modifications with counts-as rules. In (López and Luck [2003]; López et al. [2006]) electronic institutions are specified in the Z specification language where legislation norms restrict legislative actions. The conditions for legislation norms are less expressive than our proposal and the authors do not consider the interdependency between changing rules in the past/present/future and the built social

reality. On the other hand, in Boella and van der Torre [2004] rule modifications ascribed by counts-as rules are formalised where there is such a potential interdependency, but the setting is non-temporal.

Our proposal is thematically related to work in the institutional/normative reasoning sphere, in particular work on: 1. constitutive rule classes (Grossi et al. [2005, 2006, 2008]; Grossi [2008, 2011]), 2. norm change postulates (Boella and van der Torre [2004]), 3. detecting and/or resolving norm inconsistencies (Jiang et al. [2015]; Jiang [2015]; Kollingbaum et al. [2007]; Corapi et al. [2011]; Li [2014]; Vasconcelos et al. [2008]), rectifying *non-compliant* institutions (King et al. [2015a,b]) and 4. temporal norm updates (Alechina et al. [2013]; Knobbout et al. [2014]). However, these papers do not look at rule change legality ascribed by constitutive rules over time.

7 Conclusions

This paper answers “when do rule changes count-as legal rule changes?” with a novel formalism. Our framework formalises reasoning about institutional rule change over time ascribed by counts-as rules. A novel semantics defines how an institution evolves from one social reality to the next and from one version of rules to another. Under the proposed semantics counts-as rules define the past/present/future social reality. In turn, rule modifications change counts-as rules in the past/present/future and therefore the constructed social reality.

A rule change counts-as a legal rule change if and only if - 1. the rule change is ascribed by counts-as rules, conditional on a context which can include the potential changes to the social reality the rule modification *would* make. 2. the rule change results are consistent with the context the rule change is conditional on. In particular, taking into account the rule modification’s past/present/future effect on counts-as rules, any changes to previous rule modifications, and the rule modification being ‘undone’ by future modifications. Legal rule changes always occur. Meeting our desiderata, illegal rule changes do not occur and the institution continues to operate ‘as usual’.

There are many avenues for future work. First, extending the framework to deal with further cases. In particular, rules which explicitly block retroactive modifications altogether (e.g. [USC, Art. 1 Sec. 9 Cl. 3] “No Bill of Attainder or ex post facto Law shall be passed”). Preventing retroactive modifications can be expressed as a lack of a rule ascribing past rule changes, but not rules which *block* past rule changes. Second, the fixed-point institutional model characterisation can be implemented in any adequately expressive language. One possibility is Answer-Set Programming (Gebser et al. [2011b]; Gelfond and Lifschitz [1988]) as used in the InstAL framework (Cliffe et al. [2007]). Third, agent planning for rule changes, such as by building on existing agent-planning in Answer-Set Programming (Gebser et al. [2011a]; Lifschitz [1999]). Fourth, looking at agents bringing about legal rules, known as social commitments (e.g. promises, contracts), through locutions (Austin [1975]). In particular, looking at how social commitments can be created with the special role of ascribing changes to social commitments, such as by building on Event Calculus based frameworks (Chesani et al. [2012]; Günay and Yolum [2012]).

ACKNOWLEDGEMENTS

This research was supported by the SHINE² project of TU Delft. The authors would like to thank the anonymous referees for their input.

² <http://shine.tudelft.nl>

REFERENCES

- Natasha Alechina, Mehdi Dastani, and Brian Logan. Reasoning about normative update. *IJCAI International Joint Conference on Artificial Intelligence*, pages 20–26, 2013.
- Giulia Andrighetto, Guido Governatori, Pablo Noriega, and Leendert van der Torre. *Normative Multi-Agent Systems*, volume 4. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2013.
- John Langshaw Austin. *How To Do Things With Words*. Oxford university press, 1975.
- Carlo Biagioli. Towards a legal rules functional micro-ontology. In *1st LegOnt Workshop on Legal Ontologies.*, 1997.
- Guido Boella and Leendert van der Torre. Regulative and Constitutive Norms in Normative Multiagent Systems. *KR'04*, pages 255–265, 2004.
- Federico Chesani, Paola Mello, Marco Montali, and Paolo Torroni. Representing and monitoring social commitments using the event calculus. *Autonomous Agents and Multi-Agent Systems*, 27(1):85–130, jun 2012.
- Owen Cliffe, Marina De Vos, and Julian Padget. Answer set programming for representing and reasoning about virtual institutions. *Computational Logic in Multi-Agent Systems*, pages 60–79, 2007.
- Owen Cliffe. *Specifying and Analysing Institutions in Multi-Agent Systems Using Answer Set Programming*. PhD thesis, University of Bath, 2007.
- Domenico Corapi, Alessandra Russo, Marina De Vos, Julian Padget, and Ken Satoh. Normative design using inductive learning. *TPLP*, 4-5:783–799, 2011.
- Council of Europe. European Convention on Human Rights, 1953.
- Finance Act 2008, Chapter 9 (United Kingdom), 2008.
- Martin Gebser, Roland Kaminski, Murat Knecht, and Torsten Schaub. Plasp: A prototype for PDDL-based planning in ASP. *Logic Programming and Nonmonotonic Reasoning*, pages 358–363, 2011.
- Martin Gebser, Benjamin Kaufmann, and Roland Kaminski. Potassco: The Potsdam answer set solving collection. *AI Communications*, 24(2):107 – 124, 2011.
- Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In *ICLP/SLP*, pages 1070 – 1080, 1988.
- Guido Governatori and Antonino Rotolo. Changing Legal Systems: Legal Abrogations and Annulments in Defeasible Logic. *Logic Journal of IGPL*, 18(1):157–194, 2010.
- Guido Governatori, Monica Palmirani, Regis Riveret, Antonino Rotolo, and Giovanni Sartor. Norm modifications in defeasible logic. In *In Proceedings of JURIX'05*, pages 13–22, 2005.
- Davide Grossi, John-Jules Meyer, and Frank Dignum. Modal logic investigations in the semantics of counts-as. *ICAAIL '05: Proceedings of the 10th international conference on Artificial intelligence and law*, pages 1–19, 2005.
- Davide Grossi, John Jules Ch Meyer, and Frank Dignum. Classificatory aspects of counts-as: An analysis in modal logic. *Journal of Logic and Computation*, 16(5):613–643, 2006.
- Davide Grossi, J. J Ch Meyer, and Frank Dignum. The many faces of counts-as: A formal analysis of constitutive rules. *Journal of Applied Logic*, 6(2):192–217, 2008.
- Davide Grossi. Pushing Anderson's Envelope: The Modal Logic of Ascription. In *9th International Conference on Deontic Logic in Computer Science (DEON 2008)*, pages 263–277, 2008.
- Davide Grossi. Norms as ascriptions of violations: An analysis in modal logic. *Journal of Applied Logic*, 9(2):95–112, 2011.
- A Günay and P Yolum. Detecting conflicts in commitments. *Declarative Agent Languages and Technologies IX*, pages 51–66, 2012.
- Jie Jiang, Jeremy Pitt, and Ada Diaconescu. Rule Conflicts in Holonic Institutions. *2015 IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops*, pages 49–54, 2015.
- Jie Jiang. *Organizational Compliance: An Agent-based Model for Designing and Evaluating Organizational Interactions*. PhD thesis, TU Delft, Delft University of Technology, 2015.
- Thomas C King, Tingting Li, Marina De Vos, Virginia Dignum, Catholijn M Jonker, Julian Padget, and M Birna Van Riemsdijk. A Framework for Institutions Governing Institutions. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)*, pages 473–481, 2015.
- Thomas C King, Tingting Li, Marina De Vos, Catholijn M Jonker, Julian Padget, and M Birna Van Riemsdijk. Revising Institutions Governed by Institutions for Compliant Regulations. In *Coordination, Organizations, Institutions and Norms, LNCS 9628*. 2015.
- Max Knobbout, Mehdi Dastani, and John-jules Ch Meyer. Reasoning about Dynamic Normative Systems. *Logics in Artificial Intelligence, LECTURED NOTES IN COMPUTER SCIENCE*, 8761:628–636, 2014.
- Martin J. Kollingbaum, Wamberto W. Vasconcelos, Andres García-Camino, and Timothy J. Norman. Managing conflict resolution in norm-regulated environments. *ESAW 2007*, 2007.
- Tingting Li. *Normative Conflict Detection and Resolution in Cooperating Institutions*. PhD thesis, University of Bath, 2014.
- Vladimir Lifschitz. Answer set planning. *16th International Conference on Logic Programming*, (2):23–37, 1999.
- Fabiola López Y López and Michael Luck. Modelling Norms for Autonomous Agents. In *Proceedings of The Fourth Mexican Conference on Computer Science*, pages 238–245. IEEE Computer Society, 2003.
- Fabiola López y López, Michael Luck, and Mark D'Inverno. A normative framework for agent-based systems. *Computational and Mathematical Organization Theory*, 12(2-3):227–250, oct 2006.
- Javier Morales, M Lopez-Sanchez, Juan A. Rodriguez-Aguilar, Michael Wooldridge, and Wamberto Vasconcelos. Minimality and Simplicity in the On-line Automated Synthesis of Normative Systems. In *Proceedings of the 13th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2014)*, pages 109–116, 2014.
- Javier Morales, Maite López-Sánchez, Juan Antonio Rodríguez-Aguilar, Michael Wooldridge, and Wamberto Vasconcelos. Synthesising Liberal Normative Systems. In *Proceedings of the 15th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2015)*, pages 433–441, 2015.
- Padmore v IRC (1987) STC 36 affirmed by the Court of Appeal (1989) STC 493.
- John R. Searle. *Speech acts: An essay in the philosophy of language*. Cambridge university press, 1969.
- John R. Searle. What is an institution? *Journal of Institutional Economics*, 1:1–22, 2005.
- Peter Suber. *The Paradox of Self-Amendment: A Study of Law, Logic, Omnipotence, and Change*. Peter Lang International Academic Publishers, 1990.
- The United States Constitution.
- 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, nov 2008.

Constant Time EXpected Similarity Estimation for Large-Scale Anomaly Detection

Markus Schneider¹ and Wolfgang Ertel and Günther Palm

Abstract. A new algorithm named *EXpected Similarity Estimation* (EXPOSE) was recently proposed to solve the problem of *large-scale anomaly detection*. It is a non-parametric and distribution free kernel method based on the Hilbert space embedding of probability measures. Given a dataset of n samples, EXPOSE takes $\mathcal{O}(n)$ time to build a model and $\mathcal{O}(1)$ time per prediction.

In this work we describe and analyze a simple and effective stochastic optimization algorithm which allows us to drastically reduce the learning time of EXPOSE from previous *linear* to *constant*. It is crucial that this approach allows us to determine the number of iterations based on a desired accuracy, *independent of the dataset size* n . We will show that the proposed stochastic gradient descent algorithm works in general possible infinite-dimensional Hilbert spaces, is easy to implement and requires no additional step-size parameters.

1 INTRODUCTION

Anomaly detection is used in a variety of fields [13, 14] such as network intrusion detection [15], credit card fraud detection [3], medical diagnosis [30, 21] and failure detection in industrial environments [34]. Commonly, “anomaly detection refers to the problem of finding patterns in data that do not conform to expected behavior. These non-conforming patterns are often referred to as *anomalies*” [8]. The challenge in detecting anomalies is that anomalous instances or events are by definition *rare*. “Virtually all [anomaly] detection algorithms create a model of the normal patterns in the data, and then compute an [anomaly] score of a given data point on the basis of the deviations from these patterns” [2].

The *EXpected Similarity Estimation* (EXPOSE) algorithm was proposed to solve the problem of large-scale anomaly detection where the dataset sizes are too immense to be processed by standard techniques [26]. As explained later in detail, the EXPOSE anomaly detection classifier

$$\eta(y) = \langle \phi(y), \mu[\mathbb{P}] \rangle$$

calculates a score (the likelihood of y belonging to the class of normal data) using the inner product between a feature map ϕ and the kernel mean map $\mu[\mathbb{P}]$ of the distribution \mathbb{P} (Fig. 1). Given a dataset of size n , the authors provide a methodology to train this classifier in linear time with $\mathcal{O}(n)$ computational complexity. The question arises if it is possible to improve the linear training time and create an algorithm which is *completely independent of the dataset size*.

The answer to this question is positive if a high accuracy sample estimate of $\mu[\mathbb{P}]$ does not improve the anomaly detection performance.

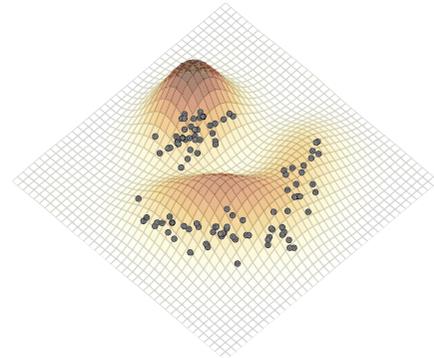


Figure 1. Sketch of the EXPOSE scores $\eta(y)$ in \mathbb{R}^2 , some samples (black dots) from \mathbb{P} .

As Bousquet and Bottou [6] observed, for most machine learning applications there is no need for a high accuracy estimation of an empirical cost function since it is itself an approximation of the expected costs and therefore contains errors. With this idea in mind we will show that it is possible to achieve any desired accuracy (the maximal deviation from the optimal EXPOSE model) with a *fixed* number of samples and that this number is *independent* of the datasets size n .

1.1 Contributions & Related Work

In the next section we derive a methodology to build an ϵ -accurate model w of $\mu[\mathbb{P}]$ using a fixed random subset of the training data by means of stochastic optimization.

Definition 1. We say an algorithm finds an ϵ -accurate solution w of a real valued objective function f if

$$f(w) \leq \inf f + \epsilon$$

for a given $\epsilon > 0$.

We will show that for the proposed objective function f we can achieve a rate of convergence of $\mathcal{O}(1/t)$ that is

$$f(w_t) - f(\mu[\mathbb{P}]) \leq \mathcal{O}(1/t),$$

where w_t only needs access to $t \in \mathbb{N}$ independent random samples from \mathbb{P} . The key observation is that for any given an $\epsilon > 0$ we can reach $\|w_t - \mu[\mathbb{P}]\| < \epsilon$ in a *fixed* number of iterations *independent* of the dataset size. Moreover, it can be shown that (without further assumptions) the $\mathcal{O}(1/t)$ rate is optimal for stochastic optimization [1].

¹ Institute of Neural Information Processing, University of Ulm, Germany
Institute for Artificial Intelligence, University of Applied Sciences Ravensburg-Weingarten, Germany

Before we start with a detailed problem description we would like to put the choice of a stochastic optimization approach for EXPOSE in the context of other machine learning methods which already use such techniques with great success.

Stochastic optimization and especially stochastic gradient (SG) methods [6, 25], are widely used to train machine learning models on very large-scale datasets, since a single iteration of SG requires only little computational time compared to a full gradient calculation. SG techniques are used for example to train support vector machines [27], logistic regression [5] and lasso models [28]. However, *this is the first time that EXPOSE is considered as an optimization problem* and we will show that the derived algorithm is of general interest for applications of the kernel mean map $\mu[\mathbb{P}]$. More sophisticated optimization techniques such as projected gradient descent [7] or Nesterov’s accelerated gradient descent [17, 18] are also applicable in principle, however a single gradient evaluation has already a linear computational complexity and is therefore slower than the original proposed approach. Other stochastic gradient methods [25] can obtain a better convergence rate for an objective composed of a sum of smooth functions. However, this requires multiple passes over the datasets is therefore of no benefit.

Our main contributions are:

- An ϵ -accurate approximation of EXPOSE model which reduces the training time complexity from *linear* to *constant*.
- A formal definition of the empirical kernel mean map as an infinite-dimensional stochastic optimization problem.
- A detailed theoretical analysis and proofs of ϵ -accurate solutions.
- Experimental evaluation on three large-scale datasets.

2 PROBLEM DESCRIPTION

EXPOSE is a probabilistic anomaly detector which assumes that the *normal* (non-anomalous) instances are distributed according to some probability measure \mathbb{P} . We assume there is a random variable X which maps into a measurable input space $(\mathcal{X}, \mathcal{X})$ and denote realizations of X by $X = x$. Moreover we will operate in a reproducing kernel Hilbert space (RKHS) \mathcal{H} associated with the kernel $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$. The function $\phi: \mathcal{X} \rightarrow \mathcal{H}$ with

$$k(x, y) = \langle \phi(x), \phi(y) \rangle$$

is called *feature map* denoted by $\phi(x) = k(x, \cdot)$. To avoid technical difficulties we assume that the input space \mathcal{X} is separable and the kernel k is continuous on \mathcal{X} such that ϕ is measurable.

Given these preliminaries, EXPOSE calculates a score which can be interpreted as the likelihood of a query point belonging to the distribution of normal data \mathbb{P} . This is done in the following way.

Definition 2 (Expected Similarity Estimation). *The expected similarity of $y \in \mathcal{X}$ with respect to the (Borel probability) distribution \mathbb{P} is defined as*

$$\eta(y) = \mathbb{E}[\phi(y)] = \int_{\mathcal{X}} k(y, x) d\mathbb{P}(x), \quad (1)$$

where $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a continuous reproducing kernel.

Intuitively the query point y is compared to all other points of the distribution \mathbb{P} using the kernel as a similarity measure. We will see that this equation can be rewritten as an inner product between the feature map $\phi(y)$ and the kernel mean map $\mu[\mathbb{P}]$ of \mathbb{P} .

The key idea of the kernel mean map [10, 29, 32, 11] is to embed a probability distribution into a reproducing kernel Hilbert space (RKHS) where it is then in a more accessible form and can be manipulated efficiently.

Definition 3 (Kernel Embedding). *The kernel embedding or kernel mean map of a measure distribution \mathbb{P} is given in terms of the Bochner integral*

$$\mathbb{P} \mapsto \mu[\mathbb{P}] = \int_{\mathcal{X}} k(\cdot, x) d\mathbb{P}(x),$$

where $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is the associated reproducing kernel.

We see that μ maps every distribution to a single point in the RKHS \mathcal{H} . This mapping is injective if k is characteristic [32, 9, 31]. However, we emphasize that the characteristic property is not relevant for the EXPOSE anomaly detection algorithm and we therefore explicitly allow kernel mean embeddings which are not injective. To facilitate the further analysis, we assume that the kernel k is measurable and bounded in expectation² i.e. there is a constant $c > 0$ such that

$$\int_{\mathcal{X}} \sqrt{k(x, x)} d\mathbb{P}(x) \leq c$$

which is sufficient to show that $\mu[\mathbb{P}]$ exists for all Borel probability measures on \mathcal{X} [31]. Equation (1) can now be rewritten in terms of the kernel embedding

$$\begin{aligned} \eta(y) &= \int_{\mathcal{X}} k(y, x) d\mathbb{P}(x) \\ &= \langle \phi(y), \mu[\mathbb{P}] \rangle. \end{aligned}$$

where we interchanged the integral and inner product. This is possible since ϕ is strong integrable and therefore coincide with the weak integral, a property we will need several times.

Lemma 1. *If f is strong (Bochner) integrable then f is weak (Pettis) integrable and the two integrals coincide. [4, Theorem 11.50]*

In anomaly detection, we cannot expect to know the distribution of normal data \mathbb{P} explicitly. However we assume to have access to a sample $(X_1, \dots, X_n) = (x_1, \dots, x_n)$ from \mathbb{P} . The empirical measure after n draws $\mathbb{P}_n = \frac{1}{n} \sum_{i=1}^n \delta_{x_i}$ can now act as a proxy for \mathbb{P} and we can use \mathbb{P}_n to construct an approximation $\mu[\mathbb{P}_n]$ of $\mu[\mathbb{P}]$ as

$$\mu[\mathbb{P}_n] = \frac{1}{n} \sum_{i=1}^n \phi(x_i)$$

which is called *empirical kernel embedding* [29]. With the empirical kernel embedding $\mu[\mathbb{P}_n]$ we can derive an empirical version of EXPOSE substituting $\mu[\mathbb{P}]$ by $\mu[\mathbb{P}_n]$ whenever the distribution \mathbb{P} is not directly accessible and has to be estimated from samples (X_1, \dots, X_n) . Obviously, the empirical kernel embedding has a $\mathcal{O}(n)$ computational complexity and is responsible for the *linear* training time of EXPOSE which we tackle in the remainder of this work.

In the next section, we will look at the EXPOSE classifier from the perspective of a stochastic optimization problem. We will show that this allows us to replace $\mu[\mathbb{P}_n]$ by an ϵ -accurate approximation of $\mu[\mathbb{P}]$ which can be computed in *constant* time. We expect this to be beneficial for all techniques which use the kernel mean map and not just EXPOSE. However the main focus of this work is to reduce the EXPOSE training time complexity.

² Bounded in expectation is weaker than the assumption of k being bounded.

3 STOCHASTIC OPTIMIZATION

Let \mathcal{H} be a separable Hilbert space and H be a closed and convex subset of \mathcal{H} . The classic stochastic approximation algorithm [24] tries to solve

$$\min_{w \in H} \mathbb{E}[f(w, X)] = \min_{w \in H} \int_{\mathcal{X}} f(w, x) d\mathbb{P}(x)$$

where X a random variable taking values in the $(\mathcal{X}, \mathcal{X})$ as defined in the previous section. All expectations are taken with respect to this random variable need to exist. To solve this problem, stochastic approximation techniques make the following two assumptions.

- It is possible to generate independent samples x_1, x_2, \dots from the probability distribution \mathbb{P} .
- Given a point $(w, x) \in \mathcal{H} \times \mathcal{X}$ we assume to have access to a *stochastic gradient* of f at w .

A *stochastic gradient* $\nabla f(w, X)$ is a random variable whose expectation is a gradient of f *i.e.* it has the property that $\nabla f(w) = \mathbb{E}[\nabla f(w, X)]$. The stochastic approximation algorithm then creates the sequence $(w_t), t \in \mathbb{N}$ as

$$w_{t+1} = \Pi_H(w_t - \gamma_t \nabla f(w, X))$$

with the aim to minimize $\mathbb{E}[f(w, X)]$ starting at some $w_1 \in H$. Here (γ_t) is a sequence of positive step sizes and the operator

$$\Pi_H(w) = \arg \min_{v \in H} \|w - v\|$$

is called the *projection* of w onto H . $\Pi_H(w)$ is unique in H and the function $w \mapsto \Pi_H(w)$ is *non-expanding* which means that

$$\|\Pi_H(w) - \Pi_H(v)\| \leq \|w - v\|$$

for all w and v in \mathcal{H} . We will denote the optimal solution to the optimization problem (which may not exist and is not necessary unique) by $\tilde{w} = \min_{w \in H} \mathbb{E}[f(w, X)]$.

Most literature on this topic is concerned with the optimization of an objective function defined on some finite-dimensional Euclidean vector space. For example Nemirovski et al. [16] consider H to be a non-empty closed bounded convex subset of $\mathcal{H} = \mathbb{R}^d$ and show that stochastic approximation can obtain a $\mathcal{O}(1/t)$ convergence rate if the objective function f is differentiable and α -strongly convex on \mathcal{H} .

Definition 4 (α -Strongly Convex Function). *Let H be a convex subset of a normed space. A function $f: H \rightarrow \mathbb{R}$ is called α -strongly convex if for some $\alpha > 0$*

$$f(\lambda u + (1 - \lambda)v) \leq \lambda f(u) + (1 - \lambda)f(v) - \frac{\alpha}{2} \lambda(1 - \lambda) \|u - v\|^2$$

for all $u, v \in \text{dom}(f)$ and $\lambda \in [0, 1]$.

This implies that a function f is α -strongly convex if and only if the function

$$u \mapsto f(u) - \frac{\alpha}{2} \|u\|^2$$

is convex. Nemirovski et al. require in addition, that the stochastic gradients $\nabla f(w, X)$ are bounded in expectation *i.e.* there exists a constant $c > 0$ such that

$$\mathbb{E}[\|\nabla f(w, X)\|^2] \leq c^2$$

for all $w \in H$. Under these conditions, Nemirovski et al. demonstrate that

$$\begin{aligned} \mathbb{E}[f(t) - f(\tilde{w})] &\leq \mathcal{O}(1/t) \quad \text{and} \\ \mathbb{E}[\|w_t - \tilde{w}\|^2] &\leq \mathcal{O}(1/t). \end{aligned}$$

Without further assumptions on the objective function f , these rates are unimprovable [1].

4 INFINITE-DIMENSIONAL STOCHASTIC OPTIMIZATION

This section lays the theoretical foundation to prove the rate of convergence for the EXPOSE optimization problem. We will see that if a minimizer exists and the gradient of the objective function f is bounded we can achieve rates similar to the ones derived by Nemirovski et al. However, since our optimization problem is infinite-dimensional, the main challenge will be to overcome the technical difficulties that arise in such spaces.

In finite-dimensions, closed and bounded sets are *compact*, guaranteed by the Heine–Borel theorem. Compact sets have several desirable properties *i.e.* compactness of a nonempty set L suffice to show that a continuous function $f: L \rightarrow \mathbb{R}$ attains its minimum on L which means there is some $\tilde{w} \in L$ such that $f(\tilde{w}) = \inf_{w \in L} f(w)$. Furthermore, if (w_t) is a *minimizing sequence* for f , that is

$$\lim_{t \rightarrow \infty} f(w_t) = \inf f$$

and if (w_t) converges to \tilde{w} , then $\tilde{w} \in \arg \min f$. This is easy to see from the Bolzano–Weierstrass theorem: Since (w_t) is bounded, $f(w_t)$ is bounded and there exists a subsequence (w_{t_k}) which converges to some \tilde{w} and because L is closed, \tilde{w} is an element of L . Since (w_t) is a minimizing sequence, it follows from the continuity of f that $f(w_{t_k}) \rightarrow f(\tilde{w}) = \inf f$. However, the situation is entirely different in infinite-dimensional spaces like the Hilbert space \mathcal{H} we have to consider in our optimization problem.

In infinite-dimensional Banach spaces, closed balls are not compact. [33]

Moreover, in this situation, closed and bounded sets need not even be *sequentially compact*. That is not every sequence in the set has a converging subsequence that converges to a point in that set. In infinite-dimensional optimization we can try to surrogate Bolzano–Weierstrass theorem by the property that a bounded sequence in a reflexive Banach space always has a *weakly convergent* subsequence [33, Theorem 2.C].

Definition 5 (Weak Convergence). *A sequence (w_t) in a Banach space \mathcal{H} converges weakly to w if*

$$\lim_{t \rightarrow \infty} \langle u^*, w_t \rangle_* = \langle u^*, w \rangle_*$$

for all u^* in the dual of \mathcal{H}^* of \mathcal{H} and is denoted by $w_t \rightharpoonup w$.

4.1 Strongly Convex Convergence Rates

We start to derive convergence rates for general strongly convex functions with Lipschitz-continuous gradients and then apply them to EXPOSE. The derivation follows closely the one for finite-dimensional literature [18, 7, 16] with additional assumptions where necessary. The next lemma plays an important role in the derivation of the central theorems about the convergence rate of strongly convex functions with domain in infinite-dimensional Hilbert spaces (theorems 1 and 2).

Lemma 2. Let \mathcal{H} be an (infinite-dimensional) Hilbert space, $H \subset \mathcal{H}$ and $f: H \rightarrow \mathbb{R}$ be continuous and α -strongly convex. If \tilde{w} is a global minimizer of f , then

$$\langle \nabla f(v), v - \tilde{w} \rangle \geq \alpha \|\tilde{w} - v\|^2$$

for all v in H and any gradient $\nabla f(v)$ of $f(v)$.³

Lemma 2 leads directly to a second important inequality bounding the distance from the minimizer in terms of the gradient.

Lemma 3. Under the prerequisites of Lemma 2

$$\|v - \tilde{w}\|^2 \leq \frac{\|\nabla f(v)\|^2}{\alpha^2}$$

for all $v \in H$.

We are now in the position to state one of the main theorems.

Theorem 1. Let \mathcal{H} be a Hilbert space, H a non-empty closed convex subset of \mathcal{H} and $f: H \rightarrow \mathbb{R}$ be α -strongly convex. Assume the minimizer $\tilde{w} \in H$ exists and $\nabla f(w_t, X)$ is strongly (Bochner) integrable, then the stochastic approximation sequence (w_t)

$$w_{t+1} = \Pi_H(w_t - \gamma_t \nabla f(w_t, X))$$

converges strongly to \tilde{w} with rate

$$\mathbb{E}[\|w_t - \tilde{w}\|^2] \leq \frac{\|\nabla f(w_t)\|^2}{\alpha^2 t}$$

for any $w_t \in H$ and $t \in \mathbb{N}$.

Proof. We will use the facts that Π_H is non-expanding and $\tilde{w} \in H$ by definition.

$$\begin{aligned} \|w_t - \tilde{w}\|^2 &= \|\Pi_H(w_t - \gamma_t \nabla f(w_t, X)) - \tilde{w}\|^2 \\ &= \|\Pi_H(w_t - \gamma_t \nabla f(w_t, X)) - \Pi_H(\tilde{w})\|^2 \\ &\leq \|w_t - \gamma_t \nabla f(w_t, X) - \tilde{w}\|^2 \\ &= \|w_t - \tilde{w}\|^2 + \gamma_t^2 \|\nabla f(w_t, X)\|^2 \\ &\quad - 2\gamma_t \langle w_t - \tilde{w}, \nabla f(w_t, X) \rangle \end{aligned}$$

We will take expectations on both sides. As $\nabla f(w_t, X)$ is assumed to be strongly integrable we can interchange the inner product and expectation operator (lemma 1) and therefore

$$\mathbb{E}[\langle w_t - \tilde{w}, \nabla f(w_t, X) \rangle] = \langle w_t - \tilde{w}, \mathbb{E}[\nabla f(w_t, X)] \rangle.$$

Furthermore, by Jensen's inequality

$$\mathbb{E}[\|\nabla f(w_t, X)\|^2] \leq \|\mathbb{E}[\nabla f(w_t, X)]\|^2$$

and hence applying lemma 2 yields

$$\begin{aligned} \mathbb{E}[\|w_t - \tilde{w}\|^2] &\leq \|w_t - \tilde{w}\|^2 + \gamma_t^2 \|\nabla f(w_t)\|^2 - 2\gamma_t \alpha \|\tilde{w} - v\|^2 \\ &= (1 - 2\gamma_t \alpha) \|w_t - \tilde{w}\|^2 + \gamma_t^2 \|\nabla f(w_t)\|^2 \\ &= (1 - 2t^{-1}) \|w_t - \tilde{w}\|^2 + \frac{\|\nabla f(w_t)\|^2}{\alpha^2 t^2} \end{aligned}$$

where we used $\gamma_t = \frac{1}{\alpha t}$ in the last step. For $t = 1$ we know that

$$\mathbb{E}[\|w_1 - \tilde{w}\|^2] \leq \frac{\|\nabla f(w_1)\|^2}{\alpha^2}$$

by lemma 3. The claim follows by an induction argument. \square

³ A proof can be found in the appendix.

Theorem 1 implies a $\mathcal{O}(1/\sqrt{t})$ rate of convergence for the sequence $(w_t) \rightarrow \tilde{w}$, whenever $\|\nabla f(w_t)\|^2$ is bounded. Furthermore, the step size determined to be $\gamma_t = \frac{1}{\alpha t}$ and there is no need to tune it manually. Using the previous argumentation we can also derive a bound for the objective function f .

Theorem 2. Assume the prerequisites of theorem 1 are fulfilled. Assume additionally that f is Fréchet differentiable with β -Lipschitz continuous stochastic gradient $\nabla f(w) = \mathbb{E}[\nabla f(w, X)]$ and let $\nabla f(w, X)$ be strongly integrable as before, then

$$\mathbb{E}[f(w_t) - f(\tilde{w})] \leq \frac{\beta}{2} \frac{\|\nabla f(w_t)\|^2}{\alpha^2 t}$$

for all $w_t \in H$.

Proof. Since $\nabla f(w)$ is β -Lipschitz continuous it holds⁴ that

$$f(v) - f(w) \leq \langle \nabla f(w), v - w \rangle + \frac{\beta}{2} \|w - v\|^2$$

for all $v, w \in H$. Furthermore, $\langle \nabla f(\tilde{w}), v - \tilde{w} \rangle = 0$ if \tilde{w} is a minimizer of f in H by Fermat's rule. Combined with theorem 1 and lemma 1 this yields

$$\mathbb{E}[f(w_t) - f(\tilde{w})] \leq \frac{\beta}{2} \mathbb{E}[\|w_t - v\|^2] \leq \frac{\beta}{2} \frac{\|\nabla f(w_t)\|^2}{\alpha^2 t}$$

which concludes the proof. \square

It follows that the rate of convergence of the objective function f is $\mathcal{O}(1/t)$.

5 CONVEX EXPOSE

The previous results are independent of EXPOSE and can be used for general infinite-dimensional optimization problems.

In this section we look at the EXPOSE anomaly predictor from the perspective of such a stochastic optimization problem to find an ϵ -accurate constant time approximation of $\mu[\mathbb{P}]$. Hereby we apply the bounds derived in the previous sections and show that there exists an optimal solution to our optimization problem and that this solution is unique. To achieve this goal we need to show that the objective function is strongly convex and the derivative is β -Lipschitz and Bochner integrable.

Obviously, if $\mu[\mathbb{P}] \in H \subset \mathcal{H}$, then the kernel mean map is the solution of the following optimization problem

$$\begin{aligned} \min_{w \in H} g(w) &= \min_{w \in H} \|\mu[\mathbb{P}] - w\|^2 \\ &= \min_{w \in H} \langle w, w \rangle - 2\langle \mu[\mathbb{P}], w \rangle + \langle \mu[\mathbb{P}], \mu[\mathbb{P}] \rangle \\ &= \min_{w \in H} \frac{1}{2} \langle w, w \rangle - \langle \mu[\mathbb{P}], w \rangle. \end{aligned}$$

This is equivalent to the *stochastic optimization problem* where we minimize over the expectation of the objective function f

$$\min_{w \in H} \mathbb{E}[f(w, X)] = \min_{w \in H} \int_{\mathcal{X}} f(w, x) d\mathbb{P}(x),$$

where f is defined as $f(w, X) = \frac{1}{2} \langle w, w \rangle - \langle \phi(X), w \rangle$. For the remainder of this work we are concerned with this objective functions.

⁴ [20, Proposition Lemma 1.30]

5.1 Existence & Uniqueness

In the following let $H \subset \mathcal{H}$ be a ball with radius c defined as

$$H = \{g \in \mathcal{H} \mid \|g\| \leq c\}$$

which is a *closed bounded convex* set. Then $\mu[\mathbb{P}] \in H$ since

$$\begin{aligned} \|\mu[\mathbb{P}]\| &= \left\| \int_{\mathcal{X}} \phi(x) \, d\mathbb{P}(x) \right\| \\ &\leq \int_{\mathcal{X}} \|\phi(x)\| \, d\mathbb{P}(x) \\ &\leq \int_{\mathcal{X}} \sqrt{k(x, x)} \, d\mathbb{P}(x) \\ &\leq c \end{aligned}$$

by assumption that k bounded in expectation and shows the *existence* of a minimizer in H . Our next concern is the uniqueness of $\tilde{w} = \mu[\mathbb{P}]$ which requires f to have certain properties. We start with smoothness and convexity.

Proposition 1. *The objective function $f(w, X) = \frac{1}{2}\langle w, w \rangle - \langle \phi(X), w \rangle$ is α -strongly convex and its gradient $\nabla f(w, X) = w - \phi(X)$ is β -Lipschitz with $\alpha = \beta = 1$.⁵*

The sufficient conditions for w^* to be unique are given by [20, Corollary 2.19] which states the following.

Corollary 1. *Let H be a subset of \mathcal{H} . If $f: H \rightarrow \mathbb{R}$ is proper, convex, coercive and lower-semicontinuous then $\arg \min f$ is non-empty and weakly compact. If, moreover, f is strictly convex, then $\arg \min f$ is a singleton.*

Proof of the uniqueness of \tilde{w} . All Hilbert spaces are reflexive. Since f is continuous, proper ($\text{dom}(f) \neq \{\}$) and strongly convex it is also convex, coercive and lower-semicontinuous. \square

5.2 Convergence Rates of the Kernel Mean Map

In order to derive exact rates of convergence for EXPOSE it remains to bound the stochastic gradient $\nabla f(w_t, X)$ in theorem 1 and theorem 2. The main contribution of this section is the next theorem.

Theorem 3. *Let k be a continuous kernel which is bounded in expectation i.e. $\int_{\mathcal{X}} \sqrt{k(x, x)} \, d\mathbb{P}(x) \leq c$ and \mathcal{H} be the corresponding RKHS. Let H be the ball in \mathcal{H} with radius c , then*

$$\begin{aligned} \mathbb{E}[\|w_t - \tilde{w}\|^2] &\leq \frac{4c^2}{t} \quad \text{and} \\ \mathbb{E}[f(w_t) - f(\mu[\mathbb{P}])] &\leq \frac{2c^2}{t} \end{aligned}$$

for all $t \in \mathbb{N}$.

Proof. We first note that for $w_t \in H$ for all $t \in \mathbb{N}$ since $w_1 \in H$ by definition and for subsequent $t \geq 2$ the projection operator Π_X guarantees that $w_t \in H$ which implies $\|w_t\| \leq c$. By definition of the EXPOSE gradient we get

$$\begin{aligned} \mathbb{E}[\|\nabla f(w_t, X)\|^2] &= \mathbb{E}[\|w_t - \phi(X)\|^2] \\ &\leq \|w_t\|^2 + \mathbb{E}[\|\phi(X)\|^2] + 2\|w_t\| \mathbb{E}[\|\phi(X)\|] \\ &\leq c^2 + \mathbb{E}[k(X, X)] + c\mathbb{E}[\sqrt{k(X, X)}] \\ &\leq 4c^2 \end{aligned}$$

which also shows that $\nabla f(w_t, X)$ is strongly integrable as required by theorem 1 and theorem 2 in which be plug in the upper bound of ∇f to conclude the proof. \square

⁵ A proof can be found in the appendix.

5.3 Convergence of EXPOSE

Since w_t converges strongly to \tilde{w} as shown in theorem 3 we also have the weak convergence [20]

$$\lim_{t \rightarrow \infty} \langle u, w_t \rangle = \langle u, \tilde{w} \rangle, \quad \forall u \in H$$

and especially

$$\lim_{t \rightarrow \infty} \langle \phi(y), w_t \rangle = \langle \phi(y), \mu[\mathbb{P}] \rangle = \eta(y)$$

for all $y \in \mathcal{X}$ which justifies the use of w_t as a proxy for $\mu[\mathbb{P}]$. The final stochastic optimization procedure for EXPOSE is summarized in Algorithm 1.

Algorithm 1 A stochastic optimization procedure for EXPOSE

REQUIRE:

1: T : the number of iterations or ϵ : accuracy

ALGORITHM:

2: Set $w_1 \leftarrow 0$

3: FOR $t \leftarrow 1, 2, \dots, T$ DO

4: Sample x_t uniformly from \mathbb{P}

5: Set $\gamma_t \leftarrow \frac{1}{t}$

6: Set $f(w_t, X) \leftarrow w_t - \phi(x_t)$

7: Update $w_{t+1} \leftarrow w_t - \gamma_t \nabla f(w_t, X)$

8: Project $w_{t+1} \leftarrow w_{t+1} \cdot \max\{1, c\|w_{t+1}\|\}^{-1}$

9: RETURN w_{T+1}

We emphasize that the stochastic optimization procedure presented here is relatively simple and requires only a few lines of code to implement. It also does not introduce additional parameters since the step-size is known to be $\gamma_t = \frac{1}{t}$. Step-sizes are crucial and difficult to determine in most optimization algorithms as they have a significant effect on the results. The bound of the kernel function is typically known and the number of iterations T is a trade off between computing time and accuracy. Alternatively, the number of iterations T can be calculated given a desired accuracy ϵ . The projection operator $\Pi_H(w)$ in the last step, projecting w onto the ball H , takes a form which can efficiently be computed.

6 EXPERIMENTAL EVALUATION

In this section we present experimental results demonstrating the benefit of the proposed approach. This is challenging since the true data distribution \mathbb{P} is unknown it is therefore not possible to determine a closed form solution of $\mu[\mathbb{P}]$. After all we use the empirical kernel mean map $\mu[\mathbb{P}_n]$ as a surrogate for $\mu[\mathbb{P}]$ to evaluate the behavior of

$$\|w_t - \mu[\mathbb{P}_n]\| \approx \|w_t - \mu[\mathbb{P}]\|$$

as the number of iterations t increase. For sufficiently large sample sizes n we can expect $\mu[\mathbb{P}_n]$ to be a good proxy for $\mu[\mathbb{P}]$ by the law of large numbers. While it is theoretically possible to calculate $\|w_t - \mu[\mathbb{P}_n]\|$ explicitly, this operation is of order $\mathcal{O}(n^2)$ and therefore intractable for large n . We face the dilemma that with a small sample size n we cannot expect $\mu[\mathbb{P}_n]$ to be a good proxy for $\mu[\mathbb{P}]$ and with a large n we are not able to compute $\|w_t - \mu[\mathbb{P}_n]\|$. This requires us to replace the explicit feature maps ϕ by their approximate counterpart $\hat{\phi}$ in the following experiments.

The idea of *approximate feature maps* is to find approximations $\hat{\phi}: \mathcal{X} \rightarrow \mathbb{R}^r$ of ϕ such that

$$k(x, y) \approx \langle \hat{\phi}(x), \hat{\phi}(y) \rangle$$

for all $x, y \in \mathcal{X}$ and $r \in \mathbb{N}$. We will utilize the Random Kitchen Sinks (RKS) approach [22, 23] which is based on Bochner’s theorem for translation invariant kernels such as the squared exponential, the Laplace, the Matérn and many others kernel functions.⁶ In all experiments we apply the squared exponential kernel with 20 000 kernel expansion⁷. The squared exponential is a general purpose kernel like the Euclidean distance is the standard metric. However other choices pose a possibility to include domain and expert knowledge about the problem.

We emphasize that the use of approximate feature maps has no impact on the previous theoretical analysis and all bounds hold in infinite-dimensional RKHSs and corresponding ϕ . The sole purpose of these maps are the experimental evaluations.

6.1 Setup & Datasets

The following datasets, which all have purposely very different feature characteristics, are used to evaluate anomaly detection algorithms. We refer to [26] for a more detailed description of the datasets and their feature characteristic.

- The MNIST database contains 8 100 000 images of handwritten digits. We use the raw pixel values which results in an input space dimension of 784.
- KDDCUP is an intrusion detection dataset which contains 4 898 431 connection records of network traffic. As in [26] we rescale the 34 continuous features to $[0, 1]$ and apply a binary encoding for the 7 symbolic features.
- The third dataset contains 600 000 instances of the *Google Street View House Numbers* (SVHN) [19] where we use the *Histogram of Oriented Gradients* (HOG) with a cell size of 3 to get a 2592-dimensional feature vector.

The kernel bandwidth σ^2 used for these datasets are 7.0, 5.6 and 7.8 respectively, which we found to yield a reasonable anomaly detection performance⁸. The other datasets used in [26] are too small to benefit from the stochastic approximation approach and are hence omitted.

Since SVHN and MNIST are multi-class and not anomaly detection datasets we use all instances of digit 1 as normal data and add 1% of randomly sampled images from the remaining digits as anomalies⁹. At each iteration of Algorithm 1 we sample without replacement an instance from the training dataset which is assumed to comprise independent realizations from \mathbb{P} . We then update the model w_t according to the algorithm.

In addition to the convergence rate of the model, we will examine the anomaly detection scores

$$\begin{aligned} \eta(y) &= \langle \phi(y), \mu[\mathbb{P}_n] \rangle \quad \text{and} \\ \eta_t(y) &= \langle \phi(y), w_t \rangle \end{aligned}$$

calculated by the original EXPOSE predictor and the stochastic optimization approximation, respectively. For this purpose we periodically evaluate η and η_t on 10 000 test instances which contain 50% normal and 50% anomalous instances. These test instances are not part of the training datasets and dedicated to evaluate the classifier.

⁶ We refer to the literature and the references therein for a detailed discussion of these techniques.

⁷ Using more expansions result in better kernel approximations as the Monte Carlo estimate becomes more accurate. A value between 10 000 and 20 000 is sufficient for most applications.

⁸ Here, we are not interested to tune these parameters towards an optimal anomaly detection rate.

⁹ A different normal/anomaly setup had no significant impact on the experimental results.

6.2 Evaluation

The experimental results with approximate feature maps are shown in Fig. 2. The first plot in each column contains traces of the objective function $f(w_t) - f(\tilde{w})$, where we estimated \tilde{w} by $\mu[\mathbb{P}_n]$. We see that the stochastic optimization algorithm already reaches a reasonable low objective after a few thousand iterations. More important, we observe a similar effect in the second row when comparing $\|w_t - \tilde{w}\|$. We get near to \tilde{w} relatively fast, but it takes much more samples to estimate \tilde{w} with a high accuracy. However, a high accuracy estimation is not necessary to obtain good anomaly detection performance as shown in the following. To measure the anomaly detection rate, we first plug w_t and \tilde{w} into the EXPOSE estimators η_t and η respectively and calculate scores for all instances of the test dataset. The differences between these scores are shown in the third row. We see again that the stochastic optimization approximation η_t yields similar scores as η using the full model. The last row illustrates the development of the area under ROC [12] as the optimization routine performs more iterations. *After only a few thousand iterations η_t reaches the same predictive performance as the original EXPOSE predictor η .* This confirms that a high accuracy approximation of $\mu[\mathbb{P}]$ does not necessarily lead to a better predictor. The key is, that for a given ϵ we can reach $\|w_t - \mu[\mathbb{P}]\| < \epsilon$ in a fixed number of iterations, *independent* of the dataset size n which reduced the computational complexity from $\mathcal{O}(n)$ to $\mathcal{O}(1)$ as claimed.

To see the performance gain in seconds, we stopped the algorithms at a relatively high accuracy $\|w_t - \mu[\mathbb{P}]\| \leq \epsilon = 0.005$ and compare the full model η with the approximation η_t (table 1).

Table 1. Comparison of the η and the stochastic approximation η_t

	TIME [sec]			AUC	
	η	η_t		η	η_t
SVHN	437	32	■ —	0.890	0.890
MNIST	248	19	■ —	0.991	0.990
KDDCUP	961	81	■ —	0.981	0.981

We observe that the training time is significant shorter while the predictive performance (the area under ROC) is still very close to the one obtained by the full model. It is important to note that the computational complexity of η_t is constant which means that even if the datasets grow in size, we would not need more training time to achieve the same accuracy.

7 SUMMARY

This is the first time the EXPOSE anomaly detection algorithm is viewed from the perspective of a stochastic optimization problem which enables us to find an ϵ -accurate approximation of the kernel mean map $\mu[\mathbb{P}]$ in *constant time*, independent of the training dataset size n . Such constant time algorithms are the key to tackle large-scale machine learning problems.

To achieve this goal we take existing stochastic optimization approaches and port them to infinite-dimensional problems. Moreover we carefully choose an objective function f such that an optimal solution can be shown to *exist* and to be *unique*. Hereby the choice of the working subset H in \mathcal{H} is of crucial importance to bound the gradient and to automatically derive the step size. It is also essential that f is chosen such that it is strongly convex and obeys a Lipschitz

continuous gradient with the right constants. Despite all these constraints we tried to minimize our assumptions. We only require the input space to be separable and the kernel function to be continuous and bounded in expectation.

As a result, the proposed approximation reduces the computational complexity of EXPOSE and the empirical kernel mean map $\mu^{[\mathbb{P}_n]}$ from the previous $\mathcal{O}(n)$ to $\mathcal{O}(1)$ whenever an ϵ -accurate estimation is sufficient. More precisely, we are able to determine the number of necessary stochastic optimization iterations T for a user defined error threshold ϵ such that $\mathbb{E}[\|w_T - \mu^{[\mathbb{P}]}\|] < \epsilon$. The intuition is that a very high accuracy estimation $\mu^{[\mathbb{P}]}$ does not necessarily result in a better anomaly detection performance and hence there is no benefit in spending more computational resources. This intuition is also confirmed experimentally on three large-scale datasets, where we reach the same anomaly detection performance long before all data is incorporated into the model. We emphasize that, unlike other optimization problems in this class, EXPOSE does not have a regularization parameter. This is important as the authors of Pegasos noticed that

[...] the runtime to achieve a predetermined suboptimality threshold would increase in proportion to λ [the regularization parameter]. Very small values of λ (small amounts of regularization) result in rather long runtimes. [27]

We expect our contribution to have significant implications for large-scale applications such as anomaly detection problems and other techniques, which are based on the kernel mean embedding.

APPENDIX

Definition 6 (Strong Integral). Let $(\mathcal{X}, \mathcal{X}, \mathbb{P})$ be a σ -finite measure space and let $\phi: \mathcal{X} \rightarrow \mathcal{H}$ be measurable. Then ϕ is strong integrable (Bochner integrable) over a set $\mathcal{D} \in \mathcal{X}$ if and only if its norm $\|\phi\|$ is Lebesgue integrable over \mathcal{D} , that is,

$$\int_{\mathcal{D}} \|\phi\| d\mathbb{P}(x) < \infty.$$

If ϕ is strong integrable over each $\mathcal{D} \in \mathcal{X}$ we say that ϕ is strong integrable. [4, Theorem 11.44]

Definition 7 (Weak Integral). Let $(\mathcal{X}, \mathcal{X}, \mathbb{P})$ be a σ -finite measure space. A function $\phi: \mathcal{X} \rightarrow \mathcal{H}$ is weakly integrable over a set $\mathcal{D} \in \mathcal{X}$ if there exists some $\lambda \in \mathcal{H}$ satisfying

$$\langle f, \lambda \rangle = \int_{\mathcal{D}} \langle f, \phi(x) \rangle d\mathbb{P}(x)$$

for each $f \in \mathcal{H}$. The weak integral is denoted by

$$\lambda = \oint_{\mathcal{D}} \phi d\mathbb{P}(x)$$

and the unique element $\lambda \in \mathcal{H}$ is called weak integral of ϕ over \mathcal{D} . If the integral exists for each $\mathcal{D} \in \mathcal{X}$ we say that ϕ is weakly integrable. [4, Section 11.10]

Proof of lemma 2

Proof. By definition of α -strong convexity we know that¹⁰

$$f(w) \geq f(v) + \langle \nabla f(v), w - v \rangle + \frac{\alpha}{2} \|w - v\|^2$$

¹⁰ [20, Proposition 3.23]

and therefore

$$\begin{aligned} f(v) - f(w) &\leq \langle \nabla f(v), v - w \rangle - \frac{\alpha}{2} \|w - v\|^2 \\ \langle \nabla f(v), v - w \rangle &\geq f(v) - f(w) + \frac{\alpha}{2} \|w - v\|^2 \end{aligned}$$

for all v, w in H . If \tilde{w} is a global minimizer of f , then

$$f(v) - f(\tilde{w}) \geq 0$$

for all $v \in H$ which concludes the proof. \square

Proof of lemma 3

Proof. By the Cauchy–Schwarz inequality we obtain

$$\begin{aligned} \alpha^2 \|v - \tilde{w}\|^4 &\leq \langle v - \tilde{w}, \nabla f(v) \rangle^2 \\ &\leq \|v - \tilde{w}\|^2 \|\nabla f(v)\|^2 \\ \alpha^2 \|v - \tilde{w}\|^2 &\leq \|\nabla f(v)\|^2 \end{aligned}$$

which concludes the proof. \square

Proof of proposition 1

Proof. We know that f is α -strongly convex with $\alpha = 1$ since

$$\begin{aligned} w \mapsto f(w, X) - \frac{\alpha}{2} \|w\|^2 &= \frac{1}{2} \langle w, w \rangle - \langle \phi(X), w \rangle - \frac{\alpha}{2} \|w\|^2 \\ &= -\langle \phi(X), w \rangle \end{aligned}$$

is convex in $w \in \mathcal{H}$. Furthermore, $Df(w, X): z \mapsto \langle w - \phi(X), z \rangle_*$ is the Fréchet derivative of f at $w \in \mathcal{H}$ since

$$\lim_{h \rightarrow 0} \frac{\|f(w + h, X) - f(w, X) - \langle Df(w, X), h \rangle_*\|}{\|h\|} = 0$$

using the notation $\langle \cdot, \cdot \rangle_*$ for the dual map. By Riesz' representation theorem we can identify $Df(w, X): z \mapsto \langle w - \phi(X), z \rangle_*$ with an element in \mathcal{H} which we denote by $\nabla f(w, X) = w - \phi(X)$. This function is β -Lipschitz with constant $\beta = 1$ since

$$\begin{aligned} \|\nabla f(w, X) - \nabla f(v, X)\| &= \|w - \phi(X) - v + \phi(X)\| \\ &= \|w - v\| \end{aligned}$$

for all w and v in \mathcal{H} . \square

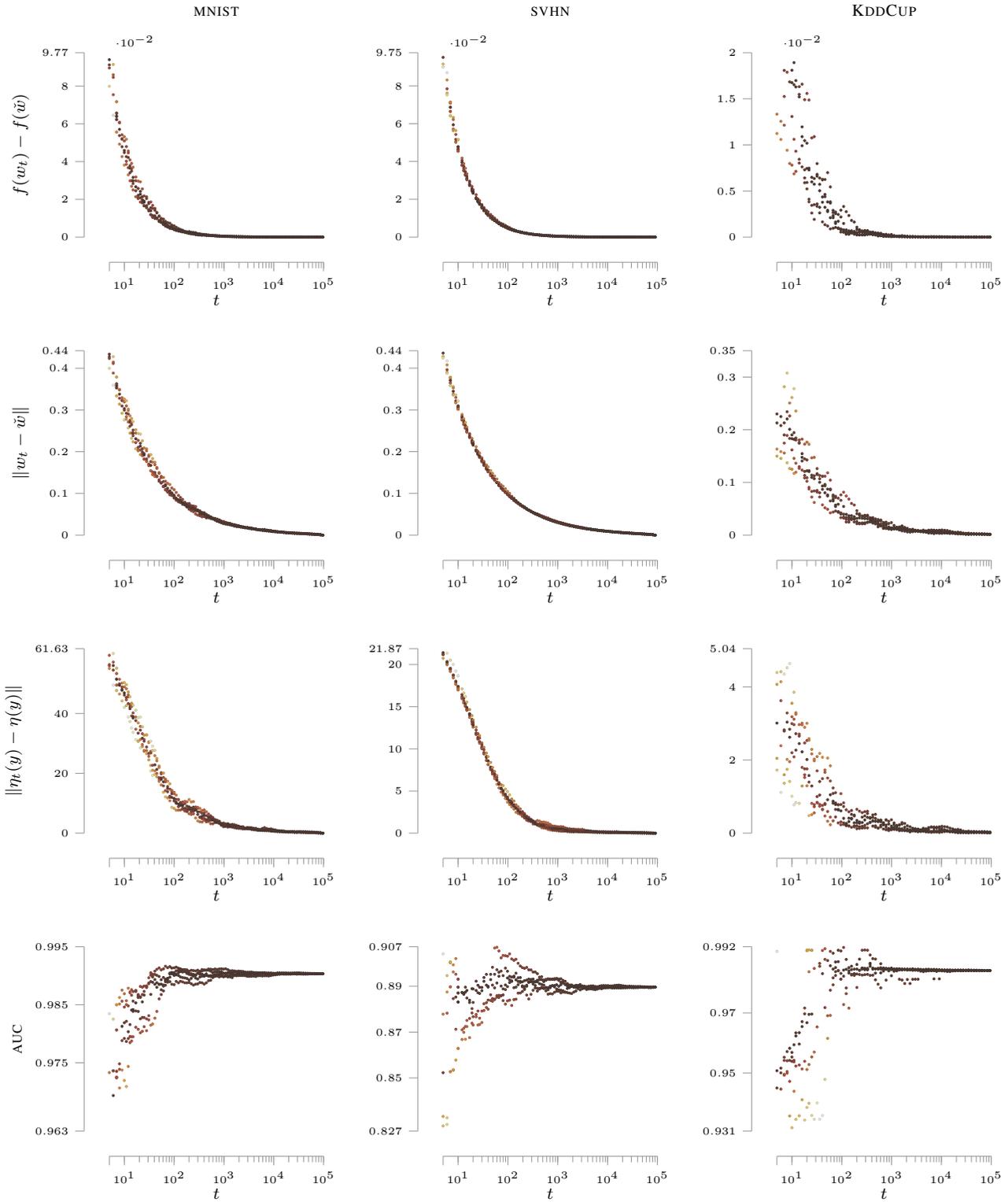


Figure 2. Evaluation of the stochastic optimization approach for EXPOSE. Each column refers to a different dataset. Each experiment was repeated 5 times represented by a dot. A darker color indicates a value closer to the mean of the 5 repetitions.

References

- [1] Alekh Agarwal, Peter Bartlett, Pradeep Ravikumar, and Martin Wainwright, ‘Information-Theoretic Lower Bounds on the Oracle Complexity of Convex Optimization Convex optimization’, in *Advances in Neural Information Processing Systems*, pp. 1–9, (2011).
- [2] Charu C Aggarwal, *Outlier analysis*, Springer, 2013.
- [3] Emin Aleskerov, Bernd Freisleben, and Bharat Rao, ‘Cardwatch: A neural network based database mining system for credit card fraud detection’, in *Computational Intelligence for Financial Engineering*, (1997).
- [4] Charalambos D Aliprantis and Kim Border, *Infinite dimensional analysis: a hitchhiker’s guide*, Springer Science & Business Media, 2006.
- [5] Francis Bach, ‘Adaptivity of averaged stochastic gradient descent to local strong convexity for logistic regression’, *The Journal of Machine Learning Research*, **15**(1), 595–627, (2014).
- [6] Olivier Bousquet and Léon Bottou, ‘The tradeoffs of large scale learning’, in *Advances in neural information processing systems*, pp. 161–168, (2008).
- [7] Stephen Boyd and Lieven Vandenberghe, *Convex optimization*, Cambridge University Press, 2004.
- [8] Varun Chandola, Arindam Banerjee, and Vipin Kumar, ‘Anomaly detection: A survey’, *ACM Computing Surveys (CSUR)*, **41**(3), 1–58, (2009).
- [9] Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Bharath K Sriperumbudur, ‘Characteristic kernels on groups and semigroups’, in *Advances in Neural Information Processing Systems*, pp. 473–480, (2009).
- [10] Arthur Gretton, Karsten M Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alexander Johannes Smola, ‘A kernel method for the two-sample-problem’, *Advances in neural information processing systems*, (2006).
- [11] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola, ‘A kernel two-sample test’, *The Journal of Machine Learning Research*, **13**(1), 723–773, (2012).
- [12] James A Hanley and Barbara J McNeil, ‘The meaning and use of the area under a receiver operating characteristic (ROC) curve’, *Radiology*, **143**(1), 29–36, (1982).
- [13] Victoria J Hodge and Jim Austin, ‘A survey of outlier detection methodologies’, *Artificial Intelligence Review*, **22**(2), 85–126, (2004).
- [14] Stephen M Kent, ‘Sloan digital sky survey’, in *Science with Astronomical Near-Infrared Sky Surveys*, 27–30, Springer, (1994).
- [15] Vipin Kumar, ‘Parallel and distributed computing for cybersecurity’, *IEEE Distributed Systems Online*, **6**(10), 1, (2005).
- [16] Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro, ‘Robust stochastic approximation approach to stochastic programming’, *SIAM Journal on Optimization*, **19**(4), 1574–1609, (2009).
- [17] Yurii Nesterov, ‘A method of solving a convex programming problem with convergence rate $O(1/k^2)$ ’, *Soviet Mathematics Doklady*, **27**(2), 372–376, (1983).
- [18] Yurii Nesterov, *Introductory lectures on convex optimization*, volume 87, Springer Science & Business Media, 2004.
- [19] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Ng, ‘Reading digits in natural images with unsupervised feature learning’, in *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, p. 4, (2011).
- [20] Juan Peypouquet, *Convex Optimization in Normed Spaces: Theory, Methods and Examples*, Springer, 2015.
- [21] Marcel Prastawa, Elizabeth Bullitt, Sean Ho, and Guido Gerig, ‘A brain tumor segmentation framework based on outlier detection’, *Medical image analysis*, **8**(3), 275–283, (2004).
- [22] Ali Rahimi and Benjamin Recht, ‘Random features for large-scale kernel machines’, in *Advances in neural information processing systems*, pp. 1177–1184, (2007).
- [23] Ali Rahimi and Benjamin Recht, ‘Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning’, in *Advances in neural information processing systems*, pp. 1313–1320, (2008).
- [24] Herbert Robbins and Sutton Monro, ‘A stochastic approximation method’, *The annals of mathematical statistics*, 400–407, (1951).
- [25] Nicolas L Roux, Mark Schmidt, and Francis R Bach, ‘A stochastic gradient method with an exponential convergence rate for finite training sets’, in *Advances in Neural Information Processing Systems*, pp. 2663–2671, (2012).
- [26] Markus Schneider, Wolfgang Ertel, and Günther Palm, ‘Expected Similarity Estimation for Large Scale Anomaly Detection’, in *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2015)*, pp. 1–8. IEEE, (2015).
- [27] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter, ‘Pegasos: Primal estimated sub-gradient solver for SVM’, in *Mathematical Programming*, volume 127, pp. 3–30, (2011).
- [28] Shai Shalev-Shwartz and Ambuj Tewari, ‘Stochastic methods for l_1 -regularized loss minimization’, *The Journal of Machine Learning Research*, **12**, 1865–1892, (2011).
- [29] Alex J Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf, ‘A Hilbert space embedding for distributions’, in *Algorithmic Learning Theory*, pp. 13–31. Springer, (2007).
- [30] Clay Spence, Lucas Parra, and Paul Sajda, ‘Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model’, in *Mathematical Methods in Biomedical Image Analysis, 2001. MMBIA 2001. IEEE Workshop on*, pp. 3–10. IEEE, (2001).
- [31] Bharath K Sriperumbudur, Kenji Fukumizu, and Gert R G Lanckriet, ‘Universality, characteristic kernels and RKHS embedding of measures’, *The Journal of Machine Learning Research*, **12**, 2389–2410, (2011).
- [32] Bharath K Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Gert Lanckriet, and Bernhard Schölkopf, ‘Injective Hilbert space embeddings of probability measures’, *Proceedings of the 21st Annual Conference on Learning Theory*, 111–122, (2008).
- [33] Eberhard Zeidler, *Applied Functional Analysis*, Springer New York, 1995.
- [34] Bin Zhang, Chris Sconyers, Carl Byington, Romano Patrick, Marcos E Orchard, and George Vachtsevanos, ‘A probabilistic fault detection approach: application to bearing fault detection’, *IEEE Transactions on Industrial Electronics*, **58**(5), 2011–2018, (2011).

AUC Maximization in Bayesian Hierarchical Models

Mehmet Gönen¹

Abstract. The area under the curve (AUC) measures such as the area under the receiver operating characteristics curve (AUROC) and the area under the precision-recall curve (AUPR) are known to be more appropriate than the error rate, especially, for imbalanced data sets. There are several algorithms to optimize AUC measures instead of minimizing the error rate. However, this idea has not been fully exploited in Bayesian hierarchical models owing to the difficulties in inference. Here, we formulate a general Bayesian inference framework, called Bayesian AUC Maximization (BAM), to integrate AUC maximization into Bayesian hierarchical models by borrowing the pairwise and listwise ranking ideas from the information retrieval literature. To showcase our BAM framework, we develop two Bayesian linear classifier variants for two ranking approaches and derive their variational inference procedures. We perform validation experiments on four biomedical data sets to demonstrate the better predictive performance of our framework over its error-minimizing counterpart in terms of average AUROC and AUPR values.

1 INTRODUCTION

In binary classification problems, we are given a sample of N independent and identically distributed training instances $\mathbf{X} = \{\mathbf{x}_n \in \mathcal{X}\}_{n=1}^N$ and their class labels $\mathbf{y} = \{y_n \in \{-1, +1\}\}_{n=1}^N$. We then use \mathbf{X} and \mathbf{y} to learn usually a parametric function that can be used to predict the class labels of unseen test instances. Let $\mathbf{f} = \{f_n \in \mathbb{R}\}_{n=1}^N$ be the output values of this parametric function when evaluated on the training instances. The output values can be, for example, posterior probabilities assigned to one of the classes in neural networks or discriminant outputs in support vector machines. During training, the classification parameters used to generate the output values are selected by optimizing an objective function, which usually contains a loss function defined on \mathbf{f} and \mathbf{y} such as the hinge loss and squared error loss to minimize the expected error rate on test instances.

The error rate is by far the most commonly used performance measure to compare different classification models. However, the area under the curve (AUC) measures such as the area under the receiver operating characteristics curve (AUROC) and the area under the precision-recall curve (AUPR) are better suited to imbalanced binary classification problems. As noted by many earlier studies [5, 9, 11, 21], minimizing the error rate may not lead to better AUC measures. To the best of our knowledge, there is not a full-Bayesian algorithm to optimize AUC measures owing to the difficulties in inference.

In this work, we study AUC maximization for Bayesian hierarchical models and propose a novel inference framework, called Bayesian AUC Maximization (BAM), to optimize AUC measures

with a full-Bayesian treatment. To this aim, we borrow the pairwise and listwise ranking ideas from the information retrieval literature and show how they can help us maximize AUROC values in Bayesian hierarchical models. We demonstrate the better predictive performance of our framework on four biomedical data sets by comparing it to an error-minimizing baseline algorithm.

2 MODELING AUC MAXIMIZATION USING CATEGORICAL DISTRIBUTIONS

To be able to model AUC maximization in Bayesian hierarchical models, we first write AUROC as a function of the output values and then show two possible strategies to represent this function using random variables from the categorical distributions (also known as generalized Bernoulli distribution or multinomial distribution with a single trial).

It is very well-known that AUROC is equal to the value of the Wilcoxon-Mann-Whitney statistic in the discrete case [7]:

$$\text{AUROC}(\mathbf{f}) = \frac{1}{|\mathcal{P}||\mathcal{N}|} \sum_{n \in \mathcal{P}} \sum_{o \in \mathcal{N}} \delta(f_n > f_o),$$

where $\mathcal{P} = \{n: y_n = +1\}$, $\mathcal{N} = \{n: y_n = -1\}$, $|\cdot|$ gives the cardinality of the input set, and $\delta(\cdot)$ represents the Kronecker delta function that returns 1 if its argument is true and 0 otherwise.

The first strategy to represent AUROC using the categorical distributions is similar to the pairwise ranking models in the information retrieval literature, which force the output value of a relevant document to be larger than that of an irrelevant document for all relevant-irrelevant document pairs [3, 6, 10]. The Wilcoxon-Mann-Whitney statistic considers all pairs defined between the positive and negative instances, which we can represent using auxiliary random variables drawn from categorical distributions with two possible outcomes and their respective probabilities calculated using the softmax function:

$$z_{n,o} | f_n, f_o \sim \mathcal{C} \left(z_{n,o}; \{n, o\}, \frac{[\exp(f_n) \quad \exp(f_o)]}{\exp(f_n) + \exp(f_o)} \right), \quad (1)$$

where $(n, o) \in \mathcal{P} \times \mathcal{N}$ and $\mathcal{C}(\cdot; \mathbf{E}, \boldsymbol{\pi})$ denotes the categorical distribution with the event set \mathbf{E} and the event probabilities $\boldsymbol{\pi}$. Figure 1 illustrates the pairwise ranking idea applied on a toy data set with two positive and two negative instances, which requires four categorical random variables to be defined. To maximize AUROC during training, we can treat $z_{n,o}$ variables as observed variables and set each of them to the index of the positive instance involved, i.e., $z_{n,o} = n$.

The pairwise ranking strategy requires $|\mathcal{P}| \times |\mathcal{N}|$ categorical random variables to be added, whereas we can reduce this number to $\min(|\mathcal{P}|, |\mathcal{N}|)$ using the listwise ranking models in the information retrieval literature, which force the output value of a relevant document to be larger than those of all irrelevant documents at once

¹ Department of Industrial Engineering, Koç University, İstanbul, Turkey, email: mehmetgonen@ku.edu.tr

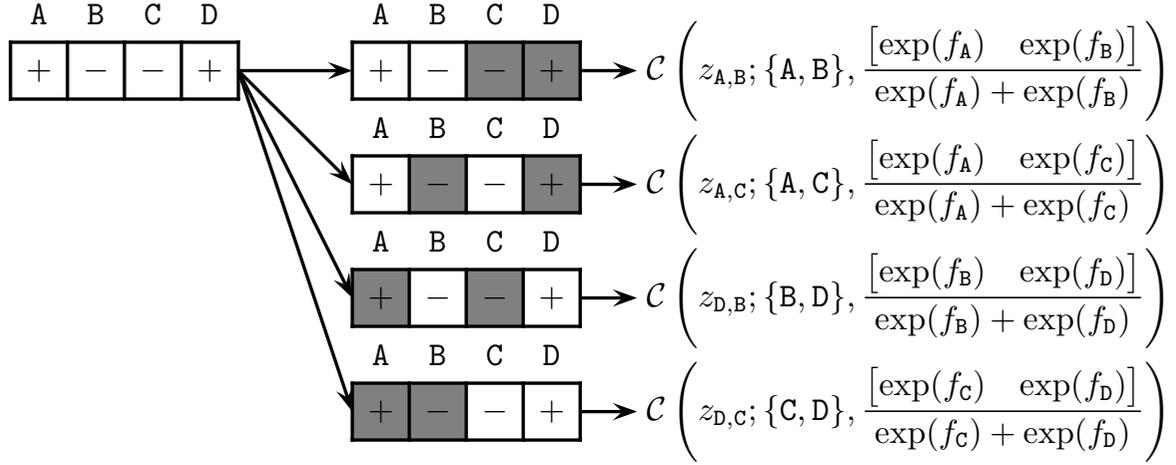


Figure 1. Pairwise ranking applied to modeling AUC maximization using categorical distributions.

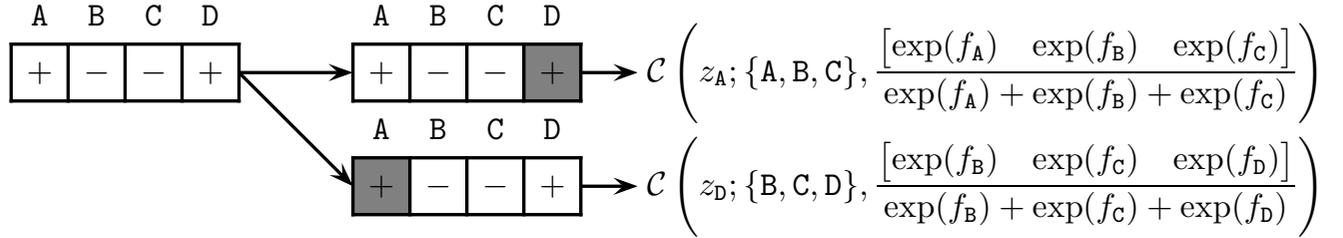


Figure 2. Listwise ranking applied to modeling AUC maximization using categorical distributions.

[4, 13, 20]. Without loss of generality, we assume that $|\mathcal{P}| < |\mathcal{N}|$ in the following. For each positive instance, we can add an auxiliary random variable drawn from a categorical distribution with $(|\mathcal{N}|+1)$ possible outcomes and their respective probabilities calculated using the softmax function:

$$z_n | \{f_o\}_{o \in \mathcal{W}_n} \sim \mathcal{C} \left(z_n; \mathcal{W}_n, \left[\frac{\exp(f_o)}{\sum_{p \in \mathcal{W}_n} \exp(f_p)} \right]_{o \in \mathcal{W}_n} \right), \quad (2)$$

where $n \in \mathcal{P}$ and $\mathcal{W}_n = \{n\} \cup \mathcal{N}$. Figure 2 illustrates the listwise ranking idea applied on a toy data set with two positive and two negative instances, which requires two categorical random variables to be defined. To maximize AUROC during training, we can treat z_n variables as observed variables and set each of them to the index of the positive instance involved, i.e., $z_n = n$. Reducing the number of categorical random variables from $|\mathcal{P}| \times |\mathcal{N}|$ to $\min(|\mathcal{P}|, |\mathcal{N}|)$ would help us make our inference procedures more effective as detailed later.

3 BAYESIAN AUC MAXIMIZATION

To showcase our framework, without loss of generality, we use Bayesian probit regression model as our baseline method [1]. We first describe this model briefly and then give detailed derivations for our two AUC-maximizing variants of this model.

3.1 Bayesian Probit Regression as Baseline Linear Classifier

The distributional assumptions of Bayesian probit regression model are defined as

$$\gamma \sim \mathcal{G}(\gamma; \alpha_\gamma, \beta_\gamma),$$

$$\begin{aligned} b | \gamma &\sim \mathcal{N}(b; 0, \gamma^{-1}), \\ \eta_d &\sim \mathcal{G}(\eta_d; \alpha_\eta, \beta_\eta) \quad \forall d, \\ w_d | \eta_d &\sim \mathcal{N}(w_d; 0, \eta_d^{-1}) \quad \forall d, \\ f_n | b, \mathbf{w}, \mathbf{x}_n &\sim \mathcal{N}(f_n; \mathbf{w}^\top \mathbf{x}_n + b, 1) \quad \forall n, \\ y_n | f_n &\sim \delta(f_n y_n > \nu) \quad \forall n, \end{aligned} \quad (3)$$

where $\{f_n\}_{n=1}^N$ is the set of output values introduced to make the inference procedures efficient [1]. The nonnegative margin parameter ν is introduced to resolve the scaling ambiguity and to place a low-density region between two classes, similar to the margin idea in support vector machines. $\mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ represents the normal distribution with the mean vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$. $\mathcal{G}(\cdot; \alpha, \beta)$ denotes the gamma distribution with the shape parameter α and the scale parameter β .

We can approximate the required posterior as

$$\begin{aligned} p(\gamma, \boldsymbol{\eta}, b, \mathbf{w}, \mathbf{f} | \mathbf{X}, \mathbf{y}) \\ \approx q(\gamma) q(\boldsymbol{\eta}) q(b, \mathbf{w}) q(\mathbf{f}), \end{aligned}$$

and define each factor in the ensemble just like its full conditional distribution:

$$\begin{aligned} q(\gamma) &= \mathcal{G}(\gamma; \alpha(\gamma), \beta(\gamma)), \\ q(\boldsymbol{\eta}) &= \prod_{d=1}^D \mathcal{G}(\eta_d; \alpha(\eta_d), \beta(\eta_d)), \\ q(b, \mathbf{w}) &= \mathcal{N} \left(\begin{bmatrix} b \\ \mathbf{w} \end{bmatrix}; \boldsymbol{\mu}(b, \mathbf{w}), \boldsymbol{\Sigma}(b, \mathbf{w}) \right), \\ q(\mathbf{f}) &= \prod_{n=1}^N \mathcal{TN}(f_n; \boldsymbol{\mu}(f_n), \boldsymbol{\Sigma}(f_n), \rho(f_n)), \end{aligned}$$

where $\alpha(\cdot)$, $\beta(\cdot)$, $\mu(\cdot)$, and $\Sigma(\cdot)$ denote the shape parameter, the scale parameter, the mean vector, and the covariance matrix for their arguments, respectively. $\mathcal{TN}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \rho(\cdot))$ represents the truncated normal distribution with the mean vector $\boldsymbol{\mu}$, the covariance matrix $\boldsymbol{\Sigma}$, and the truncation rule $\rho(\cdot)$ such that $\mathcal{TN}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \rho(\cdot)) \propto \mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ if $\rho(\cdot)$ is true and $\mathcal{TN}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \rho(\cdot)) = 0$ otherwise.

We can bound the likelihood using Jensen's inequality:

$$\begin{aligned} \log p(\mathbf{y}|\mathbf{X}) &\geq \mathbb{E}_q[\log p(\boldsymbol{\gamma}, \boldsymbol{\eta}, b, \mathbf{w}, \mathbf{f}, \mathbf{y}|\mathbf{X})] - \mathbb{E}_q[\log q(\boldsymbol{\gamma}, \boldsymbol{\eta}, b, \mathbf{w}, \mathbf{f})], \end{aligned}$$

where $\mathbb{E}_q[\cdot]$ denotes the posterior expectations, and optimize this bound by maximizing with respect to each factor until convergence, leading to the following update equations:

$$\alpha(\gamma) = \alpha_\gamma + \frac{1}{2}, \quad \beta(\gamma) = \left(\beta_\gamma^{-1} + \frac{1}{2} \mathbb{E}_q[b^2] \right)^{-1}, \quad (4)$$

$$\alpha(\eta_d) = \alpha_\eta + \frac{1}{2}, \quad \beta(\eta_d) = \left(\beta_\eta^{-1} + \frac{1}{2} \mathbb{E}_q[w_d^2] \right)^{-1}, \quad (5)$$

$$\Sigma(b, \mathbf{w}) = \begin{bmatrix} \mathbb{E}_q[\gamma] + N & \mathbf{1}^\top \mathbf{X}^\top \\ \mathbf{X} \mathbf{1} & \text{diag}(\mathbb{E}_q[\boldsymbol{\lambda}]) + \mathbf{X} \mathbf{X}^\top \end{bmatrix}^{-1}, \quad (6)$$

$$\boldsymbol{\mu}(b, \mathbf{w}) = \Sigma(b, \mathbf{w}) \begin{pmatrix} \mathbf{1}^\top \\ \mathbf{X} \end{pmatrix} \mathbb{E}_q[\mathbf{f}], \quad (7)$$

$$\Sigma(f_n) = 1, \quad \rho(f_n) \triangleq f_n y_n > \nu, \quad (8)$$

$$\boldsymbol{\mu}(f_n) = \Sigma(f_n) \mathbb{E}_q[\mathbf{w}^\top \mathbf{x}_n + b], \quad (9)$$

where $\mathbf{1}$ denotes a vector of ones of appropriate dimension.

3.2 AUC-Maximizing Linear Classifier Variant Using Pairwise Ranking

To develop our linear classification variant with pairwise ranking flavor, we first start by replacing (3) in our baseline Bayesian probit regression model with (1). In this modified model, the update equations for $\boldsymbol{\gamma}$, $\boldsymbol{\eta}$, b , and \mathbf{w} remain intact because they are assumed to be independent from \mathbf{f} in the posterior approximation. However, we can not have closed-form update equations for the parameters of the output values $\{f_n\}_{n=1}^N$ owing to the softmax function in (1). Instead, we update these parameters by solving a series of unconstrained optimization problems on the lower bound.

The lower bound we need to maximize to update the parameters of the output values is

$$\begin{aligned} \mathcal{L}_q(\mathbf{f}) &= \sum_{n=1}^N (\mathbb{E}_q[\log p(f_n|b, \mathbf{w}, \mathbf{x}_n)] - \mathbb{E}_q[\log q(f_n)]) \\ &\quad + \sum_{n \in \mathcal{P}} \sum_{o \in \mathcal{N}} \mathbb{E}_q[\log p(z_{n,o}|f_n, f_o)] + \text{const.}, \end{aligned}$$

where the first two terms are log-likelihood and negative entropy terms for the output values, whereas the third term stems from the auxiliary random variables introduced in (1). The log-likelihood term can be decomposed as

$$\begin{aligned} \mathbb{E}_q[\log p(f_n|b, \mathbf{w}, \mathbf{x}_n)] &= \mathbb{E}_q \left[-\frac{1}{2} \log(2\pi) - \frac{1}{2} (f_n - \mathbf{w}^\top \mathbf{x}_n - b)^2 \right] \\ &= -\frac{1}{2} \log(2\pi) - \frac{1}{2} \mathbb{E}_q[f_n^2] + \mathbb{E}_q[f_n] \mathbb{E}_q[\mathbf{w}^\top \mathbf{x}_n + b] \\ &\quad - \frac{1}{2} \mathbb{E}_q[(\mathbf{w}^\top \mathbf{x}_n + b)^2], \end{aligned}$$

where $\mathbb{E}_q[f_n] = \mu(f_n)$ and $\mathbb{E}_q[f_n^2] = \mu(f_n)^2 + \Sigma(f_n)$. The negative entropy term is

$$\mathbb{E}_q[\log q(f_n)] = -\frac{1}{2} (\log(2\pi\Sigma(f_n)) + 1).$$

The last term with the softmax function can be lower bounded with a local variational approximation, which uses a linear Taylor expansion of the log function [2]:

$$\begin{aligned} \mathbb{E}_q[\log p(z_{n,o}|f_n, f_o)] &= \mathbb{E}_q \left[\log \left(\frac{\exp(f_n)}{\exp(f_n) + \exp(f_o)} \right) \right] \\ &= \mathbb{E}_q[f_n] - \mathbb{E}_q[\log(\exp(f_n) + \exp(f_o))] \\ &\geq \mathbb{E}_q[f_n] - \left(\frac{(\mathbb{E}_q[\exp(f_n)] + \mathbb{E}_q[\exp(f_o)])}{\zeta_{n,o}} - 1 + \log(\zeta_{n,o}) \right), \end{aligned}$$

where $\{\zeta_{n,o}\}_{n \in \mathcal{P}, o \in \mathcal{N}}$ is the set of variational parameters introduced and $\mathbb{E}_q[\exp(f_n)] = \exp(\mu(f_n) + \Sigma(f_n)/2)$.

We can now write the lower bound as a function of $\{\mu(f_n)\}_{n=1}^N$, $\{\Sigma(f_n)\}_{n=1}^N$, and $\{\zeta_{n,o}\}_{n \in \mathcal{P}, o \in \mathcal{N}}$, and optimize the lower bound with respect to each of these three sets of parameters separately.

3.2.1 Optimizing $\mathcal{L}_q(\mathbf{f})$ with respect to $\{\zeta_{n,o}\}_{n \in \mathcal{P}, o \in \mathcal{N}}$

Given $\{\mu(f_n)\}_{n=1}^N$ and $\{\Sigma(f_n)\}_{n=1}^N$, the optimal values for $\{\zeta_{n,o}\}_{n \in \mathcal{P}, o \in \mathcal{N}}$ can be found as

$$\zeta_{n,o}^* = \exp\left(\mu(f_n) + \frac{\Sigma(f_n)}{2}\right) + \exp\left(\mu(f_o) + \frac{\Sigma(f_o)}{2}\right). \quad (10)$$

3.2.2 Derivatives of $\mathcal{L}_q(\mathbf{f})$ with respect to $\{\mu(f_n)\}_{n=1}^N$

Given $\{\Sigma(f_n)\}_{n=1}^N$ and $\{\zeta_{n,o}\}_{n \in \mathcal{P}, o \in \mathcal{N}}$, we can find the first and second derivatives of the lower bound with respect to $\{\mu(f_n)\}_{n \in \mathcal{P}}$ as follows:

$$\begin{aligned} \frac{\partial \mathcal{L}_q(\mathbf{f})}{\partial \mu(f_n)} &= -\mu(f_n) + \mathbb{E}_q[\mathbf{w}^\top \mathbf{x}_n + b] + |\mathcal{N}| \\ &\quad - \sum_{o \in \mathcal{N}} \frac{1}{\zeta_{n,o}} \exp\left(\mu(f_n) + \frac{\Sigma(f_n)}{2}\right) \\ \frac{\partial^2 \mathcal{L}_q(\mathbf{f})}{\partial \mu(f_n)^2} &= -1 - \sum_{o \in \mathcal{N}} \frac{1}{\zeta_{n,o}} \exp\left(\mu(f_n) + \frac{\Sigma(f_n)}{2}\right), \end{aligned}$$

and with respect to $\{\mu(f_o)\}_{o \in \mathcal{N}}$ as follows:

$$\begin{aligned} \frac{\partial \mathcal{L}_q(\mathbf{f})}{\partial \mu(f_o)} &= -\mu(f_o) + \mathbb{E}_q[\mathbf{w}^\top \mathbf{x}_o + b] \\ &\quad - \sum_{n \in \mathcal{P}} \frac{1}{\zeta_{n,o}} \exp\left(\mu(f_o) + \frac{\Sigma(f_o)}{2}\right), \\ \frac{\partial^2 \mathcal{L}_q(\mathbf{f})}{\partial \mu(f_o)^2} &= -1 - \sum_{n \in \mathcal{P}} \frac{1}{\zeta_{n,o}} \exp\left(\mu(f_o) + \frac{\Sigma(f_o)}{2}\right). \end{aligned}$$

3.2.3 Derivatives of $\mathcal{L}_q(\mathbf{f})$ with respect to $\{\Sigma(f_n)\}_{n=1}^N$

Given $\{\mu(f_n)\}_{n=1}^N$ and $\{\zeta_{n,o}\}_{n \in \mathcal{P}, o \in \mathcal{N}}$, we can find the first and second derivatives of the lower bound with respect to $\{\Sigma(f_n)\}_{n \in \mathcal{P}}$ as follows:

$$\begin{aligned} \frac{\partial \mathcal{L}_q(\mathbf{f})}{\partial \Sigma(f_n)} &= -\frac{1}{2} - \sum_{o \in \mathcal{N}} \frac{1}{2\zeta_{n,o}} \exp\left(\mu(f_n) + \frac{\Sigma(f_n)}{2}\right) + \frac{1}{2\Sigma(f_n)}, \\ \frac{\partial^2 \mathcal{L}_q(\mathbf{f})}{\partial \Sigma(f_n)^2} &= - \sum_{o \in \mathcal{N}} \frac{1}{4\zeta_{n,o}} \exp\left(\mu(f_n) + \frac{\Sigma(f_n)}{2}\right) - \frac{1}{2\Sigma(f_n)^2}, \end{aligned}$$

and with respect to $\{\mu(f_o)\}_{o \in \mathcal{N}}$ as follows:

$$\frac{\partial \mathcal{L}_q(\mathbf{f})}{\partial \Sigma(f_o)} = -\frac{1}{2} - \sum_{n \in \mathcal{P}} \frac{1}{2\zeta_{n,o}} \exp\left(\mu(f_o) + \frac{\Sigma(f_o)}{2}\right) + \frac{1}{2\Sigma(f_o)},$$

$$\frac{\partial^2 \mathcal{L}_q(\mathbf{f})}{\partial \Sigma(f_o)^2} = -\sum_{n \in \mathcal{P}} \frac{1}{4\zeta_{n,o}} \exp\left(\mu(f_o) + \frac{\Sigma(f_o)}{2}\right) - \frac{1}{2\Sigma(f_o)^2}.$$

As our overall inference scheme, we first perform closed-form variational updates for γ , $\boldsymbol{\eta}$, b , and \mathbf{w} as given in (4)–(7) at each iteration. However, to optimize the parameters of the output values \mathbf{f} , we replace (8)–(9) with a series of unconstrained optimization problems. At each iteration, we perform the following four steps to update these parameters:

- (i) update $\{\zeta_{n,o}\}_{n \in \mathcal{P}, o \in \mathcal{N}}$ using (10),
- (ii) optimize $\{\mu(f_n)\}_{n=1}^N$ using $\{\partial \mathcal{L}_q(\mathbf{f})/\partial \mu(f_n)\}_{n=1}^N$ and $\{\partial^2 \mathcal{L}_q(\mathbf{f})/\partial \mu(f_n)^2\}_{n=1}^N$,
- (iii) update $\{\zeta_{n,o}\}_{n \in \mathcal{P}, o \in \mathcal{N}}$ using (10),
- (iv) optimize $\{\Sigma(f_n)\}_{n=1}^N$ using $\{\partial \mathcal{L}_q(\mathbf{f})/\partial \Sigma(f_n)\}_{n=1}^N$ and $\{\partial^2 \mathcal{L}_q(\mathbf{f})/\partial \Sigma(f_n)^2\}_{n=1}^N$.

In the steps (ii) and (iv), we use minFunc Matlab package by Mark Schmidt, which uses a quasi-Newton strategy with limited-memory BFGS updates and is publicly available at <https://goo.gl/Vrd5DL>. Note that the step (iv) needs to be performed in the log-domain to ensure the non-negativity of the variance parameters.

3.3 AUC-Maximizing Linear Classifier Variant Using Listwise Ranking

We follow the same strategy to develop our linear classification variant with listwise ranking flavor and start by replacing (3) in our baseline Bayesian probit regression model with (2). Different from the pairwise ranking variant, we now need fewer auxiliary random variables, and the lower bound becomes

$$\mathcal{L}_q(\mathbf{f}) = \sum_{n=1}^N (\mathbb{E}_q[\log p(f_n|b, \mathbf{w}, \mathbf{x}_n)] - \mathbb{E}_q[\log q(f_n)])$$

$$+ \sum_{n \in \mathcal{P}} \mathbb{E}_q[\log p(z_n|\{f_o\}_{o \in \mathcal{W}_n})] + \text{const.},$$

where the first two terms are the same as before. The third term with the softmax function can again be lower bounded with a local variational approximation:

$$\mathbb{E}_q[\log p(z_n|\{f_o\}_{o \in \mathcal{W}_n})] = \mathbb{E}_q \left[\log \left(\frac{\exp(f_n)}{\sum_{o \in \mathcal{W}_n} \exp(f_o)} \right) \right]$$

$$= \mathbb{E}_q[f_n] - \mathbb{E}_q \left[\log \left(\sum_{o \in \mathcal{W}_n} \exp(f_o) \right) \right]$$

$$\geq \mathbb{E}_q[f_n] - \left(\sum_{o \in \mathcal{W}_n} \frac{1}{\zeta_n} \mathbb{E}_q[\exp(f_o)] - 1 + \log(\zeta_n) \right),$$

where $\{\zeta_n\}_{n \in \mathcal{P}}$ is the set of variational parameters introduced. We expect the lower bound approximation in the listwise setting with

significantly fewer variational parameters to be much tighter than that of the pairwise setting.

We can again write the lower bound as a function of the mean, covariance, and variational parameters, and optimize them separately.

3.3.1 Optimizing $\mathcal{L}_q(\mathbf{f})$ with respect to $\{\zeta_n\}_{n \in \mathcal{P}}$

Given $\{\mu(f_n)\}_{n=1}^N$ and $\{\Sigma(f_n)\}_{n=1}^N$, the optimal values for $\{\zeta_n\}_{n \in \mathcal{P}}$ can be found as

$$\zeta_n^* = \sum_{o \in \mathcal{W}_n} \exp\left(\mu(f_o) + \frac{\Sigma(f_o)}{2}\right). \quad (11)$$

3.3.2 Derivatives of $\mathcal{L}_q(\mathbf{f})$ with respect to $\{\mu(f_n)\}_{n=1}^N$

Given $\{\Sigma(f_n)\}_{n=1}^N$ and $\{\zeta_n\}_{n \in \mathcal{P}}$, we can find the first and second derivatives of the lower bound with respect to $\{\mu(f_n)\}_{n=1}^N$ as follows:

$$\frac{\partial \mathcal{L}_q(\mathbf{f})}{\partial \mu(f_n)} = -\mu(f_n) + \mathbb{E}_q[\mathbf{w}^\top \mathbf{x}_n + b] + \delta(y_n = +1)$$

$$- \sum_{o \in \mathcal{B}_n} \frac{1}{\zeta_o} \exp\left(\mu(f_n) + \frac{\Sigma(f_n)}{2}\right),$$

$$\frac{\partial^2 \mathcal{L}_q(\mathbf{f})}{\partial \mu(f_n)^2} = -1 - \sum_{o \in \mathcal{B}_n} \frac{1}{\zeta_o} \exp\left(\mu(f_n) + \frac{\Sigma(f_n)}{2}\right),$$

where \mathcal{B}_n is \mathcal{P} if $y_n = -1$ and $\{n\}$ otherwise.

3.3.3 Derivatives of $\mathcal{L}_q(\mathbf{f})$ with respect to $\{\Sigma(f_n)\}_{n=1}^N$

Given $\{\mu(f_n)\}_{n=1}^N$ and $\{\zeta_n\}_{n \in \mathcal{P}}$, we can find the first and second derivatives of the lower bound with respect to $\{\Sigma(f_n)\}_{n=1}^N$ as follows:

$$\frac{\partial \mathcal{L}_q(\mathbf{f})}{\partial \Sigma(f_n)} = -\frac{1}{2} - \sum_{o \in \mathcal{B}_n} \frac{1}{2\zeta_o} \exp\left(\mu(f_n) + \frac{\Sigma(f_n)}{2}\right) + \frac{1}{2\Sigma(f_n)},$$

$$\frac{\partial^2 \mathcal{L}_q(\mathbf{f})}{\partial \Sigma(f_n)^2} = -\sum_{o \in \mathcal{B}_n} \frac{1}{4\zeta_o} \exp\left(\mu(f_n) + \frac{\Sigma(f_n)}{2}\right) - \frac{1}{2\Sigma(f_n)^2}.$$

Different from our pairwise variant, we now need to update $\{\zeta_n\}_{n \in \mathcal{P}}$ using (11) in the steps (i) and (iii) during the variational inference.

4 EXPERIMENTS

To illustrate the effectiveness of our AUC-maximizing variants with pairwise ranking (BAM_P) and with listwise ranking (BAM_L), we report their results on four biomedical data sets (i.e., two cancer and two HIV data sets) and compare them to the error-minimizing baseline algorithm (i.e., Bayesian probit regression; BPROBIT). We implement these three algorithms in Matlab, and our implementations are publicly available at <https://goo.gl/DYh7ZR>.

We use AUROC and AUPR values from repeated random subsampling validation experiments to compare the classification performance of the algorithms. For each data set, we create 100 random train/test splits to obtain robust results. For each replication, the training set is defined by randomly selecting 75% of the data points with stratification on the phenotype, and the remaining 25% of the samples are used as the test set. The training set is normalized to have

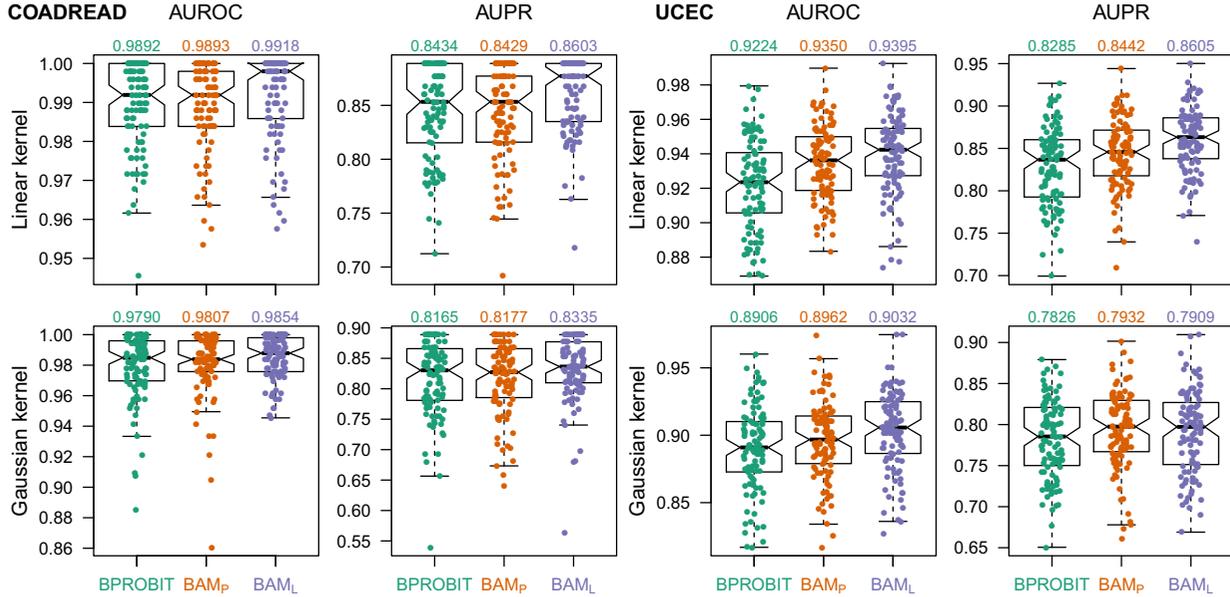


Figure 3. Classification results on two cancer data sets. The box-and-whisker plots show the results of the error-minimizing baseline algorithm (BPROBIT), our AUC-maximizing variant with pairwise ranking (BAM_P), and our AUC-maximizing variant with listwise ranking (BAM_L) over 100 replications in repeated random subsampling validation experiments. The numbers above the figures give the average performance values for each experiment.

zero mean and unit standard deviation, and the test set is then normalized using the mean and the standard deviation of the original training set.

Owing to the high dimensional inputs in our applications, we represent data points using an empirical kernel map, i.e., replacing \mathbf{x}_n with $[k(\mathbf{x}_1, \mathbf{x}_n) \dots k(\mathbf{x}_N, \mathbf{x}_n)]^\top$, which is the main idea behind relevance vector machines [18]. This step reduces the dimensionality of the input space from D to N . We perform experiments with the linear and Gaussian kernels, where we normalize the linear kernel to have unit diagonal entries, and the kernel width of the Gaussian kernel is selected as the average pairwise distance between the training instances.

The hyper-parameter values are selected as $(\alpha_\gamma, \beta_\gamma) = (1, 1)$ and $(\alpha_\eta, \beta_\eta) = (1, 1)$ for all algorithms, and $\nu = 1$ for BPROBIT. We perform at most 200 iterations or stop when the improvement in the lower bound between successive iterations is less than 0.001% during variational inference.

4.1 Classification Results on Cancer Data Sets

Micro-satellite instability is a hypermutable phenotype caused by the loss of DNA mismatch repair activity. It is frequently observed in several tumor types such as colorectal, endometrial, gastric, ovarian, and sebaceous carcinomas [19]. Tumors with micro-satellite instability do not respond to chemotherapeutic strategies developed for micro-satellite stable tumors, leading to its clinical importance. That is why we address the problem of predicting micro-satellite instability status of cancer patients from their gene expression data. We use two publicly available data sets provided by the Cancer Genome Atlas (TCGA) consortium: (i) colon and rectum adenocarcinoma (COADREAD) patients [16] and (ii) uterine corpus endometrial carcinoma (UCEC) patients [17].

The phenotype values of cancer patients for both data sets are downloaded from the TCGA website (<https://tcga-data.nci.nih.gov>), which groups the patients into three categories: (i) micro-satellite instability high (MSI-H), (ii) micro-satellite insta-

bility low (MSI-L), and (iii) micro-satellite stable (MSS). The pre-processed genomic characterizations of primary tumors from the patients (i.e., mRNA gene expression) are downloaded from <http://dx.doi.org/10.7303/syn300013>, where 20530 normalized gene expression intensities are provided for each profiled primary tumor. We remove the patients with missing phenotype value or genomic data from further analysis. At the end, there are 261 (37 MSI-H, 43 MSI-L, and 181 MSS) and 330 (108 MSI-H, 27 MSI-L, and 195 MSS) patients with available phenotype value and genomic data for COADREAD and UCEC data sets, respectively. We run binary classification experiments to separate MSI-H patients from others (i.e., MSI-L and MSS), which is in agreement with the earlier studies that combine MSI-L and MSS tumors into the same group [19].

Figure 3 compares the performance of the baseline algorithm and our two variants on two cancer data sets in terms of AUROC and AUPR over 100 replications, and reports the average AUROC and AUPR values for each experiment. We clearly see that our AUC-maximizing variants obtain better classification results than the error-minimizing baseline in most scenarios (and comparable results in few cases). Note that our listwise variant BAM_L obtains better AUROC values than our pairwise variant BAM_P in all scenarios, possibly owing to its tighter lower bound approximation.

4.2 Classification Results on HIV Data Sets

Predicting the effect of a drug using pretreatment genomic information is a current computational challenge in modern medicine. For example, HIV Drug Resistance Database (HIVDB) contains phenotype (i.e., drug susceptibility results) and genotype (i.e., amino acid sequences) information about HIV-1 [14], which is publicly available at <http://hivdb.stanford.edu>. On HIVDB, we address the problem of predicting drug susceptibility of reverse transcriptase sequences obtained from HIV patients using the genotype information. We extract all sequences originated from subtype B strains and treated with Zalcitabine (known as DDC) or Emtricitabine (known

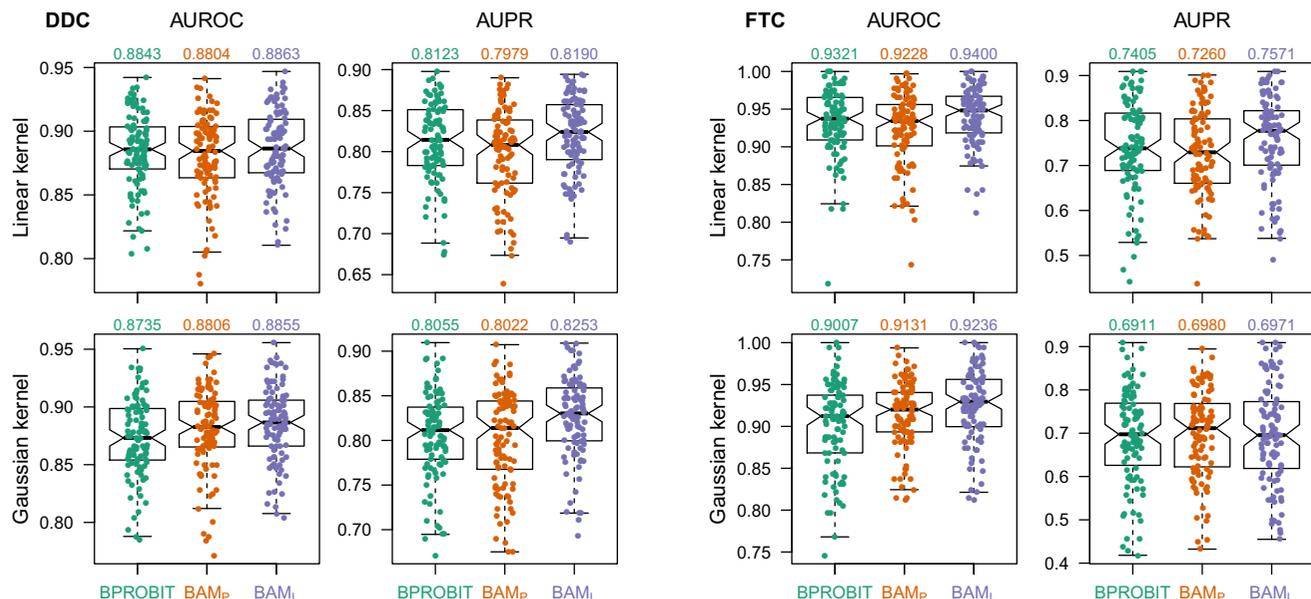


Figure 4. Classification results on two HIV data sets. The box-and-whisker plots show the results of the error-minimizing baseline algorithm (BPROBIT), our AUC-maximizing variant with pairwise ranking (BAM_p), and our AUC-maximizing variant with listwise ranking (BAM_L) over 100 replications in repeated random subsampling validation experiments. The numbers above the figures give the average performance values for each experiment.

as FTC). We remove the sequences with no phenotype or genotype information, leading to two final data sets with 472 (174 susceptible and 298 resistant) and 165 (46 susceptible and 119 resistant) sequences for DDC and FTC, respectively.

We use drug susceptibility results measured using the PhenoSense method for these two nucleoside analogs. Drug susceptibility results are given as fold change:

$$\text{IC}_{50} \text{ ratio} = \frac{\text{IC}_{50} \text{ of an isolate}}{\text{IC}_{50} \text{ of a standard wild-type control isolate}},$$

where IC₅₀ of a resistant or wild-type control isolate gives its half maximal inhibitory concentration. We label sequences as “resistant” or “susceptible” using drug-specific cutoff values as done similarly in the earlier studies [8, 15]. The cutoff is set to 1.5 for DDC and to 3.0 for FTC. For each reverse transcriptase, genotype information is extracted from the amino acid sequence of positions 1–240. Amino acid differences from the subtype B consensus wild-type sequence are considered as mutations. There are 856 and 520 unique mutations for DDC and FTC, which means that sequences can be represented as 856- or 520-dimensional binary vectors.

Figure 4 compares the performance of the baseline algorithm and our two variants on two HIV data sets. We see that our listwise variant BAM_L improves AUROC and AUPR values compared to the baseline algorithm in all scenarios, whereas our pairwise variant BAM_p can consistently improve AUROC values only with the Gaussian kernel but not with the linear kernel. Similar to the results on cancer data sets, our listwise variant BAM_L shows better performance than our pairwise variant BAM_p in all scenarios.

5 DISCUSSION

We introduce a novel Bayesian inference framework to optimize AUC measures in Bayesian hierarchical models. This full-Bayesian treatment is made possible by borrowing the pairwise and listwise ranking ideas from the information retrieval literature. To showcase our framework, we develop two linear classification algorithms by

modifying Bayesian probit regression model and derive their variational inference procedures. We then illustrate the practical importance of our framework on four biomedical data sets by validation experiments. These results show that our algorithms can obtain better AUROC and AUPR values compared to the baseline error-minimizing algorithm.

To bound the softmax function, we use a simple local variational approximation with a linear Taylor expansion of the log function [2]. An interesting topic for future research is to replace this bound with a much tighter bound such as the one proposed by [12] to further improve the generalization performance of our framework.

REFERENCES

- [1] James H. Albert and Siddhartha Chib, ‘Bayesian analysis of binary and polychotomous response data’, *Journal of the American Statistical Association*, **88**(422), 669–679, (1993).
- [2] David M. Blei and John D. Lafferty, ‘A correlated topic model of science’, *The Annals of Applied Statistics*, **1**(1), 17–35, (2007).
- [3] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, et al., ‘Learning to rank using gradient descent’, in *Proceedings of the 22nd International Conference on Machine Learning*, (2005).
- [4] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li, ‘Learning to rank: From pairwise approach to listwise approach’, in *Proceedings of the 24th International Conference on Machine Learning*, (2007).
- [5] Corinna Cortes and Mehryar Mohri, ‘AUC optimization and error rate minimization’, in *Advances in Neural Information Processing Systems 16*, (2003).
- [6] Yoav Freund, Raj Iyer, Robert E. Schapire, and Yoram Singer, ‘An efficient boosting algorithm for combining preferences’, *Journal of Machine Learning Research*, **4**(Nov), 933–969, (2003).
- [7] James A. Hanley and Barbara J. McNeil, ‘The meaning and use of the area under a receiver operating characteristic (ROC) curve’, *Radiology*, **143**(1), 29–36, (1982).
- [8] Dominik Heider, Robin Senge, Weiwei Cheng, and Eyke Hüllermeier, ‘Multilabel classification for exploiting cross-resistance information in HIV-1 drug resistance prediction’, *Bioinformatics*, **29**(16), 1946–1952, (2013).
- [9] Alan Herschtal and Bhavani Raskutti, ‘Optimising area under the ROC curve using gradient descent’, in *Proceedings of the 21st International Conference on Machine Learning*, (2004).

- [10] Thorsten Joachims, 'Optimizing search engines using clickthrough data', in *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (2002).
- [11] Thorsten Joachims, 'A support vector for multivariate performance measures', in *Proceedings of the 22nd International Conference on Machine Learning*, (2005).
- [12] David A. Knowles and Tom Minka, 'Non-conjugate variational message passing for multinomial and binary regression', in *Advances in Neural Information Processing Systems 24*, (2011).
- [13] Yanyan Lan, Tie-Yan Liu, Zhiming Ma, and Hang Li, 'Generalization analysis of listwise learning-to-rank algorithms', in *Proceedings of the 26th International Conference on Machine Learning*, (2009).
- [14] Soo-Yon Rhee, Matthew J. Gonzales, Rami Kantor, Bradley J. Betts, Jaideep Ravela, et al., 'Human immunodeficiency virus reverse transcriptase and protease sequence database', *Nucleic Acids Research*, **31**(1), 298–303, (2003).
- [15] Soo-Yon Rhee, Jonathan Taylor, Gauhar Wadhera, Asa Ben-Hur, Douglas L. Brutlag, et al., 'Genotypic predictors of human immunodeficiency virus type 1 drug resistance', *Proceedings of the National Academy of Sciences of the United States of America*, **103**(46), 17355–17360, (2006).
- [16] The Cancer Genome Atlas Network, 'Comprehensive molecular characterization of human colon and rectal cancer', *Nature*, **487**(7407), 330–337, (2012).
- [17] The Cancer Genome Atlas Research Network, 'Integrated genomic characterization of endometrial carcinoma', *Nature*, **497**(7447), 67–73, (2013).
- [18] Michael E. Tipping, 'Sparse Bayesian learning and the relevance vector machine', *Journal of Machine Learning Research*, **1**(Jun), 211–244, (2001).
- [19] Eduardo Vilar and Stephen B. Gruber, 'Microsatellite instability in colorectal cancer — The stable evidence', *Nature Reviews Clinical Oncology*, **7**(3), 153–162, (2010).
- [20] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li, 'Listwise approach to learning to rank - Theory and algorithm', in *Proceedings of the 25th International Conference on Machine Learning*, (2008).
- [21] Lian Yan, Robert Dodier, Michael C. Mozer, and Richard Wolniewicz, 'Optimizing classifier performance via an approximation to the Wilcoxon-Mann-Whitney statistics', in *Proceedings of the 20th International Conference on Machine Learning*, (2003).

Validating Cross-Perspective Topic Modeling for Extracting Political Parties' Positions from Parliamentary Proceedings

Janneke M. van der Zwaan¹ and Maarten Marx and Jaap Kamps²

Abstract.

In the literature, different topic models have been introduced that target the task of viewpoint extraction. Because, generally, these studies do not present thorough validations of the models they introduce, it is not clear in advance which topic modeling technique will work best for our use case of extracting viewpoints of political parties from parliamentary proceedings. We argue that the usefulness of methods like topic modeling depend on whether they yield valid and reliable results on real world data. This means that there is a need for validation studies. In this paper, we present such a study for an existing topic model for viewpoint extraction called cross-perspective topic modeling [11]. The model is applied to Dutch parliamentary proceedings, and the resulting topics and opinions are validated using external data. The results of our validation show that the model yields valid topics (content and criterion validity), and opinions with content validity. We conclude that cross-perspective topic modeling is a promising technique for extracting political parties' positions from parliamentary proceedings. Second, by exploring a number of validation methods, we demonstrate that validating topic models is feasible, even without extensive domain knowledge.

1 Introduction

Over the last fifteen years, topic modeling has been established as a method for uncovering hidden structure in document collections. Traditionally, these methods learn probability distributions over words along a single dimension of topicality, and do not take into account other dimensions, such as sentiment, perspective, or theme [20]. More recently, various extensions to topic modeling have been proposed that do allow for multiple dimensions. In our work, we are interested in opinion or viewpoint extraction from text.

Different topic models have been proposed that target the task of viewpoint extraction, each of which is based on different assumptions about what an opinion or viewpoint consists of, and impose different requirements on the data. Section 2 provides an overview of these methods, and their particularities. Studies introducing new topic models usually only

evaluate model fit based on held-out perplexity and provide anecdotal evidence that the results make sense. Systematic validations of new algorithms are rare. As a result of this, it is not clear in advance which topic modeling technique will work best for our use case of extracting viewpoints of political parties from parliamentary proceedings.

It can be argued that the usefulness of methods like topic modeling depend on whether they yield valid and reliable results on real world data. In order for researchers from other domains to trust and use methods such as topic modeling, insight into the strengths and limitations of individual techniques is indispensable. In domains applying these text mining techniques, e.g., political science, the need for validation is already recognized [14]. However, from a computer science perspective, validation of methods is also important, because insight into why methods are successful or not can help to improve existing methods and inform the design of new methods.

This paper presents a validation of one particular topic model for viewpoint extraction: the cross-perspective topic model [11]. The cross-perspective topic model learns topics and opinions from a corpus that is (or can be) divided in perspectives (e.g., political parties). Topics are learned from topic words (nouns) in the entire corpus, whereas opinions are learned from opinion words (adjectives) for each perspective separately. The model is applied to Dutch parliamentary proceedings from 1999 to 2012. The Dutch multiparty political system allows us to apply cross-perspective topic modeling to data consisting of more than just two or three perspectives. We demonstrate that cross-perspective topic modeling yields valid topics. While opinions extracted from the parliamentary proceedings are representative of the political parties' positions, these positions were uncorrelated with classical left/right rankings of the parties. These results indicate that cross-perspective topic modeling is a promising technique for extracting political parties' positions from parliamentary proceedings.

This paper is organized as follows. Section 2 introduces the related work by reviewing existing topic models for viewpoint extraction. In section 3, we provide an in-depth explanation of cross-perspective topic modeling. Section 4 presents the design of the validation study. Topic and opinion validity are assessed in section 5. The results are discussed in section 6. Finally, we present our conclusions in section 7.

¹ Netherlands eScience Center, The Netherlands, email: j.vanderzwaan@esciencecenter.nl

² University of Amsterdam, The Netherlands, email: {maartenmarx, j.kamps}@uva.nl

2 Related Work

Opinion mining or sentiment analysis is concerned with extracting subjective information from text. Pang and Lee provide a broad introduction to this diverse research area [19]. This section provides a review of topic models for opinion or viewpoint extraction, and discusses the need for validation studies.

2.1 Topic Models for Viewpoint Extraction

The Joint Topic and Perspective (JPT) model assumes lexical variation in ideological text can be attributed to the topic and the author's point of view [18]. Consequently, for each word in the vocabulary two weights are learned: a topical and an ideological weight. The model needs a collection of texts on the same topic that is divided into two contrasting perspectives. In essence, this method learns a single topic per perspective; so, there are two topics in total.

The Joint Topic Viewpoint (JTV) model proposed by Traibelsi and Zaïane assumes documents contain expressions of one or more divergent viewpoints [25]. Each term in a document is assigned a topic and a viewpoint. The model generates a probability distribution over terms for each topic-viewpoint pair. This means that 'objective' (i.e., substantive) and 'subjective' (i.e., viewpoint-specific) information are mixed. Although, theoretically, the number of perspectives can be > 2 , for the experiments presented in the paper it is set to 2 (i.e., the viewpoints are 'supporting' and 'opposing').

Gottipati et al. propose a topic model to infer topics and positions (pro/con) by exploiting the hierarchical structure in which debates are organized on Debatepedia³ [12]. The model learns to classify terms either as named entities, 'general' position terms, 'topic-specific' position terms, 'topic' terms, or 'background' terms. The positions are limited to pro and con.

The Topic-Aspect (TAM) model was designed to capture a text's underlying perspectives [20]. In addition to a mixture component to filter 'background' words, the model assigns words either to an aspect-neutral or aspect-dependent distribution. This means 'objective' and aspect-specific information is separated. The number of aspects (perspectives) is a parameter of the model. The model learns the perspectives from the data, so documents do not need to be labeled; in fact, it is assumed that documents contain mixtures of perspectives.

The Viewpoint and Opinion Discovery Unification Model (VODUM) uses heuristics to learn topics, viewpoints, and opinions from text [23]. A viewpoint is defined as a standpoint on a set of topics, and an opinion is a wording that is specific to a topic and a viewpoint. VODUM separates topic and opinion words based on their part of speech tag. In addition, words in the same sentence are assumed to belong to the same topic, and all text in a document is assumed to belong to the same viewpoint. These constraints help to improve model fit.

Generally, topic models for viewpoint extraction target slightly differing tasks, ranging from finding arguments or evidence for or against standpoints to discovering words indicative of viewpoints or perspectives, and are usually designed to exploit characteristics of the particular data used

(e.g., document structure as with documents from Debatepedia). Although work that presents new viewpoint extraction topic models includes evaluations of the results produced by these models, the evaluations typically are limited. Quantitative evaluations usually assess model fit, based on held-out perplexity (i.e., [11, 18, 23, 25]). However, it is not surprising that topic models that exploit particularities of the data and/or tasks, generally result in models with a better fit to the data. Also, evaluations of model fit do not take into account the extent to which topics and viewpoints learned from data make sense at all. In fact, topics from a model with lower perplexity are not necessarily more meaningful to humans than topics from a model with higher perplexity [10]. We argue that in order to gain insight into the contents of document collections, topic modeling results must be semantically meaningful to humans. Therefore, it is necessary to evaluate performance in other ways than just calculating perplexity. In this study, we use additional, domain specific data to validate topic modeling results.

Qualitative evaluations of topic modeling results presented in the related work are anecdotal, and do not go beyond presenting anecdotal results for topics and opinions/viewpoints (i.e., [11, 12, 20, 23, 25]). Again we contend that more thorough evaluations are required in order for the results to be useful representations of documents in collections.

This paper presents a validation study of the results of cross-perspective topic modeling [11]. There were multiple reasons to select this method and not one of the others. The cross-perspective topic model yields explicit representations of viewpoints, which allows us to quantify differences between the viewpoints. This is helpful for representations that can be compared directly to external data. Also, the CPT model allows for more than two perspectives on a topic, whereas many other models aim to learn pro/con stances towards topics only. Although this results in a counterintuitive notion of what an opinion is (i.e., probability distribution over words vs. arguments for or against), it allows for a more nuanced representation of opinions and differences between opinions. Finally, CPT is conceptually simpler than some of the other models, which makes it easier to implement.

2.2 Assessing Validity

Grimmer and Stewart note that 'all automated [text mining] methods are based on incorrect models of language' [14]. While this does not imply that the results of these methods are therefore useless, it does mean that output of automated text mining methods must be validated before their results can be trusted. In the introduction, we argued that validation studies are relevant for both domain scientists that use these methods to explore text corpora, and computer scientists that work on developing new text mining methods and improving existing ones.

Validity refers to the extent to which a measure measures what it is intended to measure [9]. Table 1 lists different types of validity. According to Grimmer and Stewart, 'to validate the output of an unsupervised method, scholars must combine experimental, substantive, and statistical evidence to demonstrate that the measures are as conceptually valid as measures from an equivalent supervised model' [14].

³ <http://www.debatepedia.org/>

Type	Description
Face	The extent to which results appear to be valid.
Content	The extent to which a method for measuring a latent construct represents all of its facets.
Criterion	Correlation between a measure and other measures that reflect the same concept.
Construct	The extent to which a measure behaves as expected in a theoretical context.

Table 1: Types of validity (adapted from [9]).

Quinn et al. performed a validation study of topic modeling on speech in the US Senate [21]. They show that, with the exception of ‘procedural’ topics, words from the same topic generally have common substantive meaning, that hierarchical clusterings of topics yield meaningful semantic relationships between topics, that topics found in speeches correlate with roll-calls and hearings, and that there is correlation between topics found in speeches and exogenous events. However, the different steps of the validation are mostly qualitative. In addition, the topic modeling method used is simpler than cross-perspective topic modeling; a speech can be assigned to a single topic only, and opinions are not taken into account.

In this paper, we address content validity and criterion validity of topics and opinions extracted using cross-perspective topic modeling.

3 Cross-Perspective Topic Modeling

The cross-perspective topic model [11] is an extended form of Latent Dirichlet Allocation (LDA) [6]. Topics are learned by doing LDA on the topic words (nouns) in the corpus. Opinions are learned from a separate LDA process using opinion words (adjectives, verbs, and adverbs). A topic is a probability distribution over topic words. An opinion is a probability distribution over opinion words. While the topics are shared among the entire corpus, opinions depend on the perspective a document belongs to. A document can only belong to a single perspective, and the division of the corpus in perspectives is fixed and must be known in advance.

The imaginary process for generating documents is: one first selects a topic, based on the topic mixture of that document. Then a topic word is drawn from the topic. This procedure is repeated until all topic words have been selected. Next, one selects an opinion based on the frequency of topic words associated with the topics in the document. The more words associated with a certain topic, the higher the chance that the corresponding opinion will be selected. The contents of the opinion (i.e., probabilities of opinion words) depend on the generator’s perspective. Next, an opinion word is drawn from the selected opinion. This procedure is again repeated until all opinion words have been selected. More formally, this generative process can be described as follows.

1. Draw a perspective-independent multinomial topic word distribution ϕ from $\text{Dirichlet}(\beta)$ for each topic z
2. Draw a perspective-specific multinomial opinion word distribution ϕ_o^i from $\text{Dirichlet}(\beta_o^i)$ for each opinion x for perspective c^i
3. For each document d choose a topic mixture θ from $\text{Dirichlet}(\alpha)$

4. For each topic word w in document d
 - (a) Draw a topic z from $\text{Multinomial}(\theta)$
 - (b) Draw a word w from $\text{Multinomial}(\phi)$ conditional on z
5. For each opinion word o in document $d \in c^i$
 - (a) Draw an opinion x from $\text{Uniform}(z_{w_1}, z_{w_2}, \dots, z_{w_{N_w(d)}})$
 - (b) Draw an opinion word o^i from $\text{Multinomial}(\phi_o^i)$ conditional on x^i

There are $2+C$ parameters that need to be estimated for the cross-perspective topic model: the document-topic distribution θ , the topic-word distribution ϕ , and, for every perspective, the opinion-word distribution ϕ_o^c . As Fang et al. [11], we chose to implement a Gibbs sampler to estimate the parameters [13]. Gibbs sampling is a type of Markov Chain Monte Carlo (MCMC) algorithm [1]. MCMC algorithms aim to construct a Markov chain that have the target posterior as the stationary distribution. In Gibbs sampling, new assignments of variables are sequentially sampled by drawing from the distributions conditioned on the current values of all other variables. To estimate parameters of opinions, additional Markov chains are introduced to simulate the generation of opinions. The sampling equations of the topic variable z for each topic word w_i is as follows. The notation used in these equations is explained in table 2.

$$p(z_i = k | w_i = v, \mathbf{z}_{-i}, \mathbf{w}_{-i}, \alpha, \beta) \propto \frac{n_{k(d)-i} + \alpha}{\sum_{k=1}^K n_{k(d)-i} + K\alpha} \times \frac{n_{v(k)-i} + \beta}{\sum_{v=1}^V n_{v(k)-i} + V\beta}$$

The sampling equation of opinion variable x^c for each opinion word o_i is

$$p(x_i^c = s | o_i = r, \mathbf{x}_{-i}^c, \mathbf{o}_{-i}, \beta_o) \propto \frac{n_{r(s)-i} + \beta_o^c}{\sum_{r=1}^T n_{r(s)-i} + T\beta_o^c} \times \frac{n_{s(d)}}{N_{w(d)}}$$

For every sample thus obtained, the relevant parameters are estimated using the following equations:

$$\theta_{kd} = \frac{n_{k(d)} + \alpha}{\sum_{k=1}^K n_{k(d)} + K\alpha}$$

$$\phi_{vk} = \frac{n_{v(k)} + \beta}{\sum_{v=1}^V n_{v(k)} + V\beta} \quad \phi_{rs}^c = \frac{n_{r(s)} + \beta_o^c}{\sum_{r=1}^T n_{r(s)} + T\beta_o^c}$$

Our implementation of a Gibbs sampler for cross-perspective topic modeling is available online⁴.

4 Study Design

As mentioned in section 2, most existing work on topic models for viewpoint extraction only addresses ‘face validity’ and model fit. This paper assesses content validity and criterion validity of topics and opinions extracted using cross-perspective topic modeling. In order to do so, we need to determine to what extent topics and opinions correspond to political subjects and political parties’ ideological positions.

⁴ <https://github.com/NLeSC/cptm>

Symbol	Description
w, o	Topic word and opinion word
d, v, r, k, s, c	Variable instances; d for document, v for topic word, r for opinion word, k for topic, s for opinion, c for perspective
D, K, C	The number of documents, topics, and perspectives
V, T	The size of the topic and opinion vocabulary
\mathbf{z}, \mathbf{x}	Topic and opinion
$\mathbf{w}_{-i}, \mathbf{z}_{-i}, \mathbf{o}_{-i}$	The vector values of \mathbf{w}_i , \mathbf{z}_i , and \mathbf{o}_i on all dimensions except i
$N_{w(d)}$	The number of topic words in document d
$n_{k(d)[-i]}$	The number of times topic k has occurred in document d [except the current instance]
$n_{v(k)[-i]}$	The number of times word v is assigned to topic k [except the current instance]
$n_{r(s)[-i]}$	The number of times word r is assigned to opinion s [except the current instance]
$n_{s(d)}$	The number of times opinion s occurs in document d
θ	$D \times K$ matrix containing the document-topic distribution
ϕ	$K \times V$ matrix containing the topic-word distribution
ϕ_o^c	$K \times T$ matrix containing the opinion-word distribution for perspective c

Table 2: Notation used in the cross-perspective topic model (adapted from [11]).

This section presents the design of our validation study. After introducing the research questions, we provide a description of the dataset and experiments.

4.1 Research Questions

To assess validity of the topics, the following research questions need to be answered.

- Do the topics learned from the parliamentary proceedings cover all relevant political subjects? (content validity)
- Can the topics learned from the parliamentary proceedings be used to predict the political subject of texts? (criterion validity)

To assess content validity of the topics, we need to determine whether the topics extracted from the data cover all relevant political subjects. As a set of ‘all relevant political subjects’, we use the Comparative Agendas Project (CAP) main coding categories [5]. The 21 CAP main coding categories are listed in table 4. We check whether there is at least one topic covering each CAP coding category by training a text classifier on manually coded data, and use this classifier to predict CAP codes for the topics extracted from our data. To assess criterion validity, we apply our topic models to the manually coded data, and use the results to predict CAP codes. Performance of this ‘classifier’ is compared to two other text classifiers; a Naive Bayes classifier and Support Vector Machine (SVM). To make it a fair comparison, the classifiers are trained using the topic words (nouns) only.

To assess validity of the opinions, the following research questions are addressed.

- Are party opinions learned from the parliamentary proceedings representative of party manifestos? (content validity)

- Is there substantial correlation between party rankings generated from the opinions and rankings from domain experts? (criterion validity)

To assess content validity of the opinions, we need to determine whether the topics and associated opinions are representative of a party’s ideological position. In order to do so, we estimate opinion word perplexity of party manifestos⁵. We can conclude the opinions have construct validity if there is a correspondence between the perspective that has lowest perplexity for a manifesto and the political party that published it. With regard to criterion validity, we test whether we can use the parties’ opinions to rank them on a left/right scale. As a gold standard of the left/right scale we use expert rankings of political party ideology from the Chapel Hill Expert Survey (CHES) [4]. To generate rankings from our data, we apply principal components analysis (PCA) to the parties’ opinions and project them on the first few principal components. We can conclude the opinions have criterion validity if there is substantial correlation between these and the CHES rankings.

4.2 Data and Experiments

The data used for validity assessment consists of Dutch parliamentary proceedings from the House of Parliament and Senate between September 21, 1999 and September 11, 2012⁶. It contains 20,594 documents in total. Each document contains speeches that are tagged with a political party. These tags are used to divide the corpus in perspectives. For this study, we made two divisions of the data. In the first division, there is a perspective for each political party. The *parties* dataset consists of 11 perspectives. For the second division, *parties/time*, the data is divided by party and government term. This set contains 59 perspectives. The numbers of documents per perspectives for the two datasets are listed in table 3. Figure 1 presents a timeline of the government terms and the period covered by the dataset. The light blue lines represent the government terms, while the dark line represents the time period covered by the parliamentary proceedings.

Party name	<i>parties</i>	<i>parties/time</i>					
		K.II	B.I	B.II	B.III	B.IV	R.I
CDA	6416	1165	296	1715	240	1788	1212
CU	2783	332	138	673	77	828	735
D66	4151	941	236	1010	118	813	1033
GL	4960	986	262	1335	163	1176	1038
LPF	846	13	151	666	14	2	-
PvdA	6590	1134	300	1864	263	1685	1344
PvdD	667	-	-	-	20	285	362
PVV	2179	-	-	-	57	1105	1017
SGP	2669	531	139	767	120	712	400
SP	5506	611	199	1307	203	1867	1319
VVD	5976	1054	252	1552	227	1770	1121

Table 3: Number of documents per perspective in the *parties* and *parties/time* datasets.

⁵ Party manifestos were obtained through The Manifesto Project (MP) [17]. The MP provides manually coded versions of party manifestos. For the validation, we only used the texts.

⁶ <http://ode.politicalmashup.nl/data/summarise/folia/>

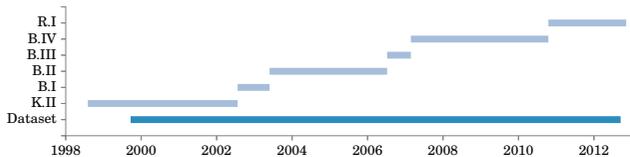


Figure 1: Timeline of period covered by the government terms and parliamentary proceedings dataset.

All text was part-of-speech (POS) tagged and lemmatized using Frog [26]. Nouns are saved as topic words. Because preliminary experiments showed that verbs and adverbs mostly add noise to the opinions, only adjectives are used as opinion words. The topic and opinion vocabularies were filtered. Terms occurring less than six times, the top 100 most frequent terms, and the top 100 terms that occur in the most documents were removed. The topic vocabulary contains 38,145 terms and the opinion vocabulary contains 6245 terms.

The cross-perspective topic model has $2 + C$ Dirichlet hyper parameters: α , β and β^i ; $i \in$ perspectives. α affects the number of topics found in a document (lower α leads to fewer topics per document), whereas β affects the number of words a topic consists of (lower β leads to topics with fewer words). Because the values of these parameters mostly affect the convergence of Gibbs sampling and not the results [13], we fix them to $\alpha = 50/K$, and $\beta = \beta^i = 0.02$ (cf. [13]). Based on previous experience with the Dutch parliamentary proceedings, the number of topics (K) is set to 100. Gibbs sampling is done for 200 iterations, and the final θ , topics, and opinions are calculated by taking the average of every tenth iteration starting from iteration 80.

5 Results

In this section, we answer the research questions formulated in section 4. Sections 5.1 and 5.2 address topic and opinion validity respectively.

5.1 Topic Validity

This section addresses content and criterion validity of the topics. For the assessment of content validity, topics are divided into two sets: high quality topics and low quality topics. Topic quality is determined by calculating topic coherence measure *NPMI* [7, 22] using the Dutch Wikipedia as a reference corpus [27]. Topics are considered to be of high quality if their *NPMI* score is above the mean.

5.1.1 Content Validity

In order to determine whether all political subjects are covered by our two sets of 100 topics, we train a text classifier that predicts CAP main categories based on manually coded data. The dataset we use are manually coded parliamentary

questions^{7 8} [24]. To train text classifiers, we selected the parliamentary questions texts from September 21, 1999 to September 11, 2012. One text that was not coded properly was removed. The resulting dataset consists of 834 texts. Table 4 lists the percentages of texts coded with each CAP main coding category. There are three CAP codes that do not occur in this dataset: 9, 18, and 23. These are excluded from analysis. Note that CAP codes 11 and 22 do not exist and are therefore also excluded.

CAP	Description	% texts
1	Domestic Macroeconomic Issues	3.36
2	Civil Rights, Minority Issues, and Civil Liberties	8.63
3	Health	10.55
4	Agriculture	2.88
5	Labor and Employment	9.47
6	Education	8.03
7	Environment	2.40
8	Energy	2.52
9	Immigration and Refugee Issues	0.00
10	Transportation	5.52
12	Law, Crime, and Family Issues	12.95
13	Social Welfare	7.19
14	Community Development and Housing Issues	4.08
15	Banking, Finance, and Domestic Commerce	3.48
16	Defense	3.36
17	Space, Science, Technology, and Communications	1.68
18	Foreign Trade	0.00
19	International Affairs and Foreign Aid	7.67
20	Government Operations	5.64
21	Public Lands, Water Management, and Territorial Issues	0.60
23	Cultural Policy Issues	0.00

Table 4: CAP main codes and percentages of texts in the parliamentary questions dataset.

Topic words (nouns) extracted from the texts were used to train two classifiers: a Naive Bayes classifier and SVM. Results reported in table 5 were obtained through 5-fold cross validation. Performance of the SVM is significantly better than performance of the Naive Bayes classifier (Welch's two-sided t-test, $p < 0.05$). Based on these results the SVM was selected to map topics to coding categories.

Text classification was performed on the top 10 topic words of our two sets of 100 topics. Figures 2 and 3 show the numbers of topics that were mapped to the different CAP coding categories for all topics and the high quality topics. The figures show that all CAP coding categories are covered by the topics.

⁷ In addition to the parliamentary questions data, there is a second dataset available: the Queen's speeches [8]. However, a text classification experiment using the Queen's speeches resulted in very low performance (Naive Bayes $F_1 \approx 0.06$; SVM $F_1 \approx 0.07$). This can be explained by the fact that the Queen's speeches data consists of coded sentences that are too short to learn anything from (20.59 words on average; std = 10.40). We therefore decided not to use the Queen's speeches data to assess content validity.

⁸ The parliamentary questions data has the disadvantage that the texts are also part of the parliamentary proceedings. However, given that there is no other suitable manually coded dataset available, and that these texts make up a very small part of the parliamentary proceedings, this data was used to assess content validity.

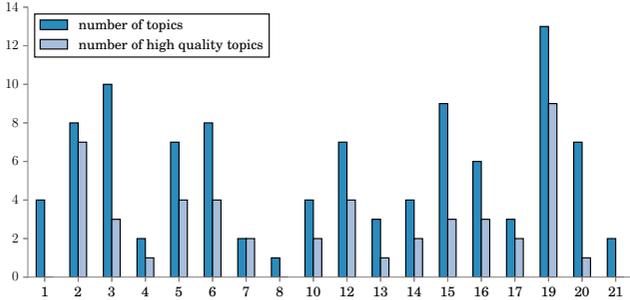


Figure 2: Number of topics and high quality topics from the *parties* dataset that is predicted for each main CAP code.

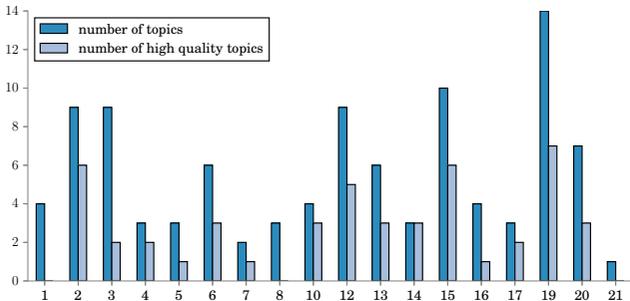


Figure 3: Number of topics and high quality topics from the *parties/time* dataset that is predicted for each main CAP code.

With regard to the high quality topics: the *parties* dataset does not contain high quality topics for CAP codes 1, 8, and 21; in the *parties/time* dataset there are no high quality topics for CAP code 21. As shown in table 4, these coding categories are relatively rare in the parliamentary questions data (0.60% - 3.36%). Based on these results, we conclude that topics extracted using cross-perspective topic modeling have content validity.

5.1.2 Criterion Validity

To assess criterion validity, we use the mapping between topics and CAP coding categories described in the previous section to predict CAP codes of the manually coded parliamentary questions texts. In order to do so, we estimate θ for the parliamentary questions texts using ϕ^{topic} obtained through the experiments. Then, the most important topic found for a text is mapped to a CAP category. Because the results of topic modeling are probabilistic, this procedure was repeated 10 times. Classification performance is calculated using accuracy and F_1 . Table 5 presents the results of the Naive Bayes classifier (baseline), SVM, and the topic model classifiers.

All differences in performance are statistically significant at $p < 0.05$, except the differences between the two topic model classifiers. The results for the topic model classifiers are higher than the results obtained with the Naive Bayes classifier (baseline) and lower than the performance of the SVM. When interpreting the results, it is important to keep in mind that the Naive Bayes and SVM are algorithms for supervised learning, whereas topic modeling is unsupervised.

	Accuracy	F_1
Naive Bayes	0.447 ± 0.026	0.377 ± 0.028
SVM	0.603 ± 0.030	0.585 ± 0.032
Topic models <i>parties</i>	0.537 ± 0.009	0.529 ± 0.008
Topic models <i>parties/time</i>	0.550 ± 0.008	0.548 ± 0.008

Table 5: Machine learning performance of parliamentary questions data

We are not so much interested in classifier performance per se, but instead investigate whether there is meaningful correlation between the most important topic in a text and manually assigned CAP codes. Because the SVM was used to map topics to CAP codes, its performance effectively is an upper bound for the performance of the topic model classifiers. The closer performance of the topic model classifiers is to performance of the SVM, the better. Based on the results found, we conclude that the topics have criterion validity.

5.2 Opinion Validity

This section addresses content and criterion validity of the opinions.

5.2.1 Content Validity

To assess content validity of the opinions, we use party manifestos as implicit, but complete representations of political parties' viewpoints on the topics they find most important. To assess whose opinion is expressed in party manifesto d , we need to calculate

$$\operatorname{argmax}_{i \in C} p(d|o^i)$$

Assuming equal probabilities for each perspective (political party), $p(d|o^i) \propto p(o^i|d)$. $p(o^i|d)$ is calculated as the opinion word perplexity for party manifesto document d :

$$\text{perplexity}(d) = \exp - \frac{\log(p(\mathbf{o}))}{N_o}$$

where

$$p(\mathbf{o}) = \prod_{i=1}^{N_o} \sum_{k=1}^K p(o_i|z_i = k)p(z_i = k|d)$$

In this equation \mathbf{o} is the set of opinion words in document d ; N_o is the number of opinion words in document d ; $p(o_i|z_i = k)$ is learned from the original experiments on the parliamentary proceedings; and $p(z_i = k|d)$ is estimated from the party manifestos using parameters estimated in the original experiments.

Dutch party manifestos from 2006 onwards are available in the Party Manifesto Project dataset. We downloaded manifestos for the elections in 2006, 2010, and 2012 for all parties except LPF, which is not present in the Party Manifesto Project dataset. Documents were subjected to the same pre-processing procedure as the parliamentary proceedings, and per document opinion word perplexity was calculated as described above.

As shown in table 6, the party with lowest perplexity for the *parties* dataset is correct for 66.67% of the party manifestos. The confusion matrix in figure 4 shows that mistakes

are made all over the political spectrum. First, the opinions of two confessional parties CDA and CU can't be distinguished. Also, SGP, which is a more conservative confessional party is confused with CU for one of the three manifestos. On the left side of the political spectrum, PvdD is confused with GL and/or SP, which are all left-wing parties. Parties closer to the center of the political spectrum, D66, VVD, and PvdA, are also confused. These results are in line with a common preconception of Dutch politics that although there is a multiparty system, the differences between individual parties are small.

	Accuracy
<i>parties</i>	0.667
<i>parties/time previous</i>	0.300
<i>parties/time next</i>	0.100
<i>parties/time parties</i>	0.567

Table 6: Accuracy of predicting the parties of party manifestos.

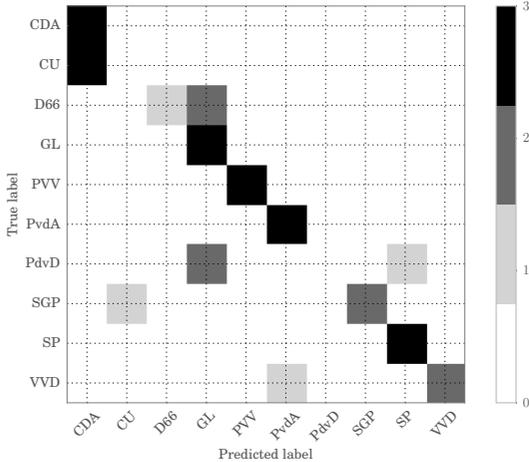


Figure 4: Confusion matrix of predictions for party manifestos based on perplexity calculated using parameters learned from the *parties* data.

For the *parties/time* data, the time parameter has to be taken into account. Because it is not clear in advance whether party manifestos represent a party's viewpoints of the previous or next government term, accuracy was calculated for both these possibilities. The results in table 6 show that political parties' opinions of the government term before an election are more similar to the party manifestos than the opinions of the next government term (accuracy of 0.300 and 0.100 respectively). Figure 5 shows the confusion matrix of the *parties/time previous* results. The black squares on the diagonal show that there is correlation between the actual party and government term and what is predicted. The results are distorted by two vertical lines, one over B.IV-CDA, the other over B.IV-PvdA. Showing the dominance of center parties (CDA and PvdA), and the government term B.IV, which is the longest government term in the time period we have party manifestos for (2006–2012). These results can be explained by

the fact that there are more documents available for government term B.IV than for B.III.

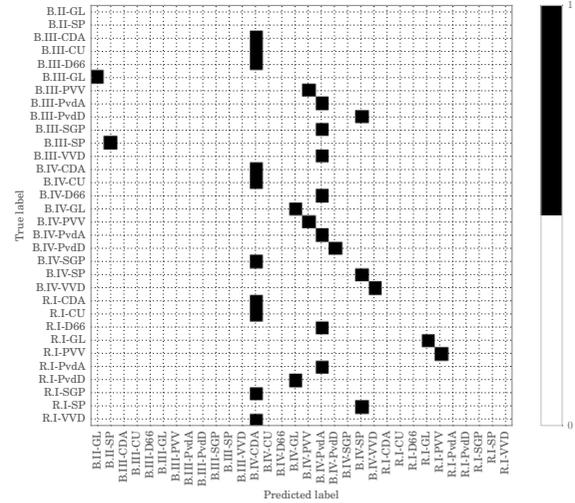


Figure 5: Confusion matrix of predictions for party manifestos based on perplexity calculated using parameters learned from the *parties/time* data.

When removing the time dimension from the predictions, accuracy increases to 0.567. This is slightly lower than accuracy for the *parties* opinion perplexity experiment. The confusion matrix is very similar to figure 4 and is not displayed. Based on the results found, we conclude that the opinions have content validity.

5.2.2 Criterion Validity

To assess criterion validity of the opinions, we use the opinions to generate rankings of perspectives and compare these rankings to rankings of political parties in the CHES dataset [4]. The CHES dataset is based on expert knowledge, and contains estimates of political party positions on different subjects, including European integration, ideology, and policy issues for national parties in different European countries. The survey is repeated every few years. For this study, we use data from 1999, 2002, 2006, and 2010. Traditionally, one of the most important scales political parties are ranked on is the left/right spectrum. The CHES dataset contains two variables that are relevant to this scale: *lrgen*, which measures ideological stance (left/right spectrum), and *lrecon*, which measures ideological stance on economic issues. Rankings generated from the opinions are compared to rankings based on these two variables.

Rankings of the different perspectives based on the opinions learned from the *parties* and *parties/time* data are generated by doing PCA on the opinions. For each dataset, we create rankings of perspectives by projecting the opinions on the first 5 principal components. CHES rankings for for *parties* are generated by averaging *lrgen* and *lrecon* over parties. For the *parties/time* data, years are mapped to government terms, and averaged over party/government term combinations.

To compare the rankings, we calculate Kendall's Tau [15] and Spearman's r [16]. The results for the *parties* dataset are presented in table 7. There are very few significant results.

We conclude that there is no linear relation between opinions from the *parties* and CHES *lrgen* or *lrecon*. Table 8 presents the results for the *parties/time* data. These results are very similar to the results for *parties*. Again we conclude that a linear relation between opinions from the *parties* and CHES *lrgen* or *lrecon* does not exist. This means that there is no criterion validity with regard to the left/right distinction.

	PC 1	PC 2	PC 3	PC 4	PC 5
Kendall's Tau					
<i>lrgen</i>	0.382	0.236	0.127	-0.273	0.127
<i>lrecon</i>	0.164	0.164	0.055	-0.636 [†]	-0.091
Spearman's r					
<i>lrgen</i>	0.364	0.136	-0.882 [†]	0.055	0.164
<i>lrecon</i>	0.209	0.173	-0.827 [†]	0.427	0.200

Table 7: Correlation between opinions learned from the *parties* dataset projected on the first five PCA components and CHES *lrgen* and *lrecon* ([†] statistically significant at $p < 0.05$).

	PC 1	PC 2	PC 3	PC 4	PC 5
Kendall's Tau					
<i>lrgen</i>	-0.191	0.094	0.037	-0.077	0.009
<i>lrecon</i>	-0.048	0.031	0.066	-0.060	0.140
Spearman's r					
<i>lrgen</i>	0.153	0.123	0.042	0.010	0.537 [†]
<i>lrecon</i>	0.093	0.045	0.076	0.038	0.643 [†]

Table 8: Correlation between opinions learned from the *parties/time* dataset projected on the first five PCA components and CHES *lrgen* and *lrecon* ([†] statistically significant at $p < 0.05$).

6 Discussion

In this paper, we explore a number of validation methods, and demonstrate that validating the results of topic modeling is feasible, even without extensive domain knowledge. The results of our study reveal that cross-perspective topic modeling is a promising technique for extracting political parties' positions from parliamentary proceedings. We have shown that the topics have content and criterion validity, and the opinions have content validity. However, in order to be able to apply cross-perspective topic modeling, the data must be divided (or at least dividable) into perspectives. Because not all datasets meet these requirements, the CPT model certainly does not solve all viewpoint extraction tasks.

For criterion validity of the opinions, we tried to use opinions to rank political parties on a left/right scale. The results indicate that the differences between the opinions of political parties are more complicated. There are two possible solutions to this problem. First, there might be other valid domain-specific interpretations of the rankings generated by applying PCA to the opinions, such as standpoints towards European integration, or socio-cultural liberal-conservative dimensions. Unfortunately, because for Dutch political parties this data is not available in the CHES data, we have been unable to test these hypotheses. Generally speaking, solving the criterion validity problem of political parties' positions, requires additional domain knowledge.

Another way that might help to correct the rankings generated from the opinions is by improving the quality of topics and opinions, as it is known topics learned from the parliamentary proceedings are noisy [3]. In the original paper, Fang et al. used an elaborate method involving supervised machine learning to select sentences containing opinion words [11]. The resources required to do this for Dutch data do not exist, and would therefore need separate validation. However, higher quality opinions could lead to better results. Another possibility to improve opinion quality is to impose constraints on the dataset, as done in VODUM [23]. Especially the constraint that words in the same sentence are assigned to the same topic might help to reduce noise in topics and opinions. Finally, topic and opinion quality could be improved by applying postprocessing techniques. For example, parsimonization might be used to select high quality topic and opinion terms [2].

In addition to content and criterion validity, there is also construct validity. Construct validity refers to the extent to which a measure behaves as expected in a theoretical context. Assessing this aspect of validity requires extensive domain knowledge, which is why we did not include construct validity in our study. What we have shown, however, is that extensive domain knowledge is not required in order to validate a topic model. What is required, of course, is the availability of external data to validate against.

The validation of new topic modeling methods is also impeded by the fact that researchers who introduce new models rarely provide implementations of these models. Of the work discussed in section 2, only an implementation of VODUM [23] was made available. To facilitate validation studies, providing access to source code of new algorithms would be very helpful.

7 Conclusion

This paper presented a validation study of cross-perspective topic modeling [11] using Dutch parliamentary proceedings. The results show that the method yields valid topics (content and criterion validity). While opinions were found to be representative of the political parties' positions as expressed in party manifestos (content validity), we were unable to find correlation between opinions and positions on the left/right political spectrum (criterion validity). Further work is required to determine whether differences between opinions correlate with other politically meaningful dimensions. We also propose to investigate the effect of improving topic and opinion quality on the validation results.

The second contribution of this paper is that we show validation studies are feasible, even without extensive domain knowledge. We contend that in order for topic models to be useful, the results must be semantically meaningful to humans. Because anecdotal qualitative evaluations and/or assessments of model fit fail to capture this essential aspect, validation of results is required before researchers from other domains will apply these methods.

Acknowledgements

The authors would like to thank Kostas Gemenis and Andreas Warntjen for valuable suggestions with regard to this study.

References

- [1] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan, 'An introduction to MCMC for machine learning', *Machine learning*, **50**(1-2), 5–43, (2003).
- [2] H. Azarbyonad, M. Dehghani, T. Kenter, M. Marx, J. Kamps, and M. de Rijke, 'Measuring Topical Diversity of Text Documents Using Hierarchical Parsimonization', (Under submission).
- [3] H. Azarbyonad, F. Saan, M. Dehghani, M. Marx, and J. Kamps, 'Are Topically Diverse Documents Also Interesting?', in *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pp. 215–221, (2015).
- [4] R. Bakker, E. Edwards, L. Hooghe, S. Jolly, J. Koedam, F. Kostelka, G. Marks, J. Polk, J. Rovny, G. Schumacher, M. Steenbergen, M. Vachudova, and K. Zilovic. 1999-2014 Chapel Hill Expert Survey Trend File. <http://chesdata.eu/>, 2015.
- [5] S. Bevan. Gone Fishing: The Creation of the Comparative Agendas Project Master Codebook. <http://sbevan.com/cap-master-codebook.html>, 2014.
- [6] D. M. Blei, A. Y. Ng, and M. I. Jordan, 'Latent dirichlet allocation', *the Journal of machine Learning research*, **3**, 993–1022, (2003).
- [7] G. Bouma, 'Normalized (pointwise) mutual information in collocation extraction', in *Proceedings of the International Conference of the German Society for Computational Linguistics and Language Technology (GSCL '09)*, pp. 31–40, (2009).
- [8] G. Breeman, D. Lowery, C. Poppelaars, S. L. Resodihardjo, A. Timmermans, and J. de Vries, 'Political attention in a coalition system: Analysing Queen's speeches in the Netherlands 1945–2007', *Acta Politica*, **44**(1), (2009).
- [9] E. G. Carmines and R. A. Zeller, *Reliability and validity assessment*, Sage publications, 1979.
- [10] J. Chang, S. Gerrish, C. Wang, J. L. Boyd-Graber, and D. M. Blei, 'Reading tea leaves: How humans interpret topic models', in *Advances in Neural Information Processing Systems*, pp. 288–296, (2009).
- [11] Y. Fang, L. Si, N. Somasundaram, and Z. Yu, 'Mining contrastive opinions on political texts using cross-perspective topic model', in *the fifth ACM international conference on Web search and data mining*, pp. 63–72, (2012).
- [12] S. Gottopati, M. Qiu, Y. Sim, J. Jiang, and N. Smith, 'Learning topics and positions from debatepedia', in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1858–1868, (2013).
- [13] T. L. Griffiths and M. Steyvers, 'Finding scientific topics', *Proceedings of the National Academy of Sciences*, **101**(1), 5228–5235, (2004).
- [14] J. Grimmer and B. M. Stewart, 'Text as data: The promise and pitfalls of automatic content analysis methods for political texts', *Political Analysis*, **21**(3), 267–297, (2013).
- [15] M. G. Kendall, 'A new measure of rank correlation', *Biometrika*, **30**(1), 81–93, (1938).
- [16] E. L. Lehmann and H. J. M. D'Abbrera, *Nonparametrics: Statistical Methods Based on Ranks*, Springer, 1998.
- [17] P. Lehmann, T. Mattheiß, N. Merz, S. Regel, and A. Werner. Manifesto Corpus. Version: 2015-5. <https://manifestoproject.wzb.eu/>, 2015.
- [18] W. H. Lin, E. Xing, and A. Hauptmann, 'A joint topic and perspective model for ideological discourse', in *Machine Learning and Knowledge Discovery in Databases*, 17–32, Springer Berlin Heidelberg, (2008).
- [19] B. Pang and L. Lee, 'Opinion mining and sentiment analysis', *Foundations and trends in information retrieval*, **2**(1-2), 1–135, (2008).
- [20] M. Paul and R. Girju, 'A two-dimensional topic-aspect model for discovering multi-faceted topics', in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)*, pp. 545–549, (2010).
- [21] K. M. Quinn and B. L. Monroe, 'How to analyze political attention with minimal assumptions and costs', *American Journal of Political Science*, **54**, 209–228, (2010).
- [22] M. Röder, A. Both, and A. Hinneburg, 'Exploring the space of topic coherence measures', in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pp. 399–408, (2015).
- [23] T. Thonet and G. Cabanac, 'VODUM: a Topic Model Unifying Viewpoint, Topic and Opinion Discovery', in *Advances in Information Retrieval*, 533–545, Springer, (2016).
- [24] A. Timmermans and G. Breeman, 'Morality issues in the Netherlands: coalition politics under pressure', in *Morality Politics in Western Europe*, 35–61, Palgrave Macmillan UK, (2012).
- [25] A. Trabelsi and O. R. Zaiane, 'Mining contentious documents using an unsupervised topic model based approach', in *Data Mining (ICDM), 2014 IEEE International Conference on*, pp. 550–559, (2014).
- [26] A. van den Bosch, B. Busser, S. Canisius, and W. Daelemans, 'An efficient memory-based morphosyntactic tagger and parser for Dutch', *LOT Occasional Series*, **7**, 191–206, (2007).
- [27] J. M. van der Zwaan, M. Marx, and J. Kamps. Palmetto position storing Lucene index of Dutch Wikipedia. <http://dx.doi.org/10.5281/zenodo.46377>, 2016.

Finite Unary Relations and Qualitative Constraint Satisfaction

Peter Jonsson¹

Abstract. Extending qualitative CSPs with the ability of restricting selected variables to finite sets of possible values has been proposed as an important research direction with important applications. Complexity results for this kind of formalisms have appeared in the literature but they focus on concrete examples and not on general principles. We propose three general methods. The first two methods are based on analysing the given CSP from a model-theoretical perspective, while the third method is based on directly analysing the growth of the representation of solutions. We exemplify our methods on temporal and spatial formalisms including Allen's algebra and RCC5.

1 INTRODUCTION

Qualitative reasoning has a long history in artificial intelligence and the combination of qualitative reasoning and constraint reasoning has been a very productive field. A large number of constraint-based formalisms for qualitative reasoning have been invented, most notably within temporal and spatial reasoning, and they have been investigated from many different angles. Recently, there has been a strong interest in combining different qualitative CSPs. Wolter and Zakharyashev [45] refer to temporal and spatial reasoning when they write the following motivation.

The next apparent and natural step would be to combine these two kinds of reasoning.

The importance of such a step for both theory and applications is beyond any doubt.

It has also been noted that another (but related) line of research is highly relevant. Cohn and Renz [17] write the following.

One problem with this [constraint-based] approach is that spatial entities are treated as variables which have to be instantiated using values of an infinite domain. How to integrate this with settings where some spatial entities are known or can only be from a small domain is still unknown and is one of the main future challenges of constraint-based spatial reasoning.

That is, they regard the question of how to extend constraint formalisms (in particular, spatial formalisms) with constants and other unary relations² as being very important; the same observation has been made in a wider context by Kreutzmann and Wolter [32]. Unfortunately, this question has not received the same amount of attention as the question of how to handle combined formalisms. Let us

consider finite-domain CSPs for a moment so let D denote a finite domain of values and let $D_f = \{U \mid U \subseteq D\}$, i.e. the finite set of unary relations over D . For every finite constraint language Γ over D , the computational complexity of $\text{CSP}(\Gamma \cup D_f)$ is known due to results by Bulatov [10]. This is an important complexity result in finite-domain constraint satisfaction and it has been reproven several times using different methods [1, 11]. The situation is radically different when considering infinite-domain CSPs where similar powerful results are not known. This can, at least partly, be attributed to the fact that infinite-domain CSPs constitute a much richer class of problems than finite-domain CSPs: for every computational problem X , there is an infinite-domain constraint language Γ_X such that X and $\text{CSP}(\Gamma_X)$ are polynomial-time Turing equivalent [3]. For finite domain CSPs, we know that the problem is in NP and that the majority of computational problems cannot be captured by finite-domain CSPs.

Nevertheless, there are concrete examples where interesting qualitative and/or infinite-domain CSPs have been extended with finite unary relations. A very early example is the article by Jonsson & Bäckström [28] where several temporal formalisms (including the point algebra and Allen's interval algebra) are extended by unary relations (and also other relations). A more recent example is the article by Li et al. [35] where the point algebra and Allen's algebra are once again considered, together with the cardinal relation algebra, and RCC-5 and RCC-8 with two-dimensional polygonal regions. The results for the temporal formalisms by Jonsson & Bäckström are not completely comparable with the results by Li et al.: Jonsson & Bäckström's approach is based on linear programming while Li et al. use methods based on enforcing consistency. Consistency-enforcing methods have certain advantages such as lower time complexity and easier integration with existing constraint solving methods. At the same time, the linear programming method allows for more expressive extensions with retained tractability. Both consistency-based and LP-based methods have attracted attention lately, cf. Giannakopoulou et al. [22] and Kreutzmann and Wolter [32], respectively, and generalisations of the basic concepts have been proposed and analysed by de Leng and Heintz [18].

Our approach is different: instead of studying concrete examples, we study basic principles and aim at providing methods that are applicable to various constraint formalisms. We present three different methods. The first two methods are based on analysing the given CSP from a model-theoretical perspective, i.e. we investigate properties such as model-completeness and homogeneity. The third method is more of a toolbox for proving that the size of solutions grows in a controlled way, and that problems consequently are in NP. We illustrate the methods on both temporal and spatial formalisms (including Allen's algebra and RCC-5). The reader may find it strange that we

¹ Department of Computer and Information Science, Linköping University, SE-58183 Linköping, Sweden. Email: peter.jonsson@liu.se

² Finite unary relations are sometimes referred to as *landmarks* in the AI literature. We will use the standard mathematical term throughout the paper.

mostly consider extensions with constant relations. The explanation is the close connection between problems extended with constants and with finite unary relations: if one of them is in NP, then both are in NP (see Lemma 3). Most problems under consideration become NP-hard when adding unary relations containing at least three elements: for example, if the constraint language contains the disequality relation \neq , then NP-hardness follows from a straightforward reduction from 3-COLOURABILITY. However, this is not always true if we only add constants to the language. Thus, we can extract more information by considering constants instead of finite unary relations. The same viewpoint is taken by, for instance, Li et al. [35].

The paper has the following structure. We introduce the basic concepts from CSPs and logic together with some information about homomorphisms in Section 2. The three different methods are presented in Sections 3, 4, and 5, respectively. We conclude the paper with a brief discussion in Section 6.

2 PRELIMINARIES

This section is divided into three parts where we consider constraint satisfaction, logic, and automorphisms of relational structures, respectively.

2.1 Constraint satisfaction problems

We begin by presenting CSPs in terms of *homomorphisms*. This view is the most common in the literature on finite-domain CSP and it will provide us with certain advantages: some of the properties that we consider later on are inherently based on homomorphisms. One should note, however, that there is no fundamental difference with the more common AI viewpoint that constraint satisfaction is about assigning values to variables in a way that satisfy certain constraints. In fact, we will use both viewpoints in the sequel.

A *relational signature* τ is a set of *relation symbols* R_i with an associated *arities* $k_i \in \mathbb{N}$. A *(relational) structure* Γ over *relational signature* τ (also called τ -*structure*) is a set D_Γ (the *domain*) together with a relation $R_i^\Gamma \subseteq D_\Gamma^{k_i}$ for each relation symbol R_i of arity k_i . If the reference to the structure Γ is clear, we may omit the superscript in R_i^Γ . We sometimes use the shortened notation \bar{x} for a vector (x_1, \dots, x_n) of any length.

Let Γ and Δ be τ -structures. A *homomorphism* from Γ to Δ is a function f from D_Γ to D_Δ such that for each n -ary relation symbol R in τ and each n -tuple $\bar{a} = (a_1, \dots, a_n)$, if $\bar{a} \in R^\Gamma$, then $(f(a_1), \dots, f(a_n)) \in R^\Delta$.

Let Γ be a (possibly infinite) structure with a (possibly infinite) relational signature τ . Then the *constraint satisfaction problem (CSP)* for Γ is the following computational problem.

CSP(Γ)

INSTANCE: A τ -structure Δ .

QUESTION: Is there a homomorphism from Δ to Γ ?

In the homomorphism perspective on CSPs, the structure Γ is typically called the *template* of the constraint satisfaction problem CSP(Γ). The reader should be aware that several different names are used in the literature; *constraint language* is probably the most common within AI.

A homomorphism from a given τ -structure Δ to Γ is called a *solution* of Δ for CSP(Γ). It is in general not clear how to represent solutions for CSP(Γ) on a computer; however, for the definition of the problem CSP(Γ) we do not need to represent solutions since we

only have to decide the *existence* of solutions. To represent an input structure Δ of CSP(Γ), we need to fix a suitable representation of the relation symbols in the signature τ . We will see in the forthcoming sections that the choice of representation is very important. Given a particular representation of relation symbols, we let $\|\Delta\|$ denote the size of an input structure Δ .

Example 1 (k -COLOURABILITY). For $k \geq 1$, the k -COLOURABILITY problem is the computational problem of deciding for a given finite graph G whether the vertices can be coloured by k colours or not such that adjacent vertices get different colours. It is well-known that the k -colouring problem is NP-hard for $k \geq 3$ and tractable when $k \leq 2$. For $k \geq 1$, let K_k denote the complete loop-free graph on k vertices. We view undirected graphs as τ -structures where τ contains a single binary relation symbol E which denotes a symmetric and anti-reflexive relation. Then the k -COLOURABILITY problem can be viewed as CSP($\{K_k\}$).

Example 2 (Digraph acyclicity). Consider the problem CSP($\{<\}$) where $<$ is the binary order relation of the set \mathbb{Q} of rational numbers. Let $G = (V, A)$ be a directed graph. It is easy to see that there is a homomorphism from G to $(\mathbb{Q}; <)$ if and only if G contains no directed cycle. Thus, CSP($\{<\}$) is solvable in polynomial time since cycle detection in directed graphs can be carried out in polynomial (in fact, linear) time.

Clearly, we can equivalently define the instances of the CSP(Γ) problem as a tuple (V, C) where V is a set of variables and C is a set of constraints of the form $R(x_{i_1}, \dots, x_{i_k})$ where $R \in \Gamma$, k is the arity of R , and $x_{i_1}, \dots, x_{i_k} \subseteq V$. In this case, a solution is a function from V to the domain of Γ satisfying $(f(x_{i_1}), \dots, f(x_{i_k})) \in R$ for every $R(x_{i_1}, \dots, x_{i_k}) \in C$.

Let D be a value domain with a particular representations and let $\|d\|$ denote the size of the representation of $d \in D$. We let $D_c = \{\{d\} \mid d \in D\}$ and $D_f = \{D' \subseteq D \mid D' \text{ is finite}\}$. Given a representation of the elements in D , we always represent the members of D_f as sets of elements in D and we may assume that the size of D_f is linear in the sizes of its elements. Other ways of representing D_f are possible but they are outside the scope of this paper. If Γ is a constraint language with domain D , then CSP($\Gamma \cup D_c$) is the problem CSP(Γ) extended with constants and CSP($\Gamma \cup D_f$) is the problem CSP(Γ) extended with finite unary relations. The next lemma is basically Proposition 1(iii) in Li et al. [35] extended to arbitrary constraint languages.

Lemma 3 *CSP($\Gamma \cup D_c$) is in NP if and only if CSP($\Gamma \cup D_f$) is in NP.*

Proof. There is a trivial polynomial-time reduction from CSP($\Gamma \cup D_c$) to CSP($\Gamma \cup D_f$) so we consider the other direction. Let $I = (V, C)$ be an arbitrary instance of CSP($\Gamma \cup D_f$). Assume I has a solution $s : V \rightarrow D$. Each constraint $U(x) \in C$ with $U \in D_f$ can be replaced by the constraint $\{s(v)\}(v)$. The resulting instance I' is an instance of CSP($\Gamma \cup D_c$), it is satisfiable, and $\|I'\| \leq \|I\|$. The problem CSP($\Gamma \cup D_c$) is in NP so the satisfiability of I' can be polynomial-time verified by a certificate X . A polynomial-time verifiable certificate for I is thus the tuple (I', X) . \square

Lemma 3 allows us to, for example, concentrate on CSP($\Gamma \cup D_c$) instead of CSP($\Gamma \cup D_f$) when proving membership in NP.

2.2 Logic

First-order formulas φ over the signature τ (or, in short, τ -formulas) are as usual inductively defined using the logical symbols of univer-

sal and existential quantification, disjunction, conjunction, negation, equality, bracketing, variable symbols and the symbols from τ . The semantics of a first-order formula over some τ -structure is defined in the ordinary Tarskian style. A τ -formula without free variables is called a τ -sentence. We write $\Gamma \models \varphi$ if and only if the τ -structure Γ is a model for the τ -sentence φ , that is, satisfies φ ; this notation is lifted to sets of sentences in the usual way.

One can use first-order formulas over the signature τ to define relations over a given τ -structure Γ : for a formula $\varphi(x_1, \dots, x_k)$ where x_1, \dots, x_k are the free variables of φ the corresponding relation R is the set of all k -tuples $(t_1, \dots, t_k) \in D_\Gamma^k$ such that $\varphi(t_1, \dots, t_k)$ is true in Γ . In this case we say that R is *first-order definable* over Γ . Note that our definitions are always parameter-free, i.e. we do not allow the use of domain elements in them. We say that the τ -structure Γ admits *quantifier elimination* if every relation with a first-order definition in Γ has a quantifier-free definition in Γ . We also say that a set of formulas T admit quantifier elimination if each $F \in T$ has a logically equivalent quantifier-free formula.

A first-order τ -formula $\phi(x_1, \dots, x_n)$ is called *existential* if it is of the form

$$\exists x_{n+1}, \dots, x_m. \psi$$

where ψ is a quantifier-free first-order formula. A subset of existential formulas is of particular interest to us: a first-order τ -formula $\phi(x_1, \dots, x_n)$ is called *primitive positive* if it is of the form

$$\exists x_{n+1}, \dots, x_m. \psi_1 \wedge \dots \wedge \psi_l$$

where ψ_1, \dots, ψ_l are *atomic τ -formulas*, i.e., formulas of the form

1. $R(y_1, \dots, y_k)$ with $R \in \tau$ and $y_i \in \{x_1, \dots, x_m\}$ or
2. $y = y'$ for $y, y' \in \{x_1, \dots, x_m\}$.

If the relation R has a primitive positive definition in Γ , then we say that R is *pp-definable* in Γ , and we define $\langle \Gamma \rangle$ to be the set of relations that are pp-definable in Γ . It is well-known [27] (and not hard to prove) that if Γ is a structure and a relation R is pp-definable in Γ , then there is a polynomial-time reduction from $\text{CSP}(\Gamma \cup \{R\})$ to $\text{CSP}(\Gamma)$. This explains why pp-definitions are important when studying the complexity of CSP problems. To exemplify, consider the constraint language $\Gamma = \{\{1, 2, 3, 4\}, \neq\}$ with the natural numbers as its domain. We see that the binary relation $K_4 = \{(x, y) \in \{1, 2, 3, 4\}^2 \mid x \neq y\}$ (from Example 1) is pp-definable in Γ since

$$K_4(x, y) \Leftrightarrow \{1, 2, 3, 4\}(x) \wedge \{1, 2, 3, 4\}(y) \wedge x \neq y.$$

and it follows that $\text{CSP}(\Gamma)$ is NP-hard.

It is worth mentioning that many of the operations in relational algebra can be viewed as pp-definitions. Let R and S denote binary relations. Then, the converse R^\sim has the pp-definition $R^\sim(x, y) \Leftrightarrow R(y, x)$, the intersection $R \cap S$ has the pp-definition $(R \cap S)(x, y) \Leftrightarrow R(x, y) \wedge S(x, y)$, and the composition $R \circ S$ has the pp-definition $(R \circ S)(x, y) \Leftrightarrow \exists z. R(x, z) \wedge S(z, y)$.

2.3 Automorphisms

Keeping the homomorphism definition of CSPs in mind may be helpful in the rest of this section. Let Γ and Δ denote two relational τ -structures. An injective homomorphism that additionally preserve the complement of each relation is called an *embedding*. Homomorphisms from Γ to Γ are called *endomorphisms* of Γ . An *automorphism* of Γ is a bijective endomorphism whose inverse is also an

endomorphism; that is, they are bijective embeddings of Γ into Γ . The set containing all endomorphisms of Γ is denoted $\text{End}(\Gamma)$ while the set of all automorphisms is denoted $\text{Aut}(\Gamma)$.

Example 4 Let $R_+ = \{(x, y, z) \in \mathbb{Z}^3 \mid x + y = z\}$. For arbitrary $a \in \mathbb{Z}$, let $e_a : \mathbb{Z} \rightarrow \mathbb{Z}$ be defined as $e_a(n) = a \cdot n$. Let $e : \mathbb{Z} \rightarrow \mathbb{Z}$ be an arbitrary endomorphism of $(\mathbb{Z}; R_+)$; e is a homomorphism so $(e(x), e(y), e(z)) \in R_+$ whenever $(x, y, z) \in R_+$ and, more generally, $e(\sum_{i=1}^k x_i) = \sum_{i=1}^k e(x_i)$ when $x_1, \dots, x_k \in \mathbb{Z}$. Arbitrarily choose $n \in \mathbb{Z}$ and note that

$$e(n) = e(\underbrace{1 + \dots + 1}_{n \text{ times}}) = n \cdot e(1).$$

It follows that $\text{End}((\mathbb{Z}; R_+)) = \{e_a \mid a \in \mathbb{Z}\}$. Note that e_a has an inverse if and only if $a \in \{-1, 1\}$. Thus, $\text{Aut}((\mathbb{Z}; R_+)) = \{e_a \mid a \in \{-1, 1\}\}$.

A useful observation is that if (V, C) is an instance of $\text{CSP}(\Gamma)$ with a solution $s : V \rightarrow D$, then $s' : V \rightarrow D$ defined by $s'(x) = \alpha(s(x))$ is a solution to (V, C) for every α in $\text{Aut}(\Gamma)$ or $\text{End}(\Gamma)$. If a function $s : V \rightarrow D$ is *not* a solution to (V, C) , then $s'(x) = \alpha(s(x))$ is not a solution for any $\alpha \in \text{Aut}(\Gamma)$ while s' may or may not be a solution if $\alpha \in \text{End}(\Gamma) \setminus \text{Aut}(\Gamma)$.

In the following, let G be a set of permutations of a set X . We say that G is a *permutation group* if the identity permutation is in G and for arbitrary $g, f \in G$, the functions $x \mapsto g(f(x))$ and $x \mapsto g^{-1}(x)$ are also in G . In other words, G is closed under function composition and inversion. If Γ is a τ -structure, then $\text{Aut}(\Gamma)$ is a permutation group on the set D_Γ . For $n \geq 1$, the *orbit* of $(t_1, \dots, t_n) \in X^n$ under G is the set $\{(\alpha(t_1), \dots, \alpha(t_n)) \mid \alpha \in G\}$. Clearly, the orbits of n -tuples under G partition the set X^n , that is, every $(t_1, \dots, t_n) \in X^n$ lies in precisely one orbit under G .

Example 5 Consider once again the structure (\mathbb{Z}, R_+) from Example 4. It is obvious that $\{e_1, e_{-1}\}$ forms a (trivial) group under function composition. If $a \in \mathbb{Z}$, then the orbit of (a) equals $\{a, -a\}$ so $(\mathbb{Z}; R_+)$ admits an infinite number of different orbits under its automorphism group.

A (first-order) *theory* is a set of first-order sentences. When the first-order sentences are over the signature τ , we say that T is a τ -*theory*. The (full) *theory* of a τ -structure Δ (denoted $\text{Th}(\Delta)$) is the set of τ -sentences ϕ such that $\Delta \models \phi$. A *model* of a τ -theory T in a τ -structure Δ such that Δ satisfies all sentences in T . Theories that have a model are called *satisfiable*. We now define a central concept: a satisfiable first-order theory T is ω -*categorical* if all countable models of T are isomorphic, and a structure is ω -categorical if its first-order theory is ω -categorical. All ω -categorical structures that appear in this article will be countably infinite, we make the convention that ω -categorical structures are countably infinite. Note that the first-order theory of a finite structure does not have infinite models so finite structures are ω -categorical. One of the first infinite structures that were found to be ω -categorical (by Cantor [15]) is the linear order of the rational numbers $(\mathbb{Q}; <)$. There are many characterisations of ω -categoricity and the most important one is in terms of the automorphism group.

Definition 6 A permutation group G over a countably infinite set X is *oligomorphic* if G has only finitely many orbits of n -tuples for each $n \geq 1$.

An accessible proof of the following theorem can be found in Hodges' book [25].

Theorem 1 (Engeler, Ryll-Nardzewski, Svenonius) *Let Γ be a countably infinite structure Γ with countable signature. The following are equivalent.*

1. Γ is ω -categorical,
2. $\text{Aut}(\Gamma)$ is oligomorphic, and
3. a relation is first-order definable in Γ if and only if it is preserved by the automorphisms of Γ .

Example 5 immediately implies that $(\mathbb{Z}; R_+)$ is not an ω -categorical structure. Consider the structure $(\mathbb{Z}; <)$. One can verify that $\text{Aut}((\mathbb{Z}; <)) = \{x \mapsto x + a \mid a \in \mathbb{Z}\}$. Hence, $(\mathbb{Z}; <)$ is not ω -categorical (despite the fact that $(\mathbb{Q}; <)$ is indeed ω -categorical): the orbits of $(0, 0)$, $(0, 1)$, $(0, 2)$, \dots are distinct.

We conclude this section by presenting a result that connects first-order definability with ω -categoricity.

Theorem 2 (Thm. 7.3.8 in Hodges [25]) *If Γ is an ω -categorical structure and Δ is first-order definable in Γ , then Δ is ω -categorical, too.*

3 METHOD I: MODEL-COMPLETE CORES

Our first method is based on analysing a given constraint language Γ with respect to its endo- and automorphisms. We first need to introduce the concept of *homomorphically equivalent CSPs*. Let Γ and Δ denote two relational τ -structures. A bijective homomorphism from Γ to Δ is called an *isomorphism*. If Γ and Δ are isomorphic, then it is clear that $\text{CSP}(\Gamma)$ and $\text{CSP}(\Delta)$ are the same computational problem. However, Γ and Δ may be non-isomorphic and still have the same CSP. This is, for instance, the case when there simultaneously exists a homomorphism from Γ to Δ and a homomorphism from Δ to Γ . In this case, we say that Γ and Δ are *homomorphically equivalent* and this defines an equivalence relation on structures. We note that there are structures that have the same CSP even when they are not homomorphically equivalent. Consider for example the structures $(\mathbb{Z}; <)$ and $(\mathbb{Q}; <)$. They have the same CSP and there is a homomorphism from $(\mathbb{Z}; <)$ to $(\mathbb{Q}; <)$ but there is no homomorphism from $(\mathbb{Q}; <)$ to $(\mathbb{Z}; <)$.

For ω -categorical structures Γ , the equivalence classes have interesting properties: the homomorphic equivalence class of Γ contains a distinguished member Δ which is up to isomorphism uniquely given by two properties: Δ is a *core* and Δ is *model-complete*. A relational structure Γ is a core if all endomorphisms (i.e. homomorphisms from Γ to Γ) are embeddings. Cores are important when studying the complexity of finite-domain CSPs: we refer to the textbook by Hell and Nešetřil [23] that extensively covers cores in the context of graph homomorphisms and to Bulatov et al. [13] that covers cores in general finite-domain CSPs. Model completeness is a central concept in model theory: a structure Γ is model-complete if every formula in $\text{Th}(\Gamma)$ is equivalent to an existential formula modulo T . This may be viewed as a limited notion of quantifier elimination.

Consider the relation $<$ over the rationals \mathbb{Q} . The structure $(\mathbb{Q}; <)$ admits quantifier elimination [33] so every formula in $\text{Th}(\{<\})$ is equivalent to a quantifier-free formula (and, naturally, an existential formula). It follows that $(\mathbb{Q}; <)$ is model-complete, and that every Γ that is first-order definable in $(\mathbb{Q}; <)$ is model-complete, too. The structure $(\mathbb{Q}; <)$ is also a core. Let $e : \mathbb{Q} \rightarrow \mathbb{Q}$ be an endomorphism of $(\mathbb{Q}; <)$, i.e. if $a < b$, then $e(a) < e(b)$. Clearly, e is injective and it preserves the relation \geq (that is, the negation of $<$) since if $a > b$, then $e(a) > e(b)$ and if $a = b$, then $e(a) = e(b)$. However, there are relations R that are first-order definable in $(\mathbb{Q}; <)$ and $(\mathbb{Q}; R)$ is not

a core. One trivial example is the equality relation $=$. The function $x \mapsto 1$ is obviously an endomorphism of $=$ but it is not injective and thus not an embedding. We have the following important result.

Theorem 3 (Bodirsky [5]) *Every ω -categorical structure Δ is homomorphically equivalent to a model-complete core structure Γ which is unique up to isomorphism. Moreover, Γ is ω -categorical and the orbits of n -tuples are pp-definable in Γ for all $n \geq 1$.*

Since homomorphically equivalent structures have the same CSP, one can focus on ω -categorical structures that have these properties. The fact that we can pp-define the orbits of n -tuples will now become highly important.

Theorem 4 *Let Γ be a constraint language over the domain D . Assume the following:*

1. Γ is a model-complete ω -categorical core and
2. the domain elements are represented in a way such that given a vector $\vec{d} = (d_1, \dots, d_n) \in D^n$, a pp-definition in Γ of the orbit of \vec{d} can be generated in polynomial time (in the size of the representation of d_1, \dots, d_n).

Then, $\text{CSP}(\Gamma)$ and $\text{CSP}(\Gamma \cup D_c)$ are polynomial-time equivalent.

Proof. Let $\Gamma' = \Gamma \cup D_c$. The reduction from $\text{CSP}(\Gamma)$ to $\text{CSP}(\Gamma')$ is trivial so we concentrate on the other direction. Let $I' = (V', C')$ be an instance of $\text{CSP}(\Gamma')$. Assume without loss of generality that if $\{d_i\}(x)$ is in C' , then there is no variable $y \neq x$ such that $\{d_i\}(y) \in C'$; if so, the constraint $\{d_i\}(y)$ can be removed and the variable y be replaced by x . Normalising an instance in this way can easily be done in polynomial-time. We assume (without loss of generality) that the only constraints in C' with relations from D_c are $\{d_1\}(x_1), \dots, \{d_m\}(x_m)$. This can be achieved in polynomial time by renaming of variables.

Compute (in polynomial time) the formula $F(x_1, \dots, x_m)$ for the orbit of (d_1, \dots, d_m) . Define $I = (V, C)$ such that C equals C' extended with $F(x_1, \dots, x_m)$ and the constant relations removed. Let V denote V' expanded with the existentially quantified variables in $F(x_1, \dots, x_m)$. Note that I can be constructed in polynomial time and it is an instance of $\text{CSP}(\Gamma)$.

If the instance I has no solution, then it follows immediately that I' does not have a solution—one can view I as being a relaxation of I' since the formula $F(x_1, \dots, x_m)$ is, in particular, satisfiable when $x_1 = d_1, \dots, x_m = d_m$. If the instance I has a solution $s : V \rightarrow D$, then we claim that there is a solution $s' : V' \rightarrow D$ to I' , too. Since F describes the orbit of (d_1, \dots, d_m) , there is an automorphism α of Γ such that $\alpha(s'(x_i)) = d_i$, $1 \leq i \leq m$. This implies that $\alpha(s'(x))$ restricted to the set V is a solution to I . \square

By Theorem 3, we know that orbit-defining formulas always can be pp-defined in Γ under the given assumptions. Whether these can be generated or not in polynomial time is a completely different question, though. We give an example based on constraint languages that are first-order definable in $(\mathbb{Q}; <)$. Such constraint languages are sometimes called *temporal constraint languages*. They are well-studied in the literature and, in fact, the computational complexity of $\text{CSP}(\Gamma)$ is known for every finite Γ [4]. A concrete example of a temporal constraint language is the point algebra PA: we see that $x \leq y \Leftrightarrow (x < y) \vee (x = y)$ and $x \neq y \Leftrightarrow \neg(x = y)$. Furthermore, temporal constraint languages are ω -categorical due to Theorem 2 and it is known (by Junker and Ziegler [31], also see Cameron [14]) that there are five possible choices of $\text{Aut}(\Gamma)$. We concentrate on

the (for our purposes) most interesting case when $< \in \langle \Gamma \rangle$ and $\text{Aut}(\Gamma) = \text{Aut}(\mathbb{Q}; <)$. Arbitrarily choose such a language Γ and assume (without loss of generality) that $< \in \Gamma$. We know that Γ is model-complete so we assume that Γ is a core (for instance, Γ may be the point algebra PA). We represent all members of \mathbb{Q} in the natural way, i.e. as (a/b) where $a, b \neq 0$ are integers written in binary.

The automorphisms of $(\mathbb{Q}; <)$ are the bijective functions $f : \mathbb{Q} \rightarrow \mathbb{Q}$ that are monotonously increasing. The orbit of 1-tuples equals \mathbb{Q} while the orbit of a 2-tuple (a, b) with $a < b$ equals $\{(x, y) \in \mathbb{Q}^2 \mid x < y\}$. More generally, the orbit of a k -tuple (a_1, \dots, a_k) with $a_1 < a_2 < \dots < a_k$ equals

$$\{(x_1, \dots, x_k) \in \mathbb{Q}^k \mid x_1 < x_2 < \dots < x_k\}$$

so the orbit-defining formulas can be generated in polynomial time. Theorem 4 is thus applicable and $\text{CSP}(\Gamma \cup \mathbb{Q}_c)$ is polynomial-time equivalent to $\text{CSP}(\Gamma)$. In particular, $\text{CSP}(\Gamma \cup \mathbb{Q}_c)$ is in P if $\text{CSP}(\Gamma)$ is in P, and $\text{CSP}(\Gamma \cup \mathbb{Q}_c)$ is in NP if $\text{CSP}(\Gamma)$ is in NP.

This example shows that ω -categoricity is indispensable. Theorem 4 combined with the tractability of $\text{CSP}(\mathbb{Q}; <, \neq)$ implies that $\text{CSP}(\Gamma_{\mathbb{Q}})$ is in P when $\Gamma_{\mathbb{Q}}$ denotes $(\mathbb{Q}; <, \neq)$ extended with the unary relations in \mathbb{Q}_c . Recall that $(\mathbb{Z}; <)$ and $(\mathbb{Z}; <, \neq)$ are not ω -categorical and define $\Gamma_{\mathbb{Z}}$ by expanding $(\mathbb{Z}; <, \neq)$ with \mathbb{Z}_c . The problem $\text{CSP}(\Gamma_{\mathbb{Z}})$ is NP-hard since the relation $\{0, 1, 2\}$ can be pp-defined via $x \in \{0, 1, 2\} \Leftrightarrow -1 < x \wedge x < 3$, and the problem $\text{CSP}(\mathbb{Z}; \{0, 1, 2\}, \neq)$ is NP-hard since there is an obvious polynomial-time reduction from 3-COLOURABILITY.

4 METHOD II: HOMOGENEITY

Homogeneous structures have been intensively studied in mathematics and logics (for instance, in connection with combinatorics, model theory, and group theory) and they are becoming more and more relevant in the study of CSPs. Homogeneous structures have useful properties such as admitting quantifier elimination and they are ω -categorical whenever the structure contains a finite number of relations and the domain is countably infinite. Examples include $(\mathbb{Q}; <)$, the *random* (or Rado) graph, and certain structures with connections to phylogenetic reconstruction problems. There are also many structures that are well-studied in AI that can be represented by homogeneous structures: examples include Allen's algebra [24] and RCC-8 [9]. We need some machinery before providing the formal definition. Let D be the domain of a relational τ -structure Γ and arbitrarily choose $S \subseteq D$. Then the *substructure induced by S in Γ* is the τ -structure Δ with domain S such that $R^\Delta = R^\Gamma \cap S^n$ for each n -ary $R \in \tau$; we also write $\Gamma[S]$ for Δ . The structure Γ is called *homogeneous* if every isomorphism $f : D_1 \rightarrow D_2$ between finite induced substructures of Γ can be extended to an automorphism of Γ , i.e. there exists an automorphism α such that $f(x) = \alpha(x)$ when $x \in D_1$. One should note that homogeneity is a more “fragile” concept than ω -categoricity. For instance, Γ being homogeneous and Δ being first-order definable in Γ does not necessarily imply that Δ is homogeneous.

To simplify the presentation, we will turn our attention to binary constraints and *partition schemes*; this concept was introduced by Ligozat & Renz [37] and it has been highly influential in CSP research. Let D be a non-empty domain. Given a finite family $\mathcal{B} = \{R_1, \dots, R_k\}$ of binary relations over D , we say that \mathcal{B} is *jointly exhaustive* (JE) if $\bigcup \mathcal{B} = D^2$ and that \mathcal{B} is *pairwise disjoint* (PD) if $R_i \cap R_j = \emptyset$ whenever $1 \leq i \neq j \leq k$. If \mathcal{B} is simultaneously JE and PD, then \mathcal{B} forms a partition of the set D^2 .

Definition 7 Let D be a non-empty domain and let $\mathcal{B} = \{R_1, \dots, R_k\}$ be a set of binary relations over D . We say that \mathcal{B} is a *partition scheme* if the following holds:

1. \mathcal{B} is JEPD,
2. the equality relation $\text{EQ}_D = \{(x, x) \in D^2\}$ is in \mathcal{B} , and
3. for every $R_i \in \mathcal{B}$, the converse relation R_i^{\smile} is in \mathcal{B} .

It is important to note that if \mathcal{B} is a partition scheme over a domain D , then for arbitrary $d, d' \in D$ there exists exactly one $B \in \mathcal{B}$ such that $(d, d') \in B$. Given a finite set of binary relations $\mathcal{B} = \{R_1, \dots, R_k\}$, we follow notational conventions from [16, 30] and define $\mathcal{B}^{\vee=}$ as the set of all unions of relations from \mathcal{B} . The set $\mathcal{B}^{\vee=}$ and the problem $\text{CSP}(\Gamma)$ where $\Gamma \subseteq \mathcal{B}^{\vee=}$ are the natural objects that are studied in connection with partition schemes.

Theorem 5 Let $\mathcal{B} = \{B_1, \dots, B_k\}$ be a partition scheme over the domain D . Assume the following:

1. $\mathcal{B}^{\vee=}$ is homogeneous, and
2. the domain elements are represented in a way such that given two elements $a, b \in D$, it is possible to find (by using an algorithm A) the unique B_i , $1 \leq i \leq m$, such that $(a, b) \in B_i$ in polynomial time (measured in the size of the representations of a and b).

If $\mathcal{B} \subseteq \Gamma \subseteq \mathcal{B}^{\vee=}$, then $\text{CSP}(\Gamma)$ and $\text{CSP}(\Gamma \cup D_c)$ are polynomial-time equivalent.

Proof. Let $\Gamma' = \Gamma \cup D_c$. The reduction from $\text{CSP}(\Gamma)$ to $\text{CSP}(\Gamma')$ is trivial so we concentrate on the other direction. Let $I' = (V', C')$ be an instance of $\text{CSP}(\Gamma')$. We assume without loss of generality (just as in the proof of Theorem 4) that the only constraints in C' with relations from D_c are $\{d_1\}(x_1), \dots, \{d_m\}(x_m)$.

Construct an instance $I = (V, C)$ of $\text{CSP}(\mathcal{B}^{\vee=})$ as follows: let

- $V = V'$,
- $\widehat{C} = \{A(d_i, d_j)(x_i, x_j) \mid 1 \leq i \neq j \leq m\}$, and
- $C = (C' \cup \widehat{C}) \setminus \{\{d_1\}(x_1), \dots, \{d_m\}(x_m)\}$.

The instance $I = (V, C)$ can obviously be generated in polynomial time.

If the instance I' has a solution, then it follows immediately that I has a solution—the constraints in \widehat{C} are satisfiable by the assignment $x_1 = d_1, \dots, x_m = d_m$.

If the instance I has a solution $s : V \rightarrow D$, then we claim that there is a solution $s' : V \rightarrow D$ to I' , too. Let $S = \{s(x_1), \dots, s(x_m)\}$ and $T = \{d_1, \dots, d_m\}$. The set T contains m elements by our initial assumptions and the set S contains m elements due to the constraints in \widehat{C} ; all variables in $\{x_1, \dots, x_m\}$ are assigned distinct values since none of the constraints in \widehat{C} allows equality (due to the fact that \mathcal{B} is a partition scheme and d_1, \dots, d_m are distinct values). Thus, $f : S \rightarrow T$ is a well-defined bijective function if we let $f(s(x_i)) = d_i$, $1 \leq i \leq m$. We continue by proving the following claim.

Claim: f is an homomorphism from $B[S]$ to $B[T]$ when $B \in \mathcal{B}$. Arbitrarily choose a tuple $(a, b) \in B[S]$. By the choice of S , we know that $a = s(x_i)$ and $b = s(x_j)$ for some distinct $1 \leq i, j \leq m$. We see that

$$(f(a), f(b)) = (f(s(x_i)), f(s(x_j))) = (d_i, d_j).$$

We know that $d_i, d_j \in T$ so it remains to show that $(d_i, d_j) \in B$. If $A(d_i, d_j) = B$, then we are done. If $A(d_i, d_j) = B' \neq B$, then

$B'(x_i, x_j) \in \widehat{C} \subseteq C$ so $(s(x_i), s(x_j)) \in B'$. This contradicts that $a = s(x_i)$, $b = s(x_j)$, and $(a, b) \in B$ since $B \cap B' = \emptyset$.

We show that f is a homomorphism from $\mathcal{B}^{\vee} = [S]$ to $\mathcal{B}^{\vee} = [T]$. Since f is bijective, it follows that f is an isomorphism between $\mathcal{B}^{\vee} = [S]$ and $\mathcal{B}^{\vee} = [T]$, too. Arbitrarily choose a relation $R \in \mathcal{B}^{\vee}$ where $R = B_1 \cup \dots \cup B_p$ and $B_i \in \mathcal{B}$, $1 \leq i \leq p$. Arbitrarily choose $(a, b) \in R[S]$. The tuple (a, b) is a member of some relation B_i in $\{B_1, \dots, B_p\}$. By the Claim, $(f(a), f(b)) \in B_i[T]$ so $(f(a), f(b)) \in R[T]$ since $B_i \subseteq R$. It follows that f is a homomorphism from $R[S]$ to $R[T]$ since (a, b) was arbitrarily chosen in $R[S]$. This, in turn, implies that f is a homomorphism $\mathcal{B}^{\vee} = [S]$ to $\mathcal{B}^{\vee} = [T]$ since R was arbitrarily chosen in \mathcal{B}^{\vee} .

Since \mathcal{B}^{\vee} is a homogeneous structure, the function f can be extended to an automorphism α of \mathcal{B}^{\vee} . It follows that the function $s' : V \rightarrow D$ defined such that $s'(x) = \alpha(s(x))$ is a solution to I' ; merely note that $s'(x_i) = d_i$, $1 \leq i \leq m$. \square

Consider Allen's algebra \mathcal{A} with domain \mathbb{I} where intervals are represented as (I^-, I^+) where $I^- < I^+$, $I^-, I^+ \in \mathbb{Q}$, and the members of \mathbb{Q} are represented as in Section 3. Hirsch [24] has shown that \mathcal{A} is a homogeneous structure and the second precondition of Theorem 5 is clearly satisfied with the given representation. We conclude that $\text{CSP}(\mathcal{A} \cup \mathbb{I}_c)$ and $\text{CSP}(\mathcal{A} \cup \mathbb{I}_f)$ are NP-complete problems since $\text{CSP}(\mathcal{A})$ is NP-complete. We can also conclude that $\text{CSP}(\mathcal{H} \cup \mathbb{I}_c)$ is in P when \mathcal{H} is the ORD-Horn subclass [40] since \mathcal{H} contains all 13 basic relations. More examples of homogeneous structures that are relevant for computer science are described in, for example, Bodirsky [5], Bodirsky and Chen [6], and Bodirsky and Wöfl [9].

5 METHOD III: SMALL SOLUTIONS

The methods in Section 3 and 4 provide polynomial-time equivalences between $\text{CSP}(\Gamma)$ and $\text{CSP}(\Gamma \cup D_c)$ under certain conditions. In this section, we will instead analyse the constraint language $\Gamma \cup D_c$ directly. The main result will be weaker than in the previous two sections since we will only be able to prove membership in NP. On the other hand, the approach is applicable also without ω -categoricity.

Let Γ be an arbitrary constraint language with domain D , and assume that the relations in Γ and the elements in D are represented in some fixed way. We say that Γ has the *small solution property* if there exists a polynomial p (that only depends on the choice of Γ) such that for every satisfiable instance $I = (V, C)$ of $\text{CSP}(\Gamma)$, there exists a solution $s : V \rightarrow D$ such that $\|s(v)\| \leq p(\|I\|)$ for every $v \in V$.

Lemma 8 *Let Γ denote a constraint language over the domain D . Assume that*

1. Γ has the small solution property and
2. there exists an algorithm A and a polynomial q such that for arbitrary k -ary $R \in \Gamma$ and $d_1, \dots, d_k \in D$, algorithm A can verify whether $(d_1, \dots, d_k) \in R$ or not in time $O(q(\|R\| + \sum_{i=1}^k \|d_i\|))$.

Then $\text{CSP}(\Gamma)$ is in NP.

Proof. Let (V, C) denote an arbitrary instance of $\text{CSP}(\Gamma)$. To show that $I = (V, C)$ is satisfiable, non-deterministically guess a solution $s : V \rightarrow D$ such that $\|s(v)\| \leq p(\|I\|)$ for every $v \in V$ (where p denotes a fixed polynomial). Such a solution exists since Γ has the small solution property, and the size of s is consequently polynomially bounded in $\|I\|$. The solution s can thus be verified in polynomial time with the aid of algorithm A . \square

The small solution property is particularly useful in connection with partition schemes.

Lemma 9 *Let \mathcal{B} be a partition scheme with domain D such that precondition (2) of Lemma 8 is satisfied. If $\mathcal{B} \cup D_c$ has the small solution property, then $\mathcal{B}^{\vee} \cup D_c$ has the small solution property and both $\text{CSP}(\mathcal{B}^{\vee} \cup D_c)$ and $\text{CSP}(\mathcal{B}^{\vee} \cup D_f)$ are in NP.*

Proof. Let $I = (V, C)$ denote an instance of $\text{CSP}(\mathcal{B}^{\vee} \cup D_c)$ with solution $s : V \rightarrow D$. Replace each binary constraint $x(b_1 \cup \dots \cup b_m)y \in C$ (where $\{b_1, \dots, b_m\} \subseteq \mathcal{B}$) with the constraint $x\{b_i\}y$, $1 \leq i \leq m$, and $(s(x), s(y)) \in b_i$. The resulting instance $I' = (V, C')$ is solvable, $\|I'\| \leq \|I\|$, and it is an instance of $\text{CSP}(\mathcal{B} \cup D_c)$. We know that $\mathcal{B} \cup D_c$ has the small solution property so there is a solution $s' : V \rightarrow D$ such that $\|s'(v)\| \leq p(\|I'\|)$ for every $v \in V$ and some polynomial p that only depends on \mathcal{B} . Since s' is a solution to I , too, it follows that $\|s'(v)\| \leq p(\|I'\|) \leq p(\|I\|)$. Thus, $\mathcal{B}^{\vee} \cup D_c$ has the small solution property. Lemma 8 implies that $\text{CSP}(\mathcal{B}^{\vee} \cup D_c)$ is in NP and consequently Lemma 3 implies that $\text{CSP}(\mathcal{B}^{\vee} \cup D_f)$ is in NP. \square

Many well-known structures possess the small solution property. A prime example is relations R defined by linear expressions, that is, R is defined by

$$(x_1, \dots, x_k) \in R \Leftrightarrow \sum_{i=1}^k c_i \cdot x_i \leq c_0$$

or

$$(x_1, \dots, x_k) \in R \Leftrightarrow \sum_{i=1}^k c_i \cdot x_i = c_0$$

where the coefficients are in \mathbb{Z} and the variables ranges over, for instance, \mathbb{Q} or \mathbb{Z} . Given a constraint language Γ containing such relations, the small solution property for \mathbb{Q} follows from the fact that linear programming can be solved (and a concrete solution written down) in polynomial time while the property for \mathbb{Z} has been proven by Papadimitriou [41]. This example is interesting in several respects. First of all, the constants in \mathbb{Q}_c are, of course, linear. Furthermore, we know (from Example 4) that not even the language $\Gamma = \{(x, y, z) \in \mathbb{Z}^3 \mid x + y = z\}$ is ω -categorical; the same can be proved for the domain \mathbb{Q} . Thus, the methods in Section 3 and 4 are not applicable in this case.

We illustrate the small solution property with a different example: the RCC-5 formalism. The RCC formalisms [42] are designed for reasoning about spatial regions and they are the basis for a large part of the work in *qualitative spatial reasoning* (QSR). There are several variants such as RCC-23, RCC-8, and RCC-5. We concentrate on the simplest formalism RCC-5. The interpretation of the five basic relations in RCC-5 is given in Figure 1 and it is easy to see that they constitute a partition scheme. The choice of objects is important in RCC-5 and different choices may give rise to different computational problems. A (slightly degenerated) example is if the set of regions only contains one member. In this case, all basic relations except EQ are empty and this makes the CSP problem for the power set of the partition scheme tractable. If we instead assume that the regions are non-empty regular subsets of an infinite topological space, then the very same problem is NP-hard [43]. In the sequel, we consider the variant of RCC-5 where the objects are non-empty subsets of an infinite set, e.g., of \mathbb{N} . We denote this variant by $\text{RCC-5}_{\text{Set}}$ and we let \mathcal{R} be the corresponding set of basic relations. This particular interpretation is interesting since it can be viewed as the least restricted variant

of RCC-5: if there is a solution to an RCC-5 instance when the regions are taken from some set X that does not contain the empty set, then there is a solution where the regions are taken from $2^{\mathbb{N}} \setminus \{\emptyset\}$. This is well-known and it can quite easily be proved by methods similar to those used in Drakengren and Jonsson [20] (which, incidentally, also have inspired the proof of Proposition 10 below). Further discussions concerning different interpretations of RCC-5 and other spatial formalisms can be found in [6, 21, 36].

We first establish that the methods in Sections 3 and 4 are not applicable. One could do this by analysing the automorphism group and conclude that $\text{RCC-5}_{\text{set}}$ is not ω -categorical. A simpler way is the following (but it is tacitly based on the assumption $\text{P} \neq \text{NP}$). Let $\Gamma = \mathcal{R} \cup \{\neq\}$ where \neq equals $\bigcup \mathcal{R} \setminus \{\text{EQ}\}$. It is known that $\text{CSP}(\Gamma)$ is in P [29, 43]. We extend Γ with one constant: $\Gamma' = \Gamma \cup \{0, 1, 2\}$. Consider the constraints $\{y\{\text{PP}\}z, \{0, 1, 2\}(z)\}$. It is clear that if s is a solution, then $s(y) \in 2^{\{0,1,2\}} \setminus \{\emptyset, \{0, 1, 2\}\}$, i.e. there are 6 distinct possible choices for the variable y . This implies that there is a straightforward polynomial-time reduction from 6-COLOURABILITY to $\text{CSP}(\Gamma')$ (since the relation \neq is in Γ') and, consequently, that $\text{CSP}(\Gamma')$ is NP-complete. If Theorem 4 or Theorem 5 were applicable, then $\text{CSP}(\Gamma')$ would be polynomial-time solvable.

Proposition 10 *Let $D = 2^{\mathbb{N}} \setminus \{\emptyset\}$. The constraint language $\mathcal{R}^{\vee} \cup D_c$ has the small solution property and $\text{CSP}(\mathcal{R}^{\vee} \cup D_c)$ is in NP.*

Proof. By Lemma 9, it is sufficient to show that $\mathcal{R} \cup D_c$ has the small solution property. Let $I = (V, C)$ be a satisfiable instance of $\text{CSP}(\mathcal{R} \cup D_c)$ with solution $s : V \rightarrow 2^{\mathbb{N}} \setminus \{\emptyset\}$. Construct a new instance $I' = (V', C')$ as follows.

Step 1. Remove every $x\{\text{EQ}\}y$ constraint: this can be done by collapsing the variables x and y (we leave the obvious details of this step to the reader).

Step 2. Replace every $x\{\text{PP}\}y$ constraint with $y\{\text{PP}\}x$.

Step 3. Remove every $x\{\text{PO}\}y$ constraint by replacing it with

$$\begin{array}{lll} z_1\{\text{DR}\}z_2 & z_2\{\text{DR}\}z_3 & z_3\{\text{DR}\}z_1 \\ z_1\{\text{PP}\}x & z_1\{\text{DR}\}y & \\ z_2\{\text{PP}\}x & z_2\{\text{PP}\}y & \\ z_3\{\text{PP}\}y & z_3\{\text{DR}\}x & \end{array}$$

where z_1, z_2, z_3 are fresh variables.

Note that I' is still a satisfiable instance of $\text{CSP}(\mathcal{R} \cup D_c)$ and that the only non-unary relations that appear in I are DR and PP. Additionally note that if there is a solution to I with codomain S , then there is a solution to I' with codomain S .

We say that two variables u, v in I' are *PP-connected* if there exists a sequence of variables w_1, \dots, w_p such that

1. $w_1 = u$,
2. $w_p = v$, and
3. $w_i\{\text{PP}\}w_{i+1} \in C'$ for all $1 \leq i < p$.

Note that if u and v are PP-connected, then in any solution s' of I' we have that $(s'(u), s'(v)) \in \text{PP}$.

Let T denote the number of elements in the largest unary relation appearing in I . If u is PP-connected with some variable v and $U(v) \in C$, then we know that $|s'(u)| < T$ for any solution s' to I' . We prove that at most $|V'| \cdot T$ different elements are needed for representing a solution by induction over the number of variables in V' . This implies the result by reasoning as follows: we can without loss of generality assume that the set of possible values is

$2^{\{1, \dots, |V'| \cdot T\}} \setminus \{\emptyset\}$. To represent such a value, we need at most $|V'| \cdot T$ bits if we view each value as a 0/1-vector where the i :th component equals 1 if and only if i is a member of the value. Hence, $\text{CSP}(\mathcal{R} \cup D_c)$ has the small solution property since $|V| \leq \|I'\| \leq \|I\|$ and $T \leq \|I'\| \leq \|I\|$.

Basis step. If $|V'| = 1$ and $V = \{v\}$, then either one value is sufficient (if v is not constrained by a unary relation) or T values are sufficient (otherwise).

Induction hypothesis. Assume the claim holds when $|V'| = p$.

Induction step. We show the claim when $|V'| = p+1$. Choose a variable $v \in V$ such that v is maximal with respect to PP-connectedness, i.e. v is not PP-connected to any other variable. By the induction hypothesis, we need at most pT values for the instance $I' \setminus \{v\}$. If there exists $U(v) \in C$, then we need at most T values for v which gives us at most $pT + T = (p+1)T$ values in total. If there is no $U(v) \in C$, then we need at most one additional value for v so we need at most $pT + 1 \leq (p+1)T$ values in total. To see this, v may (in the worst case) be PP-connected to every other variable and v must (by the induction hypothesis) contain at least pT different values. However, it must also be a strict superset of the other variables and this is accomplished by adding one fresh element. \square

The basic proof idea of Proposition 10 is to analyse the growth of the variables that are not constrained by any unary relation. Clearly, if a variable v is constrained by a constant relation $\{c\}$, then any solution must satisfy $s(v) = c$ and $\|c\| \leq \|I\|$. Otherwise, PP-connectedness gives a way of estimating the size of the contents of the other variables. This idea can readily be extended to other classes of relations that are related to RCC-5 such as (certain variants of) *set relations* (cf. Bodirsky and Hils [7] and the references in their paper), and it can also be generalised in other directions. An interesting observation is that the NP membership results for RCC5 and RCC8 with polygonal regions in the plane by Li et al. [35] is implicitly based on the small solution property. Here, the representational size of the regions are analysed and bounded by exploiting a particular parameter that is related to embeddings of planar graphs in the plane. Another interesting observation is that Li [34] uses concepts that are similar to PP-connectedness when constructing different realisations of the RCC8 formalism. This may indicate that the approach taken in the proof of Proposition 10 may quite easily be adapted to other spatial formalisms.

We conclude this section by a few observations concerning the small solution property. First of all, it is important to realise that the converse of Lemma 8 does not necessarily hold. To see this, define the rapidly increasing function

$$\text{Tower}(n) = \underbrace{2^{2^{\cdot^{\cdot^{\cdot}}}}}_{n \text{ times}}.$$

Clearly, $\log(\text{Tower}(n))$ grows faster than any polynomial in n . Now consider the constraint language $\Gamma = \{U_1, U_2, \dots\}$ where $U_i = \{x \in \mathbb{N} \mid x = \text{Tower}(i)\}$. Checking if there an instance of $\text{CSP}(\Gamma)$ is satisfiable or not can trivially be solved in polynomial time if U_1, U_2, \dots are represented in a reasonable way—for instance, if U_i is represented by the number i written in binary. Thus, $\text{CSP}(\Gamma)$ is in NP, too. It is obvious, though, that Γ does not have the small solution property if we represent the natural numbers in binary.

Another important observation is that it is *not* sufficient to verify that Γ itself has the small solution property—one need to verify that $\Gamma \cup D_c$ has the small solution property. We exemplify by using the

$$\begin{aligned}
X\{\text{PP}\}Y &\Leftrightarrow X \subset Y \\
X\{\text{PP}^{\sim}\}Y &\Leftrightarrow X \supset Y \\
X\{\text{DR}\}Y &\Leftrightarrow X \cap Y = \emptyset \\
X\{\text{PO}\}Y &\Leftrightarrow \exists a, b, c : a \in X, a \notin Y, b \in X, b \in Y, c \notin X, c \in Y \\
X\{\text{EQ}\}Y &\Leftrightarrow X = Y
\end{aligned}$$

Figure 1. The five basic relations of RCC-5.

relation $R = \{(x, y) \in \mathbb{N}^2 \mid x = 2^{y-1}\}$. The constraint language $\{R\}$ has the small solution property since every instance has the solution that assigns 1 to every variable. However, $\text{CSP}(\{R, \{2\}\})$ does not have small solutions. Consider the instance (V, C) where $V = \{x_0, \dots, x_n\}$ and

$$C = \{\{2\}(x_0), R(x_1, x_0), R(x_2, x_1), \dots, R(x_n, x_{n-1})\}.$$

It is easy to verify that (V, C) is solvable and every solution $s : V \rightarrow \mathbb{N}$ must satisfy $s(x_n) = \text{Tower}(n)$.

Finally, we want to emphasise once again that the choice of exact interpretation and representation of relations and domain elements is extremely important. Bodirsky and Chen [6] have presented an interpretation of RCC-5 that is homogeneous. In this case, adding constants preserves computational complexity (up to polynomial-time reductions) by Theorem 5 (given that relations and domain elements are represented in a suitable way). We know from earlier examples that this does not hold for RCC-5_{Set}.

6 DISCUSSION

We have presented three different methods for analysing the complexity of qualitative CSPs extended with finite unary relations, and identifying additional general methods for studying the complexity of such CSPs is an obvious research direction. One should observe that restricting oneself to *finite* unary relations may be reasonable in certain cases but not in others. For instance, a substantial part of the literature on temporal reasoning is concerned with TCSPs and the simple temporal problem (STP) [19]: the basic binary relations here are expressions $a \leq x - y \leq b$ (where $a, b \in \mathbb{Q}$ and x, y are variables) and unary relations $a \leq x \leq b$ (which are either constants or infinite unary relations depending on the choice of a and b). It is easy to see that extending a formalism with (non-trivial) infinite unary relations may yield an easier computational problem than adding finite unary relations. For instance, PA extended with the finite unary relation $\{0, 1, 2\}$ is NP-hard since the disequality relation \neq is in PA, while PA extended with the infinite unary relation $\{x \in \mathbb{Q} \mid 0 \leq x \leq 2\}$ is tractable [28]. Thus, it would be interesting to study the computational complexity of CSPs extended with non-finite unary relations.

Our methods I and II are based on certain model-theoretical properties of the underlying constraint languages. While methods based on model theory and universal algebra have been very common when studying CSPs from the viewpoint of theoretical computer science [2, 4, 12], such methods have been less popular within the AI community (with some notable exceptions such as Huang [26]). Thus, we take the opportunity to discuss these methods in slightly more detail.

Method I. (model-complete cores) The main obstacle for applying method I is the need for computing orbit-defining formulas efficiently. In fact, it is not even known if this problem is decidable

or not in the general case. Studying this problem is a very important future research direction. In cases where we do not know how to efficiently generate orbit-defining formulas, there are (at least) two possible workarounds. The first one is proposed by Bodirsky [5, Sec. 7]: if the set of possible constants is finite, then an orbit-defining formula for these constants can be computed off-line and subsequently be used without additional cost. Another workaround is to sacrifice polynomial-time equivalence and allow more time for computing the orbit-defining formula. If the problem at hand is NP-hard, then a (preferably mildly) exponential algorithm can be acceptable. In both cases, algorithmic methods for generating orbit-defining formulas would be helpful. We note, on the positive side, that related definability problems have recently been successfully addressed, cf. Bodirsky et al. [8]. Their methods are interesting since they combine methods taken from universal algebra, Ramsey theory, and topological dynamics.

Method II. (homogeneity) We have chosen to present the results when the constraint language is restricted to partition schemes. This is convenient but not inherently necessary—generalisations to (for instance) higher-arity relations are possible. One should consequently not view our results as the only possible way of exploiting homogeneity: how to exploit homogeneity must be decided on a case-by-case basis.

Given a structure Γ , it may be difficult to verify that it is indeed homogeneous. Here, one should note that if Γ contains a finite number of relations, the domain of Γ is countably infinite, and Γ is homogeneous, then Γ is ω -categorical. This is a consequence of Theorem 1 and the details are to be found in Macpherson [39]. A first step is thus to verify the ω -categoricity of Γ , and this can quite often be accomplished by using Theorem 1. If Γ is ω -categorical, then Γ is homogeneous *if and only if* every formula in $\text{Th}(\Gamma)$ is equivalent to a quantifier-free formula (see, for instance, Macpherson [39]). This gives an alternative way of proving homogeneity than using the automorphism-based definition directly. This also clarifies the connections between method I and method II: recall that Γ is model-complete if and only if every formula in $\text{Th}(\Gamma)$ is equivalent to an existential formula.

Another approach for using homogeneity is to construct suitable homogeneous structures “from scratch”. The main tool for this is *Fraïssé amalgamation*. The details are outside the scope of this article: Macpherson [39] outlines the approach and concrete constructions for RCC5 and RCC8 can be found in Bodirsky and Chen [6] and Bodirsky and Wöflf [9], respectively. One should note that amalgamation is quite common in the literature on CSPs and related problems; however, it is often referred to as the *patchwork property* [26, 38, 44]

ACKNOWLEDGEMENTS

We thank Manuel Bodirsky, Fredrik Heintz, and Johan Thapper for providing valuable input to this paper.

REFERENCES

- [1] Libor Barto. The dichotomy for conservative constraint satisfaction problems revisited. In *Proc. 26th Annual IEEE Symposium on Logic in Computer Science (LICS-2011)*, pages 301–310, 2011.
- [2] Libor Barto and Marcin Kozik. Constraint satisfaction problems solvable by local consistency methods. *J. ACM*, 61(1):3:1–3:19, 2014.
- [3] Manuel Bodirsky and M. Grohe. Non-dichotomies in constraint satisfaction complexity. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP-2008)*, pages 184–196, 2008.
- [4] Manuel Bodirsky and Jan Kára. The complexity of temporal constraint satisfaction problems. *J. ACM*, 57(2), 2010.
- [5] Manuel Bodirsky. Cores of countably categorical structures. *Logical Methods in Computer Science*, 3(1):1–16, 2007.
- [6] Manuel Bodirsky and Hubie Chen. Qualitative temporal and spatial reasoning revisited. *J. Log. Comput.*, 19(6):1359–1383, 2009.
- [7] Manuel Bodirsky and Stefan Wöflfl. Tractable set constraints. *J. Artif. Intell. Res. (JAIR)*, 45:731–759, 2012.
- [8] Manuel Bodirsky, Michael Pinsker, and Todor Tsankov. Decidability of definability. *J. Symb. Log.*, 78(4):1036–1054, 2013.
- [9] Manuel Bodirsky and Stefan Wöflfl. RCC8 is polynomial on networks of bounded treewidth. In *Proc. 22nd International Joint Conference on Artificial Intelligence (IJCAI-2011)*, pages 756–761, 2011.
- [10] Andrei Bulatov. Tractable conservative constraint satisfaction problems. In *Proceedings of the 18th annual IEEE symposium on logic in computer science*, pages 321–???, 2003.
- [11] Andrei Bulatov. Conservative constraint satisfaction re-revisited. *Journal of Computer and System Sciences*, 82(2):347–356, 2016.
- [12] Andrei Bulatov, Peter Jeavons, and Andrei Krokhin. Classifying the computational complexity of constraints using finite algebras. *SIAM J. Comput.*, 34(3):720–742, 2005.
- [13] Andrei A. Bulatov, Andrei A. Krokhin, and Peter G. Jeavons. Classifying the complexity of constraints using finite algebras. *SIAM Journal on Computing*, 34:720–742, 2005.
- [14] Peter J. Cameron. *Oligomorphic permutation groups*. Cambridge University Press, Cambridge, 1990.
- [15] Georg Cantor. Über unendliche, lineare Punktmannigfaltigkeiten. *Mathematische Annalen*, 23:453–488, 1884.
- [16] David Cohen, Peter Jeavons, Peter Jonsson, and Manolis Koubarakis. Building tractable disjunctive constraints. *Journal of the ACM*, 47(5):826–853, 2000.
- [17] Anthony G. Cohn and Jochen Renz. Qualitative spatial representation and reasoning. In *Handbook of Knowledge Representation*, pages 551–596. Elsevier, 2008.
- [18] Daniel de Leng and Fredrik Heintz. Qualitative spatio-temporal stream reasoning with unobservable intertemporal spatial relations using landmarks. In *Proc. 30th AAAI Conference on Artificial Intelligence (AAAI-2016)*, pages ...–..., 2016.
- [19] Rina Dechter, Itay Meiri, and Judea Pearl. Temporal constraint networks. *Artif. Intell.*, 49:61–95, 1991.
- [20] Thomas Drakengren and Peter Jonsson. Reasoning about set constraints applied to tractable inference in intuitionistic logic. *Journal of Logic and Computation*, 8(6):855–875, 1998.
- [21] Ivo Dütsch, Hui Wang, and Stephen McCloskey. A relation-algebraic approach to the region connection calculus. *Theor. Comput. Sci.*, 255(1–2):63–83, 2001.
- [22] Stella Giannakopoulou, Charalampos Nikolaou, and Manolis Koubarakis. A reasoner for the RCC-5 and RCC-8 calculi extended with constants. In *Proc. 28th AAAI Conference on Artificial Intelligence (AAAI-2014)*, pages 2659–2665, 2014.
- [23] Pavol Hell and Jaroslav Nešetřil. *Graphs and Homomorphisms*. Oxford University Press, Oxford, 2004.
- [24] Robin Hirsch. Expressive power and complexity in algebraic logic. *Journal of Logic and Computation*, 7(3):309–351, 1997.
- [25] Wilfrid Hodges. *Model theory*. Cambridge University Press, 1993.
- [26] Jinbo Huang. Compactness and its implications for qualitative spatial and temporal reasoning. In *Proc. 13th International Conference on Knowledge Representation and Reasoning (KR-2012)*, 2012.
- [27] Peter Jeavons. On the algebraic structure of combinatorial problems. *Theoretical Computer Science*, 200:185–204, 1998.
- [28] Peter Jonsson and Christer Bäckström. A unifying approach to temporal constraint reasoning. *Artif. Intell.*, 102(1):143–155, 1998.
- [29] Peter Jonsson and Thomas Drakengren. A complete classification of tractability in RCC-5. *J. Artif. Intell. Res.*, 6:211–221, 1997.
- [30] Peter Jonsson and Victor Lagerkvist. Upper and lower bounds on the time complexity of infinite-domain CSPs. In *Proc. 21st International Conference on Principles and Practice of Constraint Programming (CP-2015)*, pages 183–199, 2015.
- [31] Markus Junker and Martin Ziegler. The 116 reducts of $(\mathbb{Q}, <, a)$. *J. Symb. Log.*, 73(3):861–884, 2008.
- [32] Arne Kreuzmann and Diedrich Wolter. Qualitative spatial and temporal reasoning with AND/OR linear programming. In *Proc. 21st European Conference on Artificial Intelligence (ECAI-2014)*, pages 495–500, 2014.
- [33] C. Langford. Some theorems on deducibility. *Annals of Mathematics*, 28:16–40, 1927.
- [34] Sanjiang Li. On topological consistency and realization. *Constraints*, 11(1):31–51, 2006.
- [35] Sanjiang Li, Weiming Liu, and Sheng-sheng Wang. Qualitative constraint satisfaction problems: An extended framework with landmarks. *Artif. Intell.*, 201:32–58, 2013.
- [36] Sanjiang Li and Mingsheng Ying. Extensionality of the RCC8 composition table. *Fundam. Inform.*, 55(3–4):363–385, 2003.
- [37] Gérard Ligozat and Jochen Renz. What is a qualitative calculus? A general framework. In *Proceedings of PRICAI*, pages 53–64, 2004.
- [38] Carsten Lutz and Maja Milicic. A tableau algorithm for description logics with concrete domains and general tboxes. *J. Autom. Reasoning*, 38(1–3):227–259, 2007.
- [39] Dugald Macpherson. A survey of homogeneous structures. *Discrete Mathematics*, 311(15):1599–1634, 2011.
- [40] Bernhard Nebel and Hans-Jürgen Bürckert. Reasoning about temporal relations: A maximal tractable subclass of Allen’s interval algebra. *J. ACM*, 42(1):43–66, 1995.
- [41] Christos Papadimitriou. On the complexity of integer programming. *J. ACM*, 28(4):765–768, 1981.
- [42] David A. Randell, Zhan Cui, and Anthony G. Cohn. A spatial logic based on regions and connection. In *KR*, pages 165–176, 1992.
- [43] Jochen Renz and Bernhard Nebel. On the complexity of qualitative spatial reasoning: A maximal tractable fragment of the region connection calculus. *Artif. Intell.*, 108(1–2):69–123, 1999.
- [44] Michael Sioutis and Manolis Koubarakis. Consistency of chordal RCC-8 networks. In *Proc. 24th International Conference on Tools with Artificial Intelligence (ICTAI-2012)*, pages 436–443, 2012.
- [45] Frank Wolter and Michael Zakharyashev. Spatio-temporal representation and reasoning based on RCC-8. In A. G. Cohn, F. Giunchiglia, and B. Selman, editors, *Proceedings of the 7th International Conference on Principles of Knowledge Representation and Reasoning*, pages 3–14. Morgan Kaufmann, 2000.

Dynamic Choice of State Abstraction in Q-Learning

Marco Tamassia and Fabio Zambetta and William L. Raffe and Florian ‘Floyd’ Mueller and Xiaodong Li¹

Abstract. Q-learning associates states and actions of a Markov Decision Process to expected future reward through online learning. In practice, however, when the state space is large and experience is still limited, the algorithm will not find a match between current state and experience unless some details describing states are ignored. On the other hand, reducing state information affects long term performance because decisions will need to be made on less informative inputs. We propose a variation of Q-learning that gradually enriches state descriptions, after enough experience is accumulated. This is coupled with an ad-hoc exploration strategy that aims at collecting key information that allows the algorithm to enrich state descriptions earlier. Experimental results obtained by applying our algorithm to the arcade game Pac-Man show that our approach significantly outperforms Q-learning during the learning process while not penalizing long-term performance.

1 Introduction

Planning and learning under uncertainty are fundamental problems in artificial intelligence. A framework to address such problems is the Markov Decision Process (MDP) [1]. MDPs are based on the Markov assumption which states that it is sufficient to know the current state of the environment to make predictions about the outcome of actions. One way for an agent to learn useful information about the environment dynamics is by interacting with it, in a sequence of observations of state and action. Based on the Markov assumption, Temporal Difference (TD) algorithms [2] encode useful information about the environment in the form of associations of states to utilities. For example, Q-learning, one of the most popular TD algorithms, associates state-action pairs to future rewards [3]. When a TD agent needs to make a decision, it will choose the action that is likely to yield the highest long-term utility value according to previous experience.

If states are rich in information, in the early stages of the learning process, the agent knowledge of the environment is sparse. If the agent considers every feature making up the state, it will take a considerable amount of time to learn associations for all of the many possible states, especially if outcomes are stochastic. To quote Andre and Russell, “Without state abstraction, every trip from A to B is a new trip” [4]. During this learning period, an agent may make blind decisions due to “details” in the state preventing an exact match with past experience. TD agents lack the ability to use knowledge of states similar to the current one. This research problem falls under the umbrella of Transfer learning [5].

The most common approach to address this issue is to use linear approximation [6], [7]. In this instance, an agent only has to learn the weights of the linear transformation mapping state features to utility.

However, a linear approximation may not be sufficient if non-linear dynamics exist in the environment. In this case, sparsity issues are

addressed by stripping states of “superfluous” details. This process is called *state abstraction*, and it consists in aggregating states. By only considering the most important information and ignoring details, two states that are effectively different will appear the same, inducing a partitioning of the state space. The drawback of this, however, is that if the information used to encode states is not rich enough, the agent will not be able to make informed decisions. Examples of this approach are coarse coding and tile coding [1].

State abstraction has attracted attention in the reinforcement learning community in the past two decades. Most of the literature on the subject focuses on choosing an abstraction prior to the actual learning [8]–[11]. McCallum’s work, however, explored online state abstraction, which is also the focus of our work. [12]–[14].

In this paper we propose an algorithm that shifts from coarse partitionings to more fine-grained ones through time. The choice of which partitioning to use is done at every step and can be different from state to state, allowing for more flexible learning. The criteria used by our algorithm to decide when to enrich state information is to compare the confidence interval of utility estimates. The idea is that, at the beginning of the process, most decisions are made using coarse partitionings while, in the long run, more choices are made with more informative partitionings. To our knowledge, this is the first attempt to combine both coarse and fine-grained partitionings online.

We evaluate our algorithm by comparing it with Q-learning in the context of the video game Pac-Man. Our experiments show that the proposed algorithm produces better performance than fixed state-size Q-learning during the learning phase. We also propose a strategy to direct exploration in a way that allows the algorithm to switch to fine-grained abstractions earlier. Experiments show that this strategy produces better performance than the standard ϵ -Greedy.

The paper is structured as follows: in Section 2 we introduce Markov Decision Processes and Q-learning and provide an overview of the relevant literature; in Section 3 we introduce our algorithm and our ad-hoc exploration strategy; in Section 4 we detail the setup of our experiments and in Section 5 we report the results of our experiments.

2 Background and Notation

A Markov Decision Process is formally defined by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, T, R, \gamma)$, where \mathcal{S} is the set of all possible *states* of the environment; \mathcal{A} is the set of *actions* the agent can perform; $T : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{P}(\mathcal{S})$ associates a state-action pair to a *next state probability distribution*; $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ associates a state-action pair to a *reward*; $\gamma < 1$ is a *discount factor* used to decrease the value of future rewards.

An agent interacting with the environment, tries to maximize the cumulative reward collected over time. While interacting with the environment, the agent collects information: at time t , after observing state s_t , it decides which action a_t to perform and observes the reward r_t it receives and the next state s_{t+1} to which the environment tran-

¹ RMIT University, Melbourne, Australia, email: first.last@rmit.edu.au

sitions. Collected information can be used to make informed future decisions.

An effective way to achieve optimal decision making is to compute, for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$, an estimate $Q^*(s, a)$ of the expected cumulative reward to be expected by taking action a while in state s and behaving optimally afterwards. Then, the optimal policy π^* is defined as: $\pi^*(s) = \arg \max_{a \in \mathcal{A}} Q^*(s, a)$. The Q-learning algorithm [3] updates of the estimates after each step of the agent according to the following rule:

$$Q(s_t, a_t) \stackrel{\alpha}{\leftarrow} r_{t+1} + \gamma \max_{a \in \mathcal{A}} Q(s_{t+1}, a) - Q(s_t, a_t),$$

where $x \stackrel{\alpha}{\leftarrow} y$ is short for $x \leftarrow x + \alpha y$ and $0 < \alpha \leq 1$ is a learning factor.

Since the agent starts the process with no information about the environment, it needs to balance exploration of the environment with exploitation of the current knowledge. This is to avoid focusing on what the agent believes to be the best action and missing the actual best action due to incomplete knowledge. A common exploration strategy, which we also use in this work, is called ϵ -greedy and is defined as follows:

$$\pi_\epsilon(s) = \begin{cases} \arg \max_{a \in \mathcal{A}} Q(s, a) & \text{with prob. } 1 - \epsilon \\ \text{random}(\mathcal{A}) & \text{with prob. } \epsilon \end{cases},$$

where $0 \leq \epsilon \leq 1$ balances exploration and exploitation.

Q-learning is said to learn “off-policy”; this means that, in the long run, its estimates $Q(\cdot, \cdot)$ approximate the exact values $Q^*(\cdot, \cdot)$ regardless of what policy the agent is following². This allows an agent to learn the optimal policy while using a sub-optimal exploration-oriented policy, such as ϵ -greedy.

Related work

Significant research effort has been directed in state abstraction. However, most of the work has been focused on choosing an abstraction prior to the learning phase. Many papers fall in this area: [8] use hypothesis testing to discover useful abstractions using Q-values learned in multiple runs; [9] select the features that are useful to reproduce the behavior of a given set of demonstrations; [10] use time-series data to select among the models provided by a human expert; [11] provide theoretical guarantees on the used abstraction, but their approach requires the use of value iteration, an expensive, model-based algorithm [1].

In the recent years there has been work at the intersection of Deep Learning and Reinforcement Learning. In the work by Mnih *et al.*, neural networks are able to learn features of the game state from very unstructured data (such as pixels) [15]. We propose a different approach: rather than learning features over time from raw data, we suggest that, given a set of meaningful features, an algorithm can use more and more of them over time to make its decisions, and that this can help achieve better performance at the early stages of the learning process, when knowledge of the environment is still scarce.

Literature covers two different, principled approaches at online learning of state abstractions:

- adaptively expanding memories to store past information, which is helpful when past information is relevant to make good decisions [12]–[14];
- adaptively split tiles in tile coding [1], [16]–[18], which works well when state features range widely in ordinal values.

The first of these approaches are all works of McCallum. In [12], he expands temporal memory to distinguish variations in rewards, and does so via hypothesis testing; this approach, however, is slow because memory is expanded one step at a time. In [13] he proposes to store raw history, so when memory is expanded, history can be re-analysed to properly compute values; this approach, reportedly, does not handle noise very well. In [14], he proposes to use, along with stored history, a tree to know how deep (how far back in history) one needs to look to distinguish situations: branches are added when a statistical test says that samples come from two different distributions³.

The second approach focuses on tile coding. Tile coding (TC) [1], [16] is a technique to extract useful features from large state spaces including widely varying ordinal features. The idea is to produce multiple discretizations (tilings) of the state features with different offsets: for each discretization, every tile becomes a state feature all of which but one are set to zero. To expand on this, Whiteson *et al.* propose to adaptively split tiles so as to maximize changes in the value function or in the policy during learning [17]. More recently, a paper by Scopes and Kudenko proposes to split tiles that are closer to the optimal transition path and suggests that perpendicular tiles to the optimal path can achieve better performance than square tiles.

The approach we propose complements the above mentioned, because we focus on augmenting the set of features used to describe the state, as opposed to adding information from the past or augmenting the granularity within the features. the video game Pacman is one example of scenario where the mentioned approaches would not work well. To an agent that already has information about food and ghosts, it is more useful to know where the closest power capsule is than it is to know where food and ghosts were in the past. Tile coding (and derivatives) are also ill-suited to Pacman because the features have few possible values each, so they would not allow for many tiles; furthermore, half of the features are not ordinals, making TC inapplicable on them. TC could possibly be applied to the original features, but this would mean counting pills per tile, and this, to the best of our knowledge, has not been tried before and deserves a deep analysis per se.

3 Dynamic Abstraction Choice

In reality, because environments are often stochastic, a number of trials are necessary for each state and action pair to evaluate reasonable estimates. In particular, in the early stages of an agent’s life, its knowledge is rather sparse, often leading to blind decisions. To deal with this, a common approach is not to model the entire MDP, but a simplified one obtained by reducing the state space size via state aggregation [5], [20]. By means of carefully engineered state aggregations, Q-learning generalizes well over the little information it has.

However, it is desirable that in the long-run, the agent makes its decisions considering all the nuances of each state, rather than based on coarse aggregations. More information on the state allows the agent to make more informed decisions.

We propose a novel algorithm to achieve the advantages of both situations, at the cost of a slight increase in processing time. In the following, we refer to abstractions as functions mapping a state to an aggregation of states. Each abstraction induces an abstracted state space of smaller size than the original one and, consequently, a smaller Q-table. However, such Q-tables do not need to be memorized since they can be inferred by appropriately aggregating entries of the original Q-table.

² as long as there is a non-zero probability of visiting each state

³ The test used is Kolmogorov-Smirnov [19]

Algorithm 1: Multi-Abstraction Q-learning algorithm for abstraction shifting. \uplus indicates a multiset sum; a multiset is a set where information about the number of occurrences of each element is preserved. \bar{X} indicates the sample average. Procedure CI computes the confidence interval of the mean of the given sample.

Input : Learning rate α
Input : Exploration parameter ϵ
Input : Abstractions $\beta_1 > \beta_2 > \dots > \beta_m$
Input : Default Q-value, initQ
Input : Significance level for t-tests, σ

```

1 for  $s, a \in \mathcal{S} \times \mathcal{A}$  do
2    $Q(s, a) \leftarrow \text{initQ}$ 
3    $H(s, a) \leftarrow \text{empty list}$  // history of  $Q(s, a)$ 
4 end
5  $s \leftarrow \text{observe state}$ 
6 repeat
7    $j^* \leftarrow m$ 
8   found  $\leftarrow \text{false}$ 
9   for  $j \leftarrow 1 \dots m$  do
10     $\xi \leftarrow \{s' \in \mathcal{S} \mid \beta_j(s') = \beta_j(s)\}$  // siblings
11    for  $a \in \mathcal{A}$  do
12      $X_a^j \leftarrow \uplus_{s' \in \xi} H(s', a)$  // samples of siblings
13      $\perp_a^j, \top_a^j \leftarrow \text{CI}(\sigma, X_a^j)$  // lower and upper bounds
14    end
15     $a^* \leftarrow \arg \max_{a \in \mathcal{A}} \bar{X}_a^j$ 
16    if  $\perp_{a^*}^j > \top_a^j$  for all  $a \in \mathcal{A}, a \neq a^*$  then
17      $j^* \leftarrow j - 1$ 
18     found  $\leftarrow \text{true}$ 
19     go to line 22
20    end
21  end
22   $a^* \leftarrow \begin{cases} \arg \max_{a \in \mathcal{A}} \bar{X}_a^{j^*} & \text{with prob. } 1 - \epsilon \\ \text{random}(\mathcal{A}) & \text{with prob. } \epsilon \end{cases}$ 
23  perform action  $a^*$ 
24   $s' \leftarrow \text{observe state}$ 
25   $r \leftarrow \text{receive reward}$ 
26   $\hat{q} \leftarrow r + \gamma \max_{a' \in \mathcal{A}} Q(s', a')$ 
27   $Q(s, a^*) \leftarrow \hat{q} - Q(s, a^*)$ 
28  append  $\hat{q}$  to  $H(s, a^*)$ 
29   $s \leftarrow s'$ 
30 until apocalypse

```

The algorithm we propose, “Multi-Abstraction Q-learning”, is presented in pseudocode in Algorithm 1. Multi-Abstraction Q-learning is given a list of abstractions of decreasing granularity, and maintains the Q-table associated with the original state representation. At decision time, the algorithm chooses the most granular abstraction whose Q-values are precise with sufficient confidence.

Formally, an abstraction is defined as $\beta_i : \mathcal{S} \rightarrow \mathcal{S}_i$. We also introduce an ordering for abstractions, based on their granularity. Formally, an abstraction β is more granular than a second abstraction β' (denoted $\beta > \beta'$) if both the following conditions hold:

- Any two states $s, s' \in \mathcal{S}$ mapped to the same abstracted state by (the more granular) abstraction β are also mapped to the same abstracted state by (the more coarse) abstraction β' . Formally:

$$\forall s, s' \in \mathcal{S} . \beta(s) = \beta(s') \Rightarrow \beta'(s) = \beta'(s')$$

- There is at least a pair of states $s, s' \in \mathcal{S}$ that are mapped to different abstracted states by (the more granular) abstraction β but that are mapped to the same abstracted state by (the more coarse) abstraction β' . Formally:

$$\exists s, s' \in \mathcal{S} . \beta(s) \neq \beta(s') \wedge \beta'(s) = \beta'(s')$$

Algorithm 1 is given a list of abstractions of decreasing granularity. The algorithm decides which action to take by choosing the most suitable abstraction at every time step. The chosen abstraction is the most granular one that provides a high confidence that the action with the highest estimated Q-value is actually the best one. Confidence of a state-action pair (s, a) is computed by running a t-test over the history $H(s, a)$ of values that the Q-table entry $Q(s, a)$ has assumed. The steps used to test the confidence of an abstraction are as follows:

1. the action a^* with the highest estimated Q-value is found;
2. the boundaries \perp_a, \top_a of the confidence intervals of all actions a are calculated;
3. for all $a \neq a^*$, test whether $\perp_{a^*} > \top_a$: each test is passed with a $1 - \sigma$ confidence level;
4. if all the tests are passed, it is reasonable to assume that the true Q-value of a^* is actually the highest.

Optimization

The procedure described in Algorithm 1 has some significant inefficiencies. However, notice that they can be overcome and have been introduced in the listing for the purpose of clarity. In the following paragraphs, we briefly explain how to address these issues.

There are two main bottlenecks. The first is at line 12, where the union operation iterates over all the states to find the siblings. This adds a significant amount of computational time to compute the same information repeatedly. In fact, caching the state space partitioning (i.e. the sets of “siblings”) for each abstraction is a better solution. By doing so, the time complexity of retrieving such information is constant at the cost of a linear (in the number of states and abstractions) increase in memory complexity.

The second bottleneck is the procedure at line 13 which computes the confidence intervals. The procedure iterates over the whole set X_a^j of samples at every invocation to compute their mean and variance. An alternative, more efficient solution is to store mean and variance of the Q-value for every state-action pair and update them online [21]. It is, then, possible to efficiently compute mean and variance of a virtual union set by aggregating the stored statistics [22]. This modification makes storing the history of Q-values unnecessary, significantly reducing the space requirements of the algorithm.

Confidence driven exploration

While ϵ -greedy is expected to shrink the confidence intervals in the long run through random exploration, it has no awareness of their existence. It is reasonable to suppose that using a different exploration strategy making use of this knowledge would produce better results. Such a strategy would bias the exploration so as to perform actions whose confidence intervals are preventing the use of the next abstraction. We propose a variation on the traditional ϵ -greedy strategy that integrates such bias. The close-form definition is slightly cumbersome; so, with clarity in mind, we provide the pseudocode instead. The pseudocode in Algorithm 2 describes such procedure and is meant to replace line 22 of Algorithm 1.

Algorithm 2: C.I. driven exploration.

```

1 return  $a^*$            with prob.  $1 - \epsilon_{CI} - \epsilon_R$ 
1 return  $\text{random}(\mathcal{A})$   with prob.  $\epsilon_R$ 
  go to line 11       with prob.  $\epsilon_{CI}$ 
2 if  $j^* = 1$  // most granular abstraction already in use then
3   return  $a^*$  // no exploration needed
4 end
5 if found // acceptable abstraction found then
6    $\hat{j} \leftarrow j^* - 1$  // use it
7 else
8    $\hat{j} \leftarrow m$  // use the most coarse one
9 end
10  $a^* \leftarrow \arg \max_{a \in \mathcal{A}} \overline{X_a^{\hat{j}}}$ 
11  $K_1 = \left\{ a \in \mathcal{A} \mid \overline{T_a^{\hat{j}}} > \overline{X_{a^*}^{\hat{j}}} \right\}$ 
12  $K_2 = \left\{ a \in \mathcal{A} \mid \overline{X_a^{\hat{j}}} > \underline{\hat{j}}_{a^*} \right\}$ 
13 if  $K_1 \neq \emptyset$  and  $K_2 \neq \emptyset$  then
14   return  $\text{random}(K_1 \cup \{a^*\})$ 
15 else if  $K_1 \neq \emptyset$  then
16   return  $\text{random}(K_1)$ 
17 else
18   return  $a^*$ 
19 end

```

This procedure requires two parameters ϵ_{CI} and ϵ_R , which set the probability of exploring versus exploiting, similarly to ϵ -greedy. Unlike ϵ -greedy, however, this procedure performs a second type of exploration. That is, with probability ϵ_{CI} , it selects an action whose confidence interval needs to be reduced in order for the next abstraction to be usable.

Figure 1 shows a qualitative representation of the reasoning behind Algorithm 2. Depicted are the different possible situations in which confidence intervals of actions a^* (action with the highest mean) and a overlap. On top of each subfigure, the behavior of the algorithm in lines 11–12 is reported. Notice that in many cases there will be multiple actions matching the criteria of a : in such cases the choice is random among them. For the sake of conciseness, in the remainder of this section as well as in Algorithm 2 we will refer to the confidence interval of the estimate of the average Q-value of action a simply as the confidence interval of a .

Line 11 of Algorithm 2 captures cases shown in figures 1a, 1b, 1e, 1f, 1g, where the upper bound of the confidence interval of some action a is higher than the average Q-value of the best action a^* . Such actions are stored in set K_1 . Line 12 of Algorithm 2 captures cases shown in figures 1c, 1d, 1e, 1f, 1g, where the average Q-value of some action a is higher than the lower bound of the confidence interval of the best action a^* . Such actions are stored in set K_2 . Algorithm 2 makes the simplifying assumption that further samples will shrink the confidence intervals without moving the average value. Notice that this does not introduce bias since the average value is an unbiased estimate of the mean value. With this assumption in mind, the procedure selects:

- a random action from K_1 if $K_1 \neq \emptyset$ and $K_2 = \emptyset$ because the only way to remove the overlaps of the confidence intervals is to shrink those of the actions in K_1 ;

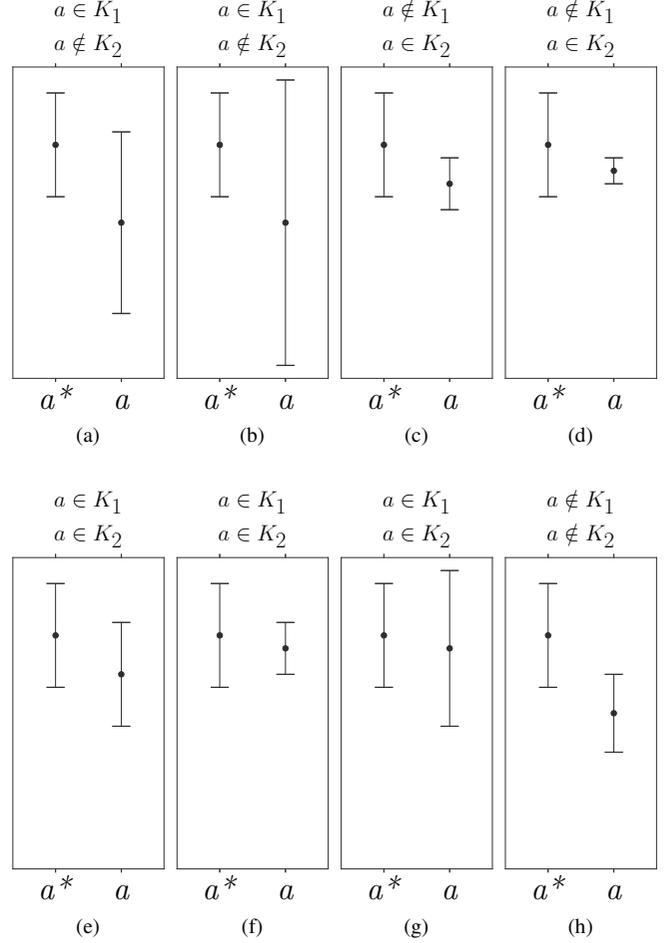


Figure 1: A visual qualitative representation of the possible situations in which confidence intervals overlap. Here, a^* is the action with the highest mean and a is another action. When confidence intervals overlap, the confidence level cannot be guaranteed. On top of each subfigure, the decision made by Algorithm 2 at lines 11–12 is indicated.

- a^* if $K_1 = \emptyset$ and $K_2 \neq \emptyset$ because, while choosing actions from K_2 would also be an effective way to remove the overlaps, choosing a^* is the choice with the highest expected future reward;
- a random action from $K_1 \cup \{a^*\}$ if $K_1 \neq \emptyset$ and $K_2 \neq \emptyset$ because of the same reasons explained above.

Notice that the case where $K_1 = \emptyset$ and $K_2 = \emptyset$ is only possible if all the abstractions are usable, but this case is captured in lines 2–4. Following this strategy, confidence intervals that are overlapping will shrink until they do not overlap anymore, therefore allowing the usage of the next abstraction, until the most fine-grained abstraction is usable.

4 Experiments

We evaluate the effectiveness of our algorithm using Pac-Man, a real-time arcade retro game⁴. Games of this type are of interest to the

⁴ Additional information can be found at http://www.gamasutra.com/view/feature/3938/the_pacman_dossier.php?print=1 (checked on March 3rd, 2016).

scientific Artificial Intelligence community due to the challenges of open-endedness and tight time-constraints they pose [23]. Pac-Man has been used as test-bed in a sizable amount of literature, including [24]–[26]. We adopted the implementation currently used in UC Berkeley to teach AI, originally developed by DeNero and Klein [27]⁵. Our algorithms were implemented in Python using the Numpy library⁶ [28] and parallelized using GNU Parallel⁷ [29].

Pac-Man

In the game, the player controls Pac-Man, an agent moving in a two-dimensional environment whose purpose is to maximise a score. The score increases by collecting food pills while steering clear of ghosts, which will kill Pac-Man when colliding with it. Special capsules in the game area can be picked up by the player that make all the ghosts edible for a limited period of time. If Pac-Man collides with an edible ghost, the ghost dies and reappears at the center of the game area in a threatening (non-edible) state. The level topology used in the experiments is depicted in figure 2. This is not a standard Pac-Man level, but a simpler one provided with the codebase. We chose this because experiments require less computational time.

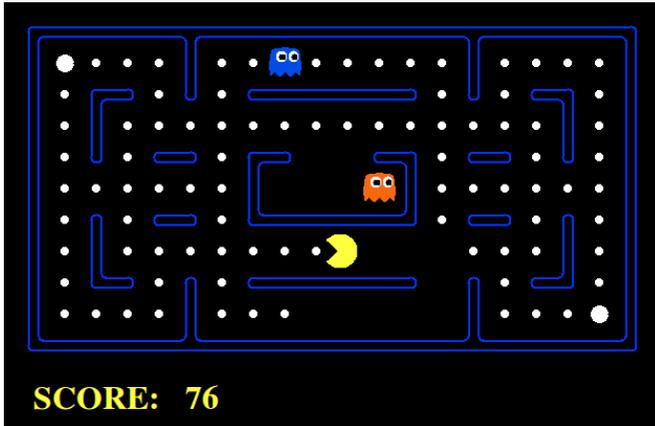


Figure 2: A screenshot of the video game used in the experiments, Pac-Man.

Players receive a score based on their performance. In the implementation we used, Pac-Man receives 10 points for each eaten pill, and it loses 1 point at each time step, while receiving 200 points every time a ghost is killed. Furthermore, 500 points are earned upon victory (i.e. when all the pills have been eaten), while 500 points are lost upon death. These scores have been chosen by the developers of the framework, and we adopt them without changes in this work.

Tests

The first question we wanted to answer is what significance setting σ yields the highest performance in Multi-Abstraction Q-learning (Algorithm 1) with ϵ_{CI} strategy. To answer this question, we tested the algorithm with $\sigma \in \{0.1, 0.2, 0.5, 0.9\}$. All the configurations used food and threatening ghosts information to describe states, successively adding edible ghosts information and, lastly, capsules information;

⁵ Currently available for download at http://ai.berkeley.edu/project_overview.html

⁶ <http://www.numpy.org/>

⁷ <http://www.gnu.org/software/parallel/>

these represents the abstractions β_i in Algorithm 1. In these tests, we set $\epsilon_{CI} = 0.05$ and $\epsilon_R = 0.05$.

Secondly, we wanted to evaluate whether shifting abstractions - from coarse to fine-grained - improves agents performance. To test this, we ran tests on Pac-Man using different agent algorithms:

- Q-learning where states included food and threatening ghosts information;
- Q-learning where states included food, threatening ghosts and edible ghosts information;
- Q-learning where states included food, threatening ghosts, edible ghosts and capsules information.

We compared the performance of these algorithms with those of the best configuration from the previous test, that with $\sigma = 0.9$. The exploration strategy used in these three configurations was ϵ -Greedy with $\epsilon = 0.1$.

We ran tests using each of these algorithms for 30000 consecutive episodes and we measured the reward collected during each of them. We repeated this 50 times and averaged the results. Agent performance is expected to improve over time, as they gather information on the environment: however, improvement rate and final performance depend upon the agent algorithm and its state representation.

The final question we wanted to answer is to what degree the performance of the other experiments are due to Multi-Abstraction Q-learning versus to the ϵ_{CI} -Greedy strategy. To test this, we ran experiments using the following algorithms:

- Multi-Abstraction Q-learning with ϵ -Greedy with $\epsilon = 0.1$;
- Q-learning with ϵ_{CI} with $\epsilon_{CI} = 0.05$ and $\epsilon_R = 0.05$.

We compared the performance of these algorithms with the performance of the best configuration in the first experiment, that with $\sigma = 0.9$. The features used in these experiments are the same used in the first set of experiments; that is, all of them: food, threatening/edible ghosts and capsules.

State space

Performance of MDPs are heavily influenced by the shape of the state space. In our experiments, the state space is the cartesian product of 8 features. Each of the features is related to objects in game; i.e., threatening/edible ghosts, pills and capsules. Features are either distance or direction information to such objects.

The “direction” feature specifies the direction that Pac-Man should follow to reach the closest object of the category. The direction information can assume five different values: one for each of the cardinal directions, plus an additional value used when there are no instances of the objects; e.g., if all the ghosts are edible, there is no threatening ghost. In the case of distance, the feature specifies $\lfloor \log_2 d \rfloor$, where d is the length of the shortest path to the closest object of the category. Shortest paths are computed by the Dijkstra algorithm for shortest path on graphs (see [30] for more information). Distance information can be null as well.

Annealing exploration

We compared different exploration strategies; i.e. ϵ -Greedy and our proposal, ϵ_{CI} -Greedy, showed in Algorithm 2. Even though we presented the naïve versions of the two strategies, in our experiments we used the simulated annealing version. This technique slowly decreases the amount of exploration as time progresses, so to gradually shift

from an exploration policy to the greedy policy over time. In ϵ -Greedy policies this is done by decreasing the value of ϵ . In ϵ_{CI} -Greedy, we similarly decrease both ϵ_R and ϵ_{CI} . The annealing schedule we chose is based on the sigmoid function, $s(x) = \frac{1}{1+e^x}$. Our schedule is defined as follows:

$$\epsilon(t) = \hat{\epsilon} \cdot s(u \cdot (m - t)),$$

where $\hat{\epsilon}$ is the maximum value for the exploration parameter, t is the current episode number, m is the desired center for the schedule and u controls the width of the function.

5 Results

In this section we discuss the results of the experiments we performed. All the figures in this section are smoothed using a moving average weighted by a Hanning function. The Hanning function is bell-shaped and smoothly zeroes at the edges. Using it to weight contributions in a moving window gives greater importance to central elements while still taking the surrounding element in account.

In the first experiment, different σ -values are compared in Multi-Abstraction Q-learning (Algorithm 1) using ϵ_{CI} -Greedy. The scores achieved by the different configurations are shown in Figure 3a. It is surprising that the lines dominating the chart are those using 0.5 and 0.9 as σ -values.

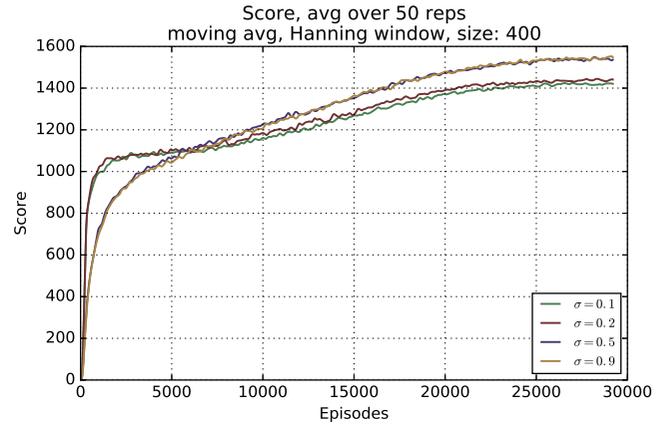
The most likely explanation for this is that 0.1 and 0.2 are too conservative values. While in normal t-tests values of 0.1 are unacceptably high, the trend here is heavily shifted. In fact, orthodox t-tests assume that the distributions are static over time. Here, however, (expected) Q-values veer from the common initial value towards their true values. For this reason, seemingly “premature” Q-values, which have a “high variance” from a t-test perspective, reliably estimate the best action.

Figures 4a and 4b show the percentage of decisions that have been made with each abstraction in successive episodes for the two configurations $\sigma = 0.2$ and $\sigma = 0.5$. There is a remarkable difference in that the former keeps using coarse abstractions throughout the learning process, while the latter barely uses any, except at the very early stages.

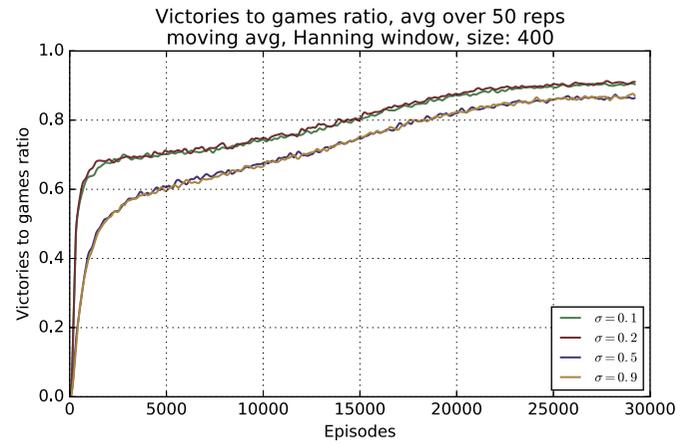
Figure 3b shows the percentage of victories for the configurations in the first experiment. The configurations winning the most often are $\sigma = 0.1$ and $\sigma = 0.2$. Considering the scores shown in Figure 3a this seems counterintuitive, because one would expect that the configurations with the highest scores are also those winning the most often. However, these numbers make sense when the structure of the game is considered: to maximize the score, Pac-Man needs to eat ghosts, but that poses an added risk in terms of winning/losing (i.e. if the ghost suddenly turns back to a threatening status). Figure 3c shows the average number of ghosts eaten in each successive episode: it can be observed that there is a significant difference between the configurations $\sigma = 0.1$ and $\sigma = 0.2$ and the configurations $\sigma = 0.5$ and $\sigma = 0.9$. The similarity of the trends showed in Figures 3c and 3a, where dominant configurations are $\sigma = 0.5$ and $\sigma = 0.9$ in both cases, supports this theory.

In the second experiment, our technique is compared with three configurations of Q-learning, each using an increasing amount of features. Figure 5 compares them to the best configuration of the first experiment, Multi-Abstraction Q-learning with $\sigma = 0.9$.

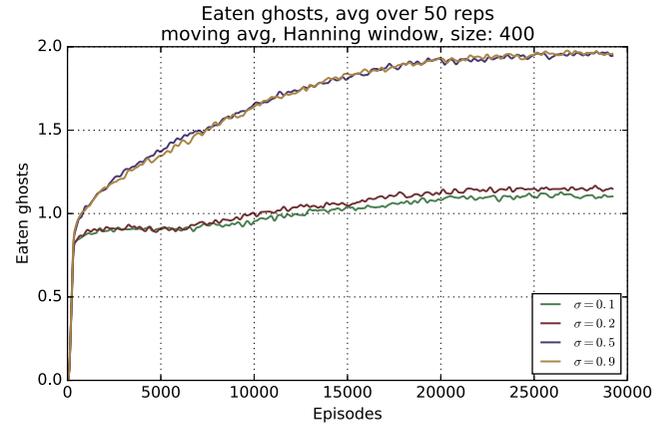
The curves show that the Q-learning configuration with the least features, at first, performs the best, showing that using more features at the beginning worsens the performance. However, this configuration is later surpassed by the Q-learning configuration using the intermediate



(a) Scores of the games



(b) Victories to total games ratio



(c) Eaten ghosts

Figure 3: Data (y axis) per successive episode (x axis) using Multi-Abstraction Q-learning with ϵ_{CI} -Greedy, varying significance parameter.

amount of features, showing that having more features pays off when sparsity fades out. Finally, the Q-learning configuration using the most features surpasses both the other two Q-learning configurations. These trends show how, in normal Q-learning, more features produce better performance at later stages at the cost of performance in the early stages.

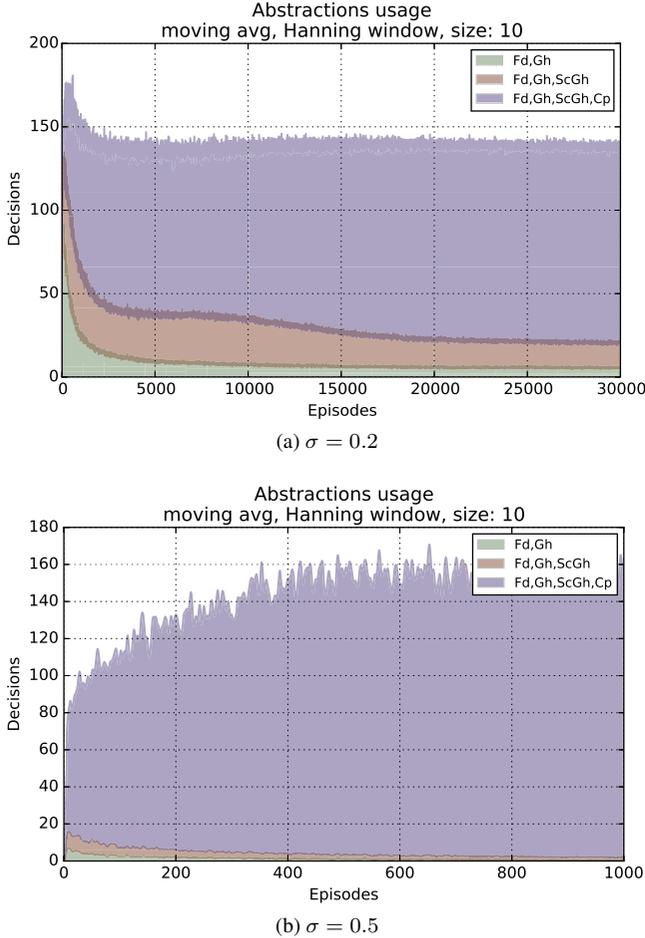


Figure 4: Decisions made with each abstraction (y axis) for each successive episode (x axis), using different values of significance in Multi-Abstraction Q-learning with ϵ_{CI} -Greedy. The second plot shows fewer episodes (x axis) because the more coarse abstractions quickly become almost unused.

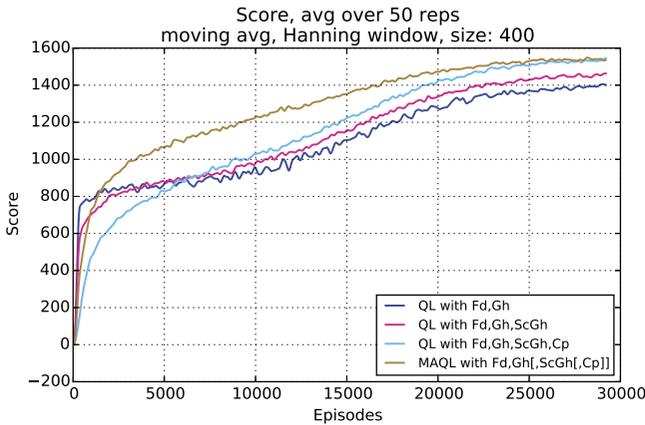


Figure 5: Score (y axis) per successive episode (x axis) using Multi-Abstraction Q-learning with ϵ_{CI} -Greedy versus standard Q-learning with different sets of features.

Except at the very beginning of the process, Multi-Abstraction Q-learning produces significantly better results than the other configurations. Importantly, it also converges to the same values as the

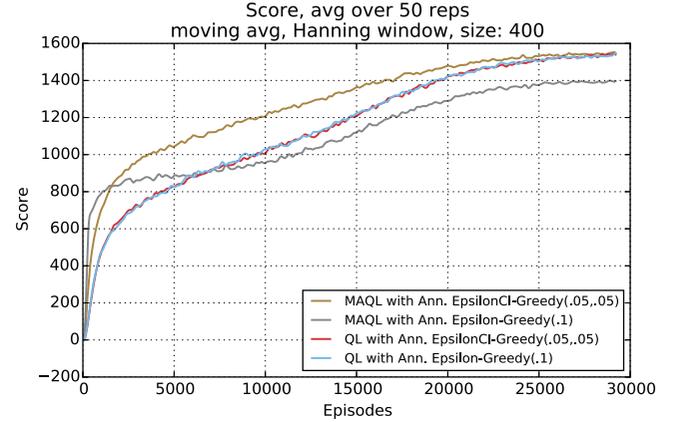


Figure 6: Score (y axis) per successive episode (x axis) using Multi-Abstraction Q-learning with ϵ -Greedy versus ϵ_{CI} -Greedy.

Q-learning configuration using all of the features: this shows that the early improvement in performance does not come at the cost of later performance.

It could be argued that our approach can be replaced by predetermined rules. In fact, the intersection points of Q-Learning performance curves in Figure 5 provide a clear indication of when it is convenient to switch abstraction. This would produce better performance than any of the three Q-Learning agents. However, because our approach allows each state to be used at a different abstraction, it is more adaptive and produces far better performance during the learning phase, as shown in Figure 5.

It could also be argued that, since the final performance of Multi-Abstraction Q-learning is the same as that of standard Q-Learning, an agent might as well just use standard Q-Learning. While this is true, the advantage of Multi-Abstraction Q-learning is an improvement in performance during the learning phase as opposed to an improvement in final performance.

Finally, notice that Algorithm 1 has the same convergence guarantees of Q-learning [31]. This is because the set of features in use at each state is expanded to the full features set within a finite amount of time. Notice that Algorithm 2 guarantees at least the same amount of exploration of ϵ -greedy.

	MQL ■	QL w/4 ■	QL w/3 ■	QL w/2 ■
Mean	1281.07	1152.11	1133.77	1099.25
Std. err	23.71	20.51	9.46	7.62

Table 1: Mean and standard error of the average reward per episode across the 50 runs of the experiments in Figure 5. Columns report values, respectively, for Multi-Abstraction Q-learning and for Q-Learning with features Fd,Gh,ScGh,Cp, Fd,Gh,ScGh and Fd,Gh.

The results of the third experiment are shown in Figure 6. The experiment shows that Multi-Abstraction Q-learning and ϵ_{CI} -Greedy do create a synergy in performance. On one side, ϵ_{CI} -Greedy does not seem to affect the performance of Q-learning; on the other side, Multi-Abstraction Q-learning without ϵ_{CI} -Greedy performs worse than Q-learning. However, when Multi-Abstraction Q-learning is used in concert with ϵ_{CI} -Greedy, they produce better performance than all other combinations.

Statistical analysis of Q-learning performance

The experiments showed that Multi-Abstraction Q-learning produces better performance than Q-Learning with any set of features. To quantitatively measure the performance of the different configurations, we computed the average reward per episode for each of the 50 runs of each of the 3 agents. This operation induces a distributions for each agent, all of which appear to be approximately normally distributed, as shown in Figure 7. Table 1 shows the means and standard errors of the data. Three two-samples t-tests (with unequal variance) have been executed to pairwise compare the agents sorted by average per-episode reward. All t-tests determined that the distributions mean are different with p -value < 0.001 , therefore confirming that both agents using learned options perform significantly better than the agent that does not use options.

6 Conclusions

In this paper we presented a novel variation of Q-learning, which we name “Multi-Abstraction Q-learning”. The algorithm we propose uses different state-abstractions for each state, increasing the level of detail over time. This allows the agent to overcome the initial sparsity in its utility estimates, typical of richer state representations. The agent can still make full use of the maximum level of detail later on in the learning process. Our experiments show that this algorithm produces better performance than standard Q-learning.

We also proposed a novel exploration strategy, ϵ_{CI} -Greedy. This strategy directs exploration to reduce sparsity of information, thereby allowing the agent to switch to more detailed abstractions earlier. Our experiments show that this strategy produces better results than standard ϵ -Greedy.

The proposed algorithm takes advantage of similarity in Q-values of similar states. In particular, factored state spaces where states in (hyper-)rectangular regions share similar values are a good fit for the algorithm. However, the algorithm just works with increasingly granular abstractions: it cannot take advantage of redundancy in regions defined by different sets of features. For example, in the case of Pac-Man, in states where a threatening ghost is nearby, food information has a low impact on Q-values, while in states where all threatening ghosts are distant, the direction from which they are likely to come has a low impact on Q-values.

To optimize the use of redundancy in these cases, the algorithm would need to consider multiple, separate sets of features. This means that the abstractions provided in input should be allowed to form a lattice, as opposed to just a list. In fact, while a list allows for only one “line of specialization”, a lattice allows for more possibilities. In other words, the proposed algorithm does not have a choice in *what* information to add over time: it can solely choose *when* to add it. Using a lattice, different sets of features could be selected for different states and they would still all converge to the abstraction with all features in the end. This is currently the strongest limitation of the algorithm and is an important direction for future research.

One drawback of our approach is its reliance on meaningful features and, hence, domain knowledge. An interesting avenue of research is to investigate a combination of our approach and a general-purpose function approximation architecture, similar in spirit to [32].

It would also be interesting to combine temporal approaches [12]–[14] and/or adaptive tile coding approaches [17], [18] with the proposed approach in more complex domains.

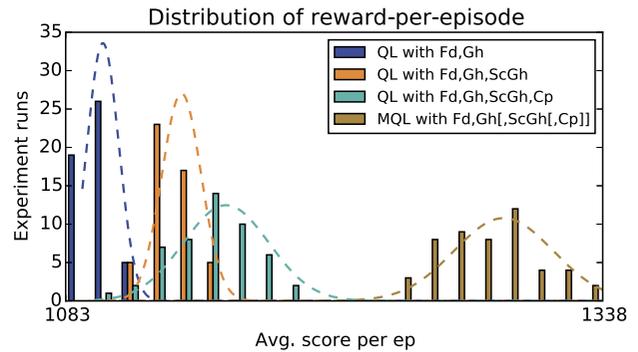


Figure 7: Histograms of the average reward per episode.

Acknowledgements

The authors acknowledge support from the ARC Grant LP130100743. We would like to thank the VxLab RMIT for their support in this work.

References

- [1] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st. Cambridge, MA, USA: MIT Press, 1998.
- [2] R. S. Sutton, “Learning to predict by the methods of temporal differences”, *Machine learning*, vol. 3, no. 1, pp. 9–44, 1988.
- [3] C. J. C. H. Watkins, “Learning from delayed rewards.”, PhD thesis, University of Cambridge, 1989.
- [4] D. Andre and S. J. Russell, “State abstraction for programmable reinforcement learning agents”, in *AAAI/IAAI*, 2002, pp. 119–125.
- [5] M. E. Taylor and P. Stone, “Transfer learning for reinforcement learning domains: A survey”, *The Journal of Machine Learning Research*, vol. 10, pp. 1633–1685, 2009.
- [6] J. N. Tsitsiklis and B. Van Roy, “An analysis of temporal-difference learning with function approximation”, *Automatic Control, IEEE Transactions on*, vol. 42, no. 5, pp. 674–690, 1997.
- [7] V. Tadi, “On the convergence of temporal-difference learning with linear function approximation”, *Machine learning*, vol. 42, no. 3, pp. 241–267, 2001.
- [8] N. K. Jong and P. Stone, “State abstraction discovery from irrelevant state variables.”, in *IJCAI*, Citeseer, 2005, pp. 752–757.
- [9] L. C. Cobo, P. Zang, C. L. Isbell Jr, and A. L. Thomaz, “Automatic state abstraction from demonstration”, in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, Citeseer, vol. 22, 2011, p. 1243.
- [10] A. Hallak, D. Di-Castro, and S. Mannor, “Model selection in markovian processes”, in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2013, pp. 374–382.
- [11] N. Jiang, A. Kulesza, and S. Singh, “Abstraction selection in model-based reinforcement learning”, in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015, pp. 179–188.
- [12] R. A. McCallum, “Overcoming incomplete perception with utile distinction memory”, in *Proceedings of the Tenth International Conference on Machine Learning*, 1993, pp. 190–196.

- [13] —, “Instance-based utile distinctions for reinforcement learning with hidden state”, in *Proceedings of the Twelfth International Conference on Machine Learning*, Citeseer, 1995, pp. 387–395.
- [14] R. A. McCallum, G. Tesauro, D. Touretzky, and T. Leen, “Instance-based state identification for reinforcement learning”, *Advances in Neural Information Processing Systems*, pp. 377–384, 1995.
- [15] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., “Human-level control through deep reinforcement learning”, *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [16] J. S. Albus, *Brains, behavior, and robotics*. 1981.
- [17] S. Whiteson, M. E. Taylor, P. Stone, et al., *Adaptive tile coding for value function approximation*. Computer Science Department, University of Texas at Austin, 2007.
- [18] P. Scopes and D. Kudenko, “Automated mixed resolution tiling”.
- [19] G. W. Corder and D. I. Foreman, *Nonparametric statistics: A step-by-step approach*. John Wiley & Sons, 2014.
- [20] T. J. Walsh, L. Li, and M. L. Littman, “Transferring state abstractions between MDPs”, in *ICML Workshop on Structural Knowledge Transfer for Machine Learning*, 2006.
- [21] B. Welford, “Note on a method for calculating corrected sums of squares and products”, *Technometrics*, vol. 4, no. 3, pp. 419–420, 1962.
- [22] D. West, “Updating mean and variance estimates: An improved method”, *Communications of the ACM*, vol. 22, no. 9, pp. 532–535, 1979.
- [23] P. Rohlfshagen and S. Lucas, “Ms pac-man versus ghost team cec 2011 competition”, in *Evolutionary Computation (CEC), 2011 IEEE Congress on*, Jun. 2011, pp. 70–77.
- [24] M. Gallagher and A. Ryan, “Learning to play pac-man: An evolutionary, rule-based approach”, in *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on*, vol. 4, Dec. 2003, 2462–2469 Vol.4.
- [25] D. Robles and S. M. Lucas, “A simple tree search method for playing ms. pac-man”, in *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*, IEEE, 2009, pp. 249–255.
- [26] S. Samothrakis, D. Robles, and S. Lucas, “Fast approximate max-n monte carlo tree search for ms pac-man”, *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 3, no. 2, pp. 142–154, 2011.
- [27] J. DeNero and D. Klein, “Teaching introductory artificial intelligence with pac-man”, in *Proceedings of the Symposium on Educational Advances in Artificial Intelligence*, 2010.
- [28] S. v. d. Walt, S. C. Colbert, and G. Varoquaux, “The numpy array: A structure for efficient numerical computation”, *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22–30, 2011.
- [29] O. Tange, “Gnu parallel - the command-line power tool”, *login: The USENIX Magazine*, vol. 36, no. 1, pp. 42–47, Feb. 2011.
- [30] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. The MIT Press, 2001.
- [31] C. J. Watkins and P. Dayan, “Q-learning”, *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.
- [32] C.-S. Chow and J. N. Tsitsiklis, “An optimal one-way multi-grid algorithm for discrete-time stochastic control”, *IEEE transactions on automatic control*, vol. 36, no. 8, pp. 898–914, 1991.

A Dialectical Proof Theory for Universal Acceptance in Coherent Logic-Based Argumentation Frameworks

Abdallah Arioua and Madalina Croitoru¹

Abstract. Given a logic-based argumentation framework built over a knowledge base in a logical language and a query in that language, the query is *universally accepted* if it is entailed from all extensions. As shown in [2, 14], universal acceptance is different from skeptical acceptance as a query may be entailed from different arguments distributed over all extensions but not necessarily skeptical ones. In this paper we provide a dialectical proof theory for universal acceptance in coherent logic-based argumentation frameworks. We prove its finiteness, soundness, completeness, consistency and study its dispute complexity. We give an exact characterization for non-universal acceptance and provide an upper-bound for universal acceptance.

1 Introduction

Dialectical proof theories have their roots in the dialogical approach to logic traditions [23]. In the Greek antiquity logic was studied in a dialogical context where two parties exchange arguments over a central claim. In modern logic the dialogical approach (or *dialogical logics*) is used to provide a game-theoretical semantics for logical systems. Proofs according to dialogical logics is a dialogue game between two parties arguing about a thesis while respecting some fixed rules. The dialogue is adversarial where one party plays the role of the defender of the thesis (proponent) and the other argues against the thesis (opponent). Each dialogue ends after a finite number of moves with a winner and a loser.

Since the work of Dung [16] many attempts have been made to adapt the dialogical approach to provide formal proof theories for formal argumentation, this is often referred to as *dialectical proof theories*. The works of [20, 28] define, similarly to dialogical logic, a dialectical proof theory as an argument game with a winning criterion alongside with a legal move function that decides the allowed moves to be played. Given an argumentation framework, a semantics x and an argument a , the objective is to prove whether the argument a is skeptically/credulously accepted under a semantics x .

The TPI (Two Party Immediate response) procedure proposed in [30] and further formalized in [17] is used for credulous and skeptical games in finite and coherent argumentation frameworks where two players exchange arguments (moves) until one of them cannot play. The justification status of the argument (skeptical/credulous) is decided with respect to the winning criterion. The turn in TPI-disputes shifts after one move with the move m_i attacks the precedent one (hence *immediate response*). Their dialectical proof theories are sound and complete. In [12], the same guideline is followed but with a refinement on the size of the proof, where [12] produces

shorter proofs than [17]. In [25] a different dialectical proof theory has been proposed for skeptical acceptance where, instead of exchanging arguments the proponent and the opponent exchange whole admissible sets. The goal is to construct a *block*, which is an admissible set of arguments that conflicts with all admissible sets around the argument in question [25, Theorem 6.7]. Following the same idea, [15] constructs such block in a *meta-argumentation framework* within a *meta-dialogue* where admissible sets are considered as moves, then the classical credulous proof theory of [12] is used as a sub-procedure to prove skeptical acceptance. In [29] a more general framework has been provided which is sound for any argumentation frameworks and it is complete for general classes of finitary argumentation frameworks including the class of finite argumentation frameworks using the notions of dispute derivation and base derivation. For skeptical preferred, the proof theory proposes to find a base then check whether it is complete or not. A base of an argument a is a set of admissible sets that (each of which) contains a such that whenever a is in an extension then there is an admissible set in the base that belongs to this extension. The base is complete if all extensions contains an admissible set from the base.

When it comes to logic-based argumentation the situation is quite different. In *logic-based argumentation we differentiate between the acceptance of an argument and the acceptance of a query*. A query is *universally accepted* under a semantics x if it is entailed from **every** extension. A query is *skeptically accepted* under a semantics x if it is entailed from a skeptically accepted argument. It is important to notice that the universal acceptance of a query does not necessarily mean that the query is skeptically accepted whereas the skeptical acceptance of a query necessarily yields the universal acceptance of the query. Skeptical acceptance can be easily handled by state of the art dialectical proof theories. However, already proposed dialectical proof theories fail when it comes to the universal acceptance as this one is not implied by skeptical acceptance.

In this paper, following [17], we propose a new TPI-like dialectical proof theory for universal acceptance. We limit the scope of the work to finite and coherent logic-based argumentation frameworks. In coherent argumentation frameworks the stable and preferred extensions coincide. Therefore, our dialectical proof theory works for all the above mentioned semantics. We show the soundness, completeness and finiteness of the proposed proof theory and analyse its dispute complexity properties.

2 The Dialectical Proof Theory

2.1 Preliminaries and Motivating Example

To facilitate the readability of this section, we first introduce necessary background notions then we shift directly to Example 1 (moti-

¹ GraphIK, INRIA: INRA, LIRMM and University Montpellier, France. Email: abdallaharioua@gmail.com

vating example) that shows how existing work cannot be applied to universal acceptance. Then we give a complete characterization of such acceptance.

We consider existential rules, widely used nowadays on the Semantic Web [27, 6, 26], they generalize certain Description Logics (such as \mathcal{EL} [5] and DL-Lite [11] families) with cyclicity notions and predicate arity. This language is a fragment of first-order logic (aka *Datalog* $^\pm$ [10]) composed of formulas built with the only logical connectives (\wedge, \rightarrow), the quantifiers (\exists, \forall) and the special constant \perp . An *atom* is of the form $p(t_1, \dots, t_k)$ where p is a predicate of arity k and the t_i are terms (variables or constants). A finite set of atoms A is called an *atomset*, we denote by $terms(A)$ (resp. $vars(A)$) the set of terms (resp. variables) that occur in A . Given atomsets A_1 and A_2 , a *homomorphism* π from A_1 to A_2 is a substitution of $vars(A_1)$ by $terms(A_2)$ such that $\pi(A_1) \subseteq A_2$. In this case we say $A_2 \models A_1$ where \models is the FOL entailment. An *existential rule* is of the form $R = \forall \vec{x} \forall \vec{y} (B \rightarrow \exists \vec{z} H)$, where B and H are conjunctions of atoms², with $vars(B) = \vec{x} \cup \vec{y}$, and $vars(H) = \vec{x} \cup \vec{z}$. B and H are respectively called the *body* and the *head* of R . A rule with an empty body (resp. head set to \perp) is called a *fact* (resp. *negative constraint*). A *boolean conjunctive query* (BCQ) Q has the form of a fact. From now on we use the general term *query* to mean BCQ. We denote variables by uppercase letters X, Y, Z, \dots , constants by lowercase letters a, b, c, \dots and predicate symbols by lowercase letters p, q, r, s, \dots .

A *knowledge base* $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ is composed of a finite sets of facts \mathcal{F} , rules \mathcal{R} and negative constraints \mathcal{N} . Facts represent factual knowledge about the world, rules represent generic rule-based knowledge, and negative constraints represent logical falsehood (e.g. $\forall X (cat(X) \wedge dog(X) \rightarrow \perp)$). We say a rule $R \in \mathcal{R}$ is applicable on a fact $F \in \mathcal{F}$ iff there is a homomorphism from F to the body of R . This application gives a new fact F' which is the head of R with instantiated variables. For instance $\forall X (p(X) \rightarrow q(X))$ is applicable on $p(a)$ and it gives $q(a)$. The application of all rules \mathcal{R} on all facts \mathcal{F} exhaustively until termination is referred to as *the chase* and it produces the set of facts $\mathcal{C1}_{\mathcal{R}}(\mathcal{F})$ (all deducible facts). We restrict our work to the *finite expansion* fragment that guarantees that the chase halts and $\mathcal{C1}_{\mathcal{R}}(\mathcal{F})$ is finite [6]. We say a query is entailed from \mathcal{K} iff $\mathcal{C1}_{\mathcal{R}}(\mathcal{F}) \models Q$. Since the chase always halts then query entailment is **decidable**. For a given \mathcal{K} , we say that a set of facts $F \subseteq \mathcal{F}$ is inconsistent (consequently \mathcal{K}) iff $\mathcal{C1}_{\mathcal{R}}(\mathcal{F}) \models \perp$.

The definition of an argument in this language is similar to the usual definition in logic-based argumentation of [7, 1].

Definition 1 (Argument). *Given a knowledge base $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$. An argument is a tuple (H, C) such that: (1) $H \subseteq \mathcal{F}$ with $\mathcal{C1}_{\mathcal{R}}(H) \not\models \perp$ (consistency), and (2) $C \in \mathcal{C1}_{\mathcal{R}}(H)$ and $H \models C$ (entailment), and (3) there is no $H' \subset H$ that verifies (1) and (2) (minimality). The support (resp. conclusion) of an argument a are denoted by $Supp(a) = H$ (resp. $Conc(a) = C$).*

We denote arguments by subscripted lowercase letters a_i, b_i , etc. However, we may use a, b, \dots when there is no ambiguity.

Arguments may attack each other with different types of attacks identified in the literature [8]. Here we focus on the *assumption attack* of [18] as it satisfies the rationality postulates [14].

Definition 2 (Attack). *An argument a attacks b iff $\exists h \in Supp(b)$ such that $\mathcal{C1}_{\mathcal{R}}(\{Conc(a), h\}) \models \perp$.*

Definition 3 (Argumentation framework). *Let $\mathcal{K} = (\mathcal{F}, \mathcal{R}, \mathcal{N})$ be a knowledge base. The corresponding argumentation framework is a*

pair $\mathcal{H} = (\mathit{Arg}(\mathcal{F}), \mathcal{U})$ where $\mathit{Arg}(\mathcal{F})$ is the set of all arguments that can be constructed from \mathcal{F} and \mathcal{U} is the attack relation.

Notation 1. *Let \mathcal{K} be a knowledge base and $\mathcal{H} = (\mathcal{A}, \mathcal{U})$ its corresponding argumentation framework such that $S \subseteq \mathcal{A}$. We denote:*

- $\mathit{range}^+(a) = \{b \mid (a, b) \in \mathcal{U}\}$, $\mathit{range}^-(a) = \{b \mid (b, a) \in \mathcal{U}\}$.
- $\mathit{range}^+(S) = \bigcup_{a \in S} \mathit{range}^+(a)$ and $\mathit{range}^-(S) = \bigcup_{a \in S} \mathit{range}^-(a)$.
- A set of arguments S attacks an argument b if there exists an argument $a \in S$ with $(a, b) \in \mathcal{U}$.

Definition 4 (Semantics). *Let \mathcal{K} be a knowledge base and $\mathcal{H} = (\mathcal{A}, \mathcal{U})$ its corresponding argumentation framework. Let $\mathcal{E} \subseteq \mathcal{A}$ and $a \in \mathcal{A}$. We say that \mathcal{E} is conflict free iff there exists no arguments $a, b \in \mathcal{E}$ such that $(a, b) \in \mathcal{U}$. \mathcal{E} defends a iff for every argument $b \in \mathcal{A}$, if we have $(b, a) \in \mathcal{U}$ then \mathcal{E} attacks b . \mathcal{E} is admissible iff it is conflict free and defends all its arguments. \mathcal{E} is a preferred extension iff it is maximal (w.r.t \subseteq) admissible set. \mathcal{E} is a stable extension iff it is conflict-free and for all $a \in \mathcal{A} \setminus \mathcal{E}$, \mathcal{E} attacks a . We denote by $\mathit{Ext}(\mathcal{H})$ the set of all extensions of \mathcal{H} under the preferred/stable semantics. An argument is skeptically accepted if it is in all extensions, credulously accepted if it is in at least one extension and rejected if it is not in any extension.*

It has been show in [14] that argumentation frameworks in our setting are *coherent*, i.e. the stable and the preferred semantics coincide.

Definition 5 (Universal acceptance [2, 14]). *Given an argumentation framework \mathcal{H} over an inconsistent knowledge base \mathcal{K} . A query Q is universally accepted in \mathcal{H} if and only if $\forall \mathcal{E} \in \mathit{Ext}(\mathcal{H})$, $\mathit{Concs}(\mathcal{E}) \models Q$ where $\mathit{Concs}(\mathcal{E}) = \bigcup_{a \in \mathcal{E}} \mathit{Conc}(a)$.*

After introducing the universal acceptance, let us explain why it is different from skeptical acceptance. Note that a query is skeptically accepted if and only if it is entailed by a conclusion of a skeptically accepted argument.

Example 1 (Motivating example). *We consider an inconsistent knowledge base \mathcal{K} : $\mathcal{F} = \{p(a), q(a), r(a)\}$, $\mathcal{R} = \{\forall X (p(X) \rightarrow s(X)), \forall X (q(X) \rightarrow s(X))\}$ and $\mathcal{N} = \{\forall X (p(X) \wedge q(X) \rightarrow \perp)\}$.*

The arguments that can be built from \mathcal{F} are:

- $a_1 = (\{p(a)\}, \{p(a)\})$, $a_2 = (\{q(a)\}, \{q(a)\})$.
- $a_3 = (\{p(a)\}, \{s(a)\})$, $a_4 = (\{q(a)\}, \{s(a)\})$.
- $a_5 = (\{p(a), r(a)\}, \{p(a), r(a)\})$.
- $a_6 = (\{q(a), r(a)\}, \{q(a), r(a)\})$.
- $a_7 = (\{p(a), r(a)\}, \{s(a), r(a)\})$.
- $a_8 = (\{q(a), r(a)\}, \{s(a), r(a)\})$.
- $a_9 = (\{r(a)\}, \{r(a)\})$.

*The attacks are $\mathcal{U} = \{(a_1, a_2), (a_1, a_4), (a_1, a_6), (a_2, a_1), (a_2, a_3), (a_2, a_5), (a_5, a_6), (a_6, a_5), (a_2, a_7), (a_1, a_8)\}$. The preferred extensions: $\mathcal{E}_1 = \{a_1, a_3, a_5, a_7, a_9\}$ and $\mathcal{E}_2 = \{a_2, a_4, a_6, a_8, a_9\}$ with $a_9 = (r(a), r(a))$ being a **skeptical argument**. As one may notice that the query $Q = s(a)$ is **not skeptically accepted but it is universally accepted**. Indeed, $Q = s(a)$ can be deduced from every extension (precisely, from the conclusions of $\{a_3, a_7\} \subset \mathcal{E}_1$ and $\{a_4, a_8\} \subset \mathcal{E}_2$). However, $Q' = r(a)$ is universally and skeptically accepted. Note that the query $Q'' = s(a) \wedge r(a)$ is also universally accepted but not skeptically accepted.*

In what follows we give a fine-grained characterization of universal acceptance that will help us to give a clear proof theory for it. It turns out that universal acceptance can be characterized using the concepts of query supporters, reduct, proponent set and block.

² We follow [6] in considering conjunctions of atoms as atomsets.

Definition 6 (Query's supporters). *Given an argumentation framework \mathcal{H} over an inconsistent knowledge base. The set of all arguments that supports the query \mathcal{Q} is defined as follows:*

$$SUP(\mathcal{Q}) = \{a \mid a \text{ is credulously accepted and } \text{Conc}(a) \models \mathcal{Q}\}$$

Definition 7 (Reduct of extension). *Given an extension $\mathcal{E} \subseteq \mathcal{A}$ and a query \mathcal{Q} . The reduct $\mathcal{E}^{\mathcal{Q}} \subseteq \mathcal{E}$ of the extension \mathcal{E} w.r.t the query \mathcal{Q} is defined as the non-empty intersection $SUP(\mathcal{Q}) \cap \mathcal{E}$. The reduct of the set of all extensions $\text{Ext}(\mathcal{H})$ w.r.t \mathcal{Q} is defined as $\text{Ext}(\mathcal{H})^{\mathcal{Q}} = \{\mathcal{E}^{\mathcal{Q}} \mid \mathcal{E} \in \text{Ext}(\mathcal{H})\}$.*

The **reduct** $\mathcal{E}^{\mathcal{Q}}$ of the extension \mathcal{E} w.r.t the query \mathcal{Q} is defined as the set of all supporters of \mathcal{Q} which belong to \mathcal{E} . This means that a complete set of reducts covers the set of all extensions.

Definition 8 (Complete reduct). *The set of all reducts $\text{Ext}(\mathcal{H})^{\mathcal{Q}}$ w.r.t a query \mathcal{Q} is complete if and only if there exists no $\mathcal{E} \in \text{Ext}(\mathcal{H})$ such that $\mathcal{E}^{\mathcal{Q}} \notin \text{Ext}(\mathcal{H})^{\mathcal{Q}}$.*

An incomplete reduct corresponds to the case where there is an extension that does not contain any supporter.

Proposition 1. *A query \mathcal{Q} is credulously accepted if and only if $\text{Ext}(\mathcal{H})^{\mathcal{Q}} \neq \emptyset$. A query \mathcal{Q} is universally accepted if and only if $\text{Ext}(\mathcal{H})^{\mathcal{Q}}$ is complete.*

The proponent set is similar to the concept of a complete base in [29]. Before defining it we need the concept of a hitting set.

Definition 9 (Hitting set). *Given a collection $\mathcal{C} = \{S_1, \dots, S_m\}$ of finite nonempty subsets of a set \mathcal{B} (the background set). A hitting set of \mathcal{C} is a set $A \subseteq \mathcal{B}$ such that $S_j \cap A \neq \emptyset$ for all $S_j \in \mathcal{C}$. A hitting set of \mathcal{C} is minimal (w.r.t \subseteq) if and only if no proper subset of it is a hitting set of \mathcal{C} . A minimum hitting set is a minimal hitting set w.r.t set-cardinality.*

Definition 10 (Proponent set). *A set of arguments $S \subseteq \mathcal{A}$ is a proponent set of \mathcal{Q} if and only if S is a minimal (w.r.t \subseteq) hitting set of $\text{Ext}(\mathcal{H})^{\mathcal{Q}}$.*

Proposition 2. *A query \mathcal{Q} is universally accepted if and only if it has a proponent set.*

It is clear that a proponent set holds the smallest set of arguments which are distributed over all extensions and support the query \mathcal{Q} . So, if one extension does not contain any supporter then the query is not universally accepted. The reason for the absence of such supporter is what we call the presence of a *block*. We follow the notion of a **block** from [25] and instantiate it in our setting. A block B is a set of arguments which are (1) all credulously accepted, (2) attack all the supporters of \mathcal{Q} , and (3) they can all together be extended to form an extension.

Definition 11 (Block). *Let \mathcal{Q} be a query and let $\mathcal{C} = \{\text{range}^-(a) \mid a \in SUP(\mathcal{Q})\}$. A set of arguments $B \subseteq \mathcal{A}$ is a block of \mathcal{Q} if and only if: (1) B is a hitting set of \mathcal{C} ; and, (2) there exists an admissible set $A \subseteq \mathcal{A}$ such that $B \subseteq A$.*

While a query may have more than one block or more than one proponent set, it is never the case that it has the two together.

Proposition 3. *A query \mathcal{Q} has a block iff \mathcal{Q} has no proponent set.*

Consequently, a query is not universally accepted iff it has a block.

2.2 Universal Dialectical Proof Theory

Given a query \mathcal{Q} and an argumentation framework \mathcal{H} , the universal dialectical proof theory is a two-person argument game between a proponent (PRO) and an opponent (OPP). The proponent and the opponent are engaged in an argumentation dialogue of *precisely defined* types of moves respecting a turn taking mechanism. The turn taking mechanism is deterministic where odd indexed moves are advanced by PRO and even index moves are advanced by OPP. The moves of the dialogue are defined in terms of speech acts: support, counter and retrace. The move $SUPPORT(a)$ advances an argument a which supports the query in question. The move $COUNTER(A)$ counterattacks the position of PRO by advancing a set of arguments that attack the previously advanced supporters. The move $RETRACE(A, i)$ is used to retrace to the stage i in the dialogue. The dialogue is asymmetric where $SUPPORT$ can only be played by PRO, whereas $COUNTER$ and $RETRACE$ ³ can only be played by OPP.

Definition 12 (Dialogue). *Let $\mathcal{H} = (\mathcal{A}, \mathcal{U})$ be an argumentation framework. A dialogue based on \mathcal{H} is a finite sequence $d_n = (m_1, \dots, m_n)$ of moves where each m_j is either:*

- **Support move:** $m_j = SUPPORT(a)$ such that $a \in \mathcal{A}$ (In this case we denote $\text{Arg}(m_j) = a$ and $\text{Sp}(m_j) = SUPPORT$).
- **Counter move:** $m_j = COUNTER(A)$ such that $A \subseteq \mathcal{A}$ (In this case we denote $\text{Arg}(m_j) = A$ and $\text{Sp}(m_j) = COUNTER$).
- **Retrace move:** $m_j = RETRACE(A, i)$ such that $A \subseteq \mathcal{A}$ and $i < j$ (In this case we denote $\text{Arg}(m_j) = A$, $\text{Sp}(m_j) = RETRACE$).

Odd-indexed (resp. even-indexed) moves are played by PRO (resp. OPP). We denote by $d \cdot d'$ and $d \cdot m$ the concatenation of the dialogues d and d' and the dialogue d with the move m respectively. The retrace move has a special parameter i called the index (denoted as $\text{Idx}(m)$). The subscript of d_n refers to the stage of the dialogue.⁴

Let \mathcal{Q} be a query, any dialogue of this dialectical theory starts by PRO advancing $SUPPORT(a)$ that supports \mathcal{Q} (i.e. $a \in SUP(\mathcal{Q})$). Then, OPP presents an argument (or a set of arguments) that attacks the previously advanced argument. Next, PRO tries to avoid this attack and reinstate the query using another argument which is not attacked by the already advanced attackers. OPP in turn, tries to extend the previous set of attackers so that it attacks all the supporters advanced so far. When OPP fails to extend the set, it retraces back and chooses another set of attackers and continues the dialogue from thereafter. By doing so OPP is somehow trying to construct a set of arguments that attack all the supporters of the query \mathcal{Q} , i.e. a *block* for \mathcal{Q} .

Following [17] we introduce the notion of a *dialectical state* which helps in controlling the dialogue.

Definition 13 (Dialectical state). *Let d_k be a dialogue at stage k . The dialectical state of d_k is a tuple $\delta_k = (\pi_k, h_k, \theta_k, \beta_k, \Delta_k)$ ⁵:*

- π_k : the set of arguments available to PRO.
- h_k : the set of arguments that have been played so far by PRO.
- θ_k : the set of arguments available to OPP.
- β_k : the current block constructed by PRO.
- Δ_k : the sets of arguments that have been shown to be not blocks.

d_0 is the empty dialogue and δ_0 is its initial dialectical state.

³ When there is no risk of ambiguity we refer to moves by their speech acts.

⁴ We may sometimes omit the subscript when it is not needed.

⁵ To be able to understand the terms think of π as the first letter of **proponent**, h as **history**, θ as **opponent** and β as **block**.

A dialectical state defines at a any stage k of the dialogue d_k the set of arguments π_k available to PRO to be used in order to support the query \mathcal{Q} . In the dialectical state, we find also the set h_k that shows the arguments so far played by PRO. In addition, it presents the set θ_k of arguments that can be used to attack the arguments previously advanced by PRO. β_k presents the currently constructed block. When OPP fails to extend the current block to another that attacks all the previously played supporters, the RETRACE move is used. By doing so we keep track of the sets of arguments that cannot be extended to blocks. These are stored in Δ_k .

At the beginning stage, when the dialogue d_0 has not yet been started, the set of available arguments π_0 for PRO ranges over all the possible supporters of the query \mathcal{Q} . The played arguments h_0 , the available arguments θ_0 , current block β_0 and Δ_0 are empty.

2.3 Dialogue Rules

The advancement of moves within the dialogue are usually controlled by a legal move function [25] which can be expressed in terms of rules, called dialogue rules. Every move depends on certain *preconditions* about the actual dialectical state and the previous move advanced by the other party. Every move also determines the next move to be played (*postcondition*).

Let d_k be a dialogue and δ_k the current dialectical state of d_k . Let m_{k+1} be a move and δ_{k+1} be the dialectical state of the dialogue $d_{k+1} = d_k \cdot m_{k+1}$ after playing the move m_{k+1} . For a given move we index preconditions (resp. effects) by the first letter of the speech act of the move followed by P (resp. E) and subscripted by a number.

Move:

$$m_{k+1} = \text{SUPPORT}(a).$$

Description:

advances an argument that supports the query in question.

Preconditions:

(SP₁) $k + 1$ is odd.

(SP₂) $a \in \pi_k$.

Postconditions:

the next move can be either COUNTER or RETRACE.

Effects:

(SE₁) $\pi_{k+1} = \pi_k / a$.

(SE₂) $h_{k+1} = h_k \cup \{a\}$.

(SE₃) $\theta_{k+1} = \text{range}^-(h_{k+1})$.

(SE₄) $\beta_{k+1} = \beta_k$.

(SE₅) $\Delta_{k+1} = \Delta_k$.

This move is advanced by PRO, therefore $k + 1$ should be odd (SP₁). It advances an argument a that supports the query \mathcal{Q} (SP₂). To respond, OPP should either use COUNTER or RETRACE.

As one may notice, the support move m_{k+1} changes the set of available arguments π_{k+1} of PRO. In fact a supporting argument ceases to be available once it is played (SE₁). In contrast it is added to the history h_{k+1} . The support move alters the set of available arguments of OPP by adding to θ_{k+1} all the arguments that can be played in the future by OPP (SE₃), which are those that can attack the advanced supporting arguments. As indicated in the postconditions of the support move, a counter move is allowed to be played next.

Move:

$$m_{k+1} = \text{COUNTER}(A).$$

Description:

this move advances a set of arguments that attacks all the arguments presented so far.

Preconditions:

(CP₁) $k + 1$ is even.

(CP₂) $A = \beta_k \cup S$ such that $S \subseteq \theta_k$ (i.e. A extends β_k by S).

(CP₃) A attacks h_k and belongs to (or is) an admissible set.

(CP₄) there is no $A' \in \Delta_k$ such that $A' \subseteq A$.

Postconditions:

the next move should be SUPPORT.

Effect:

(CE₁) $\pi_{k+1} = \pi_k / \text{range}^+(A)$.

(CE₂) $h_{k+1} = h_k$.

(CE₃) $\theta_{k+1} = \theta_k$.

(CE₄) $\beta_{k+1} = A$.

(CE₅) $\Delta_{k+1} = \Delta_k$.

This move is advanced by OPP therefore $k + 1$ should be even (CP₁). It tries to extend the current block β_k to another set of arguments that attacks all the supporters presented so far (CP₂ and CP₃). OPP does so by incorporating arguments from θ_k . The new current block ($\beta_{k+1} = A$) or one of its subsets should have not been already proven to be not a block (CP₄). After advancing m_{k+1} , π_{k+1} contains all the arguments from π_{k+1} except those which are attacked by A (CE₁), thus they are spared from further use. Note that the spared arguments may be readded afterwards, this is particularly the case when we use retrace as we shall mention later.

Since A attacks all the supporting arguments so far provided, it is considered the current block (CE₄). The sets θ_{k+1} , Δ_{k+1} and h_{k+1} are left unchanged (CE₂, CE₃ and CE₅).

After a support move, OPP can also play a retrace move. This is particularly needed when he is unable to play a counter move.

Move:

$$m_{k+1} = \text{RETRACE}(A, i).$$

Description:

this move retraces to the recent stage i from which it can extend the current block of i .

Preconditions:

(RP₁) $k + 1$ is even, $i < k + 1$ and i is odd.

(RP₂) there is no set of arguments $S \subseteq \theta_k$ such that $\beta_k \cup S$ is (or belongs to) an admissible set and attacks h_k .

(RP₃) $A = \beta_i \cup S$ such that $S \subseteq \theta_i$.

(RP₄) A attacks h_i and belongs to (or is) an admissible set.

(RP₅) there is no $A' \in \Delta_k$ such that $A' \subseteq A$.

Postconditions:

the next move should be SUPPORT.

Effect:

(RE₁) $\pi_{k+1} = \pi_i / \text{range}^+(A)$.

(RE₂) $h_{k+1} = h_i$.

(RE₃) $\theta_{k+1} = \theta_i$.

(RE₄) $\beta_{k+1} = A$.

(RE₅) $\Delta_{k+1} = \Delta_k \cup \beta_k$.

When OPP cannot extend the current block β_k with arguments from θ_k (RP₂), it should retrace back and choose other arguments.

The index i (which should be odd) determines the point of a support move from which OPP can mount another line of attack. By starting a new line of attack, OPP should opt for a new block that attacks all the supporters from stage i up to the stage 1 (RP₃) by extending β_i using θ_i . The new block $\beta_{k+1} = A$ or one of its subsets should have not been already proven to be not a block (CP₅).

When the retrace move is advanced, π_{k+1} is reset to its ancient state i in addition to excluding all the arguments that can be attacked afterwards (RE₁). The current block β_{k+1} is set to A (RE₄), and Δ_{k+1} is set to $\Delta_k \cup \beta_k$ (RE₅), i.e. the block of stage k that OPP could not extend. If one precondition is not satisfied OPP looks for other stages to build a new attack. OPP follows the procedure:

Procedure 1. Let d_n be a dialogue and m_n be the last played move such that $\text{Sp}(m_n) = \text{SUPPORT}$. If OPP cannot play a counter move m_{n+1} then it tries to play the retrace move m_{n+1} as follows:⁶

1. do $y = y - 1$ until $m_y = \text{RETRACE}(A, x)$ or $m_y = \text{SUPPORT}(a)$.
2. if $m_y = \text{RETRACE}(A, x)$ then:
 - (a) play $m_{n+1} = \text{RETRACE}(A', x)$ that respects the preconditions and exit. If there isn't such move then set $y = x$ and goto 1.
3. if $m_y = \text{SUPPORT}(a)$ then:
 - (b) play $m_{n+1} = \text{RETRACE}(A', y)$ that respects the preconditions and exit. If there is no such move then goto 1.

OPP starts by looking for the most recent retrace or support move (line 1). If a retrace move is found (line 2) then it tries to play a retrace to stage x that respects the preconditions (line a) by looking exhaustively for all possible sets A' that makes the move respect the preconditions. If he succeeds to play such move, the procedure exits. Otherwise it continues the search by setting y to x . If a support move m_y is found (line 3) then it plays a retrace with index to y . Otherwise, it continues the search for other moves from which OPP can play.

The dialogue represents a compact representation of a tree where nodes are arguments or set of arguments played by both parties. Nodes in the odd levels are played by PRO and nodes in the even level are played by OPP. This tree is called the dialogue tree. In this tree retrace moves represent branching points.

Definition 14 (Dialogue tree). Given a dialogue $d_n = (m_1, \dots, m_n)$, its dialogue tree is a labeled tree $\mathcal{T}(d_n) = (\mathcal{V}, \mathcal{D})$ where \mathcal{V} is a set of nodes and \mathcal{D} is a binary relation over \mathcal{V} . $\mathcal{T}(d_n)$ is defined as follows: while $n > 0$, $\mathcal{D}(d_n)$ is recursively defined as:

$$\begin{cases} \mathcal{D}(d_n) = \emptyset & n = 1 \\ \mathcal{D}(d_{n-1}) \cup \{(\text{Arg}(m_i), \text{Arg}(m_n))\} & n > 1, m_n = \text{RETRACE}(A, i) \\ \mathcal{D}(d_{n-1}) \cup \{(\text{Arg}(m_{n-1}), \text{Arg}(m_n))\} & n > 1, m_n \neq \text{RETRACE}(A, i) \end{cases}$$

The set of all nodes is defined as $\mathcal{V} = \{\text{Arg}(m_i) \mid m_i \in d_n\}$ with $\text{Arg}(m_1)$ as the root node of the tree. Note that $|\mathcal{T}(d_n)| = |\mathcal{V}|$ refers to the size of the tree which is equal to the number of its nodes.

Note that this dialogue tree is similar to the well-known dispute tree in the work of [17, 25]. The difference is in the nature of nodes where the nodes of OPP in the dialogue tree contain set of arguments as opposed to a dispute tree.

The dialogue tree enjoys the following properties.

Proposition 4. Let d_n be a dialogue, $\mathcal{T}(d_n)$ its dialogue tree and $\text{Pre}(\mathcal{T}(d_n))$ the pre-order traversal of $\mathcal{T}(d_n)$ with $\text{Seq}(d_n) = (c_1, \dots, c_n)$ such that $c_i = \text{Arg}(m_i)$. The following hold:

⁶ Note that y is initialized to n and $x < y$, and a, A are arbitrary (set of arguments respectively).

1. $\mathcal{T}(d_n)$ is unique.
2. $|d_n| = |\mathcal{T}(d_n)|$.
3. $\text{Pre}(\mathcal{T}(d_n)) = \text{Seq}(d_n)$.

What is left for the dialectical proof theory is to determine the termination condition.

Definition 15 (Termination and winning). A dialogue d_n is a terminated dialogue if and only if neither PRO nor OPP can play a move. The winner of d_n is the player of the last move m_n .

It is easy to determine the winner of a dialogue from its tree:

Proposition 5. Let d_n be a dialogue, $\mathcal{T}(d_n)$ its dialogue tree. The following statements are equivalent:

- the length of the rightmost path is odd.
- PRO is the winner of d_n .

We conclude the section by defining a dialectical proof.

Definition 16 (Dialectical proof). Given a query \mathcal{Q} and a terminated dialogue d_n about \mathcal{Q} . We call d_n a dialectical proof for the universal acceptance of \mathcal{Q} if and only if PRO is the winner.⁷

In Section 4 we provide the properties of the dialectical proof theory. We now give an example to better illustrate it.

3 Illustrating Example

Consider the argumentation framework \mathcal{H} of Figure 1. This argumentation framework is coherent (preferred and stable extensions coincide). Suppose that the gray-colored arguments support a query \mathcal{Q} (i.e. $\text{SUP}(\mathcal{Q}) = \{a, d, e, l, h\}$). In what follows, we show how the query \mathcal{Q} is universally accepted by providing a dialectical proof.

The dialectical proof is presented in Table 1 and its dialogue tree is shown in Figure 2.

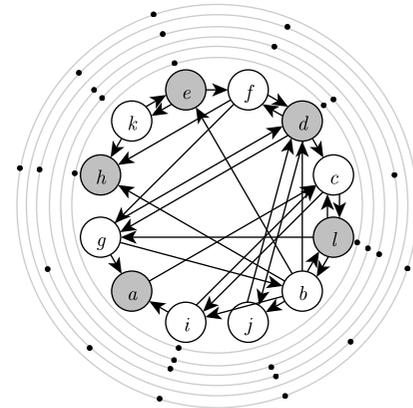


Figure 1: The argument graph. The circles are the extensions presented in an increasing order from \mathcal{E}_1 to \mathcal{E}_6 with the extension \mathcal{E}_1 being the inner circle and the extension \mathcal{E}_6 the outer circle.

At stage (0) the dialectical state is initialized as defined previously. The dialogue starts at stage (1) by PRO playing the supporter a from the available supporters in π_0 . When PRO plays a , the argument a is moved from the available supporters π_1 to the history of advanced arguments h_1 by PRO. The set of available attackers θ_1 becomes the set of all attackers that can attack h_1 . This means when the turn of

⁷ Otherwise it is called a dialectical proof for non-universal acceptance of \mathcal{Q} .

i	Move	π_i	h_i	θ_i	β_i	Δ_i
0	-	$\{a, d, e, l, h\}$	\emptyset	\emptyset	\emptyset	\emptyset
1	S(a)	$\{d, e, l, h\}$	$\{a\}$	$\{g, i\}$	\emptyset	\emptyset
2	C($\{g\}$)	$\{e, l, h\}$	$\{a\}$	$\{g, i\}$	$\{g\}$	\emptyset
3	S(l)	$\{e, h\}$	$\{a, l\}$	$\{g, i, c, b\}$	$\{g\}$	\emptyset
4	C($\{g, c\}$)	$\{e, h\}$	$\{a, l\}$	$\{g, i, c, b\}$	$\{g, c\}$	\emptyset
5	S(e)	$\{h\}$	$\{a, l, e\}$	$\{g, i, c, b, k\}$	$\{g, c\}$	\emptyset
6	R($\{i\}, 1$)	$\{d, e, l, h\}$	$\{a\}$	$\{g, i\}$	$\{i\}$	$\{\beta_5\}$
7	S(d)	$\{e, l, h\}$	$\{a, d\}$	$\{g, i, f, j, b\}$	$\{i\}$	$\{\beta_5\}$
8	C($\{i, f\}$)	$\{l, h\}$	$\{a, d\}$	$\{g, i, f, j, b\}$	$\{i, f\}$	$\{\beta_5\}$
9	S(h)	$\{l\}$	$\{a, d, h\}$	$\{g, i, f, j, b, k\}$	$\{i, f\}$	$\{\beta_5\}$
10	C($\{i, f, k\}$)	$\{l\}$	$\{a, d, h\}$	$\{g, i, f, j, b, k\}$	$\{i, f, k\}$	$\{\beta_5\}$
11	S(l)	\emptyset	$\{a, d, h, l\}$	$\{g, i, f, j, b, k, c\}$	$\{i, f, k\}$	$\{\beta_5\}$
12	R($\{i, j\}, 7$)	$\{e, l, h\}$	$\{a, d\}$	$\{g, i, f, j, b\}$	$\{i, j\}$	$\{\beta_5, \beta_{11}\}$
13	S(h)	$\{e, l\}$	$\{a, d, h\}$	$\{g, i, f, j, b, k\}$	$\{i, j\}$	Δ_{12}
14	C($\{i, j, k\}$)	$\{l\}$	$\{a, d, h\}$	$\{g, i, f, j, b, k\}$	$\{i, j, k\}$	Δ_{12}
15	S(l)	\emptyset	$\{a, d, h, l\}$	$\{g, i, f, j, b, k, c\}$	$\{i, j, k\}$	$\Delta_{12} \cup \{\beta_{14}\}$

Table 1: A dialectical proof for the query \mathcal{Q} . For space reasons S(), C() and R() denote SUPPORT(), COUNTER() RETRACE() respectively.

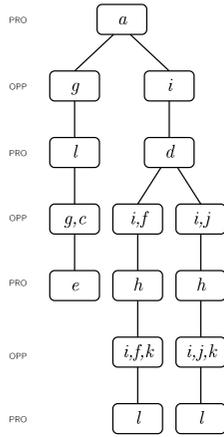
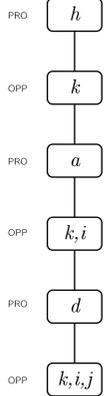
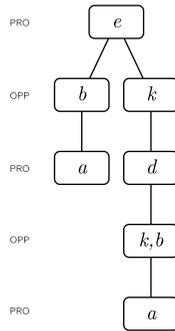


Figure 2: The dialogue tree for universal acceptance \mathcal{Q} .



(a) Dialogue tree for the non-universal acceptance of \mathcal{Q}' .



(b) Another dialogue tree for the universal acceptance of \mathcal{Q} .

Figure 3: Dialogue trees for the the illustrating example.

OPP comes at stage (2) he shall choose from this set. At stage (2) OPP advances a counter move with argument g that attacks all the advanced supporters (i.e. $h_1 = \{a\}$). After advancing such move, the argument d is removed from the set of available arguments π_2 since g attacks d , thus PRO will not be able to play d . Since $\{g\}$ attacks

i	Move	π_i	h_i	θ_i	β_i
0	-	$\{a, d, e, h\}$	\emptyset	\emptyset	\emptyset
1	S(h)	$\{a, d, e\}$	$\{h\}$	$\{k, f, b\}$	\emptyset
2	C($\{k\}$)	$\{a, d\}$	$\{h\}$	$\{k, f, b\}$	$\{k\}$
3	S(a)	$\{d\}$	$\{a, h\}$	$\{k, f, b, g, i\}$	$\{k\}$
4	C($\{k, i\}$)	$\{d\}$	$\{a, h\}$	$\{k, f, b, g, i\}$	$\{k, i\}$
5	S(d)	\emptyset	$\{d, a, h\}$	$\{k, f, b, g, i, j\}$	$\{k, i\}$
6	C($\{k, i, j\}$)	\emptyset	$\{d, a, h\}$	$\{k, f, b, g, i, j\}$	$\{k, i, j\}$

Table 2: A dialectical proof for the non-universal acceptance of \mathcal{Q}' . Note that we omit Δ_i as it is always empty in this example.

all the supporters advanced so far, it becomes the current block, i.e. $\beta_2 = \{g\}$. At stage (3), PRO responds by a support move with the argument l that is not attacked by the current block. At stage (4), OPP extends the current block $\beta_3 = \{g\}$ by the argument c which attacks l . Note that $\{g, c\}$ is a subset of the admissible set $\{g, c, e\}$. Now, $\beta_4 = \{g, c\}$ attacks all the presented supporters. At stage (5), PRO presents another unattacked supporter (i.e. e). Note that the choice of the supporters is arbitrary.

At stage (6), OPP could not extend the current block β_5 into another that attacks e too. Therefore OPP plays a retrace move R($\{i\}, 1$) that can be read as “retrace to stage (1) and play a counter move with $\{i\}$ ”. By doing so, OPP creates another line of dialogue and roll back all the changes that have been made on the dialectical state up to the stage (1). That is why at stage (6) the sets π_6 , h_6 and θ_6 are computed with respect to π_1 , h_1 and θ_1 . The current block is changed to $\{i\}$ and the ancient block β_5 is moved to $\Delta_6 = \{\beta_5\}$. The former would say that this set or any of its supper sets will never form a block. This is important to avoid unnecessary moves. The same thing happens at stage (12) where OPP retraces to stage (7) because he could not retrace to the stage (9). The current block β_{12} is set to $\{i, j\}$ which extends β_7 .

The dialogue continues until stage (15) where PRO plays a support move with argument l against which OPP could neither attack nor retrace to previous stages. For instance, OPP has not been able to extend $\{i, j, k\}$ by c because it would be conflicting (not admissible). At this stage the dialogue ends and PRO is declared as the winner.

The dialogue tree in Figure 2 shows clearly the relation between the advanced arguments played by both parties. The tree in Figure 3b is another dialogue tree for another dialogue where PRO is the winner. This can be easily observed since all leaf nodes are in odd levels.

Let us now take an example where the query is not universally accepted. Consider a query \mathcal{Q}' that happens to have the supporters $SUP(\mathcal{Q}') = \{a, d, e, h\}$. The dialogue is presented in Table 2 and its dialogue tree is shown in Figure 3a. In this example, OPP has been able to construct the block $\beta_6 = \{k, i, j\}$ in the last move which attacks all the supporters. This made PRO unable to continue the dialogue. Note that we do not allow retracing for PRO because one block is sufficient to prove the non-universal acceptance.

4 Dialectical Proof Theory Properties

4.1 Finiteness, Soundness and Completeness

As indicated in [3, 21], finiteness or termination is an important property for any dialogue, since a possibly infinite dialogue will fail to meet the intended goal. In what follows we show how our dialectical theory produces always finite dialogues.

To establish such property we need to show that for any dialogue d its dialogue tree is finite. Such result can be established by showing that the height of the tree is finite and that for each node the number of its child nodes is finite.

Lemma 1. *Let \mathcal{H} be an argumentation framework, \mathcal{D}^∞ be the set of all possible dialogues over \mathcal{H} , $\text{Height}(\mathcal{T}(d))$ is the height of the tree $\mathcal{T}(d)$ and $\mathcal{C}(v)$ is the set of all child nodes of v . Given $\mathcal{T}(d) = (\mathcal{V}, \mathcal{D})$ of any $d \in \mathcal{D}^\infty$ the following hold:*

1. $\text{Height}(\mathcal{T}(d)) \in \mathbb{N}$.
2. $\forall v \in \mathcal{V}, |\mathcal{C}(v)| \in \mathbb{N}$.

Proof. Let us suppose that $\text{Height}(\mathcal{T}(d))$ is infinite, and let P be the longest path in $\mathcal{T}(d)$ starting from the root node. This means either there are infinitely many supporting arguments used in P , or there are some infinity repeated supporting arguments used in P . The first one is impossible since we are dealing with finite argumentation framework (the set of all arguments is finite). The second is impossible since once an argument is played it cannot be advanced afterwards in the same path (see SE_1 of SUPPORT move).

Let us suppose that $|\mathcal{C}(v)|$ is infinite. This means that either (i) v is a supporting argument and it has infinitely many attacker; or (ii) v contains arguments that are advanced to attack previous supporters. The first case is impossible since the argumentation framework is finite, and the second is impossible since if it were the case then PRO would be allowed to retrace against counter moves, which is forbidden in our framework. \square

Now we can proceed to finiteness by showing the following.

Proposition 6 (Finiteness). *Let \mathcal{H} be an argumentation framework and \mathcal{D}^∞ be the set of all possible dialogues over \mathcal{H} . Then for every $d \in \mathcal{D}^\infty$: $|d| \in \mathbb{N}$.*

Proof. Let us suppose that d is infinite. This means, either (i) $\text{Height}(\mathcal{T}(d))$ is infinite; or (ii) there is a node in $\mathcal{T}(d_n)$ with infinitely many child nodes. From the previous lemma, the two cases are impossible. \square

In [3] an additional constraint has been added to finiteness, i.e. the finiteness of the moves' contents. This constraint ensures that the arguments advanced within the dialogue are finite. In our context we distinguish two cases, (i) the argument in the support moves should be finite, and (ii) the set of arguments advanced in the counter moves should be finite too. Fortunately, the two cases are verified in our argumentation framework because the set of arguments \mathcal{A} for any argumentation framework over a possibly inconsistent knowledge base (in our logical setting) is finite and the set of attackers for a given argument is finite. All in all, the argumentation framework is finite.

Before proceeding to soundness let us show that the dialectical proof theory is *consistent* in the sense that there is no two dialogues about a query \mathcal{Q} such that PRO wins in the former and loses in the later. Put differently, if one of the participant wins a dialogue about a given query \mathcal{Q} then we are sure that he will win all the other dialogues about \mathcal{Q} .

Proposition 7 (Consistency). *Let $\mathcal{D}_\mathcal{Q}^\infty \subseteq \mathcal{D}^\infty$ be the set of all dialogues about \mathcal{Q} in \mathcal{H} and let $d \in \mathcal{D}_\mathcal{Q}^\infty$. Then, if d is won by PRO (resp. OPP) then so are all $d \in \mathcal{D}_\mathcal{Q}^\infty$.*

This property is very important since we do not want to have a dialectical proof theory that is contradictory. It turns out that this property is important for soundness. In what follows, soundness is characterized by the existence of a winning dialogue (by PRO or OPP).

Proposition 8 (Soundness). *Given a dialogue d about the query \mathcal{Q} , if d is won by PRO then \mathcal{Q} is universally accepted.*

Proof. Let us proceed by contradiction. Suppose that d is won by PRO but \mathcal{Q} is not universally accepted. On the one hand, recall that if \mathcal{Q} is not universally accepted then there exists a block B against all \mathcal{Q} 's supporters. On the other hand, if PRO has won d then PRO could not find any block that attacks all supporters advanced in d . This means that either (i) OPP search was not exhaustive or (ii) there is no such block. As one can see, (ii) is in contradiction with the assumption and (i) is in contradiction with the fact that the move procedure is exhaustive. \square

If the dialectical proof theory is sound but does not provide dialectical proofs for all universally (resp. non-universally) accepted queries then it would be incomplete.

Proposition 9 (Completeness). *Given a query \mathcal{Q} . If \mathcal{Q} is universally accepted then PRO wins any dialogue about \mathcal{Q} .*

Proof. By contradiction, if \mathcal{Q} is universally accepted and PRO loses then OPP has constructed a block β_n for \mathcal{Q} . This means that \mathcal{Q} is not universally accepted, which is a contradiction. \square

In this subsection we have proved the finiteness, completeness and soundness of the proposed theory as well as its consistency.

4.2 Dispute Complexity

In this subsection we are interested in the question of how many moves the dialogue would contain for a query (at best-case) to establish its universal acceptance (non-universal acceptance). The work of [17] introduced the so-called dispute complexity for a given argument in a given argumentation framework. We adapt this definition and define the dispute complexity for a given query over a given instantiated argumentation framework as follows.

Definition 17 (Dispute complexity). *Let \mathcal{H} be an argumentation framework and \mathcal{Q} be a query. The dispute complexity $\delta(\mathcal{H}, \mathcal{Q})$ of the query \mathcal{Q} in \mathcal{H} is defined as follows:*

$$\delta(\mathcal{H}, \mathcal{Q}) = \min(|d| : d \text{ is a terminated dialogue about } \mathcal{Q} \text{ in } \mathcal{H})$$

The dispute complexity is the minimal number of moves that can be used to prove that \mathcal{Q} is universally accepted or not universally accepted. The work of [17] has given an exact characterization of such complexity for credulous acceptance by considering as an input the argumentation framework and all admissible sets. Our goal in what follows is to propose some bounds for such complexity in universal (or non-universal) acceptance.

Let \mathcal{Q} be a query, \mathcal{H} an argumentation framework such that \mathcal{Q} is not universally accepted in \mathcal{H} and $\mathcal{C} = \{\text{range}^-(a) | a \in \text{SUP}(\mathcal{Q})\}$. We will use the following notations:

- $MHS(\mathcal{H}, \mathcal{Q})$ is the set of all minimal (w.r.t \subseteq) hitting sets of \mathcal{C} .
- $MinBS(\mathcal{H}, \mathcal{Q})$ denotes the set of all minimal (w.r.t \subseteq) blocks.

- the block number of \mathcal{Q} in \mathcal{H} is the size of the minimum block:
 $\tau(\mathcal{H}, \mathcal{Q}) = \min(|B| : B \in \text{MinBS}(\mathcal{H}, \mathcal{Q}))$.
- the hitting set number is the size of the minimum hitting set of \mathcal{C} :
 $\alpha(\mathcal{H}, \mathcal{Q}) = \min(|S| : S \in \text{MHS}(\mathcal{H}, \mathcal{Q}))$.

The block number corresponds to the minimum block which is the smallest block (w.r.t set-cardinality) among all blocks. Note that it is not necessary that every minimum hitting set of \mathcal{C} is a minimum block (because a block imposes that its members have to belong to the same admissible set). Therefore it is possible to have a block which is minimum but does not correspond to any minimum/minimal hitting set. In contrast, a minimum block has to be a hitting set. We get the following straightforward relation.

Corollary 1. $\tau(\mathcal{H}, \mathcal{Q}) \geq \alpha(\mathcal{H}, \mathcal{Q})$.

In the context of a dialogue about a query \mathcal{Q} , the minimum block represents what the opponent would play in order to finish the dialogue as fast as possible. Therefore, the dispute complexity of non-universal acceptance can be characterized by such number.

Proposition 10. For any terminated dialogue d about \mathcal{Q} in an argumentation framework \mathcal{H} where \mathcal{Q} is not universally accepted:

$$\delta(\mathcal{H}, \mathcal{Q}) = 2 \times \tau(\mathcal{H}, \mathcal{Q}).$$

Sketch. If the size of the minimum block B equals n then at each stage OPP will extend his current block by advancing one attacker at each stage. Therefore, for each SUPPORT move we will have a COUNTER move that extends the current block by one argument. When the current block reaches the size n , that means OPP has played all the arguments of the minimum block, PRO will have no supporting argument to advance, thus the dialogue terminates after $2 \times n$ moves. \square

The property above provides an exact bound for the dispute complexity of the non-universal acceptance. The proposition below gives the upper bound for universal acceptance. In order to define this we define the attack degree of \mathcal{Q} in \mathcal{H} as $\text{deg}(\mathcal{H}, \mathcal{Q}) = \max(|\text{range}^-(a)| : a \in \mathcal{P}(\mathcal{Q}))$ such that $\mathcal{P}(\mathcal{Q})$ is the set all supporting arguments that belongs to at least one minimum proponent set. And the proponent number is the size of the minimum proponent set $\rho(\mathcal{H}, \mathcal{Q}) = \min(|S| : S \text{ is a proponent set of } \mathcal{Q} \text{ in } \mathcal{H})$. It is obvious that dialogues where PRO plays with minimum proponent sets are shorter than all other dialogues. Because in the latter dialogues PRO will play only the support moves that are needed to terminate the dialogue. Let $\Theta(\mathcal{H}, \mathcal{Q})$ be the size of the shortest dialogue where PRO plays only with a minimum proponent set. It is clear that $\delta(\mathcal{H}, \mathcal{Q}) \leq \Theta(\mathcal{H}, \mathcal{Q})$. To bound $\delta(\mathcal{H}, \mathcal{Q})$ we need to bound $\Theta(\mathcal{H}, \mathcal{Q})$. The latter can be bounded by imagining that the dialogue tree would have at worst-case the height equals to $2 \times \rho(\mathcal{H}, \mathcal{Q})$ and each proponent node (even-indexed) has exactly $\text{deg}(\mathcal{H}, \mathcal{Q})$ child (worst-case). Therefore, the upper-bound for $\Theta(\mathcal{H}, \mathcal{Q})$ is the size (number of nodes) of the this dialogue tree.

Proposition 11. For any dialogue d about \mathcal{Q} in an argumentation framework \mathcal{H} where \mathcal{Q} is universally accepted the following holds:

$$\delta(\mathcal{H}, \mathcal{Q}) \leq 2 \times \left(\frac{\text{deg}(\mathcal{H}, \mathcal{Q})^{\rho(\mathcal{H}, \mathcal{Q})} - 1}{\text{deg}(\mathcal{H}, \mathcal{Q}) - 1} - 1 \right)$$

Example 2. Consider the argumentation framework of Figure 1. Let $\text{SUP}(\mathcal{Q}'') = \{a, d, e, l, h, k\}$ be the set of supporters for an arbitrary universally accepted query \mathcal{Q}'' . Then, the query has two minimum proponent sets $\mathcal{P} = \{\{k, h\}, \{k, e\}\}$. The attack degree of \mathcal{Q}''

is $\max(1, 2, 3) = 3$ for k, e, h respectively. The proponent number is $\rho(\mathcal{H}, \mathcal{Q}'') = 2$. The upper-bound is:

$$\delta(\mathcal{H}, \mathcal{Q}'') \leq 2 \times \left(\frac{3^2 - 1}{3 - 1} - 1 \right) = 6$$

The real dispute complexity which corresponds to the shortest dialogue $(s(e), \mathcal{C}(\{k\}), s(k), \mathcal{R}(\{b\}, 1), s(k))$ is equal in this case to $\delta(\mathcal{H}, \mathcal{Q}'') = 5 < 6$.

It is clear that the bounds proposed for the dispute complexity of universal and non-universal acceptance are estimated in terms of the proponent and the block numbers. Unfortunately, these numbers are not given as inputs and they should be computed. It is obvious that computing such numbers is hard [19]⁸. Fortunately if one can estimate the cardinality of the minimum hitting set one can easily estimate the proponent or the block number. This can be achieved by using the results from [19] on the independence number in hypergraphs which is the complement of the hitting set number (also known as the transversal number).

5 Discussion and Conclusion

In this paper we have provided a dialectical proof theory for universal acceptance in coherent logic-based argumentation frameworks. We proved its finiteness, soundness, completeness, consistency and studied its dispute complexity.

It is important to point out that this dialectical proof theory can also be used in abstract settings like the one in [2]. In this work, Dung's abstract framework [16] is used in decision-support systems where arguments support different options (or decisions) and the final decision is computed using Dung's semantics. The author have introduced the concept of universal acceptance for a given option and shown that skeptical and universal are different. In fact, the distinction is important and practical since in certain decision making situations we may opt for an option that is supported by different arguments from different extensions but not supported by skeptical arguments (as there may be none). Our dialectical proof theory can offer an interesting feature in such settings, **explanation**. Dialectical proof theories in general provide, as argued by [25], explanation as to why a given output (option, conclusion, argument, etc.) is believed to be accepted. So, alongside to its capability of computing the accepted outputs, it can explain *why* and *how* the output are accepted. Aside from abstract settings, this intrinsic quality of explanation in our dialectical proof theory can lend itself to other domains such as (deductive) databases systems, more precisely in consistent query answering over inconsistent knowledge bases [4, 22, 9]. In fact it has been proven in [14] that the universal acceptance is equivalent to the well-known consistent query answering semantics [4], so our dialectical theory can be used in explaining why certain queries are entailed or not under the consistent query answering semantics which would have a great impact on the usability of such systems (as stipulated by [24]).

As a final remark, it seems that the concept of “arguments supporting a query” in our dialectical proof theory can somewhat be related to bipolar argumentation frameworks [13] that extends Dung's framework by a support relation *between arguments*. This will be the subject of our future work. Another future work is to look at the behavior of this dialectical proof theory on non-coherent or infinite argumentation frameworks.

⁸ In fact they are equivalent to finding a minimum proponent or block for the query, which would solve the problem in the first place.

REFERENCES

- [1] Leila Amgoud, 'Postulates for logic-based argumentation systems', *International Journal of Approximate Reasoning*, **55**(9), 2028–2048, (2014).
- [2] Leila Amgoud, Yannis Dimopoulos, and Pavlos Moraitis, 'Making decisions through preference-based argumentation', *Proceedings of the Eleventh International Conference on Principles of Knowledge Representation and Reasoning KR'08*, **8**, 963–970, (2008).
- [3] Leila Amgoud and Florence Dupin de Saint-Cyr, 'An axiomatic approach for persuasion dialogs', in *IEEE 25th International Conference on Tools with Artificial Intelligence (ICTAI'13)*, pp. 618–625. IEEE, (2013).
- [4] Marcelo Arenas, Leopoldo Bertossi, and Jan Chomicki, 'Consistent query answers in inconsistent databases', in *Proceedings of the eighteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pp. 68–79. ACM, (1999).
- [5] Franz Baader, Sebastian Brandt, and Carsten Lutz, 'Pushing the envelope', in *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI'05)*, (2005).
- [6] Jean-François Baget, Michel Leclère, Marie-Laure Mugnier, and Eric Salvat, 'On rules with existential variables: Walking the decidability line', *Artificial Intelligence*, **175**(9-10), 1620–1654, (2011).
- [7] Philippe Besnard and Anthony Hunter, *Elements of Argumentation*, MIT Press, 2008.
- [8] Philippe Besnard and Anthony Hunter, 'Constructing argument graphs with deductive arguments: a tutorial', *Argument & Computation*, **5**(1), 5–30, (2014).
- [9] Meghyn Bienvenu, 'On the complexity of consistent query answering in the presence of simple ontologies', in *Proceedings of AAAI'12 Conference on Artificial Intelligence*, (2012).
- [10] Andrea Cali, Georg Gottlob, and Thomas Lukasiewicz, 'A general datalog-based framework for tractable query answering over ontologies', *Web Semantics: Science, Services and Agents on the World Wide Web*, **14**, 57–83, (2012).
- [11] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, and Riccardo Rosati, 'Tractable reasoning and efficient query answering in description logics: The dl-lite family', *J. Autom. Reasoning*, **39**(3), 385–429, (2007).
- [12] Claudette Cayrol, Sylvie Doutre, and Jérôme Mengin, 'On decision problems related to the preferred semantics for argumentation frameworks', *Journal of logic and computation*, **13**(3), 377–403, (2003).
- [13] Claudette Cayrol and Marie-Christine Lagasque-Schiex, 'On the acceptability of arguments in bipolar argumentation frameworks', in *Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'05)*, 378–389, Springer, (2005).
- [14] Madalina Croitoru and Srdjan Vesic, 'What can argumentation do for inconsistent ontology query answering?', in *Scalable Uncertainty Management (SUM'13)*, 15–29, Springer, (2013).
- [15] Sylvie Doutre and Jérôme Mengin, 'On sceptical versus credulous acceptance for abstract argument systems', in *Logics in Artificial Intelligence*, 462–473, Springer, (2004).
- [16] 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).
- [17] Paul E Dunne and Trevor JM Bench-Capon, 'Two party immediate response disputes: properties and efficiency', *Artificial Intelligence*, **149**(2), 221–250, (2003).
- [18] Morten Elvang-Gøransson, Paul J Krause, and John Fox, 'Acceptability of arguments as logical uncertainty', in *Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'93)*, 85–90, Springer, (1993).
- [19] Alexander Eustis, *Hypergraph Independence Numbers*, Ph.D. dissertation, University of California San Diego, 2013.
- [20] Hadassa Jakobovits and Dirk Vermeir, 'Dialectic semantics for argumentation frameworks', in *Proceedings of the 7th International Conference on Artificial intelligence and Law*, pp. 53–62. ACM, (1999).
- [21] Mark W Johnson, Peter McBurney, and Simon Parsons, 'When are two protocols the same?', in *Communication in Multiagent Systems*, 253–268, Springer, (2003).
- [22] Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati, Marco Ruzzi, and Domenico Fabio Savo, 'Inconsistency-tolerant semantics for description logics', in *Proceedings of the Fourth International Conference on Web Reasoning and Rule Systems*, RR'10, pp. 103–117, Berlin, Heidelberg, (2010). Springer-Verlag.
- [23] Kuno Lorenz, 'Basic objectives of dialogue logic in historical perspective', *Synthese*, **127**(1-2), 255–263, (2001).
- [24] Deborah L McGuinness and Peter F Patel-Schneider, 'Usability issues in knowledge representation systems', in *Proceedings of AAAI'98*, pp. 608–614, (1998).
- [25] Sanjay Modgil and Martin Caminada, 'Proof theories and algorithms for abstract argumentation frameworks', in *Argumentation in artificial intelligence*, eds., Guillermo Simari and Iyad Rahwan, 105–129, Springer, (2009).
- [26] Marie-Laure Mugnier, Marie-Christine Rousset, and Federico Ulliana, 'Ontology-mediated queries for nosql databases', in *Proceedings of AAAI'16 Conference on Artificial Intelligence*, (2016).
- [27] Antonella Poggi, Domenico Lembo, Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati, 'Linking data to ontologies', in *Journal on data semantics X*, 133–173, Springer, (2008).
- [28] Henry Prakken, 'Relating protocols for dynamic dispute with logics for defeasible argumentation', *Synthese*, **127**(1-2), 187–219, (2001).
- [29] Phan Minh Thang, Phan Minh Dung, and Nguyen Duy Hung, 'Towards a common framework for dialectical proof procedures in abstract argumentation', *Journal of Logic and Computation*, **19**(6), 1071–1109, (2009).
- [30] Gerard AW Vreeswijk and Henry Prakken, 'Credulous and sceptical argument games for preferred semantics', in *Logics in Artificial Intelligence*, 239–253, Springer, (2000).

Adaptive Binary Quantization for Fast Nearest Neighbor Search

Zhujin Li¹ and Xianglong Liu^{*2} and Junjie Wu³ and Hao Su⁴

Abstract. Hashing has been proved an attractive technique for fast nearest neighbor search over big data. Compared to the projection based hashing methods, prototype based ones own stronger capability of generating discriminative binary codes for the data with complex inherent structure. However, our observation indicates that they still suffer from the insufficient coding that usually utilizes the complete binary codes in a hypercube. To address this problem, we propose an adaptive binary quantization method that learns a discriminative hash function with prototypes correspondingly associated with small unique binary codes. Our alternating optimization adaptively discovers the prototype set and the code set of a varying size in an efficient way, which together robustly approximate the data relations. Our method can be naturally generalized to the product space for long hash codes. We believe that our idea serves as a very helpful insight to hashing research. The extensive experiments on four large-scale (up to 80 million) datasets demonstrate that our method significantly outperforms state-of-the-art hashing methods, with up to 58.84% performance gains relatively.

1 Introduction

In the past decade, hashing technique has been widely studied for fast nearest neighbor search, owing to its successful applications in many areas like large-scale visual search [4, 23, 33, 25, 30], machine learning [9, 19, 26, 21], recommendation system [22], etc. As the most essential concept in hashing based nearest neighbor search, Locality-Sensitive Hashing (LSH) was first introduced in [8], which guarantees that the nearest neighbors share the similar binary codes, and thus enables fast search with compressed storage over gigantic databases.

The pioneer LSH research adopted a random projection paradigm for the metrics like l_p -norm ($p \in (0, 2]$) [1]. Due to its simple form and efficient computation, *projection based hashing* has become the most widely accepted hashing paradigm, where the data point is first projected along certain direction and further quantized to a binary value. Since the randomly generated projection vectors are independent from the data, they usually suffer from the heavy redundancy and the lack of discriminative power for nearest neighbors. To leverage the information contained in the data, recent studies attempted to learn the projection based hash function, which have shown the success in the generation of discriminative hash codes [31, 3, 13, 12, 27, 16, 33, 7, 22, 17, 34, 32, 11].

Despite the progress in the projection based hashing research, state-of-the-art methods still cannot well approximate the nearest neighbor relations using their binary codes. This is mainly because that the linear form is somewhat beyond the strong capability of capturing the data characteristics with complex inherent structures [5, 24]. Although the nonlinear mapping techniques, like kernel which uplifts the data into an informative space, have been widely used to alleviate the problem [28, 15, 14, 18], they are a little time-consuming on the one hand, and still hard to exploit the underlying data structures using the binary quantization on the other hand.

In the literature, clustering has been proved a powerful quantization method to well model the complex relationships among data using a number of prototypes. This inspires the recent hashing studies that attempt to exploit the clustering structure among data in the binary quantization. Typical methods include spherical hashing (SPH) [6] and K-means hashing (KMH) [5], both of which explicitly pursued a number of prototypes to approximate the data relations, and adopted different coding schemes to quantize the data samples based on these prototypes. Different from projection based hashing where each hash function is parameterized by the projection vectors, these *prototype based hashing* methods define the hash functions based on the discovered prototypes, and promisingly increase the search performance with much less quantization loss.

Existing prototype based hashing methods like KMH make use of the complete binary code set, which geometrically forms a hypercube with a fixed dimension and structure among the codes (or the vertices). In practice, the real-world data usually distribute with a complex structure, which can be hardly characterized by such a hypercube. A demonstrative case on a subset of SIFT-1M dataset is presented in Figure 1, where KMH as the typical prototype based hashing outperforms the state-of-the-art projection based ITQ (see Figure 1 (a) and (b)), however using 3-bit codes the Hamming distances among the prototypes (marked by stars with different colors) cannot well approximate the original ones (the hypercube is skewed in Figure 1(b)).

In fact, a better coding solution only relying on a small subset of binary codes (instead of the complete set) can largely reduce the quantization loss (see Figure 1 (c)). This is because the incomplete coding can match with the data distribution in a more reasonable way, and thus better approximate neighbor relations. Motivated by this observation, this paper proposes an adaptive binary quantization (ABQ) method that can pursue a discriminative hash function with varying number of prototypes, each of which is associated with a unique and compact binary code. The prototype set and codes are jointly discovered to respectively characterize the data distribution in original space and align the code space to the prototype distribution. Therefore, the learnt prototype based hash function can promise

¹ State Key Lab of Software Development Environment, Beihang University, China, email: lizhujin@outlook.com

² State Key Lab of Software Development Environment, Beihang University, China, email: xlliu@nlsde.buaa.edu.cn, *corresponding author

³ Beihang University, China, email: wujj@buaa.edu.cn

⁴ Stanford University, USA, email: haosu@stanford.edu

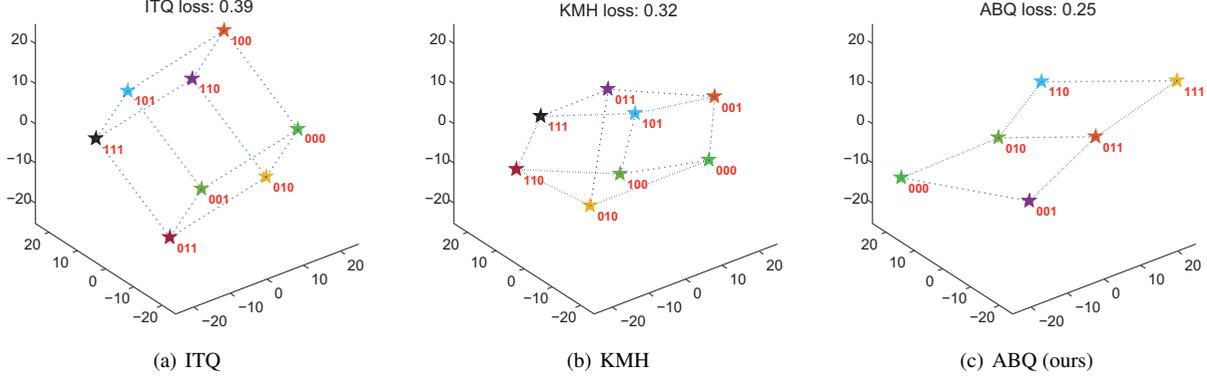


Figure 1. The geometric view of the binary quantization using different methods on a subset of SIFT-1M (projected into 3-dimensional space using PCA like ITQ method). The quantization loss is computed according to (3).

discriminative binary codes that can largely approximate the neighbor structures. We further apply product quantization to generalizing our method for long hash codes. Experimental results over four large-scale datasets demonstrate that the proposed method significantly outperforms state-of-the-art hashing methods.

2 Prototype Based Binary Quantization

Along the direction of prototype based hashing, this section will present our proposed adaptive binary quantization method (denoted as ABQ hereafter) in details.

2.1 Hash Function with Prototypes

Supposing we have a set of n training samples, we denote $\mathbf{x}_i \in \mathbb{R}^{d \times 1}$, $i = 1 \dots n$ to be the feature vector of i -th sample, where d is the feature dimension. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ be the data matrix. Our basic idea is to learn an informative binary hash function that encodes the training data \mathbf{X} into b -length hash codes $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n] \in \{-1, 1\}^{b \times n}$, which show sensitivity to neighbor structures of the data.

The literature has proved that as representative samples among the large-scale database, the prototypes show robustness to the more general metric structure for the data in high dimensional space. Subsequently, in order to capture the neighbor structure using the prototype based hashing, the hash codes should preserve the relations among the prototypes. To meet this goal, one simple, yet powerful way is to assign each prototype a unique binary code in certain order.

In particular, a set of prototypes $\mathcal{P} = \{\mathbf{p}_k | \mathbf{p}_k \in \mathbb{R}^d\}$ are learnt from the training data, and each prototype is associated with a b bit binary code $\mathbf{c}_k \in \{-1, +1\}^b$, forming a binary codebook \mathcal{C} . Then for any data point \mathbf{x} , it can be represented by its nearest prototype $\mathbf{p}_{i^*(\mathbf{x})}$ according to the specific distance function $d_o(\cdot, \cdot)$, where

$$i^*(\mathbf{x}) = \arg \min_k d_o(\mathbf{x}, \mathbf{p}_k), \quad (1)$$

and encoded by the code $\mathbf{c}_{i^*(\mathbf{x})}$ associated with $\mathbf{p}_{i^*(\mathbf{x})}$.

Subsequently, we can define the hash function $h(\mathbf{x})$ as

$$h(\mathbf{x}) = \mathbf{c}_{i^*(\mathbf{x})}. \quad (2)$$

Most of existing hashing methods attempt to pursue a series of hash functions, each of which generates a hash bit, forming a long

hash code. Therefore, these methods have to append additional constraints to reduce the redundancy among these individual bits, which usually degenerates the performance with unreasonable assumptions. Our prototype based hashing like the most related work k-means hashing [5] can exploit the complex data structure and jointly generate a number of hash bits at the same time.

Ideally, the small set of representative prototypes can reduce the computation and introduce sparsity without using the full dataset in binary quantization step. Meanwhile, they can capture the discriminative essence of the dataset with the sensitivity to metric structure and the robustness to overfitting. Therefore, choosing the positioning of the prototypes wisely can lead to a drastically reduced effort while maintaining the discriminative power of the original dataset.

2.2 Space Alignment

The binary codes encoding the data are constrained in the vertices of a hypercube with constant affinities between them. However, in practice it rarely happens that the data geometrically distribute in such a perfect structure. Therefore, an optimal binary coding strategy is highly required to jointly find the discriminative prototypes and their associated binary codes, which respectively characterize the inherent data relations, and maintain the affinities between samples in Hamming space.

Intuitively, the prototype based hash function h should approximate the relations between any two samples \mathbf{x}_i and \mathbf{x}_j using their binary codes. A straightforward way is to concentrate on the distance consistence so that codes in Hamming space will be aligned with the original data distribution. Formally, we introduce the quantization loss to measure the space alignment:

$$\mathcal{Q}(\mathbf{Y}, \mathbf{X}) = \frac{1}{n^2} \sum_{i,j=1}^n \|\lambda d_o(\mathbf{x}_i, \mathbf{x}_j) - d_h(\mathbf{y}_i, \mathbf{y}_j)\|^2 \quad (3)$$

where $d_h(\mathbf{y}_i, \mathbf{y}_j) = \frac{1}{2} \|\mathbf{y}_i - \mathbf{y}_j\|$ is the square root of the Hamming distance between $\mathbf{y}_i = h(\mathbf{x}_i)$ and $\mathbf{y}_j = h(\mathbf{x}_j)$, and λ is a constant scale parameter for the space alignment.

The above loss function involves n^2 sample pairs, which prevented the efficient learning over a large training set. As we mentioned above, the prototypes, as promising representatives of the whole data, have been proved to be able to substantially reduce the computation in many applications. Therefore, for any \mathbf{x}_i the distance from

another sample \mathbf{x}_j can be approximated as follows:

$$d_o(\mathbf{x}_i, \mathbf{x}_j) \approx d_o(\mathbf{x}_i, \mathbf{p}_{i^*(\mathbf{x}_j)}). \quad (4)$$

Motivated by the fact that the hash code of each sample \mathbf{x}_i is actually equivalent to that of its nearest prototypes, namely, $\mathbf{y}_i = \mathbf{c}_{i^*(\mathbf{x}_i)}$, the above loss function can be rewritten in a more simple and efficient way with respect to the prototypes \mathcal{P} and their binary codes \mathcal{C}

$$\mathcal{Q}(\mathcal{P}, \mathcal{C}, i^*(\mathbf{X})) = \sum_{i=1}^n \sum_{k=1}^{|\mathcal{P}|} \frac{w_k}{n^2} \|\lambda d_o(\mathbf{x}_i, \mathbf{p}_k) - d_h(\mathbf{c}_{i^*(\mathbf{x}_i)}, \mathbf{c}_k)\|^2,$$

where w_k is the number of samples represented by \mathbf{p}_k .

Note that the above approximation actually corresponds to the widely-used asymmetric distance, where the database samples are substituted by their prototypes. The literature has shown that such asymmetric approximation usually owns great power to alleviate the quantization loss. Minimizing the above loss leads to a set of prototypes that well capture the intrinsic neighbor structure among the data, and thus a discriminative coding solution that consistently preserves the original relations in Hamming space. Besides, the above loss actually enforces that the close samples in the original space can be clustered in the same group represented by one prototype, and meanwhile their hash codes also maintain the distribution in Hamming space, which together align the neighbor structures between the two spaces.

Therefore, we can formulate the hashing problem in terms of the space alignment as follows:

$$\begin{aligned} \min_{\mathcal{P}, \mathcal{C}, i^*(\mathbf{X})} \quad & \mathcal{Q}(\mathcal{P}, \mathcal{C}, i^*(\mathbf{X})) \\ \text{s.t.} \quad & \mathbf{c}_k \in \{-1, 1\}^b; \quad \mathbf{c}_k^T \mathbf{c}_l \neq b, \quad l \neq k. \end{aligned} \quad (5)$$

Here, the constraints on the binary codebook \mathcal{C} will guarantee that each prototype will be assigned a unique binary code.

It should be pointed out that here the number of prototypes or the size of the codebook isn't fixed beforehand, which is quite different from prior hashing research like [6, 5] where all possible binary codes (*i.e.*, 2^b using b bits) are assumed to be used in the binary quantization. Indeed, we adaptively decide the number in our optimization according to the data metric structure. To some extent, this strategy will avoid the rigorous and difficult alignment between the prototypes and the hypercube binary codes, and thus faithfully helps discover more consistent and discriminative prototypes and the corresponding codebook.

By solving the above problem, the prototype set \mathcal{P} can be obtained that captures the overall data distribution, which can also be reflected by the codebook \mathcal{C} . Each prototype will be associated with a distinct binary code, which together serve as a hash function that encodes those points belonging to the prototype using the corresponding binary code. For a novel sample \mathbf{x} , its hash bits can be computed fast by first determining its nearest prototype according to (1), and then assigning the binary code according to (2).

3 Alternating Optimization

To solve the above problem with respect to a small b , we present an alternating optimization solution, which pursues the near-optimal prototypes and adaptively determine the corresponding binary codes in an efficient way. For the efficiency, usually we choose a small b (*e.g.*, $b \leq 8$), and later we will discuss how to obtain a much longer hash code.

3.1 Adaptive Coding

Supposing we have the prototypes and the assignment index for each sample (see the initialization in Section 3.4), the problem turns to the discriminative binary coding that consistently keeps the distribution information of the samples in the original space. Although k-means hashing [5] as the most related work can capture the cluster structure and find an encouraging binary coding solution, its discriminative power is still limited due to the concentration on the full hypercube structure, which is beyond the true data distribution in practice.

Quite different from the previous research, we adopt an adaptive coding that directly finds the binary codes most consistent with the prototypes. Given the prototypes \mathcal{P} and the assignment of each sample, we will sequentially find a locally optimal binary code for each prototype in a greedy way. Specifically, supposing the prototypes $\mathbf{p}_1, \dots, \mathbf{p}_l$ ($1 \leq l \leq |\mathcal{P}|$) have been respectively assigned the binary codes $\mathbf{c}_1, \dots, \mathbf{c}_l$, we next select the optimal code \mathbf{c}_k for prototype \mathbf{p}_k from the set $\tilde{\mathcal{C}} = \{-1, 1\}^b - \{\mathbf{c}_1, \dots, \mathbf{c}_l\}$ of remaining hash codes. Then for \mathbf{c}_k , the objective function in (5) turns to

$$\begin{aligned} \min_{\mathbf{c}_k \in \tilde{\mathcal{C}}} \quad & \sum_{i^*(\mathbf{x}_i)=k} \sum_{k' \neq k} w_{k'} \|\lambda d_o(\mathbf{x}_i, \mathbf{p}_{k'}) - d_h(\mathbf{c}_k, \mathbf{c}_{k'})\|^2 \\ & + \sum_{i^*(\mathbf{x}_i) \neq k} w_k \|\lambda d_o(\mathbf{x}_i, \mathbf{p}_k) - d_h(\mathbf{c}_{i^*(\mathbf{x}_i)}, \mathbf{c}_k)\|^2. \end{aligned} \quad (6)$$

Since the code space is quite limited ($|\tilde{\mathcal{C}}| \leq 2^b$), the above optimal code pursuit can be completed efficiently using exhaustive search over $\tilde{\mathcal{C}}$.

As to the first step, we can simply choose any binary code as the optimal \mathbf{c}_1 . This is because the code space is highly symmetric with a hypercube structure. After repeating $|\mathcal{P}|$ steps, we can assign each prototype a unique hash code with the minimal coding loss, and meanwhile greedily keep the original data distribution information.

3.2 Prototype Update

While the binary codebook \mathcal{C} is discovered, the prototypes \mathcal{P} should be further calibrated to simultaneously capture the data distribution and align it to the geometric structure in the code space. Therefore, the above problem turns to:

$$\min_{\mathcal{P}} \sum_{i=1}^n \sum_{k=1}^{|\mathcal{C}|} w_k \|\lambda d_o(\mathbf{x}_i, \mathbf{p}_k) - d_h(\mathbf{c}_{i^*(\mathbf{x}_i)}, \mathbf{c}_k)\|^2. \quad (7)$$

To pursue a set of prototypes that well represent the data, we adopt a two-step optimization, since the prototype discovery involves the assignment variable $i^*(\mathbf{x}_i)$. We first determine which prototype the samples belong to, and then update the position of each prototype based on the assignment.

Deriving from (7), the prototype that yields the least loss for each sample \mathbf{x}_i can be found using a simple search:

$$\min_{k' \leq |\mathcal{C}|} \sum_{k=1}^{|\mathcal{C}|} w_k \|\lambda d_o(\mathbf{x}_i, \mathbf{p}_k) - d_h(\mathbf{c}_{k'}, \mathbf{c}_k)\|^2. \quad (8)$$

With the assignment of each sample, we approximately recalculate the position of each prototype:

$$\mathbf{p}_k = \frac{1}{w_k} \sum_{i^*(\mathbf{x}_i)=k} \mathbf{x}_i, \quad 1 \leq k \leq |\mathcal{C}|. \quad (9)$$

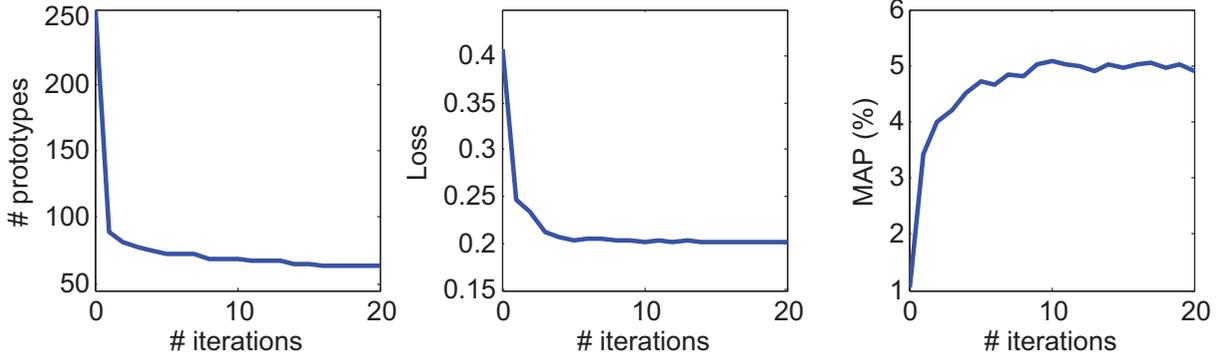


Figure 2. Demonstration of the adaptive binary quantization in one subspace ($b = 8$) on GIST-1M using 32 bits.

Algorithm 1 Adaptive Binary Quantization.

Input: Training data \mathbf{X} , and the binary code length b .

Output: Hash function h , the prototype set \mathcal{P} and the corresponding binary code set \mathcal{C} .

- 1: Initialize the assignment index $i^*(\mathbf{X})$ and the prototype set \mathcal{P} using k-means.
 - 2: Initialize the scale parameter λ according to (11).
 - 3: **repeat**
 - 4: **for** $l = 1, \dots, |\mathcal{P}|$ **do**
 - 5: Find the local optimal code \mathbf{c}_l for \mathbf{p}_l by solving (6);
 - 6: **end for**
 - 7: Update the prototype set \mathcal{P} according to (8) and (9);
 - 8: Update the distribution $i^*(\mathbf{X})$ according to (10);
 - 9: **until** convergence
-

In this step the number of the prototypes varies, *i.e.*, \mathcal{P} is shrunk, where the uninformative prototypes are eliminated. This is the most different part from the previous research. Subsequently, the prototype set can gradually adapt the binary codes to the data distribution in the alternating optimization.

Figure 2 demonstrates how the performance benefits from the adaptive prototype set, where as the number of prototypes decreases, only a subset of the binary codes in the hypercube are utilized to maximally capture the neighbor relations among data (also see Figure 1(c)), significantly reducing the quantization loss and meanwhile improving the precision of nearest neighbor search.

3.3 Distribution Update

After the prototype set \mathcal{P} is updated, the binary codebook size in the next alternating round will be also determined. Moreover, the data distribution with respect to \mathcal{P} , characterized by the variable $i^*(\mathbf{X})$, will change slightly. Since the binary coding should maximally preserve the data distribution, we further append an assignment updating step to capture the distribution variation. This can be easily done by employing a similar step in k-means:

$$i^*(\mathbf{x}_i) = \arg \min_{k \leq |\mathcal{P}|} d_o(\mathbf{x}_i, \mathbf{p}_k). \quad (10)$$

This is consistent with the hash function definition in (1), guaranteeing that the hash function can discriminatively preserve the intrinsic data relations based on the prototypes.

3.4 Algorithm Details

Algorithm 1 lists the main steps of our adaptive binary quantization, where some algorithm details are discussed as follows.

3.4.1 Initialization

To start the alternating optimization, we should first initialize the indices $i^*(\mathbf{X})$ and the prototype set \mathcal{P} . In practice this can be completed by first fixing the size of \mathcal{P} to 2^b , and then performing the classical k-means algorithm on the training data \mathbf{X} , where the cluster centers are treated as the prototypes \mathcal{P} , and each sample is assigned to its nearest prototype.

Here we simply adopt k-means clustering to initialize the prototypes. Although the quality of the prototypes depends on the clustering algorithm or seed selection in the k-means initialization phase, we found that they do not affect the overall performance much. This is mainly because the positions and the quantity of the prototypes will be refined gradually in the iterative optimization to align the data distribution to the code space, and even with a coarse initialization, one can still obtain the identical informative prototypes in a number of iterations. Besides, since it has minor effects on the performance according to our empirical results, we randomly select the order in which the prototypes are processed in the adaptive coding step, *i.e.*, Equation (6).

As to the scale parameter λ in Equation (3), it is intuitively adopted to make the distances comparable between the original and Hamming space. Since we found it usually insensitive to the binary coding process, we simply set it to a constant based on the initialization using k-means, assuming that all 2^b prototypes are assigned different binary codes:

$$\lambda = \frac{\frac{1}{2^b} \sum_{\mathbf{c}_k, \mathbf{c}_l \in \{-1, 1\}^b} d_h(\mathbf{c}_k, \mathbf{c}_l)}{\frac{1}{n} \sum_{i=1}^n \sum_{k=1}^{2^b} d_o(\mathbf{x}_i, \mathbf{p}_k)}. \quad (11)$$

3.4.2 Product Quantization

For a desired level of performance, usually a long hash code is required in many practical applications. However, for the representation power and the computational efficiency, the prototype number usually ranges from tens to hundreds at most, which makes the above algorithm only generate small codes with $b \leq 8$. Fortunately, our problem can be naturally generalized to product space for longer hash codes, following the idea of product quantization (PQ) [10, 5]. In order to generate a sufficient long code of $b^* \gg b$ length, the PQ

method divide the original space into $M = b^*/b$ subspaces, in which a small code of $b = b^*/M$ length is respectively associated with each sample and concatenated as a long one in a Cartesian product manner.

Specifically, a vector \mathbf{x} is represented as M sub-vectors in the way $\mathbf{x} = [\hat{\mathbf{x}}^{(1)}, \hat{\mathbf{x}}^{(2)}, \dots, \hat{\mathbf{x}}^{(M)}]^T$, where $\hat{\mathbf{x}}^{(m)} \in \mathbb{R}^{d \times 1}$ is the m -th sub-vector of \mathbf{x} , and its hash code $\hat{\mathbf{y}}^{(m)} \in \{-1, 1\}^{b \times 1}$ can be generated using the proposed adaptive quantization based on the sub-prototypes $\hat{\mathbf{p}}^{(m)} \in \mathbb{R}^{d \times 1}$ and the sub-codebook $\hat{\mathbf{c}}^{(m)} \in \{-1, 1\}^{b \times 1}$. The hash code \mathbf{y} for vector \mathbf{x} is the concatenation of the sub-codes of its sub-vectors: $\mathbf{y} = [\hat{\mathbf{y}}^{(1)}, \hat{\mathbf{y}}^{(2)}, \dots, \hat{\mathbf{y}}^{(M)}]$.

Recall that Equation (4) corresponds to the asymmetric distance computation (ADC) in PQ. If the original distance d_o is defined as Euclidean distance, PQ can approximate the distance between two vectors using codewords (prototypes):

$$\begin{aligned} d_o(\mathbf{x}_i, \mathbf{x}_j) &\approx d_o(\mathbf{x}_i, \mathbf{p}_{i^*(\mathbf{x}_j)}) \\ &= \sqrt{\sum_{m=1}^M d_o(\hat{\mathbf{x}}_i^{(m)}, \hat{\mathbf{p}}_{i^*(\mathbf{x}_j)}^{(m)})^2} \end{aligned} \quad (12)$$

In each subspace, the learnt codes can approximate the original distance d_o well using the Hamming based distance d_h , i.e., $\lambda d_o(\hat{\mathbf{x}}_i^{(m)}, \hat{\mathbf{p}}_k^{(m)}) \approx d_h(\hat{\mathbf{y}}_i^{(m)}, \hat{\mathbf{c}}_k^{(m)})$. Then, with the definition of the distance d_h , we have:

$$\begin{aligned} \lambda d_o(\mathbf{x}_i, \mathbf{p}_k) &\approx \sqrt{\sum_{m=1}^M \frac{1}{4} \|\hat{\mathbf{y}}_i^{(m)} - \hat{\mathbf{c}}_k^{(m)}\|^2} \\ &= \frac{1}{2} \|\mathbf{y}_i - \mathbf{c}_k\| = d_h(\mathbf{c}_{i^*(\mathbf{x}_i)}, \mathbf{c}_k) \end{aligned} \quad (13)$$

Putting (12) and (13) together, it is easy to show that the original distance between any two samples can be approximated by the Hamming based distance between their hash codes in the Cartesian space:

$$\lambda d_o(\mathbf{x}_i, \mathbf{x}_j) \approx d_h(\mathbf{c}_{i^*(\mathbf{x}_i)}, \mathbf{c}_{i^*(\mathbf{x}_j)}) \quad (14)$$

Note that the above approximation requires that the scale parameter λ remains the same across all subspaces, which holds roughly in practice over many datasets. Therefore, we set it to the average of the values computed according to Equation (11) in all subspaces.

Prior research has pointed out that equally splitting the space into M parts might result in ineffective hash codes, due to the unbalanced information distribution [5]. Usually independent subspaces are pursued to balance the information among the small codes of each sample. Therefore, in the space decomposition, we apply the eigenvalue allocation method to evenly distribute the variance using PCA projection without dimension reduction [2]. One can also further append an adaptive bit allocation to maximally capture the data information using different number (or code length) of hash bits [20].

3.4.3 Complexity

To learn the hash functions that can generate binary codes of b^* length, we need to compute the small codes in $M = b^*/b$ independent subspaces. For each subspace, there are maximally 2^b prototypes in d/M feature space.

At the training stage, the adaptive coding greedily finds the locally optimal code for each prototype over $\{-1, 1\}^b$ in $O(nd2^{2b})$ time. The prototype and distribution update steps require at most $O(nd2^{2b})$ time to compute the distances between training samples

and prototypes. Therefore, when using t (usually $t \leq 20$) iterations in the alternating optimization, totally $O(2^{2b}ndt)$ time is spent on the training. Since the code space is quite limited for each subspace ($b \leq 8$), the term 2^{2b} can be treated as a constant. Therefore, it can be considered that the training time scales linearly with respect to the size of the training set.

When it comes to the online search, for each query point the hash function needs $O(2^b d)$ time to compute the nearest prototype and $O(1)$ time for the code assignment, which is linear to the feature dimension d as most projection based hashing methods like LSH [1] and ITQ [3]. Furthermore, our method only utilizes a small subset (e.g., a quarter) of codes, which directly reduce the time consumption at the stage of hash code generation.

Compared with other lookup based methods like KMH, our method usually owns faster speed when performing online search in practice (see Table 1). The high efficiency of our method mainly benefits from the adaptive coding combined with PQ, which allows to compute and store only a small number of codewords (prototypes), while presenting a large dictionary to maximally preserve the information of data distribution in the original space.

4 Experiments

In this section we will evaluate the proposed adaptive binary quantization (ABQ) on large-scale nearest neighbor search, and compare it with several state-of-the-art hashing algorithms, including the classical projection based ones like Locality Sensitive Hashing (LSH) [1], Spectral Hashing (SH) [31], Kernelized Locality Sensitive Hashing (KLSH) [14], Anchor Graph Hashing (AGH) [18], Iterative Quantization (ITQ) [3] and Kronecker Binary Embedding (KBE) [35], and two representative prototype based ones: Spherical Hashing (SPH) [6] and K-Means Hashing (KMH) [5].

- **LSH**: LSH generates Gaussian random projection vectors and preserves the locality with high probability.
- **SH**: SH formulates the binary coding problem as a spectral embedding in the Hamming space, and generalizes the approximated solution for out-of-sample extension.
- **KLSH**: KLSH constructs randomized locality-sensitive functions with arbitrary kernel functions. We feed it the Gaussian RBF kernel $\|(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\alpha \|\mathbf{x}_i - \mathbf{x}_j\|^2)$ and 300 support samples. The kernel parameter α is tuned to an appropriate value on each dataset.
- **AGH**: AGH approximates the intrinsic structure underlying the data based on anchors, and generates hash codes based on the anchor representation.
- **ITQ**: ITQ iteratively finds the data rotation in a subspace to minimize the binary quantization error.
- **KBE**: KBE generates linear hash functions with a structured matrix, which can achieve fast hash coding over high-dimensional data. We adopt the optimized version of Kronecker projection.
- **SPH**: SPH iteratively adjusts the spherical planes to generate independent and balanced partitions, which serve as the nonlinear hash functions based on the distances to the centers. In SPH, each partition can generate a hash bit independently.
- **KMH**: KMH generates affinity affine clusters using k-means in the partitioned subspaces of the training features, and maps each cluster to a binary hash code for the out-of-sample coding.

Table 1. Hashing performance and time efficiency on SIFT-1M and GIST-1M.

		MAP			PH (32 BITS)		TIME (128 BITS)	
		32 BITS	64 BITS	128 BITS	$r = 1$	$r = 2$	TRAIN (S)	SEARCH (S)
SIFT-1M	LSH	5.43±0.30	13.00±0.82	26.04±0.68	18.89	19.70	0.03	0.02
	SH	10.70±0.58	17.84±0.37	25.30±0.59	32.20	41.93	0.25	0.25
	KLSH	7.08±0.44	15.61±0.57	29.48±0.72	23.72	23.32	0.28	0.02
	AGH	6.26±0.27	9.11±0.31	11.10±0.23	15.90	11.93	0.55	0.04
	ITQ	9.70±0.14	20.14±0.47	33.23±0.49	28.38	22.09	5.08	0.16
	SPH	8.57±0.12	18.23±0.54	31.11±0.14	26.90	30.82	8.93	0.04
	KMH	11.51±0.27	22.50±0.31	32.06±0.52	35.63	40.00	680.64	0.12
	KBE	6.43±0.31	14.73±0.61	27.65±0.57	20.62	16.97	3.28	0.02
ABQ	12.47±0.26	24.92±0.61	41.34±0.56	41.30	43.09	40.37	0.06	
GIST-1M	LSH	1.34±0.08	3.15±0.07	5.97±0.19	5.41	7.15	0.21	0.05
	SH	1.90±0.23	3.19±0.19	4.92±0.19	8.94	6.58	1.70	0.24
	KLSH	2.41±0.09	5.23±0.18	9.76±0.23	9.31	10.70	0.44	0.05
	AGH	2.09±0.15	3.05±0.10	3.98±0.14	5.55	4.13	0.90	0.09
	ITQ	4.43±0.06	6.93±0.10	9.49±0.15	14.08	17.8	5.87	0.17
	SPH	3.65±0.14	6.97±0.10	11.52±0.19	12.20	17.05	25.24	0.07
	KMH	3.58±0.18	5.57±0.07	6.92±0.07	14.77	17.39	2380.61	0.15
	KBE	-	-	6.58±0.22	-	-	13.66	0.06
ABQ	4.92±0.06	10.06±0.20	16.10±0.17	23.46	17.84	46.10	0.10	

4.1 Evaluation Protocols

To comprehensively evaluate the proposed method, we first employ two well-known large-scale datasets **SIFT-1M** (1M) and **GIST-1M** (1M) [10]. The two datasets respectively contain one million 128-D SIFT and 960-D GIST descriptors, each of which complies with a separate query subset. We respectively construct a training set of 10,000 random samples and a testing set of 1,000 random queries on both datasets. Besides, we employ another two much larger datasets **SIFT-20M** (20M) [10] and **Tiny-80M** (80M) [29], respectively consisting of 20 million 128-D SIFT and 80 million 384-D GIST features. We respectively sample 50,000 and 100,000 points as the training sets, and 3,000 random queries as the testing ones. As to the groundtruth of each query, we select the 1,000 Euclidean nearest neighbors among the database on SIFT-1M, GIST-1M and SIFT-20M, and 5,000 on Tiny-80M.

We adopt two common search schemes to evaluate the hashing performance, *i.e.*, Hamming distance ranking and hash table lookup. The former ranks all candidates based on the Hamming distances from the query, and the later treats points falling within a small Hamming radius r ($r \leq 2$) from the query code as the retrieved results. As to KMh and our ABQ with product quantization, we set $b = 4$ for SIFT features when using less than 64 bits, and $b = 8$ for all other cases. In each experiment, we run 10 times in a workstation with 2.53 GHz Xeon CPU and report the averaged performance.

4.2 Results and Discussions

4.2.1 Euclidean Nearest Neighbor Search

We first evaluate all hashing methods in the task of Euclidean nearest neighbor search over SIFT-1M and GIST-1M. We adopt both precision and recall to comprehensively study their performance. Table 1 lists the mean average precision (MAP) using Hamming distance ranking with respect to different number of hash bits. From the table we can observe that all methods increase their MAP performance when using more hash bits from 32 to 128 bits. Moreover, methods like ITQ, SPH, KMh and our ABQ, which encode the data from

the view of clustering quantization, consistently achieve much better performance than other methods like LSH, SH and AGH. This indicates that it is a promising way to discover a particular quantization strategy for binary hashing. Among all these methods, ABQ obtains the best performance, and gets significant performance gains over the best competitors, *e.g.*, using 128 bits, 24.41% over ITQ on SIFT-1M, and 39.76% over SPH on GIST-1M.

Figure 3 further plots the recall curves with respect to different number of retrieved results on both datasets, where we can get the same conclusion that ABQ performs best in all cases. The reason is mainly that compared to the baselines where the codebook is fixed, ABQ can adaptively generate the codebook of a varying size and well match the binary codes to the prototypes. For the performance of Hamming distance ranking, we compare our ABQ with the baseline methods in terms of precision, besides recall and MAP performance in the paper. Figure 3 also plots the precision curves with respect to different cutting points of the retrieved result lists on SIFT-1M and GIST-1M, where we vary the number of hash bits from 64 to 128. We can see that our ABQ performs best in all cases with significant performance superiority to other methods.

Besides Hamming distance ranking, hash table lookup is another common search strategy over the hash codes. In this case, usually a small code (*e.g.*, 32 bits for one million data) is used to avoid the memory and time consumption derived from the exponentially huge amount of indexing buckets. Table 1 further reports the precision within a small Hamming radius $r = 1$ and $r = 2$ (PH1 and PH2 for short). This is also a popular evaluation metric in practice, because with a small lookup radius, nearest neighbor search can be efficiently completed by only locating data falling in buckets with Hamming distance less than the radius from the query. Similarly, it is easy to see that the ABQ outperforms the baselines with a large margin, *e.g.*, 15.91% and 58.84% PH1 gains over KMh respectively on SIFT-1M and GIST-1M. Compared to SPH and KMh that also exploit the prototype based hash functions, the encouraging precision gains obtained by ABQ indicate that our ABQ can approximate the neighbor relations much better by encoding the data using a sub-

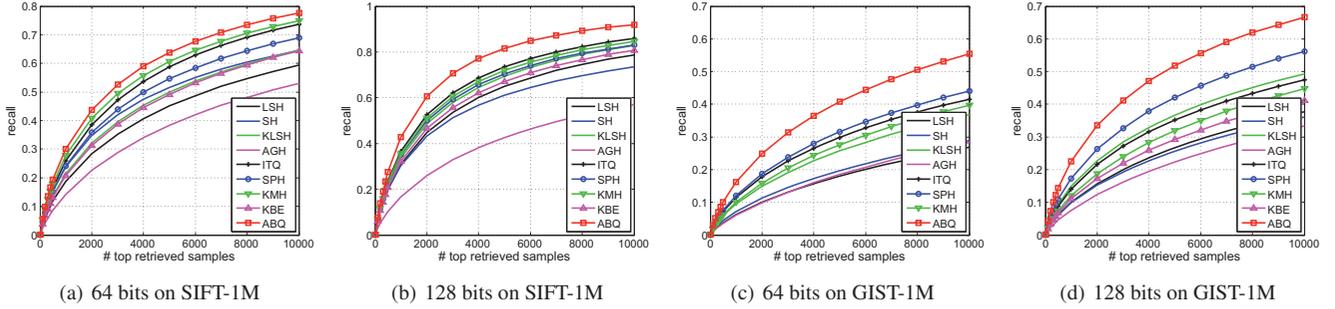


Figure 3. Recall performance of different hashing methods on SIFT-1M and GIST-1M.

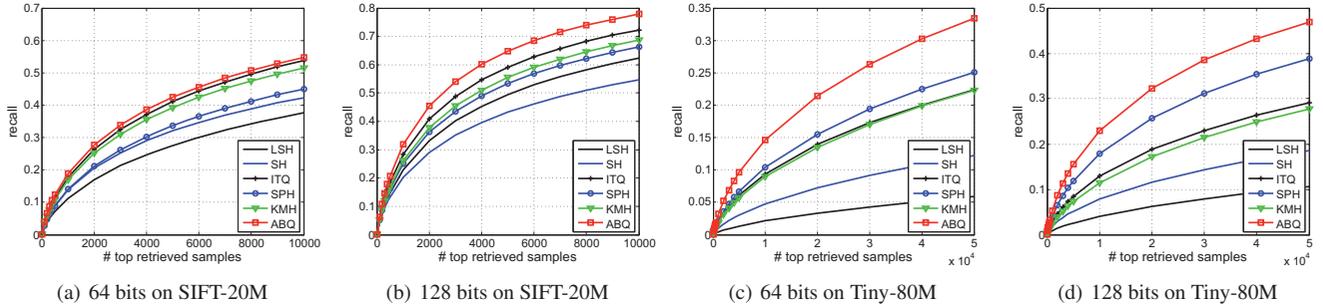


Figure 4. Recall performance of different hashing methods on SIFT-20M and Tiny-80M.

set of binary codes in Hamming space. This intuition is also visually demonstrated in Figure 1 using a subset of SIFT-1M.

4.2.2 Nearest Neighbor Search over Large Datasets

To investigate the performance of different hashing methods over more large-scale dataset, we adopt two of the largest datasets to-date: SIFT-20M and Tiny-80M. Table 2 reports the precision performance in terms of Hamming distance ranking and hash table lookup. Here, due to the facts that in practice users are more concerned about the top ranked results, and while computing the MAP of the full Hamming distance ranking list is quite time-consuming [18, 17], we present the average precision of top 1,000 returned samples ($P@1,000$) instead of MAP with respect to the varying code length (32, 64 and 128). Similar to the results in Table 1, in all cases our ABQ consistently obtains the best precision, especially on Tiny-80M dataset with remarkable superiority, *e.g.*, up to 45.87% performance gain over the best competitor SPH. As to hash table lookup, Table 2 also lists the PH1 performance using 32 bits hash table, from which we can get a similar observation that ABQ shows a better capability of capturing the neighbor structures, and thus covers much more nearest neighbors than all baselines.

Figure 4 respectively depicts the recall curves using 64 and 128 bits on SIFT-20M and Tiny-80M. Compared to all the baselines, our ABQ boosts the recall with the most significant improvement when using more hash bits, and consistently performs best in all cases. On the real-world dataset Tiny-80M, this observation can be more obvious as shown in Figure 4(c) and (d), where the recall of the top 10^4 result list increases largely from 14.61% to 22.93% with more hash bits, and meanwhile the best performance among all baselines is 17.90% achieved by SPH using 128 hash bits. This fact further demonstrates that our ABQ can faithfully boost the overall hashing

performance in terms of precision and recall, using Hamming distance ranking or hash table lookup. As to the precision performance, since the groundtruth number is fixed to a constant number, a similar observation can be obtained as recall performance, which means that the proposed method can obtain the best recall and meanwhile the best precision performance on the two much larger datasets SIFT-20M and Tiny-80M.

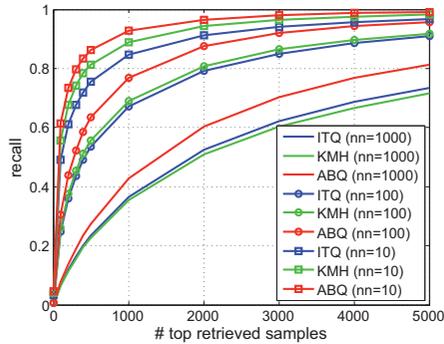
4.2.3 Groundtruth Number Effect

Prior research have pointed out that the number of groundtruth may have effects on the performance [5]. Therefore, to illustrate the robustness of our method with respect to the groundtruth number nn , we further conduct the experiments on SIFT-1M and GIST-1M by varying nn in $\{10, 100, 1000\}$. In Figure 5 we compare the recall performance of ABQ using 128 bits to those of the two state-of-the-art methods ITQ and KMH, which archived the best performance among all baselines as shown in prior experiments. In this figure, we vary the groundtruth number nn on different datasets.

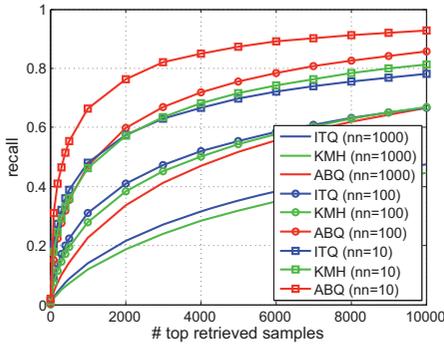
As we can see from the figure, when using more nearest neighbors as the groundtruth (from 10 to 1000), all methods decrease the recall performances. This is because that as the distance between the database point and query increases, the collision probability between them will decrease. Nevertheless, for different settings, our ABQ consistently achieves the best performance and significantly outperforms others in all cases. For instance, with $nn = 100$ on GIST-1M, ABQ can achieve much higher recall than ITQ and KMH, and even better than them with $nn = 10$. This means that our method is very robust to the task of nearest neighbor search. Besides, in all these experiments we adopt the same parameter settings, which indicates that the proposed ABQ is practical without complex parameter tuning.

Table 2. Hashing performance on SIFT-20M and Tiny-80M.

	SIFT-20M					TINY-80M				
	P@1,000			PH (32 BITS)		P@1,000			PH (32 BITS)	
	32 BITS	64 BITS	128 BITS	$r = 1$	$r = 2$	32 BITS	64 BITS	128 BITS	$r = 1$	$r = 2$
LSH	4.34	11.26	22.81	9.86	8.03	0.75	2.18	4.24	0.83	0.51
SH	8.00	13.91	20.19	22.70	17.34	2.77	5.12	9.06	3.37	1.71
ITQ	8.48	17.69	28.47	20.18	14.69	5.25	9.99	14.10	10.59	7.47
SPH	6.06	14.06	25.09	14.72	10.10	4.53	10.90	20.03	9.51	5.67
KMH	8.29	16.90	26.08	22.54	16.18	5.31	9.41	11.92	11.64	7.50
ABQ	8.95	18.92	31.86	25.97	17.59	7.51	15.93	26.56	12.67	7.57



(a) recall on SIFT-1M



(b) recall on GIST-1M

Figure 5. Recall performance of different hashing methods with respect to different number of groundtruth (10, 100, 1000) on SIFT-1M and GIST-1M.

4.2.4 Efficiency Issue

Figure 2 shows that the proposed ABQ can converge fast in less than 10 iterations. Therefore, in practice, the algorithm can achieve efficient training and support the large-scale learning. This is consistent with our complexity analysis in Section 3.4.3, *e.g.*, the training time scales linearly to the size of the training set.

Table 1 further lists the offline training time and online search time when using 128 hash bits on SIFT-1M and GIST-1M. We can see that usually the iterative binary quantization methods like ITQ, SPH, KMH and our ABQ take more training time than the others. This is mainly due to the difficulty of finding an optimal coding solution that can align the Hamming space with the original one. Among these methods, our ABQ costs much less time than KMH, while gives the best performance with a little more training time than SPH and ITQ.

Moreover, at the online search stage, only a small set of prototypes (smaller than 2^b , $b \leq 8$ in each subspace) will be checked, and thus the hashing time is very close to the prior projection based methods. Namely, it can support the real-time nearest neighbor search as the existing methods do.

5 Conclusions

Inspired by our observation that in prototype based hashing there might exist a better coding solution that only utilizes a small subset of binary codes instead of the complete set, this paper proposed an adaptive binary quantization method that jointly pursues a set of prototypes in the original space and a subset of binary codes in the Hamming space. The prototypes and the codes are correspondingly associated and together define the hash function for small hash codes. Our method enjoys fast computation and the capability of generating long hash codes in product space, with discriminative power for nearest neighbor search. The significant performance gains over existing methods were obtained in our extensive experiments on several large datasets, which encourage us to further study the effective coding for binary quantization.

6 Acknowledgment

We would like to thank the referees for their comments, which helped improve this paper considerably. This work was partially supported by the National Natural Science Foundation of China (61402026, 71322104, and 71531001), the Foundation of the State Key Laboratory of Software Development Environment (SKLSDE-2016ZX-04), and National High Technology Research and Development Program of China (SS2014AA012303).

REFERENCES

- [1] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni, ‘Locality-sensitive hashing scheme based on p-stable distributions’, in *SCG*, pp. 253–262, (2004).
- [2] Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun, ‘Optimized product quantization’, *IEEE TPAMI*, **36**(4), 744–755, (April 2014).
- [3] Yunchao Gong and S. Lazebnik, ‘Iterative quantization: A procrustean approach to learning binary codes’, in *IEEE CVPR*, pp. 817–824, (2011).
- [4] Junfeng He, Jinyuan Feng, Xianglong Liu, Tao Cheng, Tai-Hsu Lin, Hyunjin Chung, and Shih-Fu Chang, ‘Mobile product search with bag of hash bits and boundary reranking’, in *IEEE CVPR*, pp. 3005–3012, (2012).
- [5] Kaiming He, Fang Wen, and Jian Sun, ‘K-means hashing: An affinity-preserving quantization method for learning binary compact codes’, in *IEEE CVPR*, pp. 2938–2945, (2013).
- [6] Jae-Pil Heo, Youngwoon Lee, Junfeng He, Shih-Fu Chang, and Sung-Eui Yoon, ‘Spherical hashing’, in *IEEE CVPR*, pp. 2957–2964, (2012).

- [7] Long-Kai Huang, Qiang Yang, and Wei-Shi Zheng, ‘Online hashing’, in *IJCAI*, pp. 1422–1428, (2013).
- [8] Piotr Indyk and Rajeev Motwani, ‘Approximate nearest neighbors: towards removing the curse of dimensionality’, in *ACM STOC*, (1998).
- [9] Prateek Jain, Sudheendra Vijayanarasimhan, and Kristen Grauman, ‘Hashing Hyperplane Queries to Near Points with Applications to Large-Scale Active Learning’, in *NIPS*, 928–936, (2010).
- [10] Herve Jegou, Matthijs Douze, and Cordelia Schmid, ‘Product quantization for nearest neighbor search’, *IEEE TPAMI*, **33**(1), 117–128, (January 2011).
- [11] Qing-Yuan Jiang and Wu-Jun Li, ‘Scalable graph hashing with feature transformation’, in *IJCAI*, pp. 2248–2254, (2015).
- [12] Zhongming Jin, Yao Hu, Yue Lin, Debing Zhang, Shiding Lin, Deng Cai, and Xuelong Li, ‘Complementary projection hashing’, in *IEEE ICCV*, pp. 257–264, (2013).
- [13] Weihao Kong and Wu-Jun Li, ‘Isotropic hashing’, in *NIPS*, pp. 1–8, (2012).
- [14] B. Kulis and K. Grauman, ‘Kernelized locality-sensitive hashing for scalable image search’, in *IEEE ICCV*, (2009).
- [15] Brian Kulis and Trevor Darrell, ‘Learning to hash with binary reconstructive embeddings’, in *NIPS*, pp. 1–8, (2009).
- [16] X. Li, G. Lin, C. Shen, A. van den Hengel, and A. Dick, ‘Learning hash functions using column generation’, in *ICML*, (2013).
- [17] Wei Liu, Cun Mu, Sanjiv Kumar, and Shih-Fu Chang, ‘Discrete graph hashing’, in *NIPS*, (2014).
- [18] Wei Liu, Jun Wang, Sanjiv Kumar, and Shih-Fu Chang, ‘Hashing with graphs’, in *ICML*, pp. 1–8, (2011).
- [19] Wei Liu, Jun Wang, Yadong Mu, Sanjiv Kumar, and Shih-Fu Chang, ‘Compact hyperplane hashing with bilinear functions.’, in *ICML*, (2012).
- [20] Xianglong Liu, Bowen Du, Cheng Deng, Ming Liu, and Bo Lang, ‘Structure sensitive hashing with adaptive product quantization’, *IEEE TCYB*, **PP**(99), 1–12, (2015).
- [21] Xianglong Liu, Xinjie Fan, Cheng Deng, Zhujin Li, Hao Su, and Dacheng Tao, ‘Multilinear hyperplane hashing’, in *IEEE CVPR*, (2016).
- [22] Xianglong Liu, Junfeng He, Cheng Deng, and Bo Lang, ‘Collaborative hashing’, in *IEEE CVPR*, (2014).
- [23] Xianglong Liu, Junfeng He, Bo Lang, and Shih-Fu Chang, ‘Hash bit selection: a unified solution for selection problems in hashing’, in *IEEE CVPR*, (2013).
- [24] Xianglong Liu, Lei Huang, Cheng Deng, Jiwen Lu, and Bo Lang, ‘Multi-view complementary hash tables for nearest neighbor search’, in *IEEE ICCV*, (2015).
- [25] Xianglong Liu, Yadong Mu, Bo Lang, and Shih-Fu Chang, ‘Mixed image-keyword query adaptive hashing over multilabel images’, *ACM TOMM*, **10**(2), 22:1–22:21, (February 2014).
- [26] Yadong Mu, Gang Hua, Wei Fan, and Shih-Fu Chang, ‘Hash-svm: Scalable kernel machines for large-scale visual classification’, in *IEEE CVPR*, (2014).
- [27] Mohammad Norouzi and David J. Fleet, ‘Cartesian k-means’, in *IEEE CVPR*, pp. 2938–2945, (2013).
- [28] Maxim Raginsky and Svetlana Lazebnik, ‘Locality-sensitive binary codes from shift-invariant kernels’, in *NIPS*, pp. 1–8, (2009).
- [29] Antonio Torralba, Rob Fergus, and William T. Freeman, ‘80 million tiny images: A large data set for nonparametric object and scene recognition’, *IEEE TPAMI*, **30**(11), 1958–1970, (2008).
- [30] Qifan Wang, Zhiwei Zhang, and Luo Si, ‘Ranking preserving hashing for fast similarity search’, in *IJCAI*, pp. 3911–3917, (2015).
- [31] Yair Weiss, Antonio Torralba, and Rob Fergus, ‘Spectral hashing’, in *NIPS*, pp. 1–8, (2008).
- [32] Botong Wu, Qiang Yang, Wei-Shi Zheng, Yizhou Wang, and Jingdong Wang, ‘Quantized correlation hashing for fast cross-modal search’, in *IJCAI*, pp. 3946–3952, (2015).
- [33] Bin Xu, Jiajun Bu, Yue Lin, Chun Chen, Xiaofei He, and Deng Cai, ‘Harmonious hashing’, in *IJCAI*, pp. 1820–1826, (2013).
- [34] Felix Yu, Sanjiv Kumar, Yunchao Gong, and Shih-Fu Chang, ‘Circulant binary embedding’, in *ICML*, (2014).
- [35] Xu Zhang, Felix X. Yu, Ruiqi Guo, Sanjiv Kumar, Shengjin Wang, and Shih-Fu Chang, ‘Fast orthogonal projection based on kronecker product’, in *IEEE ICCV*, (2015).

Exploring Parallel Tractability of Ontology Materialization

Zhangquan Zhou¹ and Guilin Qi¹ and Birte Glimm²

Abstract. Materialization is an important reasoning service for applications built on the Web Ontology Language (OWL). To make materialization efficient in practice, current research focuses on deciding tractability of an ontology language and designing parallel reasoning algorithms. However, some well-known large-scale ontologies, such as YAGO, have been shown to have good performance for parallel reasoning, but they are expressed in ontology languages that are not parallelly tractable, i.e., the reasoning is inherently sequential in the worst case. This motivates us to study the problem of parallel tractability of ontology materialization from a theoretical perspective. That is, we aim to identify the ontologies for which materialization is parallelly tractable, i.e., in NC complexity. In this work, we focus on datalog rewritable ontology languages. We identify several classes of datalog rewritable ontologies (called *parallelly tractable classes*) such that materialization over them is parallelly tractable. We further investigate the parallel tractability of materialization of a datalog rewritable OWL fragment DHL (*Description Horn Logic*) and an extension of DHL that allows *complex role inclusion axioms*. Based on the above results, we analyze real-world datasets and show that many ontologies expressed in DHL or its extension belong to the parallelly tractable classes.

1 Introduction

The Web Ontology Language OWL³ is an important standard for ontology languages in the Semantic Web and other application areas. *Materialization* is a basic reasoning service for computing all implicit facts that follow from a given OWL ontology. Since there is an exponential growth of semantic data [18], it is challenging to perform materialization on such large-scale ontologies efficiently.

To make materialization sufficiently efficient and scalable in practice, many works employ parallel reasoning systems. For example, RDFox [19] is a parallel implementation for materialization of datalog rewritable ontology languages. Parallel reasoning is also studied for the ontology language RDFS [22, 26]. There are also parallel implementations for scalable reasoning of highly expressive ontology languages [25, 33]. However, according to [5], even for RDFS and datalog rewritable ontology languages, which have PTime-complete or higher complexity⁴ of reasoning in the worst case, they are not parallelly tractable, i.e., reasoning may be inherently sequential even on a parallel implementation. On the other hand, some well-known large-scale ontologies, such as YAGO, have been shown to have good

performance for parallel reasoning [14], but they are expressed in ontology languages that are not parallelly tractable. The theoretical results on the complexity of ontology languages can hardly explain this. While one can try out different parallel implementations to see whether an ontology can be handled by (one of) them efficiently, the aim of our study is to identify properties that make an ontology parallelly tractable and that can also guide ontology engineers in creating ontologies for which parallel tractability can be guaranteed theoretically. According to [19], many real large-scale ontologies are essentially expressed in the ontology languages that can be rewritten into datalog rules. Thus, we focus on such datalog rewritable ontology languages in this paper. Our aim is to identify the classes of datalog rewritable ontologies such that materialization over these ontologies is parallelly tractable, i.e., in the parallel complexity class NC [5]. This complexity class consists of problems that can be solved efficiently in parallel.

To show that a problem is in the NC class, one can give an NC algorithm that handles this problem in parallel computation [5]. However, current materialization algorithms (e.g., the algorithm used in RDFox [19]) are not NC algorithms, since their computational complexity is PTime-complete. Thus, we study the parallel tractability of materialization by first giving several NC algorithms that perform materialization, and then identifying the corresponding classes of datalog rewritable ontologies (called *parallelly tractable classes*) that can be handled by these NC algorithms. We next study the specific ontology language *Description Horn Logic* (DHL) [6], which is a datalog rewritable fragment of OWL, and investigate what kinds of ontologies expressed in DHL are in the parallelly tractable classes. We give a case of a DHL ontology where materialization can hardly be parallelized. Based on the analysis of this case, we propose to restrict the usage of DHL such that materialization over the restricted ontologies can be handled by the proposed NC algorithms. We further extend the results to an extension of DHL that also allows *complex role inclusion axioms*. Finally, we analyze well-known benchmarks and real-world datasets and show that many ontologies following the proposed restrictions belong to the parallelly tractable classes.

The rest of the paper is organized as follows. In Section 2, we introduce some basic notions. We then give some NC algorithms in Section 3 and Section 4. We study the parallelly tractable materialization of DHL and its extension in Section 5 and Section 6 respectively. In Section 7, we analyze real-world datasets. We then discuss related work in Section 8 and conclude in Section 9. The technical report can be found at “<https://github.com/quanzz/ECAI2016>”.

2 Preliminaries

In this section, we introduce some notions that are used in this paper.

¹ School of Computer Science and Engineering, Southeast University, email: {qzz, gqi}@seu.edu.cn

² Institution of Artificial Intelligence, University of Ulm, email: birte.glimm@uni-ulm.de

³ The latest version is OWL 2: <http://www.w3.org/TR/owl2-overview/>

⁴ We consider the data complexity for materialization here.

Datalog. We discuss the main issues in this paper using standard datalog notions. In datalog [1], a *term* is a variable or a constant. An *atom* A is defined by $A \equiv p(t_1, \dots, t_n)$ where p is a *predicate* (or *relational*) name, t_1, \dots, t_n are terms, and n is the arity of p . If all the terms in an atom A are constants, then A is called a *ground atom*. A datalog *rule* is of the form: ' $B_1, \dots, B_n \rightarrow H$ ',⁵ where H is referred to as the *head atom* and B_1, \dots, B_n the *body atoms*. Each variable in the head atom of a rule must occur in at least one body atom of the same rule. A *fact* is a rule of the form ' $\rightarrow H$ ', i.e., a rule with an empty body and the head H being a ground atom. A datalog program P consists of rules and facts. A *substitution* θ is a partial mapping of variables to constants. For an atom A , $A\theta$ is the result of replacing each variable x in A with $\theta(x)$ if the latter is defined. We call θ a *ground substitution* if each defined $A\theta$ is a ground atom. A *ground instantiation* of a rule is obtained by applying a ground substitution on all the terms in this rule with respect to a finite set of constants occurring in P . Furthermore the ground instantiation of P , denoted by P^* , consists of all ground instantiations of rules in P . The predicates occurring only in the body of some rules are called *EDB predicates*, while the predicates that may occur as head atoms are called *IDB predicates*.

DHL. DHL (short for *description horn logic*) [6] is introduced as an intersection of description logic (DL) and datalog in terms of expressivity. In what follows, **CN**, **RN** and **IN** denote three disjoint countably infinite sets of *concept names*, *role names*, and *individual names* respectively. The set of roles is defined as $\mathbf{R} := \mathbf{RN} \cup \{R^- \mid R \in \mathbf{RN}\}$ where R^- is the *inverse role* of R .

For ease of discussion, we focus on the *simple forms* of axioms shown in the left column of Table 1. These simple forms can be obtained by using well-known *structure transformation* techniques [12]. We define a DHL ontology \mathcal{O} as a triple: $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$, where \mathcal{T} denotes the TBox containing axioms of the forms (T1) and (T2); \mathcal{R} is the RBox that is a set of axioms of the forms (R1-R3); \mathcal{A} is the ABox containing *assertions* of the forms (A1) and (A2). In an axiom of either of the forms (T1-T2 and R1-R3), concepts $A_{(i)}$ and B are either concept names, *top concept* (\top) or *bottom concept* (\perp); R and $S_{(i)}$ are roles in \mathbf{R} . An axiom of the form $A \sqsubseteq B$ is a special case of (T1) where only one concept appears on the left-hand side. For an axiom of the form $A \sqsubseteq \forall R.B$ that is also allowed in DHL, we only consider its equivalent form $\exists R^-.A \sqsubseteq B$.

Table 1: Axioms and corresponding datalog rules

	Axioms	Datalog Rules
(T1)	$A_1 \sqcap A_2 \sqsubseteq B$	$A_1(x), A_2(x) \rightarrow B(x)$
(T2)	$\exists R.A \sqsubseteq B$	$R(x, y), A(y) \rightarrow B(x)$
(R1)	$S \sqsubseteq R$	$S(x, y) \rightarrow R(x, y)$
(R2)	$S \sqsubseteq R^-$	$S(x, y) \rightarrow R(y, x)$
(R3)	$R \circ R \sqsubseteq R$	$R(x, y), R(y, z) \rightarrow R(x, z)$
(R4)	$R_1 \circ R_2 \sqsubseteq R$	$R_1(x, y), R_2(y, z) \rightarrow R(x, z)$
(A1)	$A(a)$	$A(a)$
(A2)	$R(a, b)$	$R(a, b)$

In the initial work of DHL [6], *complex role inclusion axioms* (complex RIAs) of the form $R_1 \circ \dots \circ R_n \sqsubseteq R$ are not considered, although they can be naturally transformed to datalog rules. In this paper, we also consider an extension of DHL (denoted by DHL(\circ)) that allows complex RIAs. Since a complex RIA can be transformed to several axioms of the form (R4), we then require that an RBox \mathcal{R} of a DHL(\circ) ontology can contain axioms of the forms (R1-R4). Note that (R3) is actually a special case of (R4).

A DHL (or DHL(\circ)) ontology can be transformed to a datalog program (see the corresponding rules in the right column of Table 1). In what follows, for an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$, we also use $P = \langle R, \mathbf{I} \rangle$ to represent the corresponding datalog program where R is the set of rules transformed from the axioms in \mathcal{T} and \mathcal{R} , \mathbf{I} is the set of facts that are directly copied from the assertions in \mathcal{A} . Further, we use $R_1 \sqsubseteq_* R_2$ to denote the smallest transitive reflexive relation between roles such that $R_1 \sqsubseteq R_2 \in \mathcal{R}$ implies $R_1 \sqsubseteq_* R_2$ and $R_1^- \sqsubseteq_* R_2^-$. In this paper, we also use the notion of *simple role*, which is initially proposed to restrict the usage of highly expressive ontology languages [10]. Specifically, a role $S \in \mathbf{R}$ is *simple* if, (1) it has no subrole (including S) occurring on the right-hand side of axioms of the forms (R3) and (R4); (2) S^- is simple.

DHL is related with other ontology languages. First, DHL is essentially a fragment of the description logic Horn-*SHOIQ* with disallowing *nominal*, *number restriction* and right-hand *existential restriction* ($A \sqsubseteq \exists R.B$). Second, the expressivity of DHL covers that of RDFS to some extent [6]. Reasoning with RDFS ontologies is NP-complete [29] and, thus, is not parallelly tractable. However, by applying some simplifications and restrictions, RDFS statements can be expressed in DHL axioms [6].

Ontology Materialization. Based on the above representations, ontology materialization corresponds to the evaluation of datalog programs. Specifically, given a datalog program $\langle R, \mathbf{I} \rangle$, let $T_R(\mathbf{I}) = \{H\theta \mid \forall B_1, \dots, B_n \rightarrow H \in R, B_i\theta \in \mathbf{I} (1 \leq i \leq n)\}$, where θ is some substitution; further let $T_R^0(\mathbf{I}) = \mathbf{I}$ and $T_R^i(\mathbf{I}) = T_R^{i-1}(\mathbf{I}) \cup T_R(T_R^{i-1}(\mathbf{I}))$ for each $i > 0$. The smallest integer n such that $T_R^n(\mathbf{I}) = T_R^{n+1}(\mathbf{I})$ is called *stage*, and *materialization* refers to the computation of $T_R^n(\mathbf{I})$ with respect to R and \mathbf{I} . $T_R^n(\mathbf{I})$ is also called the *fixpoint* and denoted by $T_R^\omega(\mathbf{I})$. In this paper, we consider the data complexity of materialization, i.e., we *assume that the rule set R is fixed*.

NC. The parallel complexity class NC, known as Nick's Class [5], is studied by theorists as a parallel complexity class where each decision problem can be efficiently solved in parallel. Specifically, a decision problem can be solved in parallel. Specifically, a decision problem can be solved in parallel poly-logarithmic time on a parallel machine with a polynomial number of processors. We also say that an NC problem can be solved in *parallel poly-logarithmic time*. Although the NC complexity is a theoretical analysis tool, it has been shown that many NC problems can be solved efficiently in practice [5].

From the perspective of implementations, the NC problems are also highly parallel feasible for other parallel models like BSP [31] and MapReduce [11]. The NC complexity is originally defined as a class of decision problems. Since we study the problem of materialization, we do not require in this work that a problem should be a decision problem in NC. In addition, since many parallel reasoning systems (see related work in Section 8) are implemented on shared-memory platforms, we study all the issues in this work by assuming that the running machines are in shared-memory configurations.

3 Parallelly Tractable Class

Parallelly Tractable Class. Our target is to find for which kinds of ontologies (not ontology languages) materialization is parallelly tractable. Since we assume that for any datalog program $\langle R, \mathbf{I} \rangle$ the rule set R is fixed, the materialization problem is thus in data complexity PTime-complete, which is considered to be inherently sequential in the worst case [5]. In other words, the materialization problem on general datalog programs cannot be solved in parallel poly-logarithmic time unless P=NC. Thus, we say that materializa-

⁵ In datalog rules, a comma represents a Boolean conjunction ' \wedge '.

tion on a class of datalog programs is parallelly tractable if there exists an algorithm that handles this class of datalog programs and runs in parallel poly-logarithmic time (this algorithm is also called an NC algorithm). Formally, we give the following definition to identify such a class of datalog programs.

Definition 1. (Parallelly Tractable Class) Given a class \mathcal{D} of datalog programs, we say that \mathcal{D} is a parallelly tractable datalog program (PTD) class if there exists an NC algorithm that performs materialization for each datalog program in \mathcal{D} . The corresponding class of ontologies of \mathcal{D} is called a parallelly tractable ontology (PTO) class.

According to the above definition, if we find an NC algorithm A for datalog materialization, then we can identify a PTD class \mathcal{D}_A , which is the class of all datalog programs that can be handled by A . However, current materialization algorithms are not NC algorithms, since their computational complexity is PTime-complete. Thus we give our NC algorithms. In the following, we first give a parallel materialization algorithm that works for general datalog programs. We then restrict this algorithm to an NC version and identify the target PTD class.

Materialization Graph. In order to give a parallel materialization algorithm, we introduce the notion of *materialization graph*. It makes the analysis of the given algorithm convenient.

Definition 2. (Materialization Graph) A materialization graph, with respect to a datalog program $P = \langle R, \mathbf{I} \rangle$, is a directed acyclic graph denoted by $\mathcal{G} = \langle V, E \rangle$ where,

- V is the node set and $V \subseteq T_R^\omega(\mathbf{I})$;
- E is the edge set and $E \subseteq T_R^\omega(\mathbf{I}) \times T_R^\omega(\mathbf{I})$;

\mathcal{G} satisfies the following condition:

- $\forall H, B_1, \dots, B_n \in V$ such that $e(B_1, H), \dots, e(B_n, H) \in E$ and B_1, \dots, B_n are all the parents of H , we have that $B_1, \dots, B_n \rightarrow H \in P^*$.

For some derived atom H , there may exist several rule instantiations where H occurs as a head atom. This also means that H can be derived in different ways. The condition in the definition above results in only one way of deriving H being described by a materialization graph. Suppose \mathcal{G} is a materialization graph, the nodes whose in-degree is 0 are the original facts in \mathbf{I} . We call such a node an *explicit node*. We call the other nodes in \mathcal{G} the *implicit nodes*. We say that a node v is a *single-way derivable* (SWD) node if v has at most one implicit parent node; nodes with more than one implicit parent nodes are called *multi-way derivable* (MWD) nodes. The size of \mathcal{G} , denoted by $|\mathcal{G}|$, is the number of nodes in \mathcal{G} . The depth of \mathcal{G} , denoted by $\text{depth}(\mathcal{G})$, is the maximal length of a path in \mathcal{G} . We next give an example of a materialization graph.

Example 1. Consider a DHL(\circ) ontology $\mathcal{O}_{e_{x_1}}$ where the TBox is $\{\exists R.A \sqsubseteq A\}$, the RBox is $\{S \circ R \sqsubseteq R\}$ and the ABox is $\{A(b), R(a_1, b), S(a_i, a_{i-1})\}$ for $2 \leq i \leq k$ and k is an integer greater than 2. The corresponding datalog program of this ontology is $P_{e_{x_1}} = \langle R, \mathbf{I} \rangle$ where \mathbf{I} contains all the assertions in the ABox and R contains the two rules ' $R(x, y), A(y) \rightarrow A(x)$ ' and ' $S(x, y), R(y, z) \rightarrow R(x, z)$ '. The graph in Figure 1 is a materialization graph with respect to $P_{e_{x_1}}$, denoted by $\mathcal{G}_{e_{x_1}}$. The explicit nodes whose in-degree is 0 are the original facts in \mathbf{I} . Each of the implicit nodes corresponds to a ground instantiation of some rule.

For example, the node $A(a_k)$ corresponds to the ground rule instantiation ' $R(a_k, b), A(b) \rightarrow A(a_k)$ '. The size of this materialization graph is the number of nodes, that is $3k$. The depth of $\mathcal{G}_{e_{x_1}}$ is k .

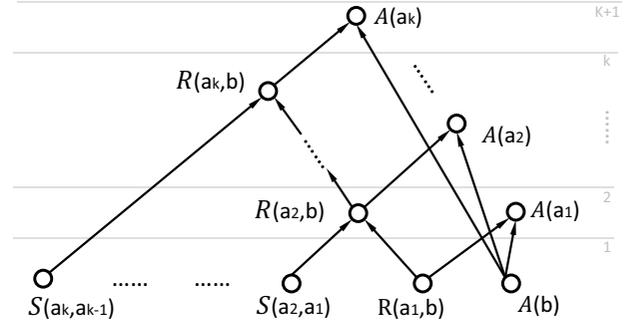


Figure 1. An example of a materialization graph.

We say that a materialization graph \mathcal{G} is a *complete materialization graph* when \mathcal{G} contains all ground atoms in $T_R^\omega(\mathbf{I})$. The set of nodes in a complete materialization graph is actually the result of materialization. Thus, the procedure of materialization can be transformed to the construction of a complete materialization graph. We pay our attention to complete materialization graphs and do not distinguish it to the notion 'materialization graph'. It should also be noted that there may exist several materialization graphs for a datalog program.

A Parallel Algorithm. In this part, we propose a parallel algorithm (Algorithm 1) that constructs a materialization graph for a given datalog program.

Algorithm 1. Given a datalog program $P = \langle R, \mathbf{I} \rangle$, the algorithm returns a materialization graph \mathcal{G} of P . Recall that P^* denotes the ground instantiation of P , which consists of all possible ground instantiations of rules in R . Suppose we have $|P^*|$ processors, and each rule instantiation in P^* is assigned to one processor.⁶ Initially \mathcal{G} is empty. The following three steps are then performed:

- (Step 1) Add all facts in \mathbf{I} to \mathcal{G} .
- (Step 2) For each rule instantiation $B_1, \dots, B_n \rightarrow H$, if the body atoms are all in \mathcal{G} while H is not in \mathcal{G} ,⁷ the corresponding processor adds H to \mathcal{G} and creates edges pointing from B_1, \dots, B_n to H .
- (Step 3) If no processor can add more nodes and edges to \mathcal{G} , terminate, otherwise iterate Step 2. \square

Example 2. We consider the datalog program $P_{e_{x_1}}$ in Example 1 again, and perform Algorithm 1 on it. Initially, all the facts $(A(b), R(a_1, b), S(a_2, a_1), \dots, S(a_k, a_{k-1}))$ are added to the result $\mathcal{G}_{e_{x_1}}$ (Step 1). Then in different iterations of Step 2, the remaining nodes are added to $\mathcal{G}_{e_{x_1}}$ by different processors. For example a processor p is allocated a rule instantiation ' $R(a_2, b), A(b) \rightarrow A(a_2)$ '. Then, processor p adds $A(a_2)$ to $\mathcal{G}_{e_{x_1}}$ after it checks that $A(b)$ and

⁶ This might not be practically feasible, but we focus on a theoretical analysis here. In practice, one can map several rule instantiations to a single processor.

⁷ Suppose that each processor can use $O(1)$ time units to access the state of ground atoms, i.e., whether this ground atom has been added to the materialization graph. This can be implemented by maintaining an index of polynomial size.

$R(a_2, b)$ are in $\mathcal{G}_{e_{x_1}}$. Algorithm 1 halts when $A(a_k)$ has been added to $\mathcal{G}_{e_{x_1}}$ (Step 3).

Lemma 1 shows the correctness of Algorithm 1 and that, for any datalog program P , Algorithm 1 always constructs a materialization graph with the minimum depth among all the materialization graphs of P . The proofs of Lemma 1 and other lemmas and theorems can be found in the technical report.

Lemma 1. *Given a datalog program $P = \langle R, \mathbf{I} \rangle$, we have*

1. Algorithm 1 halts and returns a materialization graph \mathcal{G} of P ;
2. \mathcal{G} has the minimum depth among all the materialization graphs of P .

Proof sketch. This lemma can be proved by performing an induction on $T_R^\omega(\mathbf{I})$. The stage (see the related contents in Section 2) of P is the lower-bound of the depth of the materialization graphs. Based on the previous induction, one can further check that, for the materialization graph \mathcal{G} constructed by Algorithm 1, its depth equals the depth of the stage. \square

We now discuss how Algorithm 1 can be restricted to an NC version. (I) Since Algorithm 1 does not introduce new constants and each predicate has a constant arity, one can check that $|P^*|$ is polynomial in the size of P . This also means that the number of processors is polynomially bounded. (II) The computing time of Step 1 and Step 3 occupies constant time units because of parallelism. (III) The main computation part in Algorithm 1 is the iteration of Step 2. In each iteration of Step 2, all processors work independently from each other. Thus, in theory, Step 2 costs one time unit. The whole computing time turns out to be bounded by the number of iterations of Step 2. (IV) We use the symbol ψ to denote a poly-logarithmically bounded function. The input of ψ is the size of P and the output is a non-negative integer. Based on (I, II, III, IV), for any datalog program P , if we use $\psi(|P|)$ to bound the number of iterations of Step 2, then Algorithm 1 is an NC algorithm, denoted by A_1^ψ .

Based on A_1^ψ , we can identify a class of datalog programs $\mathcal{D}_{A_1^\psi}$ where all the datalog programs can be handled by A_1^ψ . It is obvious that $\mathcal{D}_{A_1^\psi}$ is a PTD class.

We further show that $\mathcal{D}_{A_1^\psi}$ can be captured in terms of materialization graph properties based on the following theorem.

Theorem 1. *For any datalog program P , $P \in \mathcal{D}_{A_1^\psi}$ iff P has a materialization graph whose depth is upper-bounded by $\psi(|P|)$.*

Proof sketch. We can first prove that the number of iterations of Step 2 is actually the depth of the constructed materialization graph. This theorem then follows by considering Lemma 1. \square

The algorithm A_1^ψ is restricted in the sense that it cannot even work on the rather simple datalog program $P_{e_{x_1}}$ in Example 1. The graph $\mathcal{G}_{e_{x_1}}$ in Figure 1 is the unique materialization graph of $P_{e_{x_1}}$. One can also check that $\text{depth}(\mathcal{G}_{e_{x_1}}) = k$. This means that the depth of $\mathcal{G}_{e_{x_1}}$ is linearly bounded by k . On the other hand, the size of $P_{e_{x_1}}$ is a polynomial of the integer k . Thus, for any ψ that is poly-logarithmically bounded, we can always find a k large enough such that A_1^ψ terminates without constructing a materialization graph of $P_{e_{x_1}}$. However there indeed exists an NC algorithm that can handle $P_{e_{x_1}}$. We discuss this in the next section.

4 Two Optimized NC Algorithms

In this section, we discuss how to optimize Algorithm 1 such that $P_{e_{x_1}}$ can be handled. Based on the optimized variants of Algorithm 1, we can identify other PTD classes.

An Optimization Strategy. We discuss our optimization based on a general case in Example 3. We find that, in this kind of case, the construction of a materialization graph can be accelerated.

Example 3. *Consider a snapshot of Algorithm 1 in Figure 2. A materialization graph \mathcal{G} is being constructed for some datalog program $\langle R, \mathbf{I} \rangle$. The bottom nodes in the dashed box are the original facts in \mathbf{I} . In this snapshot, v_0 has been newly added to \mathcal{G} in the l^{th} ($l \geq 1$) iteration. Each of the nodes v_i ($1 \leq i \leq k$) is an SWD (single-way derivable) node. The node v' is an MWD (multi-way derivable) node. All of the nodes v_i ($1 \leq i \leq k$) and v' would be added to \mathcal{G} afterwards.*

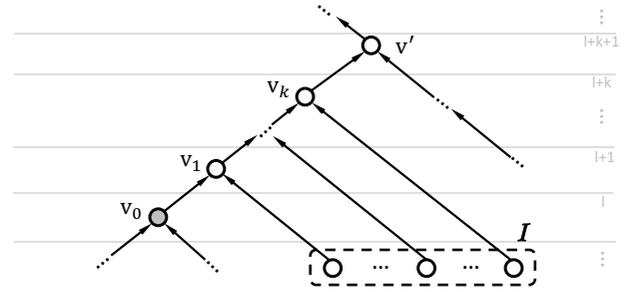


Figure 2. A partial materialization graph.

In Example 3, v_k would be added to \mathcal{G} after at least k iterations by performing Algorithm 1. Observe that v_k is reachable from v_0 through the path (v_0, v_1, \dots, v_k) . On the one hand, each node v_i ($1 \leq i \leq k$) can be added to \mathcal{G} whenever its parent v_{i-1} is in \mathcal{G} , since v_i is an SWD node, i.e., v_{i-1} is the unique implicit parent node of v_i . Since v_0 has been added to \mathcal{G} , one can add all the nodes v_i ($1 \leq i \leq k$) to \mathcal{G} right after v_0 . Based on this observation, we optimize Algorithm 1 using the following strategy:

(Strategy) *In every iteration of Step 2, for each SWD node v , we add v to \mathcal{G} immediately if v is reachable from some node that has been in \mathcal{G} through a path containing only SWD nodes.*

For an SWD node v in some materialization graph \mathcal{G} , we say that a path τ is a derivable path of v if τ starts from some node that has been in \mathcal{G} and ends in v and only contains SWD nodes. To describe the reachability between two nodes, we use a binary transitive relation $\text{rch} \subseteq T_R^\omega(\mathbf{I}) \times T_R^\omega(\mathbf{I})$, e.g., $\text{rch}(v_1, v_2)$ means that v_2 is reachable from v_1 . In each iteration of Step 2, we compute a rch relation (denoted by S_{rch}) by performing the following process:

(†) *For each rule instantiation of the form $B_1, \dots, B_i, \dots, B_n \rightarrow H$ where H is not in \mathcal{G} :*

1. if the body atoms B_1, \dots, B_n are all in \mathcal{G} , add $\text{rch}(B_1, H), \dots, \text{rch}(B_n, H)$ to S_{rch} ;
2. if B_i is the unique implicit node in the body and not yet in \mathcal{G} , add $\text{rch}(B_i, H)$ to S_{rch} . \square

We then compute the transitive closure of rch with respect to S_{rch} . From the transitive closure, we can identify such SWD nodes that

can be added to \mathcal{G} in advance. The following algorithm applies this optimization strategy.

Algorithm 2. The algorithm requires two inputs: a datalog program $P = \langle R, \mathbf{I} \rangle$ and a (partial) materialization graph \mathcal{G} that is constructed from P . The following steps are performed:

- (i) Compute a `rch` relation S_{rch} by following the above process (see (†)).
- (ii) Compute the transitive closure S_{rch}^* of S_{rch} .
- (iii) Update \mathcal{G} as follows: for any `rch`(B_i, H) $\in S_{rch}$ that corresponds to ' $B_1, \dots, B_i, \dots, B_n \rightarrow H$ ' such that `rch`(B', H), `rch`(B'', B_i) $\in S_{rch}^*$ where B', B'' are in \mathcal{G} ; If H is not in \mathcal{G} or H is in \mathcal{G} but has no parent pointing to it, add H and B_i (if B_i is not in \mathcal{G}) to \mathcal{G} , and create the edges $e(B_1, H), \dots, e(B_n, H)$ in \mathcal{G} . Do nothing for other statements `rch`(B_j, H) $\in S_{rch}$. \square

It is well known that there is an NC algorithm for computing the transitive closure [2]. Based on this result and Algorithm 2, we propose a variant of Algorithm 1:

Algorithm 3. Given a datalog program $P = \langle R, \mathbf{I} \rangle$, the algorithm returns a materialization graph \mathcal{G} of P . Initially \mathcal{G} is empty. The following steps are then performed:

- (Step 1) Add all facts in \mathbf{I} to \mathcal{G} .
- (Step 2) Compute S_{rch} by performing (i) in Algorithm 2; use an NC algorithm to compute the transitive closure S_{rch}^* (see (ii) in Algorithm 2); update \mathcal{G} by performing (iii) in Algorithm 2.
- (Step 3) If no node has been added to \mathcal{G} (in Step 2), terminate, otherwise iterate Step 2. \square

The following lemma shows the correctness of Algorithm 3.

Lemma 2. Given a datalog program $P = \langle R, \mathbf{I} \rangle$, Algorithm 3 halts and outputs a materialization graph \mathcal{G} of P .

Proof sketch. This lemma is proved in two stages: (1) the graph \mathcal{G} returned by Algorithm 3 is a materialization graph; (2) \mathcal{G} is a complete materialization graph. We prove (1) by an induction on the iterations of Step 2 in Algorithm 3. To prove (2), we use the same method as in the proof for Lemma 1 to show that all atoms in $T_R^\omega(\mathbf{I})$ have to be added to \mathcal{G} . \square

Example 4. We perform Algorithm 3 on the datalog program P_{ex_1} in Example 1. Initially, $R(a_1, b)$ is in the materialization graph \mathcal{G}_{ex_1} . In the first iteration of Step 2, all the rule instantiations are in two kinds of forms: ' $R(a_i, b), A(b) \rightarrow A(a_i)$ ' and ' $S(a_i, a_{i-1}), R(a_{i-1}, b) \rightarrow R(a_i, b)$ ' ($2 \leq i \leq k$). S_{rch} is the set $\{\text{rch}(R(a_{i-1}, b), R(a_i, b)) \mid 2 \leq i \leq k\} \cup \{\text{rch}(R(a_i, b), A(a_i)) \mid 1 \leq i \leq k\}$. In the transitive closure of S_{rch} , one can check that $\text{rch}(R(a_1, b), R(a_i, b))$, $\text{rch}(R(a_1, b), A(a_i)) \in S_{rch}^*$ ($2 \leq i \leq k$). Thus, $R(a_i, b)$ and $A(a_i)$ ($2 \leq i \leq k$) can all be added to \mathcal{G}_{ex_1} in the first iteration of Step 2.

We obtain an NC variant of Algorithm 3 analogously to the process for Algorithm 1. It can be checked that an iteration of Step 2 in Algorithm 3 costs poly-logarithmic time, since the main part is computing S_{rch}^* by an NC algorithm. Thus, if the number of iterations of Step 2 is upper-bounded by a poly-logarithmic function, Algorithm 3 is an NC algorithm. Analogously to A_1^ψ , we use A_3^ψ to denote an NC variant. Specifically, for any datalog program P , the number

of iterations of Step 2 in Algorithm 3 is bounded by $\psi(|P|)$, where ψ is a poly-logarithmically bounded function.

Based on A_3^ψ , we can identify a PTD class $\mathcal{D}_{A_3^\psi}$. The following theorem shows that $\mathcal{D}_{A_3^\psi}$ can also be captured by the properties of a materialization graph.

Theorem 2. For any datalog program P , $P \in \mathcal{D}_{A_3^\psi}$ iff P has a materialization graph \mathcal{G} such that the number of MWD nodes in each path of \mathcal{G} is upper-bounded by $\psi(|P|)$.

Proof sketch. (\Rightarrow) Suppose each materialization graph of P has a path where the number of MWD nodes is not upper-bounded by $\psi(|P|)$. This also means the number of iterations of Step 2 is not upper-bounded by $\psi(|P|)$ when constructing a materialization graph of P . Thus A_3^ψ cannot handle P .

(\Leftarrow) Suppose P has a materialization graph \mathcal{G} such that the number of MWD nodes in each path is upper-bounded by $\psi(|P|)$. It is not hard to check that A_3^ψ returns either of \mathcal{G} or the other materialization graph \mathcal{G}' that has fewer MWD nodes in each path than that of \mathcal{G} . \square

Further Optimizing Algorithm 3. Algorithm 3 can be further optimized. In step (i) of Algorithm 2, when computing S_{rch} , for the rule instantiations of the form ' $B_1, \dots, B_i, \dots, B_n \rightarrow H$ ', B_1, \dots, B_n (except B_i) are restricted to be explicit nodes (see (†)). We now extend S_{rch} by allowing that B_1, \dots, B_n (except B_i) could also be implicit nodes which have been added to the constructed materialization graph. Consider Example 3 again. If all the other implicit parents (except v_k) of v' have been added to the materialization graph, `rch`(v_k, v') can also be put in S_{rch} . This allows some MWD nodes being added to the materialization graph in advance. In this way, a derivable path represents such a path where the starting node is in the constructed materialization graph and each of the other nodes (whether or not it is an SWD node) has only one parent that is not in the constructed materialization graph. Algorithm 4 is given based on this optimization.

Algorithm 4. This algorithm is almost the same as Algorithm 3 except Step 2. Thus we only give the new step here.

(Step 2) For all rule instantiations of the form $B_1, \dots, B_i, \dots, B_n \rightarrow H$ where H is not yet in \mathcal{G} , compute S_{rch} as follows:

- (1) if all of B_1, \dots, B_n are in \mathcal{G} , add `rch`(B_1, H), ..., `rch`(B_n, H) to S_{rch} ;
- (2) if B_1, \dots, B_n (except B_i) are already in \mathcal{G} , put `rch`(B_i, H) in S_{rch} .

Compute S_{rch}^* ; update \mathcal{G} based on S_{rch}^* . \square

It is easy to prove the correctness of Algorithm 4 by referring to Lemma 2. Similarly, we use A_4^ψ to denote the NC variant of Algorithm 4, and $\mathcal{D}_{A_4^\psi}$ is the corresponding PTD class. Further, we have the following corollary. This corollary also implies that Algorithm 4 performs better than Algorithm 1 and Algorithm 3 in terms of computing time.

Corollary 1. For any poly-logarithmically bounded function ψ , we have that $\mathcal{D}_{A_1^\psi} \subseteq \mathcal{D}_{A_3^\psi} \subseteq \mathcal{D}_{A_4^\psi}$.

Proof sketch. Suppose $P \in \mathcal{D}_{A_1^\psi}$. According to Theorem 1, the depth of the materialization graph \mathcal{G} constructed by A_1^ψ is upper-bounded by $\psi(|P|)$. It is obvious that the number of MWD nodes in each path of \mathcal{G} is also upper-bounded by $\psi(|P|)$. Similarly we can prove that $\mathcal{D}_{A_3^\psi} \subseteq \mathcal{D}_{A_4^\psi}$. \square

5 Parallely Tractable Materialization of DHL

In this section, we study whether A_4^ψ can handle DHL ontologies. Unfortunately there exist DHL ontologies such that A_4^ψ does not work. In the following, we first give such an ontology to illustrate the reason why A_4^ψ cannot work. Based on the analysis of this case, we propose to restrict the usage of DHL in order to achieve parallel tractability of materialization.

Path Twisting. We find that, an unlimited usage of axioms of the form $B_1 \sqcap B_2 \sqsubseteq A$ may make it impossible for Algorithm 4 to construct a materialization graph in a poly-logarithmical number of iterations of Step 2. We use the following example to illustrate it.

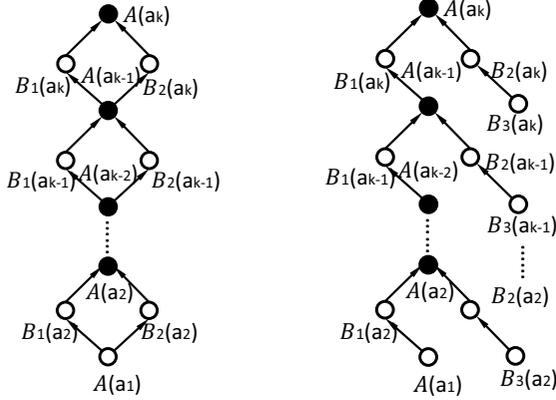


Figure 3. A partial graph of $\mathcal{G}_{e_{x_2}}$. **Figure 4.** A partial materialization graph $\mathcal{G}_{e_{x_3}}$.

Example 5. Given a DHL ontology $\mathcal{O}_{e_{x_2}}$ where its TBox contains three axioms: $B_1 \sqcap B_2 \sqsubseteq A$, $\exists S.A \sqsubseteq B_1$ and $\exists R.A \sqsubseteq B_2$; the ABox is $\{S(a_i, a_{i-1}), R(a_i, a_{i-1}), A(a_1)\}$ for $2 \leq i \leq k$ and k is an integer greater than 2. We denote the corresponding datalog program of $\mathcal{O}_{e_{x_2}}$ by $P_{e_{x_2}} = \langle R, \mathbf{I} \rangle$, where R contains three rules: ‘ $B_1(x), B_2(x) \rightarrow A(x)$ ’, ‘ $S(x, y), A(y) \rightarrow B_1(x)$ ’ and ‘ $R(x, y), A(y) \rightarrow B_2(x)$ ’. The materialization graph of $P_{e_{x_2}}$ constructed by Algorithm 4 is denoted by $\mathcal{G}_{e_{x_2}}$. Figure 3 shows a partial graph of $\mathcal{G}_{e_{x_2}}$. Note that all binary predicates in $P_{e_{x_2}}$ (S and R) are EDB predicates. We include only unary atoms in Figure 3 for clarity. Further, all MWD nodes are filled with black color.

One can check that $\mathcal{G}_{e_{x_2}}$ is the unique materialization graph of $P_{e_{x_2}}$. We focus on the partial materialization graph in Figure 3. Observe that there exists a path (e.g., $A(a_1), B_1(a_2), A(a_2), \dots, A(a_k)$) involving $k - 1$ MWD nodes. Obviously there is not a poly-logarithmical function ψ such that A_3^ψ handles $P_{e_{x_2}}$. Further, when performing Algorithm 4, it can be checked that all the $k - 1$ MWD nodes ($A(a_2), \dots, A(a_{k-1})$) have to be added to $\mathcal{G}_{e_{x_2}}$ in at least $k - 1$ iterations. Thus Algorithm 4 cannot handle $P_{e_{x_2}}$ in a poly-logarithmical number of iterations either. The intuitive reason is that, at least two paths exist starting from $A(a_1)$ to $A(a_k)$. These paths ‘twist’ mutually and share the same MWD nodes. It makes the optimization of acceleration used in Algorithm 3 and Algorithm 4 invalid. That is, for each node $A(a_i)$ ($2 \leq i \leq k$), until its parents ($B_1(a_i)$ and $B_2(a_i)$) are added to $\mathcal{G}_{e_{x_2}}$, there would not exist an available derivable path for $A(a_i)$. We use ‘path twisting’ to represent such cases.

Note that, applying the rules corresponding to either of (T2) or (R3) can also generate MWD nodes. However, we find that these

rules do not lead to the situations of ‘path twisting’. We show this in the proof of Theorem 3, which can be found in our technical report.

Simple Concept. In order to make Algorithm 4 terminate in a poly-logarithmical number of iterations, we consider restricting the usage of axioms of the form $B_1 \sqcap B_2 \sqsubseteq A$ to avoid ‘path twisting’. An intuitive idea is to ensure that *there is only one path between each two MWD nodes generated from the rules corresponding to (T1)*. We explain it using the following example where the ontology is modified from that in Example 5.

Example 6. Consider an ontology where the TBox contains three axioms: $B_1 \sqcap B_2 \sqsubseteq A$, $\exists S.A \sqsubseteq B_1$ and $B_3 \sqsubseteq B_2$; the ABox is $\{S(a_i, a_{i-1}), B_3(a_i), A(a_1)\}$ for $2 \leq i \leq k$ and k is an integer greater than 2. We denote the corresponding datalog program by $P_{e_{x_3}}$ where the rule set contains: ‘ $B_1(x), B_2(x) \rightarrow A(x)$ ’, ‘ $S(x, y), A(y) \rightarrow B_1(x)$ ’ and ‘ $B_3(x) \rightarrow B_2(x)$ ’. $P_{e_{x_3}}$ has a unique materialization graph denoted by $\mathcal{G}_{e_{x_3}}$. Figure 4 shows a partial graph of $\mathcal{G}_{e_{x_3}}$ where only unary atoms are involved, and all MWD nodes are filled with black color.

In the above example, for the axiom $B_1 \sqcap B_2 \sqsubseteq A$, all derived atoms of the form $B_2(x)$ are SWD nodes. This ensures that only one path exists between each two MWD nodes among $A(a_2), \dots, A(a_k)$. Further, when constructing $\mathcal{G}_{e_{x_3}}$, Algorithm 4 can terminate after two iterations of Step 2. Specifically, Algorithm 4 adds all SWD nodes ($B_3(a_i)$ and $B_2(a_i)$, $2 \leq i \leq k$) to $\mathcal{G}_{e_{x_3}}$ in the first iteration; after that, all the other nodes (including MWD nodes) are added to $\mathcal{G}_{e_{x_3}}$ in the second iteration (because each MWD node has a derivable path). Motivated by this example, we consider restricting the usage of the axioms $B_1 \sqcap B_2 \sqsubseteq A$ such that all atoms of the form $B_1(x)$ or $B_2(x)$ are SWD nodes. To this end, we first define *simple concepts* as follows:

Definition 3. Given an ontology $\mathcal{O} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$, a concept $A \in \text{CN}$ is simple, if (1) A does not occur on the right-hand side of some axiom; or (2) A satisfies the following conditions:

1. for each $B \sqsubseteq A \in \mathcal{T}$, B is simple;
2. for each $\exists R.B \sqsubseteq A \in \mathcal{T}$, B is simple;
3. there is no axiom of the form $B_1 \sqcap B_2 \sqsubseteq A$ in \mathcal{T} .

Based on simple concepts, we restrict DHL ontologies such that, in all axioms of the form $B_1 \sqcap B_2 \sqsubseteq A$, at least one concept of B_1 and B_2 should be a simple concept (we call it *simple-concept restriction*). Intuitively, for the restricted DHL ontologies, the situation of ‘path twisting’ would not happen. This is because, if in each axiom of the form $B_1 \sqcap B_2 \sqsubseteq A$, w.l.o.g., B_1 is a simple concept, then none of MWD ancestors of $B_1(x)$ for some x is generated from the rules corresponding to (T1).

Example 7. In the ontology of Example 5, all of A , B_1 and B_2 are non-simple concepts. In the ontology of Example 6, A and B_1 are non-simple concepts, while B_3 and B_2 are simple concepts. Further, it can be checked that, the ontology of Example 6 follows the simple-concept restriction and can be handled by A_4^ψ for some poly-logarithmical function ψ .

We define the following class of DHL ontologies based on the above restriction and give Theorem 3 to show that any DHL ontology that satisfies the simple-concept restriction can be handled by A_4^ψ for some poly-logarithmical function ψ .

Definition 4. Let \mathcal{D}_{dhl} be a class of datalog programs where each program is rewritten from a DHL ontology that follows the condition

that, for all axioms of the form $A_1 \sqcap A_2 \sqsubseteq B$, at least one concept of A_1 and A_2 should be a simple concept.

Theorem 3. *There exists a poly-logarithmically bounded function ψ s.t. $\mathcal{D}_{dhl} \subseteq \mathcal{D}_{A_4^\psi}$.*

Proof sketch. Suppose \mathcal{G} is a materialization graph of a datalog program P in \mathcal{D}_{dhl} . In \mathcal{G} , the nodes of the form $R(x, y)$ can only be derived by applying the rules corresponding to (R1-R3). All ground atoms derived from (R1) and (R2) correspond to the SWD nodes in \mathcal{G} . Thus, MWD nodes of the form $R(x, y)$ are only derived by applying (R3), which is to compute transitive closures. It can be checked that all binary atoms would be added to \mathcal{G} in poly-logarithmically many iterations of Step 2 by performing Algorithm 4. The unary atoms of the form $A(x)$ can also be added to \mathcal{G} in poly-logarithmically many iterations of Step 2 due to the simple-concept restriction. \square

6 Parallely Tractable Materialization of DHL(\circ)

In this section, we study parallely tractable materialization of DHL(\circ) ontologies. In addition to the rules in DHL, we also have to consider complex RIAs (R4). In the following, we first show that complex RIAs may also cause the situation of ‘path twisting’. Inspired by the simple-concept restriction, we then propose to restrict the usage of complex RIAs such that A_4^ψ works for some poly-logarithmical function ψ .

Restricting Usage of Complex RIAs. With complex RIAs, ‘path twisting’ may also happen when constructing a materialization graph by Algorithm 4. Consider the following example.

Example 8. *Given a DHL(\circ) ontology \mathcal{O}_{ex_4} where its TBox is empty; the RBox \mathcal{R} contains three axioms: $R_1 \circ R_2 \sqsubseteq R$, $R_3 \circ R \sqsubseteq R_1$ and $R \circ R_4 \sqsubseteq R_2$; the ABox \mathcal{A} is $\{R(a_1, a_1), R_3(a_i, a_{i-1}), R_4(a_{i-1}, a_i)\}$ for $2 \leq i \leq k$ and k is an integer greater than 2. The corresponding datalog program P_{ex_4} contains three rules: ‘ $R_1(x, y), R_2(y, z) \rightarrow R(x, z)$ ’, ‘ $R_3(x, y), R(y, z) \rightarrow R_1(x, z)$ ’ and ‘ $R(x, y), R_4(y, z) \rightarrow R_2(x, z)$ ’. The materialization graph of P_{ex_4} constructed by Algorithm 4 is denoted by \mathcal{G}_{ex_4} .*

One can check that the materialization graph \mathcal{G}_{ex_4} has the same shape as that of \mathcal{G}_{ex_2} in Figure 3. A twisted path exists in \mathcal{G}_{ex_4} involving $R(a_i, a_i)$ ($2 \leq i \leq k$) as MWD nodes. Further, all the roles R_1, R_2, R_3, R_4 and R in this example are non-transitive roles.

Inspired by what we do for axioms $B_1 \sqcap B_2 \sqsubseteq A$, we require that, for all axioms of the form $R_1 \circ R_2 \sqsubseteq R$, if R is not a transitive role, at least one of R_1 and R_2 is a simple role.⁸ Consider such an axiom $R_1 \circ R_2 \sqsubseteq R$ (denoted by α_1) where R is a transitive role. That is we also have $R \circ R \sqsubseteq R$ (denoted by α_2). By replacing R on the left-hand of α_2 using R_1 and R_2 , we can get a complex RIA in the form of $R_1 \circ R_2 \circ R_1 \circ R_2 \sqsubseteq R$ (denoted by α_3). If one of R_1 and R_2 is not a simple role, the corresponding rule of α_3 may also lead to ‘path twisting’.⁹ The reason can be explained as follows. Without loss of the generality, R_2 is a simple role while R_1 is not. For some atom $R(x, y)$, it may depend on two different MWD nodes of the predicate R_1 through the corresponding rule of α_3 . To tackle this issue, we require both of R_1 and R_2 in α_1 to be simple roles (we call

⁸ See the definition of a simple role in Section 2.

⁹ Obviously, applying the rules of α_1 and α_2 separately has the same effect to that of only applying the rule of α_3 .

the above restriction for transitive and non-transitive roles *simple-role restriction*). Combined with the simple-concept restriction, we define a class of DHL(\circ) ontologies as follows:

Definition 5. $\mathcal{D}_{dhl(\circ)}$ is a class of datalog programs where each program is rewritten from a DHL(\circ) ontology and the following conditions are satisfied:

1. for all axioms of the form $A_1 \sqcap A_2 \sqsubseteq B$, at least one concept of A_1 and A_2 should be a simple concept;
2. for all axioms of the form $R_1 \circ R_2 \sqsubseteq R$, if R is not a transitive role, at least one of R_1 and R_2 is a simple role; otherwise, both of R_1 and R_2 are simple roles.

Example 9. *For the ontology \mathcal{O}_{ex_4} in Example 8, all of the roles R_1, R_2 and R are non-simple roles. Thus, \mathcal{O}_{ex_4} does not follow the simple-role restriction because of $R_1 \circ R_2 \sqsubseteq R$. Consider the ontology \mathcal{O}_{ex_1} in Example 1 again. The role R is a non-simple role, while S is a simple role. Thus \mathcal{O}_{ex_1} follows the simple-role restriction. All the implicit nodes in \mathcal{G}_{ex_1} are SWD nodes. Thus, ‘path twisting’ cannot happen when materializing \mathcal{O}_{ex_1} by Algorithm 4.*

We further give Theorem 4 to show that A_4^ψ can handle all the datalog programs in $\mathcal{D}_{dhl(\circ)}$ for some poly-logarithmical function ψ .

Theorem 4. *There exists a poly-logarithmically bounded function ψ s.t. $\mathcal{D}_{dhl(\circ)} \subseteq \mathcal{D}_{A_4^\psi}$.*

Proof sketch. The proof idea of this theorem is similar to that of Theorem 3. Specifically, we can separate the materialization of DHL(\circ) ontologies into two parts: in the first part (Part 1), all the rules of the forms (R1-R4) are exhaustively applied; in the second part (Part 2), the rules of the forms (T1) and (T2) are then applied while the results of Part 1 serve as facts. In Part 1, since the rules of the form (R4) follow the simple-role restriction, it can be checked that all binary atoms would be added to the target materialization graph in a poly-logarithmical number of iterations of Step 2 by performing Algorithm 4. Part 2 can also be handled by A_4^ψ due to the simple-concept restriction. \square

7 Practical Usability of the Theoretical Results

In this section, we analyze different kinds of datasets including benchmarks, real-world ontologies and datasets that can be expressed in ontology languages. Based on the analysis of these datasets, we find that, ignoring imports, many of them belong to \mathcal{D}_{dhl} or $\mathcal{D}_{dhl(\circ)}$.

Benchmarks. In the Semantic Web community, many benchmarks are proposed to facilitate the evaluation of ontology-based systems in a standard and systematic way. We investigate several popular benchmarks using our results and find that the ontologies used in some benchmarks have simple structured TBoxes that can be expressed in RDFS and belong to \mathcal{D}_{dhl} . These benchmarks include SIB¹⁰ (*Social Network Intelligence BenchMark*), BSBM¹¹ (*Berlin SPARQL Benchmark*) and LODIB¹² (*Linked Open Data Integration Benchmark*). The ontology used in IIMB¹³ (*The ISLab Instance Matching Benchmark*) follows the simple-concept restriction.

In the latest version of LUBM¹⁴ (*The Lehigh University Benchmark*), there are 48 classes and 32 properties. Statements about properties, such as inverse property statements, can be rewritten into datalog rules allowed in \mathcal{D}_{dhl} . Most of the statements about classes can

¹⁰ https://www.w3.org/wiki/Social_Network_Intelligence_BenchMark

¹¹ <http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/>

¹² <http://wifo5-03.informatik.uni-mannheim.de/bizer/lodib/>

¹³ <http://islab.di.unimi.it/iimb/>

¹⁴ <http://swat.cse.lehigh.edu/projects/lubm/>

be rewritten into datalog rules that are allowed in \mathcal{D}_{dhl} . Five axioms have, however, the form $A \sqsubseteq \exists R.B$, which requires existentially quantified variables in the rule head when rewriting the axiom into a logic rule:

$$A(x) \rightarrow \exists y(R(x, y) \wedge B(y)) \quad (1)$$

Rule (1) introduces new anonymous constants. This kind of rule is not considered when using OWL RL reasoners to handle LUBM [30, 32]. On the other hand, in some cases, this kind of rule can also be eliminated when taking a rewriting approach [4]. In summary, if rules such as (1) are not considered, the materialization of a LUBM dataset can be handled by algorithm A_4^ψ .

YAGO. The knowledge base YAGO¹⁵ is constructed from Wikipedia and WordNet and the latest version YAGO3 [17] has more than 10 million entities (e.g., persons, organizations, cities, etc.) and contains more than 120 million facts about these entities. In order to balance the expressiveness and computing efficiency, a YAGO-style language, called YAGO *model*, is proposed based on a slight extension of RDFS [27]. In addition to the expressiveness of RDFS, YAGO *model* also allows stating the *transitivity* and *acyclicity* of a property. Making full use of RDFS features cannot lead to parallel tractability. However, in [27], a group of materialization rules is specified, which is more efficient. All of these rules are allowed in \mathcal{D}_{dhl} . Thus, we have that a well-constructed YAGO dataset belongs to \mathcal{D}_{dhl} .

Real Ontologies. We investigated 151 ontologies that cover many domains like biomedicine, geography, etc. These ontologies are collected from the Protege ontology library,¹⁶ Swoogle¹⁷ and Oxford ontology lib.¹⁸ All ontologies are available online.¹⁹ Among these ontologies, 111 of them belong to \mathcal{D}_{dhl} or $\mathcal{D}_{dhl(\circ)}$, and 21 DHL ontologies contain conjunctions and follow the simple-concept restriction. The remaining ontologies have simple TBoxes, i.e., no conjunction ($A_1 \sqcap A_2$) appears in these ontologies. We also find two DHL(\circ) ontologies that follow the simple-role restriction.

For ontologies that satisfy the simple-concept and simple-role restrictions, users have a guarantee of parallel tractability. On the other hand, developers and users can also refer to \mathcal{D}_{dhl} and $\mathcal{D}_{dhl(\circ)}$ when building their own ontologies.

8 Discussions and Related Work

Parallel reasoning with ontology languages has been extensively studied in the past decade.

The parallel reasoner RDFox [19] handles reasoning on datalog rewritable ontology languages. Algorithm 1 proposed in Section 3 is similar to the main algorithm for RDFox (see [19], Sections 3 and 4). A thread in RDFox handles several rule instantiations with respect to a fact. Such a thread corresponds to a group of processors in Algorithm 1 that is assigned with the rule instantiations handled by the thread. Thus the materialization of the datalog program in Example 1 is serial on RDFox. We use Algorithm 3 and Algorithm 4 to show that the datalog program in Example 1 is also parallelly tractable, i.e., belonging to $\mathcal{D}_{A_3^\psi}$ and $\mathcal{D}_{A_4^\psi}$.

The authors of [3] propose a parallel approach for RDFS encoding and reasoning and SPARQL query answering on the Cray XMT supercomputer. In [8] the authors study stream reasoning over RDF

data and SPARQL query answering using Yahoo S4. The authors in [7] report their work on RDFS reasoning on massively parallel GPU hardware. In [22], the RETE algorithm is used to improve RDFS reasoning. The authors of [26] propose a more efficient storage technique and optimize the join operations in RDFS reasoning. The above works study parallel reasoning in RDFS or its fragment ρ df [20].

Distributed parallel platforms, like MapReduce or Peer-to-Peer networks, are also used for RDFS reasoning. The representative systems are WebPIE [30], Marvin [21] and SAOR [9]. Data partitioning strategies are also studied [24, 32]. To study parallel tractability on distributed platforms, we have to discuss other issues, e.g., *network structures* and *communications*. This is not considered in this work. Parallel reasoning is also implemented for other OWL fragments, e.g., OWL RL [14], OWL EL [13], OWL QL [15], and even highly expressive languages [25, 16, 23, 33]. Parallelism can also improve the performance of reasoning in non-monotonic logics [28]. Unlike the above work, the aim of our work is not to devise an efficient parallel reasoning algorithm, but to identify ontologies that are tractable for parallel materialization.

9 Conclusions and Future Work

In this paper, we studied the problem of finding ontologies such that the materialization over them is parallelly tractable. To this end, we proposed several NC algorithms that perform materialization on datalog rewritable ontology languages. Based on these algorithms, we identified the corresponding *parallelly tractable datalog program* (PTD) classes such that materialization on the datalog programs in these classes is in the complexity class NC. We further studied two specific ontology languages, DHL and its extension DHL(\circ), and proposed two restrictions such that materialization is parallelly tractable. To verify the usefulness of our theoretical results, we analyzed different kinds of datasets, including well-known benchmarks, real-world ontologies and a famous dataset YAGO. Our analysis shows that YAGO and many real ontologies belong to the parallelly tractable class \mathcal{D}_{dhl} or $\mathcal{D}_{dhl(\circ)}$. On the other hand, developers and users can also refer to \mathcal{D}_{dhl} and $\mathcal{D}_{dhl(\circ)}$ to create large-scale ontologies for which parallel tractability is theoretically guaranteed.

In our future work, we will study in detail how to further apply the theoretical results in practice. One idea is to study the impact of the simple-concept and simple-role restrictions by analyzing more real-world ontologies. We also want to study parallelly tractable materialization on distributed systems. This is more challenging since several factors like network structure and communication should be taken into account. Finally, we plan to investigate the problem of parallel tractability of other OWL fragments, e.g., OWL RL and OWL EL.

Acknowledgement

We would like to thank the reviewers for their comments, which helped improve this paper considerably. Guilin Qi is supported by NSFC grant 61272378 and the 863 program under Grant 2015AA015406. Birte Glimm acknowledges the support of the Transregional Collaborative Research Centre SFB/TRR 62 ‘‘Companion-Technology for Cognitive Technical Systems’’ funded by the German Research Foundation (DFG).

¹⁵ <http://www.mpi-inf.mpg.de/home/>

¹⁶ http://protegewiki.stanford.edu/wiki/Protege_Ontology_Library

¹⁷ <http://swoogle.umbc.edu/>

¹⁸ <http://www.cs.ox.ac.uk/isg/ontologies/lib/>

¹⁹ <https://github.com/quanzz/ECAI2016>

REFERENCES

- [1] Serge Abiteboul, Richard Hull, and Victor Vianu, *Foundations of Databases*, Addison-Wesley, 1995.
- [2] Eric Allender, 'Reachability problems: An update', in *Proc. of CiE*, pp. 25–27, (2007).
- [3] Eric L. Goodman, Edward Jimenez, David Mizell, Sinan Al-Saffar, Bob Adolf, and David J. Haglin, 'High-performance computing applied to semantic databases', in *Proc. of ESWC*, pp. 31–45, (2011).
- [4] Bernardo Cuenca Grau, Ian Horrocks, Markus Krötzsch, Clemens Kupke, Despoina Magka, Boris Motik, and Zhe Wang, 'Acyclicity notions for existential rules and their application to query answering in ontologies', *J. Artif. Intell.*, **47**, 741–808, (2013).
- [5] Raymond Greenlaw, H. James Hoover, and Walter L. Ruzzo, *Limits to Parallel Computation: P-Completeness Theory*, Oxford University Press, New York, 1995.
- [6] Benjamin N. Grosz, Ian Horrocks, Raphael Volz, and Stefan Decker, 'Description logic programs: combining logic programs with description logic', in *Proc. of WWW*, pp. 48–57, (2003).
- [7] Norman Heino and Jeff Z. Pan, 'RDFS reasoning on massively parallel hardware', in *Proc. of ISWC*, pp. 133–148, (2012).
- [8] Jesper Hoeksema and Spyros Kotoulas, 'High-performance Distributed Stream Reasoning using S4', in *Proc. of OOR*, (2011).
- [9] Aidan Hogan, Andreas Harth, and Axel Polleres, 'Scalable authoritative OWL reasoning for the web', *Int. J. Semantic Web Inf. Syst.*, **5**(2), 49–90, (2009).
- [10] Ian Horrocks and Ulrike Sattler, 'Decidability of SHIQ with complex role inclusion axioms', *J. Artif. Intell.*, **160**(1-2), 79–104, (2004).
- [11] Howard J. Karloff, Siddharth Suri, and Sergei Vassilvitskii, 'A model of computation for mapreduce', in *Proc. of SODA*, pp. 938–948, (2010).
- [12] Yevgeny Kazakov, 'Consequence-driven reasoning for horn SHIQ ontologies', in *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, pp. 2040–2045, (2009).
- [13] Yevgeny Kazakov, Markus Krötzsch, and Frantisek Simancik, 'The incredible ELK - from polynomial procedures to efficient reasoning with \mathcal{EL} ontologies', *J. Autom. Reasoning*, 1–61, (2014).
- [14] Vladimir Kolovski, Zhe Wu, and George Eadon, 'Optimizing enterprise-scale OWL 2 RL reasoning in a relational database system', in *Proc. of ISWC*, pp. 436–452, (2010).
- [15] Domenico Lembo, Valerio Santarelli, and Domenico Fabio Savo, 'A graph-based approach for classifying OWL 2 QL ontologies', in *Proc. of DL*, pp. 747–759, (2013).
- [16] Thorsten Liebig and Felix Müller, 'Parallelizing tableaux-based description logic reasoning', in *Proc. of OTM Workshops*, pp. 1135–1144, (2007).
- [17] Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek, 'YAGO3: A knowledge base from multilingual wikipedias', in *Proc. of CIDR*, (2015).
- [18] Robert Meusel, Christian Bizer, and Heiko Paulheim, 'A web-scale study of the adoption and evolution of the schema.org vocabulary over time', in *Proc. of WIMS*, pp. 15:1–15:11, (2015).
- [19] Boris Motik, Yavor Nenov, Robert Piro, Ian Horrocks, and Dan Olteanu, 'Parallel materialisation of datalog programs in centralised, main-memory RDF systems', in *Proc. of AAAI*, pp. 129–137, (2014).
- [20] Sergio Muñoz, Jorge Pérez, and Claudio Gutierrez, 'Simple and efficient minimal RDFS', *J. Web Sem.*, **7**(3), 220–234, (2009).
- [21] Eyal Oren, Kotoulas Spyros, Anadiotis George, Siebes Ronny, ten Teije Annette, and van Harmelen Frank, 'Marvin: Distributed reasoning over large-scale Semantic Web data', *J. Web Sem.*, 305–316, (2009).
- [22] Martin Peters, Sabine Sachweh, and Albert Zündorf, 'Large scale rule-based reasoning using a laptop', in *Proc. of ESWC*, pp. 104–118, (2015).
- [23] Anne Schlicht and Heiner Stuckenschmidt, 'Distributed resolution for ALC', in *Proc. of DL*, pp. 326–341, (2008).
- [24] Ramakrishna Soma and Viktor K. Prasanna, 'A data partitioning approach for parallelizing rule based inferencing for materialized OWL knowledge bases', in *Proc. of ISCA*, pp. 19–25, (2008).
- [25] Andreas Steigmüller, Thorsten Liebig, and Birte Glimm, 'Konclude: System description', *J. Web Sem.*, **27**, 78–85, (2014).
- [26] Julien Subercaze, Christophe Gravier, Jules Chevalier, and Frédérique Laforest, 'Inferray: fast in-memory RDF inference', *J. PVLDB*, **9**(6), 468–479, (2016).
- [27] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum, 'YAGO: A large ontology from wikipedia and wordnet', *J. Web Sem.*, **6**(3), 203–217, (2008).
- [28] Ilias Tachmazidis, Grigoris Antoniou, Giorgos Flouris, Spyros Kotoulas, and Lee McCluskey, 'Large-scale Parallel Stratified Defeasible Reasoning', in *Proc. of ECAI*, pp. 738–743, (2012).
- [29] Herman J. ter Horst, 'Completeness, decidability and complexity of entailment for RDF schema and a semantic extension involving the OWL vocabulary', *J. Web Sem.*, **3**(2-3), 79–115, (2005).
- [30] Jacopo Urbani, Spyros Kotoulas, Jason Maassen, Frank van Harmelen, and Henri E. Bal, 'Webpie: A web-scale parallel inference engine using mapreduce', *J. Web Sem.*, **10**, 59–75, (2012).
- [31] Leslie G. Valiant, 'A bridging model for parallel computation', *Commun. ACM*, 103–111, (1990).
- [32] Jesse Weaver and James A. Hendler, 'Parallel materialization of the finite RDFS closure for hundreds of millions of triples', in *Proc. of ISWC*, pp. 682–697, (2009).
- [33] Kejia Wu and Volker Haarslev, 'A parallel reasoner for the description logic ALC', in *Proc. of DL*, pp. 675–690, (2012).

Student- t Process Regression with Dependent Student- t Noise

Qingtao Tang¹ and Yisen Wang and Shu-Tao Xia

Abstract. Gaussian Process Regression (GPR) is a powerful non-parametric method. However, GPR may perform poorly if the data are contaminated by outliers. To address the issue, we replace the Gaussian process with a Student- t process and introduce dependent Student- t noise in this paper, leading to a Student- t Process Regression with Dependent Student- t noise model (TPRD). Closed form expressions for the marginal likelihood and predictive distribution of TPRD are derived. Besides, TPRD gives a probabilistic interpretation to the Student- t Process Regression with the noise incorporated into its Kernel (TPRK), which is a common approach for the Student- t process regression. Moreover, we analyze the influence of different kernels. If the kernel meets a condition, called β -property here, the maximum marginal likelihood estimation of TPRD's hyperparameters is independent of the degrees of freedom ν of the Student- t process, which implies that GPR, TPRD and TPRK have exactly the same predictive mean. Empirically, the degrees of freedom ν could be regarded as a convergence accelerator, indicating that TPRD with a suitable ν performs faster than GPR. If the kernel does not have the β -property, TPRD has better performances than GPR, without additional computational cost. On benchmark datasets, the proposed results are verified.

1 INTRODUCTION

Gaussian processes are powerful Bayesian nonparametric methods with good interpretability and non-parametric flexibility. In addition, Gaussian processes have simple learning, exact inference and impressive empirical performances without manual parameter tuning [12].

In a regression problem, the basic model is $y = f(X) + \epsilon$, where y is the target vector, X is the feature matrix and ϵ is the noise. Gaussian Process Regression (GPR) assumes the latent function f is a Gaussian process and ϵ is independent and identically distributed (i.i.d.) Gaussian noise. Based on these assumptions, exact inference can be performed by the Bayes' theorem [12]. However, GPR performs poorly on data sets contaminated by outliers because of the thin-tailed property of Gaussian distribution. To address the issue, heavy-tailed distributions, e.g., the Student- t distribution, have been introduced into GPR. Generally speaking, there are two ways. The first way assumes that the noise is from an i.i.d Student- t distribution. Then a Gaussian process with the i.i.d Student- t noise is obtained. Exact inference, however, is analytically intractable. Then one has to turn to approximate inference methods, such as MCMC (Markov Chain Monte Carlo, [10]), variational approximation [6] and Laplace approximation [16]. However, these methods require additional computational cost.

The second way assumes that the latent function f is a Student- t process, which leads to the Student- t Process Regression model (TPR) [12, 14]. The problem of this way is that the sum of two independent Student- t distributions or the sum of a Student- t and a Gaussian distribution is analytically intractable. In other words, the Student- t process regression with independent Gaussian or Student- t noise is analytically intractable. Thus, Rasmussen and Williams [12] said "Allowing for independent noise contributions removes analytic tractability, which may reduce the usefulness of the t process". Later in [14, 15, 19], in order to increase the usefulness of TPR, the noise is incorporated to the kernel function, which leads to the Student- t process regression with the noise incorporated into its Kernel (TPRK). By this method, good empirical performances are achieved, however, probabilistic properties of the noise remain unknown.

In this paper, to obtain a model with robustness, reasonable computational cost and probabilistic interpretation, we propose a Student- t Process Regression with Dependent Student- t noise model (TPRD), which replaces the Gaussian process in GPR with a Student- t process and introduces dependent Student- t noise. The variance of the noise is dependent on how well the noise-free model fits the data. Owing to the novel noise, TPRD owns all the advantages of GPR, such as good interpretation, exact inference and simple hyperparameter learning. Besides, the marginal likelihood of TPRD is equivalent to that of TPRK, which indicates that TPRD gives a probabilistic interpretation to TPRK. Moreover, if the kernel has the β -property (defined later), we prove that the maximum marginal likelihood (ML) estimation of TPRD's hyperparameters is independent of the degrees of freedom ν , resulting in that TPRD, TPRK and GPR have the same predictive mean. And TPRD outperforms GPR without additional computational cost if the kernel does not have the β -property. Various experiments are conducted to evaluate the properties mentioned above.

In summary, the main contributions of this paper are as follows:

- We prove that GPR is a special case of TPRD. And closed form expressions for the marginal likelihood and predictive distribution of TPRD are derived.
- TPRD gives a probabilistic interpretation to the way of incorporating the noise into the kernel function, adopted by TPRK.
- If the kernel has the β -property, we prove that the ML estimation of TPRD's hyperparameters is independent of the degrees of freedom ν , and GPR, TPRD, TPRK have exactly the same predictive mean. But experiments show that TPRD with a suitable ν is faster than GPR.
- If the kernel does not have the β -property, empirically, TPRD obtains better performances at no additional computational cost over GPR.

¹ Tsinghua University, Beijing, China, email: tq15@mails.tsinghua.edu.cn

The rest of the paper is organized as follows: Section 2 describes GPR and proposes TPRD. Section 3 analyzes the theoretical properties of TPRD and TPRK. Section 4 presents the experimental results. Section 5 concludes the work.

2 TPR WITH DEPENDENT STUDENT-*T* NOISE

In this section, we will give a brief review to GPR and propose TPRD. Besides, we also discuss the relationships between TPRD, GPR and TPRK.

2.1 Review of GPR

In a regression problem, we have a training set \mathcal{D} of n instances, $\mathcal{D} = \{X, \mathbf{y}\}$, where $X = \{\mathbf{x}_i\}_{i=1}^n$ is the $n \times D$ design matrix with D being the dimension of attributes, and $\mathbf{y} = \{y_i\}_{i=1}^n$ denotes the output or target vector of dimension n . In GPR, the basic model is

$$y_i = f(\mathbf{x}_i) + \epsilon_i, \quad i = 1, 2, \dots, n, \quad (1)$$

where ϵ_i is the i.i.d Gaussian noise. The latent function f is given a Gaussian process prior. In practice, as n is finite, $\mathbf{f} = \{f(\mathbf{x}_i)\}_{i=1}^n$ has a multivariate Gaussian distribution as

$$p(\mathbf{f}|X, K_g) = \mathcal{N}(\mathbf{f}|\boldsymbol{\mu}_g, K_g), \quad (2)$$

where $\boldsymbol{\mu}_g$ is the mean. The subscript g indicates that the hyperparameters are of GPR. Usually, for notational simplicity, we assume $\boldsymbol{\mu}_g = \mathbf{0}$. And K_g is the covariance matrix. $(K_g)_{i,j} = \text{cov}(f(\mathbf{x}_i), f(\mathbf{x}_j)) = k(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\theta})$, where k is a kernel function, $\boldsymbol{\theta} = (\theta_1, \theta_2, \dots, \theta_l)$ is the parameters of the kernel and l is the number of parameters. As ϵ_i is i.i.d Gaussian, the likelihood is

$$p(\mathbf{y}|\mathbf{f}, \sigma_g) = \mathcal{N}(\mathbf{y}|\mathbf{f}, \sigma_g^2 I), \quad (3)$$

where σ_g^2 is the variance of the noise and I denotes the identity matrix. By the Bayes' theorem and integrating out \mathbf{f} , we can get the marginal likelihood

$$p(\mathbf{y}|X, \sigma_g, K_g) = \mathcal{N}(\mathbf{y}|\mathbf{0}, \Sigma_g), \quad (4)$$

where $\Sigma_g = K_g + \sigma_g^2 I$. Then, to learn the hyperparameters σ_g and $\boldsymbol{\theta}$, the maximum marginal likelihood can be used, which is equivalent to minimizing the negative logarithm marginal likelihood denoted by

$$-\ln p(\mathbf{y}|X, \sigma_g, K_g) = \frac{1}{2} \mathbf{y}^T \Sigma_g^{-1} \mathbf{y} + \frac{1}{2} \ln |\Sigma_g| + \frac{n}{2} \ln 2\pi. \quad (5)$$

The three terms of the negative marginal likelihood in Eq. (5) are explained in [12]: the only term involving the observed targets is the data-fit term $\frac{1}{2} \mathbf{y}^T \Sigma_g^{-1} \mathbf{y}$; $\ln |\Sigma_g|$ is the complexity penalty depending only on the kernel function and the inputs; and $\frac{n}{2} \ln 2\pi$ is a normalization constant.

After learning the hyperparameters σ_g and $\boldsymbol{\theta}$, for a known input $\mathbf{x}_* \in R^D$, the predictive distribution is given as follows [12]

$$p(y_*|\mathbf{y}) = \mathcal{N}(y_*|\mu_*, \sigma_*), \quad (6)$$

$$\mu_* = \mathbf{k}_*^T \Sigma_g^{-1} \mathbf{y}, \quad (7)$$

$$\sigma_*^2 = k(\mathbf{x}_*, \mathbf{x}_*; \boldsymbol{\theta}) - \mathbf{k}_*^T \Sigma_g^{-1} \mathbf{k}_*, \quad (8)$$

$$\mathbf{k}_* = \{k(\mathbf{x}_i, \mathbf{x}_*; \boldsymbol{\theta})\}_{i=1}^n. \quad (9)$$

Clearly, the predictive mean of GPR is a linear combination of y_i ($i = 1, 2, \dots, n$) and the predictive variance does not depend on \mathbf{y} .

2.2 Derivation of TPRD

The definition of a multivariate Student-*t* distribution is as follows.

Definition 1. An n -dimensional random vector $\mathbf{x} = (x_1, \dots, x_n)^T$ is said to have the n -variate Student-*t* distribution with degrees of freedom ν , mean vector $\boldsymbol{\mu}$, and correlation matrix R if its joint probability density function (PDF) is given by

$$St(\mathbf{x}|\nu, \boldsymbol{\mu}, R) = \frac{\Gamma[(\nu+n)/2]}{\Gamma(\nu/2)\nu^{n/2}\pi^{n/2}|R|^{1/2}} \cdot \left[1 + \frac{1}{\nu}(\mathbf{x} - \boldsymbol{\mu})^T R^{-1}(\mathbf{x} - \boldsymbol{\mu})\right]^{-(\nu+n)/2}. \quad (10)$$

We adopt the most common definition of Student-*t* distribution here, which could be found in [5, 8].

Now we introduce the Student-*t* process regression with dependent Student-*t* noise. In the regression problem Eq. (1), the latent function f is given a Student-*t* process prior, i.e., $\mathbf{f} = \{f(\mathbf{x}_i)\}_{i=1}^n$ has the PDF

$$p(\mathbf{f}|X, \boldsymbol{\theta}) = St(\mathbf{f}|\nu, \mathbf{0}, K_t) = \frac{\Gamma[(\nu+n)/2]}{\Gamma(\nu/2)\nu^{n/2}\pi^{n/2}|K_t|^{1/2}} \left[1 + \frac{1}{\nu}\mathbf{f}^T K_t^{-1}\mathbf{f}\right]^{-(\nu+n)/2}. \quad (11)$$

Just as in GPR,

$$(K_t)_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\theta}), \quad (12)$$

and we also assume the mean vector is $\mathbf{0}$ for simplicity.

Assume the noise ϵ is an n -dimensional Student-*t* distribution dependent on $p(\mathbf{f}|X, \boldsymbol{\theta})$ with the following form

$$p(\epsilon|\beta) = St\left(\epsilon|\nu+n, \mathbf{0}, \left(1 + \frac{1}{\nu}\mathbf{f}^T K_t^{-1}\mathbf{f}\right) \frac{1}{\beta} I\right). \quad (13)$$

For a given ν , $\frac{1}{\nu}\mathbf{f}^T K_t^{-1}\mathbf{f}$ is the data-fit term without considering noise by Eq. (1) and the explanation of the first term of Eq. (5). The variance of the noise depends on how well the noise-free model fits the data. To be specific, if the noise-free model fits the data well, the negative logarithm marginal likelihood is small, which implies the variance in the Eq. (13) is small. Otherwise, if the noise-free model does not fit the model well, the variance in the Eq. (13) is relatively large. And the degrees of freedom of ϵ is $n + \nu$, larger than the degrees of freedom of \mathbf{f} by n . As in practice, the number of instances n is relatively large, ϵ approximates to a multivariate Gaussian distribution with a diagonal covariance matrix, which implies that the noise ϵ_i ($i = 1, 2, \dots, n$) approximate to i.i.d Gaussian distributions. In summary, TPRD is in effect a Student-*t* process with approximate i.i.d Gaussian noise whose variance is adjusted to the data-fit term of noise-free model. Since the Student-*t* process is more robust than the Gaussian process, it is expected that TPRD performs better than GPR in some cases.

With the assumptions of Eq. (11) and Eq. (13), exact inference is achieved as follows,

$$p(\mathbf{y}|\mathbf{f}, \beta) = St\left(\mathbf{y}|\nu+n, \mathbf{f}, \left(1 + \frac{1}{\nu}\mathbf{f}^T K_t^{-1}\mathbf{f}\right) \frac{1}{\beta} I\right) = \frac{\Gamma[(\nu+2n)/2]\beta^{n/2}}{\Gamma[(\nu+n)/2](\nu+n)^{n/2}\pi^{n/2}\left(1 + \frac{1}{\nu}\mathbf{f}^T K_t^{-1}\mathbf{f}\right)^{n/2}} \cdot \left[1 + \frac{\beta}{(\nu+n)} \frac{(\mathbf{y} - \mathbf{f})^T (\mathbf{y} - \mathbf{f})}{1 + \frac{1}{\nu}\mathbf{f}^T K_t^{-1}\mathbf{f}}\right]^{-(\nu+2n)/2}. \quad (14)$$

Multiplying Eq. (11) by Eq. (14), the joint distribution of \mathbf{y} and \mathbf{f} is

$$\begin{aligned} p(\mathbf{y}, \mathbf{f} | X, \boldsymbol{\theta}, \beta) &\propto \left[1 + \frac{1}{\nu} \mathbf{f}^T K_t^{-1} \mathbf{f} \right. \\ &\quad \left. + \frac{\beta}{(\nu+n)} (\mathbf{y} - \mathbf{f})^T (\mathbf{y} - \mathbf{f}) \right]^{-(\nu+2n)/2} \\ &\propto \left[1 + \frac{\beta}{\nu+n} \mathbf{y}^T \left(I - \frac{\beta}{\nu+n} A^{-1} \right) \mathbf{y} \right. \\ &\quad \left. + (\mathbf{f} - \bar{\mathbf{f}})^T A (\mathbf{f} - \bar{\mathbf{f}}) \right]^{-(\nu+2n)/2}, \end{aligned} \quad (15)$$

where

$$\begin{aligned} A &= \frac{1}{\nu} K_t^{-1} + \frac{\beta}{\nu+n} I, \\ \bar{\mathbf{f}} &= \frac{\beta}{\nu+n} A^{-1} \mathbf{y}. \end{aligned} \quad (16)$$

By integrating out \mathbf{f} in Eq. (15), we get the marginal likelihood

$$p(\mathbf{y}) \propto \left[1 + \frac{\beta}{\nu+n} \mathbf{y}^T \left(I - \frac{\beta}{\nu+n} A^{-1} \right) \mathbf{y} \right]^{-(\nu+n)/2}, \quad (17)$$

$$p(\mathbf{y}) = \frac{\Gamma[(\nu+n)/2]}{\Gamma(\nu/2) \nu^{n/2} \pi^{n/2} |\Sigma_t|^{1/2}} \left[1 + \frac{1}{\nu} \mathbf{y}^T \Sigma_t^{-1} \mathbf{y} \right]^{-(\nu+n)/2}, \quad (18)$$

where Σ_t is the correlation matrix of \mathbf{y} and $\Sigma_t^{-1} = \frac{\nu\beta}{\nu+n} (I - \frac{\beta}{\nu+n} A^{-1})$. It's not difficult to show that $\Sigma_t = K_t + \frac{\nu+n}{\nu} \frac{1}{\beta} I$, i.e.,

$$\Sigma_t = K_t + \sigma_t^2 I, \quad \sigma_t^2 = \frac{\nu+n}{\nu} \cdot \frac{1}{\beta}. \quad (19)$$

The negative logarithm marginal likelihood of TPRD is

$$\begin{aligned} -\ln p(\mathbf{y}) &= \frac{\nu+n}{2} \ln \left[1 + \frac{1}{\nu} \mathbf{y}^T \Sigma_t^{-1} \mathbf{y} \right] + \frac{1}{2} \ln |\Sigma_t| \\ &\quad - \ln \left(\frac{\Gamma[(\nu+n)/2]}{\Gamma(\nu/2) \nu^{n/2} \pi^{n/2}} \right). \end{aligned} \quad (20)$$

Similar to the three terms of GPR's negative logarithm marginal likelihood in Eq. (5), the three terms in Eq. (20) has readily interpretable roles: the second term $\frac{1}{2} \ln |\Sigma_t|$ is the complexity penalty depending only on the kernel function and the inputs; the last term $\ln \left(\frac{\Gamma[(\nu+n)/2]}{\Gamma(\nu/2) \nu^{n/2} \pi^{n/2}} \right)$ is for normalization; the first term is related to the data-fit term $\mathbf{y}^T \Sigma_t^{-1} \mathbf{y}$. Comparing the first term in Eq. (20) with the one in Eq. (5), the main difference is that the first term in Eq. (20) is a logarithm function of $\mathbf{y}^T \Sigma_t^{-1} \mathbf{y}$ while the one in Eq. (5) is a linear function of $\mathbf{y}^T \Sigma_g^{-1} \mathbf{y}$. That implies if there are outliers in \mathbf{y} , the negative logarithm marginal likelihood of TPRD would be much less disturbed than that of GPR. So, from the view of the negative logarithm marginal likelihood, TPRD should be more robust than GPR.

After deriving the negative logarithm marginal likelihood of TPRD Eq. (20), we need to estimate the hyperparameters $\sigma_t, \boldsymbol{\theta}$ and ν . It's not difficult to derive the partial derivatives of the marginal likelihood of TPRD for each hyperparameter, which implies that the hyperparameters $\sigma_t, \boldsymbol{\theta}$ and ν can be learned by gradient-based optimization methods.

After learning the hyperparameters, we can make predictions by the following result [14].

Lemma 1 Suppose $\mathbf{y} \sim St(\nu, \boldsymbol{\mu}, K)$. $\mathbf{y}_1 \in R^{n_1}$ and $\mathbf{y}_2 \in R^{n_2}$ represent the first n_1 and remaining n_2 entries of \mathbf{y} respectively. Then $\mathbf{y}_1 | \mathbf{y}_2 \sim St(\mathbf{y}_1 | \boldsymbol{\mu}_{1|2}, K_{1|2}, \nu + n_1)$, where $\boldsymbol{\mu}_{1|2} = K_{12} K_{22}^{-1} (\mathbf{y}_2 - \boldsymbol{\mu}_2) + \boldsymbol{\mu}_1$, $K_{1|2} = \frac{\nu + \alpha_1}{\nu + n_1} (K_{22} - K_{21} K_{11}^{-1} K_{12})$, $\alpha_1 = (\mathbf{y}_1 - \boldsymbol{\mu}_1)^T K_{11}^{-1} (\mathbf{y}_1 - \boldsymbol{\mu}_1)$.

For a known input $x_* \in R^D$, we need to give the prediction y_* . As $\{\mathbf{y}, y_*\}$ is a multivariate Student-*t* distribution, by Lemma 1,

$$\begin{aligned} p(y_* | \mathbf{y}) &= St \left(y_* \mid \mathbf{k}_*^T \Sigma_t^{-1} \mathbf{y}, \frac{\nu + \alpha_t}{\nu + n} \left(k(x_*, x_*; \boldsymbol{\theta}) - \mathbf{k}_*^T \Sigma_t^{-1} \mathbf{k}_* \right) \right) \\ \mathbf{k}_* &= \{k(\mathbf{x}_i, \mathbf{x}_*; \boldsymbol{\theta})\}_{i=1}^n, \quad \alpha_t = \mathbf{y}^T \Sigma_t^{-1} \mathbf{y}. \end{aligned} \quad (21)$$

Comparing Eq. (21) with Eq. (6), we see that the form of the predictive mean of TPRD is identical to that of GPR, which implies if the kernel function and hyperparameters are the same, the predictive mean of TPRD is also the same to that of GPR. As mentioned in Section 2.1, the predictive covariance of GPR does not depend on the target vector. In contrast, from Eq. (21), we see that the predictive covariance of TPRD explicitly depends on the target vector, which implies the uncertainties are better accounted for.

2.3 Relation to GPR

Theorem 1 TPRD \rightarrow GPR when $\nu \rightarrow +\infty$.

As $\nu \rightarrow +\infty$, it's well known that a Student-*t* distribution converges to a Gaussian distribution (refer to, e.g., [9]), hence,

$$\begin{aligned} p(\mathbf{f} | X) &= St(\mathbf{f} | \nu, \mathbf{0}, K) \rightarrow \mathcal{N}(\mathbf{f} | \mathbf{0}, K), \\ p(\boldsymbol{\epsilon}) &= St \left(\boldsymbol{\epsilon} \mid \nu + n, \mathbf{0}, \left(1 + \frac{1}{\nu} \mathbf{f}^T K^{-1} \mathbf{f} \right) \frac{1}{\beta} I \right) \\ &\rightarrow \mathcal{N} \left(\boldsymbol{\epsilon} \mid \mathbf{0}, \frac{1}{\beta} I \right), \end{aligned} \quad (22)$$

i.e., TPRD \rightarrow GPR when $\nu \rightarrow +\infty$, which implies that if we choose ν by cross-validation, we can guarantee the performances of TPRD, if not better, are at least as good as that of GPR. Of course, cross-validation for hyperparameter selection requires more computational cost. But gradient-based optimization methods can roughly achieve the same performances, as showed in the experiments of Section 4.

2.4 Relation to TPRK

The Student-*t* process has been studied for a long time [11]. However, as the sum of two independent student-*t* distributions or the sum of a student-*t* distribution and a Gaussian distribution is analytically intractable, it is difficult to use Student-*t* process with noise.

In [14, 15, 19], the noise is incorporated into the kernel function, specifically, by adding a diagonal matrix to the kernel matrix. In this way, the covariance matrix of marginal likelihood equals the kernel matrix plus a diagonal matrix. In practice, this way achieved good performances. However, probabilistic properties of the noise are unknown. In [14], the negative marginal likelihood of TPRK has the form

$$\begin{aligned} -\ln p(\mathbf{y} | \nu, \sigma_a, \boldsymbol{\theta}) &= \frac{\nu+n}{2} \log \left(1 + \frac{1}{\nu-2} \mathbf{y}^T \Sigma_a^{-1} \mathbf{y} \right) \\ &\quad + \frac{1}{2} \log (|\Sigma_a|) - \ln \left(\frac{\Gamma[(\nu+n)/2]}{\Gamma(\nu/2) (\nu-2)^{n/2} \pi^{n/2}} \right), \end{aligned} \quad (23)$$

where $\Sigma_a = K_a + \sigma_a^2 I$, σ_a^2 denotes the noise incorporated to the kernel. Comparing Eq. (23) with Eq. (20), it is easy to see that if $\Sigma_a = \frac{\nu-2}{\nu} \Sigma_t$, Eq. (23) and Eq. (20) would have exactly the same form. The slight difference is caused by the fact that [14] adopts a slightly different definition of Student-t distribution, where the definition is

$$St(\nu, \boldsymbol{\mu}, R) = \frac{\Gamma(\frac{\nu+p}{2})}{\Gamma(\frac{\nu}{2})(\nu-2)^{p/2} \pi^{p/2} |R|^{1/2}} \cdot \left[1 + \frac{1}{\nu-2} (\mathbf{y} - \boldsymbol{\mu})^T R^{-1} (\mathbf{y} - \boldsymbol{\mu}) \right]^{-\frac{\nu+p}{2}} \quad (24)$$

It is not difficult to prove that if we applied this definition of Student-t distribution to Eq. (11) and Eq. (13), the marginal likelihood of TPRD would have the same form as that of TPRK. Besides, Σ_t is equivalent to Σ_a . So, TPRD is equivalent to TPRK. In fact, TPRD gives a probabilistic interpretation to TPRK. By incorporating the noise into the kernel, in fact, the noise approximates to the i.i.d Gaussian noise with variance adjusted to the data-fit term.

3 THEORETICAL ANALYSIS

GPR, TPRD and TPRK are all kernel methods. With different kernels, their performances change a lot. In this section, we will give the definition of the β -property. Then we will prove that if the kernel has the β -property, the predictive mean of TPRD has the same predictive mean as GPR does. Moreover, TPRK also has identical predictive mean as GPR does.

3.1 Maximum likelihood estimation of hyperparameters independent of ν

In this section, we will prove the maximum marginal likelihood estimation of hyperparameters $\boldsymbol{\theta}$ and σ_t is independent of ν if the kernel function has the β -property, which is defined as follows.

Definition 2. For a kernel function $k(\mathbf{x}_1, \mathbf{x}_2; \theta_1, \theta_2, \dots, \theta_l)$, where $\theta_1, \theta_2, \dots, \theta_l$ is the parameters of the kernel, if $k(\mathbf{x}_1, \mathbf{x}_2; \theta_1, \theta_2, \dots, \theta_l) = g(\theta_1)k(\mathbf{x}_1, \mathbf{x}_2; 1, \theta'_2, \dots, \theta'_l)$, where $\theta'_i (i = 2, 3, \dots, l)$ corresponds to θ_i one to one for given θ_1 , and $g(\theta_1)$ is an injective function of θ_1 with the range $(0, +\infty)$, then the kernel $k(\mathbf{x}_1, \mathbf{x}_2; \theta_1, \theta_2, \dots, \theta_l)$ is called to have the β -property.

There are several common kernels with the β -property, e.g., a diagonal squared exponential kernel [13] has the form

$$k(\mathbf{x}_1, \mathbf{x}_2; \sigma_f, \ell) = \sigma_f^2 \exp\left(-\frac{1}{2\ell^2} \mathbf{x}_1^T \mathbf{x}_2\right) = \sigma_f^2 k(\mathbf{x}_1, \mathbf{x}_2; 1, \ell), \quad \sigma_f > 0, \quad (25)$$

and a linear with diagonal weighting kernel [13]

$$k(\mathbf{x}_1, \mathbf{x}_2; \boldsymbol{\lambda}) = \mathbf{x}_1^T \Lambda^{-2} \mathbf{x}_2, \quad \Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_D) = \frac{\mathbf{x}_1^T \Lambda'^{-2} \mathbf{x}_2}{\lambda_1^2}, \quad \Lambda' = \text{diag}\left(1, \frac{\lambda_2}{\lambda_1}, \dots, \frac{\lambda_D}{\lambda_1}\right) \quad (26)$$

$$= \frac{1}{\lambda_1^2} k\left(\mathbf{x}_1, \mathbf{x}_2; 1, \frac{\lambda_2}{\lambda_1}, \dots, \frac{\lambda_D}{\lambda_1}\right), \quad \lambda_1 > 0,$$

where $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \dots, \lambda_D)$.

There are also common kernels without the β -property, e.g., a squared exponential kernel [13] has the form

$$k(\mathbf{x}_1, \mathbf{x}_2; \ell) = \exp\left(-\frac{1}{2\ell^2} \mathbf{x}_1^T \mathbf{x}_2\right). \quad (27)$$

By Eq. (19), σ_t seems to be dependent on ν , however, we have the following surprising results.

Theorem 2 *If the kernel in TPRD has the β -property, then the maximum likelihood estimation of hyperparameters $\boldsymbol{\theta}$ and σ_t is independent of ν . Furthermore, the predictive mean of TPRD is the same as that of GPR.*

Proof. The marginal likelihood of TPRD has the form

$$p(\mathbf{y}|\boldsymbol{\theta}, \sigma_t) = \frac{\Gamma[(\nu+n)/2]}{\Gamma(\nu/2)\nu^{n/2}\pi^{n/2}|\Sigma_t|^{1/2}} \left[1 + \frac{1}{\nu} \mathbf{y}^T \Sigma_t^{-1} \mathbf{y} \right]^{-\frac{(\nu+n)}{2}}, \quad (28)$$

where $\Sigma_t = K_t + \sigma_t^2 I$. As the kernel has the β -property, we have

$$\begin{aligned} \Sigma_t &= K_t + \sigma_t^2 I, & (K_t)_{ij} &= k(\mathbf{x}_i, \mathbf{x}_j; \theta_1, \theta_2, \dots, \theta_l) \\ &= g(\theta_1)K'_t + \sigma_t^2 I, & (K'_t)_{ij} &= k(\mathbf{x}_i, \mathbf{x}_j; 1, \theta'_2, \dots, \theta'_l) \\ &= \sigma_t^2 \Sigma'_t, \end{aligned} \quad (29)$$

where

$$\Sigma'_t = \lambda K'_t + I, \quad \lambda = \frac{g(\theta_1)}{\sigma_t^2}, \quad (30)$$

and $\theta'_i (i = 2, 3, \dots, l)$ depends on θ_1 and θ_i .

As $g(\theta_1)$ is an injective function, it is not difficult to show that $\sigma_t, \lambda, \theta'_2, \dots, \theta'_l$ have one-to-one mapping to $\sigma_t, \theta_1, \theta_2, \dots, \theta_l$. By Eq. (28), for a given ν , we have

$$\begin{aligned} &\max_{\sigma_t, \boldsymbol{\theta}} p(\mathbf{y}|\sigma_t, \boldsymbol{\theta}) \\ &\Leftrightarrow \max_{\sigma_t, \lambda, \theta'_2, \dots, \theta'_l} p(\mathbf{y}|\sigma_t, \lambda, \theta'_2, \dots, \theta'_l) \\ &\Leftrightarrow \min_{\sigma_t, \lambda, \theta'_2, \dots, \theta'_l} (p(\mathbf{y}|\sigma_t, \lambda, \theta'_2, \dots, \theta'_l))^{\frac{-2}{\nu+n}} \\ &\Leftrightarrow \min_{\sigma_t, \lambda, \theta'_2, \dots, \theta'_l} h(\mathbf{y}|\sigma_t, \lambda, \theta'_2, \dots, \theta'_l), \end{aligned} \quad (31)$$

where

$$\begin{aligned} h(\mathbf{y}|\sigma_t, \lambda, \theta'_2, \dots, \theta'_l) &= \left[\frac{1}{|\Sigma_t|^{1/2}} \left(1 + \frac{1}{\nu} \mathbf{y}^T \Sigma_t^{-1} \mathbf{y} \right)^{\frac{\nu+n}{2}} \right]^{\frac{-2}{\nu+n}} \\ &= |\Sigma_t|^{\frac{1}{\nu+n}} \left(1 + \frac{1}{\nu} \mathbf{y}^T \Sigma_t^{-1} \mathbf{y} \right) \\ &= \sigma_t^{\frac{2n}{\nu+n}} |\Sigma'_t|^{\frac{1}{\nu+n}} \left(1 + \frac{1}{\nu \sigma_t^2} \mathbf{y}^T \Sigma'^{-1}_t \mathbf{y} \right). \end{aligned} \quad (32)$$

Recall that σ_t and λ have one-to-one mapping to σ_t and θ_1 . Then deriving $h(\mathbf{y}|\sigma_t, \lambda, \theta'_2, \dots, \theta'_l)$ with respect to σ_t ,

$$\frac{\partial h(\mathbf{y})}{\partial \sigma_t} = \frac{2n}{\nu+n} \sigma_t^{\frac{2n}{\nu+n}-1} |\Sigma'_t|^{\frac{1}{\nu+n}} - \frac{2\sigma_t^{\frac{2n}{\nu+n}-3} |\Sigma'_t|^{\frac{1}{\nu+n}}}{\nu+n} \mathbf{y}^T \Sigma'^{-1}_t \mathbf{y}. \quad (33)$$

Letting Eq. (33) be zero, we get the maximum marginal likelihood estimation of σ_t

$$\hat{\sigma}_t = \sqrt{\frac{\mathbf{y}^T \Sigma'^{-1}_t \mathbf{y}}{n}}. \quad (34)$$

From Eq. (34) and Eq. (30), we can see that $\hat{\sigma}_t$ is determined by training data \mathcal{D} and hyperparameters $\lambda, \theta'_2, \theta'_3, \dots, \theta'_l$, which implies that $\hat{\sigma}_t$ is independent of ν . From Eq. (34) and Eq. (32), we obtain

$$h(\mathbf{y}) = \left(\frac{\mathbf{y}^T \Sigma'^{-1}_t \mathbf{y}}{n} \right)^{\frac{n}{\nu+n}} |\Sigma'_t|^{\frac{1}{\nu+n}} \left(1 + \frac{n}{\nu} \right). \quad (35)$$

Let $h'(\mathbf{y}) = \left(\frac{\mathbf{y}^T \Sigma_t'^{-1} \mathbf{y}}{n} \right)^n |\Sigma_t'|$ and $\phi = \lambda, \theta_2', \dots, \theta_l'$, similar to Eq. (31) we have

$$\begin{aligned} & \min_{\phi} h(\mathbf{y}) \\ \Leftrightarrow & \min_{\phi} \left(\frac{\mathbf{y}^T \Sigma_t'^{-1} \mathbf{y}}{n} \right)^{\frac{n}{\nu+n}} |\Sigma_t'|^{\frac{1}{\nu+n}} \\ \Leftrightarrow & \min_{\phi} h'(\mathbf{y}). \end{aligned} \quad (36)$$

Since $h'(\mathbf{y})$ is independent of ν , the solution of $\partial h'(\mathbf{y})/\partial \phi = 0$ is independent of ν .

Now, we have proved that the maximum likelihood estimation of $\sigma_t, \lambda, \theta_2', \dots, \theta_l'$ is independent of ν . As there is a one-to-one mapping between θ, σ_t and $\sigma_t, \lambda, \theta_2', \dots, \theta_l'$, the maximum likelihood estimation of θ, σ_t is independent of ν .

From Lemma 1 in Section 2.2, we know the predictive mean of TPRD has no relationship with ν , it is only determined by the training data \mathcal{D} and hyperparameters θ, σ_t . If the kernel has the β -property, the hyperparameters θ, σ_t are shown to be independent of ν , which implies that the predictive mean of TPRD is independent of ν . In other words, no matter what ν is, the prediction of TPRD does not change. As GPR is a special case of TPRD with $\nu \rightarrow +\infty$, the predictive mean of TPRD is the same as that of GPR. \square

This result might be interesting. Since in Section 2.2, we state TPRD should be more robust than GPR since the difference of their negative logarithm marginal likelihoods. Now, we prove that they have the same predictive mean with the β -property kernel. The underlying reason is that the β -property kernel has a free factor $g(\theta_1)$, which compromises their difference between the negative logarithm marginal likelihoods.

As the degrees of freedom ν does not affect the predictive mean, we can choose a suitable ν for speedup.

3.2 Predictive mean comparison

In TPRK, the marginal likelihood is different from the marginal likelihood of GPR. So, Shah, Wilson and Zoubin [14] expect that the predictive mean of TPRK would differ from that of GPR. However, we prove that when we use the maximum marginal likelihood to estimate the hyperparameters, the predictive mean of TPRK is identical to that of GPR if the kernel has the β -property.

Theorem 3 *If the maximum marginal likelihood is used to estimate the hyperparameters and the kernel has the β -property, for a given ν , the predictive mean of TPRK is the same as that of TPRD.*

Proof. Firstly, as the kernel has the β -property, we have

$$k(\mathbf{x}_1, \mathbf{x}_2; \theta_1, \theta_2, \dots, \theta_l) = g(\theta_1) k(\mathbf{x}_1, \mathbf{x}_2; 1, \theta_2', \dots, \theta_l'), \quad (37)$$

where the range of $g(\theta_1)$ is $(0, +\infty)$. We assume the maximum marginal likelihood estimation solution of TPRD's hyperparameters is $(\sigma_t^0, \theta_1^0, \theta_2^0, \dots, \theta_l^0)$. As $g(\theta_1)$ is an injective function with the range $(0, +\infty)$ and in TPRK $\nu > 2$, there is a θ_1^0 satisfying that $\frac{\nu-2}{\nu} g(\theta_1^0) = g(\theta_1^0)$. Then at $\left(\sqrt{\frac{\nu}{\nu-2}} \sigma_t^0, \theta_1^0, \theta_2^0, \dots, \theta_l^0 \right)$, we have

$$k(\mathbf{x}_1, \mathbf{x}_2; \theta_1^0, \theta_2^0, \dots, \theta_l^0) = \frac{\nu}{\nu-2} k(\mathbf{x}_1, \mathbf{x}_2; \theta_1^0, \theta_2^0, \dots, \theta_l^0) \quad (38)$$

$$\Sigma_a = K_a + \sigma_a^2 I = \frac{\nu}{\nu-2} K_t + \frac{\nu}{\nu-2} \sigma_t^2 I = \frac{\nu}{\nu-2} \Sigma_t, \quad (39)$$

which implies the marginal likelihood value of TPRK equals that of TPRD. In fact, at $\left(\sqrt{\frac{\nu}{\nu-2}} \sigma_t^0, \theta_1^0, \theta_2^0, \dots, \theta_l^0 \right)$, the marginal likelihood value of TPRK reaches the maximum. Otherwise, if there is another set of hyperparameters at which the marginal likelihood value of TPRK is larger, there is a corresponding set of hyperparameters at which the marginal likelihood of TPRD is larger, contradictory to the fact that the maximum marginal likelihood solution of TPRD is $(\sigma_t^0, \theta_1^0, \theta_2^0, \dots, \theta_l^0)$.

For a given known $x_* \in R^D$, the predictive mean of TPRK is

$$\begin{aligned} & k(x_*, X; \theta_1^0, \theta_2^0, \dots, \theta_l^0)^T \Sigma_a^{-1} \mathbf{y} \\ &= \frac{\nu}{\nu-2} k(x_*, X; \theta_1^0, \theta_2^0, \dots, \theta_l^0)^T \frac{\nu-2}{\nu} \Sigma_t^{-1} \mathbf{y} \\ &= k(x_*, X; \theta_1^0, \theta_2^0, \dots, \theta_l^0)^T \Sigma_t^{-1} \mathbf{y}. \end{aligned} \quad (40)$$

The RHS is the predictive mean of TPRD. \square

Corollary 1 *If the maximum marginal likelihood is used to estimate the hyperparameters and the kernel has the β -property, the predictive mean of TPRK is the same as the prediction of GPR.*

Proof. By Theorem 3, we know that if the maximum marginal likelihood is used to estimate the hyperparameters and the kernel has the β -property, the predictive mean of TPRK has the same predictive mean as that of TPRD. And in that case, by Theorem 2, TPRD and GRP also have the same predictive mean. We conclude that in that case, TPRK has the same predictive mean as that of GPR. \square

This result is interesting. As the negative logarithm marginal likelihood of TPRK is different from the one of GPR. The predictive mean of TPRK is expected to be different from that of GPR after learning the hyperparameters in [14]. However, by Corollary 1, TPRK and GPR have identical predictive mean if the maximum marginal likelihood is used and the kernel has the β -property. The underlying reason is that the kernel with the β -property has a free factor $g(\theta_1)$, which compromises the difference between their marginal likelihood.

4 EXPERIMENTS

Our experiments are designed to verify the following three propositions:

- If the kernel has the β -property, the ML estimation of TPRD's hyperparameters is independent of ν and the predictive mean of TPRD is the same as that of GPR. ν influences the convergence rate.
- If the kernel has the β -property and ML estimation is used, TPRK, TPRD and GPR have the same predictive mean.
- If the kernel does not have the β -property, TPRD performs better than GPR.

4.1 Data sets

We use the following seven data sets to carry out the experiments. All data are from the UCI data sets [7].

- **Servo Data.** This data set contains 4 attributes and 167 instances from a simulation of a servo system. The attributes include motor, screw, pgain and vgain. The output variable is class from 0.13 to 7.10.

- **Stock Data.** This data set includes returns of Istanbul Stock Exchange with seven other international index. There are 536 instances and 8 attributes. We randomly choose a subset of 400 instances for the experiments.
- **Wine Data [2].** This data set contains 12 attributes and 1599 instances associated with red wine. The attributes include acidity, residual sugar, pH and so on. The output variable is quality (score between 0 and 10). We randomly choose a subset of 400 instances for the experiments.
- **Airfoil Data.** Airfoil data comprises different size NACA 0012 airfoils at various wind tunnel speeds and angles of attack. It contains 1503 instances and 6 attributes. We randomly select a subset of 400 instances for the experiments.
- **Yacht Data.** This data set is used to predict the hydrodynamic performances of sailing yachts from dimensions and velocity. It contains 308 instances and 7 attributes.
- **Space Data.** This data set is to predict the number of O-rings that will experience thermal distress for a given flight when the launch temperature is below freezing point. It contains 23 instances and 4 attributes.
- **Concrete Data [18].** This data set is about the slump flow of concrete. It contains 1030 instances and 9 attributes. We randomly choose a subset of 400 instances for the experiments.

4.2 Experimental setup

We use the gradient descent algorithm [1] to get the minimum of the negative logarithm marginal likelihood. The maximum iteration number is 5000 and the stop criterion is that the absolute value of each hyperparameter's derivative is less than 10^{-8} . The initial value for step length is 0.001. And the initial value for the hyperparameters σ and $\theta_i (i = 1, 2, \dots, l)$ are 1. All the data are standardized.

4.3 ν independent of the β -property kernel

We verify the ν independent property on two kernels and two data sets. The kernels are the diagonal squared exponential kernel Eq. (25) and the linear kernel with isotropic weighting [13], which has the form $k(\mathbf{x}_1, \mathbf{x}_2; \ell) = \frac{\mathbf{x}_1^T \mathbf{x}_2}{\ell^2}$. As shown in Eq. (25) and Eq. (26), both of the kernels have the β -property. The data sets are the Servo and Stock data. We compare TPRD($\nu = 3, 10, 1000, 100000$) with GPR.

Table 1. The ML estimation results on the Servo data set with the linear kernel

Model	$\ln \ell$	$\ln \sigma$	MSE	iteration
GPR	0.4880	-0.3706	0.7947	486
TPRD($\nu=3$)	0.4880	-0.3706	0.7947	581
TPRD($\nu=10$)	0.4880	-0.3706	0.7947	339
TPRD($\nu=1000$)	0.4880	-0.3706	0.7947	384
TPRD($\nu=100000$)	0.4880	-0.3706	0.7947	418

Table 2. The ML estimation results on the Servo data set with the diagonal squared exponential kernel

Model	$\ln \ell$	$\ln \sigma_f$	$\ln \sigma$	MSE	iteration
GPR	0.5588	0.6278	-1.2413	0.3041	1500
TPRD($\nu=3$)	0.5588	0.6278	-1.2413	0.3041	767
TPRD($\nu=10$)	0.5588	0.6278	-1.2413	0.3041	598
TPRD($\nu=1000$)	0.5588	0.6278	-1.2413	0.3041	2265
TPRD($\nu=100000$)	0.5588	0.6278	-1.2413	0.3041	2004

Table 3. The ML estimation results on the Stock data set with the linear kernel

Model	$\ln \ell$	$\ln \sigma$	MSE	iteration
GPR	1.1757	-0.7758	0.1954	1406
TPRD($\nu=3$)	1.1757	-0.7758	0.1954	496
TPRD($\nu=10$)	1.1757	-0.7758	0.1954	322
TPRD($\nu=1000$)	1.1757	-0.7758	0.1954	2484
TPRD($\nu=100000$)	1.1757	-0.7758	0.1954	3386

Table 4. The ML estimation results on the Stock data set with the diagonal squared exponential kernel

Model	$\ln \ell$	$\ln \sigma_f$	$\ln \sigma$	MSE	iteration
GPR	2.7766	1.7299	-0.7949	0.1377	5000
TPRD($\nu=3$)	2.7767	1.7300	-0.7949	0.1377	1002
TPRD($\nu=10$)	2.7767	1.7300	-0.7949	0.1377	887
TPRD($\nu=1000$)	2.7752	1.7282	-0.7949	0.1377	5000
TPRD($\nu=100000$)	2.7659	1.7170	-0.7953	0.1378	5000

From Table 1-4, we can see that the ML estimation of hyperparameters of TPRD is indeed the same as that of GPR. The slight difference in Table 4 is caused by that fact the maximum iteration is reached before the optimization point is reached. And the MSE of TPRD and GPR is identical, which implies that they have the same predictive mean.

Another interesting phenomenon is that the convergence rate is indeed influenced by ν . On most data sets, when ν is set to 10, TPRD has the smallest number of iteration, faster than GPR. Empirically, we recommend that ν is set around 10. The underlying reason may be that the tail thickness is appropriate for most data when ν is around 10, considering that ν controls the thickness of the tail.

The result may be remarkable, since the computational cost is an important problem of GPR. As the main computational cost lies on solving the inverse of the covariance matrix, most efforts are focused on the covariance matrix, e.g., from the sparsity [4], distributed method [3], and exploiting the structure of covariance matrix [17]. Our model provides an iteration-less way, which may be able to combine with the covariance matrix way.

4.4 Predictive mean of GPR, TPRD and TPRK with the β -property kernel

Now, we use the experiments to verify the Theorem 3 and Corollary 1 that the predictive mean of TPRK is identical to that of GPR and TPRD with the β -property kernel and ML estimation.

We compare TPRK($\nu = 3, 10, 1000, 100000$) with GPR and TPRD($\nu = 10$) on the Servo data set. The following are the experimental results.

From Table 5 and 6, we see that on each β -property kernel, whatever ν is, the MSE of TPRK is the same as that of GPR and TPRD, which implies these three models have the same predictive mean.

Besides, different from TPRD, ν influences the ML estimation of hyperparameters of TPRK. Next, we exploit how the ν influences them. From Section 3.2, we know that if the ML estimation of TPRD's hyperparameters is $(\sigma_t^0, \theta_1^0, \theta_2^0, \dots, \theta_l^0)$, the ML estimation of TPRK's hyperparameters is $(\sigma_a^0, \theta_1^0, \theta_2^0, \dots, \theta_l^0)$, where

$$\sigma_a^0 = \sqrt{\frac{\nu}{\nu-2}} \sigma_t^0, \quad (41)$$

$$\frac{\nu-2}{\nu} g(\theta_1^0) = g(\theta_1^0). \quad (42)$$

Table 5. The ML estimation results on the Servo data set with the linear kernel

Model	$\ln \ell$	$\ln \sigma$	MSE
GPR	0.5866	-0.3603	0.6628
TPRD($\nu=10$)	0.5866	-0.3603	0.6628
TPRK($\nu=3$)	0.03733	0.1890	0.6628
TPRK($\nu=10$)	0.4751	-0.2487	0.6628
TPRK($\nu=1000$)	0.5856	-0.3593	0.6628
TPRK($\nu=100000$)	0.5866	-0.3603	0.6628

Table 6. The ML estimation results on the Servo data set with the diagonal squared exponential kernel

Model	$\ln \ell$	$\ln \sigma_f$	$\ln \sigma$	MSE
GPR	0.5977	0.5297	-1.1933	0.1397
TPRD($\nu=10$)	0.5977	0.5297	-1.1933	0.1397
TPRK($\nu=3$)	0.5977	1.079	-0.6440	0.1397
TPRK($\nu=10$)	0.5977	0.6413	-1.082	0.1397
TPRK($\nu=1000$)	0.5977	0.5307	-1.192	0.1397
TPRK($\nu=100000$)	0.5977	0.5297	-1.193	0.1397

Taking TPRK($\nu=3$) for example, we check whether the experimental results are consistent with the relationship above.

For the hyperparameters θ_1^0 and θ_1^0 , from Table 6, we know

$$\begin{aligned}
\frac{\nu-2}{\nu}g(\theta_1^0) &= \frac{\nu-2}{\nu}(\sigma_f)^2 \\
&= \frac{\nu-2}{\nu}\exp(2\sigma_f) \\
&= \frac{3-2}{3}\exp(2 \times 1.079) \\
&= 2.8846 \\
g(\theta_1^0) &= (\theta_1^0)^2 \\
&= \exp(2 \times 0.5297) \\
&= 2.8846,
\end{aligned} \tag{43}$$

which verifies the Eq. (42).

For σ_a^0 and σ_t^0 , we have

$$\begin{aligned}
\sqrt{\frac{\nu}{\nu-2}}\sigma_t^0 &= \sqrt{\frac{3}{3-2}}\exp(-1.1933) \\
&= 0.5252 \\
\sigma_a^0 &= \exp(\ln \sigma_a^0) \\
&= \exp(-0.6440) \\
&= 0.5252,
\end{aligned} \tag{44}$$

which is consistent with the Eq. (41).

As the ML estimation of TPRD's hyperparameters is not influenced by ν , from the relationship, we know that ν does not affect the ML estimation of hyperparameters of TPRK, except σ_a and θ_1^0 . From Table 6, it is clear that ν indeed does not affect the estimation of ℓ .

4.5 Robustness of TPRD with non- β -property kernel

Now, we evaluate the robustness of TPRD on all the data sets (Section 4.1) with the squared exponential kernel, which does not have the β -property. Table 7 reports the MSE of TPRD and GPR. The MSE of data sets on which TRPD is significantly better than GPR is in boldface. We see that on the data set Airfoil, Concrete, Servo and Stock, TPRD has similar performances with GPR. Then

in each training dataset, 5% of the instances are chosen randomly and each value of the target variable in these instances is randomly added or subtracted by 3 standard derivations of the target variable. It's clear that TPRD performs more robustly than GPR when the data are contaminated by the outliers. And on the data sets Space, Yacht, Wine, TPRD outperforms GPR. Just as we state in Section 2.2, the negative logarithm marginal likelihood of TPRD is a logarithm function of the data-fit term, while that of GPR is a linear function. Therefore, TPRD is more robust than GPR theoretically and empirically, with the non- β -property kernel.

Table 7. The MSE of GPR and TPRD

Data Set	GPR MSE	TPRD MSE
Airfoil	0.3153	0.3108
Airfoil with outliers	0.5524	0.3557
Concrete	0.1545	0.1665
Concret with outliers	0.5503	0.2706
Servo	0.1541	0.1550
Servo with outliers	0.8616	0.2451
Stock	0.2744	0.2756
Stock with outliers	0.7833	0.2928
Space	0.4258	0.3386
Yacht	0.0544	0.0458
Wine	0.8362	0.6580

5 CONCLUSION

We have proposed a Student-*t* Process Regression with Dependent Student-*t* noise model (TPRD) in this paper, which is proved to be a generalization of GPR. In addition, TPRD gives a probabilistic interpretation to the Student-*t* Process Regression with noise incorporated into the Kernel (TPRK). In fact, by incorporating the noise into the kernel, the noise approximates to the Gaussian noise with the variance adjusted to the data-fit term. More importantly, we analyze the influence of different kernels on TPRD and TPRK. Specifically, if the kernel has the β -property, the ML estimation of TPRD's hyperparameters is independent of ν and we also discuss how ν influences the ML estimation of the TPRK's hyperparameters. Besides, the predictive mean of TPRD, TPRK and GPR is identical, which is not expected by [14]. In that case, empirically, ν is a convergence accelerator and TPRD can be faster than GPR. On the other hand, if the kernel does not have the β -property, owing to the dependent noise, experimental results show that TPRD achieves significantly better performances than GPR.

ACKNOWLEDGMENTS

This research is supported in part by the Major State Basic Research Development Program of China (973 Program, 2012CB315803), the National Natural Science Foundation of China (61371078), and the Research Fund for the Doctoral Program of Higher Education of China (20130002110051).

REFERENCES

- Conference on Artificial Intelligence and Statistics (AISTATS-10), volume 9, p. 964, (2010).
- [1] Stephen Boyd and Lieven Vandenberghe, *Convex optimization*, Cambridge university press, 2004.
 - [2] Paulo Cortez, Antnio Cerdeira, Fernando Almeida, Telmo Matos, and Jos Reis, ‘Modeling wine preferences by data mining from physicochemical properties’, *Decision Support Systems*, **47**(4), 547 – 553, (2009). Smart Business Networks: Concepts and Empirical Evidence.
 - [3] Marc Deisenroth and Jun Wei Ng, ‘Distributed gaussian processes’, in *Proceedings of The 32nd International Conference on Machine Learning (ICML-15)*, pp. 1481–1490, (2015).
 - [4] Yarin Gal and Richard Turner, ‘Improving the gaussian process sparse spectrum approximation by representing uncertainty in frequency inputs’, in *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pp. 655–664, (2015).
 - [5] Somesh Das Gupta and Jun Shao. *Mathematical statistics*, 2000.
 - [6] Malte Kuss, *Gaussian process models for robust regression, classification, and reinforcement learning*, Ph.D. dissertation, TU Darmstadt, 2006.
 - [7] M. Lichman. UCI machine learning repository, 2013.
 - [8] Alexander J McNeil, ‘Multivariate t distributions and their applications’, *Journal of the American Statistical Association*, **101**(473), 390–391, (2006).
 - [9] Saralees Nadarajah and Samuel Kotz, ‘Mathematical properties of the multivariate t distribution’, *Acta Applicandae Mathematica*, **89**(1-3), 53–84, (2005).
 - [10] Radford M Neal, ‘Monte carlo implementation of gaussian process models for bayesian regression and classification’, Technical report, Dept. of statistics and Dept. of Computer Science, University of Toronto, (1997).
 - [11] Anthony O’Hagan, ‘Bayes–hermite quadrature’, *Journal of statistical planning and inference*, **29**(3), 245–260, (1991).
 - [12] Carl Edward Rasmussen, ‘Gaussian processes for machine learning’, (2006).
 - [13] Carl Edward Rasmussen and Hannes Nickisch, ‘The gpml toolbox version 3.5’, (2015).
 - [14] Amar Shah, Andrew Wilson, and Zoubin Ghahramani, ‘Student-t processes as alternatives to gaussian processes’, in *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics (AISTATS-14)*, pp. 877–885, (2014).
 - [15] Arno Solin and Simo Särkkä, ‘State space methods for efficient inference in student-t process regression’, in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics (AISTATS-15)*, pp. 885–893, (2015).
 - [16] Jarno Vanhatalo, Pasi Jylänki, and Aki Vehtari, ‘Gaussian process regression with student-t likelihood’, in *Advances in Neural Information Processing Systems (NIPS-09)*, pp. 1910–1918, (2009).
 - [17] Andrew Wilson and Hannes Nickisch, ‘Kernel interpolation for scalable structured gaussian processes (kiss-gp)’, in *Proceedings of The 32nd International Conference on Machine Learning (ICML-15)*, pp. 1775–1784, (2015).
 - [18] I-Cheng Yeh, ‘Modeling slump flow of concrete using second-order regressions and artificial neural networks’, *Cement and Concrete Composites*, **29**(6), 474–480, (2007).
 - [19] Yu Zhang and Dit-Yan Yeung, ‘Multi-task learning using generalized t process’, in *Proceedings of the Thirteenth International*

An Extension of the Owen-Value Interaction Index and Its Application to Inter-Links Prediction

Piotr L. Szczepański¹ and Tomasz P. Michalak^{2,3} and Talal Rahwan⁴ and Michael Wooldridge²

Abstract. Link prediction is a key problem in social network analysis: it involves making suggestions about where to add new links in a network, based solely on the structure of the network. We address a special case of this problem, whereby the new links are supposed to connect different communities in the network; we call it the *inter-links prediction problem*. This is particularly challenging as there are typically very few links between different communities. To solve this problem, we propose a local node-similarity measure, inspired by the *Owen-value interaction index*—a concept developed in cooperative game theory and fuzzy systems. Although this index requires an exponential number of operations in the general case, we show that our local node-similarity measure is computable in polynomial time. We apply our measure to solve the inter-links prediction problem in a number of real-life networks, and show that it outperforms all other local similarity measures in the literature.

1 INTRODUCTION

Link prediction is one of the key problems in social network analysis [32, 28, 45]. Informally, it involves making recommendations about where to add new links in a network, based solely on the structure of that network. Link prediction has many applications, such as (i) identifying potential customers in online shops [10]; (ii) discovering the interactions between proteins in biological networks [5]; and (iii) finding hidden connections between terrorists [23].

The problem of link prediction is strongly associated with the notion of *similarity* between nodes in a network [32]: the greater the similarity between two nodes, the greater the likelihood of having a link between them. Broadly speaking, computing the similarity between any two nodes may either involve *local* or *global* information about those nodes. Each approach has its relative strengths and weaknesses. In particular, compared to local measures, global ones generally yield better results but are harder to compute, which limits their applicability to small networks (more details can be found in Section 7). In this paper, we restrict our attention to the problem of link prediction based on local information.

Some researchers [43, 50] have suggested exploiting the fact that, in real-life networks, nodes tend to form densely-connected groups, or *communities* [15], and that nodes from the same community are more likely to be connected.

We address a new problem, whereby we are given a network and a community structure, and want to recommend new links between different communities. We call these “*inter-links*” (as opposed to *intra-*

links, which connect nodes belonging to the same community). To see why this new problem is significant, consider some applications of the general link-prediction problem:

- One of the most lucrative business applications of link prediction is product recommendation in e-commerce [29], whereby any customer viewing a certain product is presented with a list of similar products. In this context, besides obviously-similar products, it may be worthwhile to also recommend some other products that are different yet relevant. This can be modeled as the problem of recommending inter-links between products belonging to different categories, or “communities”.
- Another promising application of link prediction is to recommend new collaborations in academic networks [52]. While current tools focus on recommending collaborations between academics from the same field of study, such tools can benefit from identifying any promising collaborations between members of different communities, e.g., to foster interdisciplinary research and promote the creation of diverse teams.

Our approach to the inter-link prediction problem draws inspiration from the field of cooperative game theory. Concepts from network science may be understood in a cooperative game theoretic setting as follows:

- a *node* is represented as a *player*;
- a *group centrality* [13] is represented as a *characteristic function* that assigns to each group a real value reflecting its *payoff*, or power, according to some metric;
- a *community* (or a subset of nodes) is a *coalition* (or a subset of players), and the *community structure* corresponds to a *coalition structure*.

With this mapping in place, it is possible to measure the similarity between any two nodes using the *interaction index* [18]—a game-theoretic concept that measures the interaction between two players by analyzing the payoffs of the many possible coalitions in the game. At its core, an interaction index is built around a *payoff-division scheme* (more on this in Section 2). Among the many schemes that can be used for this purpose, one particularly attractive family of schemes is *Semivalues* [19]; by using it, we obtain the *Semivalue interaction index*. This particular index was recently proposed as a local measure of node similarity [45]. Although this measure was shown to be useful for link prediction, it does not take into account the underlying community structure. To overcome this issue, our idea is to use the *Owen value* [37]—a payoff division scheme inherently designed to handle situations where there is an underlying coalition structure; the resulting node-similarity measure is the *Owen-value interaction index*. We also propose a family of schemes that generalize

¹ Warsaw University of Technology, Poland

² University of Oxford, UK

³ University of Warsaw, Poland

⁴ Masdar Institute, UAE

the Owen value, namely *Coalitional Semivalues* [47]; the resulting node-similarity measure is the *Coalitional-Semivalued interaction index*. To the best of our knowledge, we are the first to study this latter interaction index.

In summary, the contribution of this paper is as follows:

- We formulate the problem of inter-link prediction in networks with a community structure;
- We introduce an extension of Owen-value interaction index called the Coalitional-Semivalued interaction index, and use it to define the new local similarity measure on networks;
- We propose polynomial time algorithms to efficiently compute the Owen-value and Coalitional-Semivalued interaction index on networks;
- We empirically evaluate our measure by applying it to solve the inter-links prediction problem for a number of real-life networks, and show that it outperforms other local node-similarity measures.

The remainder of this paper is organized as follows. In Section 2, we introduce some basic notations and concepts from graph theory and cooperative games theory. We formally define the new node-similarity measure in Section 5, and analyze its computational complexity in Section 4. An efficient algorithm is then presented in Section 5. The experimental results are presented in Section 6. A brief discussion of related bodies of literature is presented in Section 7, before concluding the paper.

2 PRELIMINARIES

In this section we introduce the key definitions and notation used throughout the paper.

Network notation: A graph (or a network) is denoted by $G(V, E)$, where $V = \{v_1, \dots, v_{|V|}\}$ is the set of nodes and E is the set of edges. We will sometimes write G instead of $G(V, E)$ for brevity. In this paper we consider only undirected and unweighted graphs. We will often use v and u to denote two arbitrary nodes. For any two nodes, $v, u \in V$, the distance (i.e., the length of the shortest path) between them will be denoted by $d(v, u)$. A *community* in a network is a subset of nodes, whereas a *community structure* is an exhaustive and disjoint set of communities.

A *centrality index* (or simply a *centrality*) measures the importance of individual nodes. One of the fundamental centrality indices is *degree centrality* [14, 13, 36], which simply measures the importance of a node, v , based solely on the degree of v —the number of nodes within 1 step from v . This can be generalized to k steps, resulting in what is known as *k-steps degree centrality*. The notion of centrality can also be generalized to groups of nodes, resulting in what is known as *group centrality* [13]. One such group centrality that we will focus on in this paper is *k-steps group degree centrality*.

Definition 1 Given a network G , an integer $k \in \{1, \dots, |V|\}$, and a community $S \subseteq V$, the *k-steps group degree centrality* of S is:

$$\left| \left\{ v \in V : \min_{u \in S} d(u, v) \leq k \right\} \setminus S \right| \quad (1)$$

Some authors [35, 45] interpret the above formula as a *sphere of influence* of the community S in the network. From this perspective, the parameter k can be interpreted as the “diameter” of this sphere.

Coalitional games: A *coalitional game in characteristic function form* (also called a *cooperative game*) is comprised of a set of players $N = \{1, 2, \dots, |N|\}$ and a *characteristic function* $\nu : 2^N \rightarrow \mathbb{R}$

which evaluates each *coalition* $C \subseteq N$ of players, under the assumption that $\nu(\emptyset) = 0$. We often refer to $\nu(C)$ as the *value*, or *payoff*, of C .

Semivalues: This is a family of payoff-division schemes, or *solution concepts*, designed to specify how the payoff of any given coalition should be divided among its members [11]. It is centered around the notion of *marginal contribution*; for every player, $i \in N$, and every coalition, $C \subseteq N$, the marginal contribution of i to C is:

$$\text{MC}(C, i) = \nu(C \cup \{i\}) - \nu(C).$$

Now, let $\beta : \{0, \dots, |N| - 1\} \rightarrow [0, 1]$ be a discrete probability distribution, where $\beta(k)$ is the probability that a coalition of size k is drawn from the set of all possible coalitions whose size is no more than $|N| - 1$. Then, a Semivalued is defined as follows:

Definition 2 Given a game, (N, ν) , and a discrete probability distribution, $\beta : \{0, \dots, |N| - 1\} \rightarrow [0, 1]$, $\sum_{0 \leq k < |N|} \beta(k) = 1$, the *Semivalued of a player*, $i \in N$, is:

$$\text{SEMI}_i(N, \nu) = \sum_{0 \leq k < |N|} \beta(k) \mathbb{E}[\text{MC}(X^k, i)], \quad (2)$$

where X^k is a coalition of size k drawn uniformly from $\{C : C \subseteq N \setminus \{i\} \wedge |C| = k\}$, and $\mathbb{E}[\cdot]$ is the *expected-value operator*.

The first Semivalued to appear in the literature was the *Shapley value* [42], which is now recognized as a fundamental concept in cooperative game theory due to its many desirable properties, see, e.g., [7]. Another prominent Semivalued is the *Banzhaf power index* [3], which has also been studied extensively. Those two Semivalueds are defined by the following β -functions:

$$\beta^{\text{Shapley}}(i) = \frac{1}{|N|} \quad \text{and} \quad \beta^{\text{Banzhaf}}(i) = \frac{\binom{|N|-1}{i}}{2^{|N|-1}}.$$

Interaction index: One possible way to interpret the *synergy* (or added value) that results from the interaction between players i and j is as follows: $S(i, j) = \nu(\{i, j\}) - \nu(\{i\}) - \nu(\{j\})$. One can also measure such synergy with respect to any particular coalition, $C \subseteq N$, as follows:

$$S(C, i, j) = \text{MC}(C, \{i, j\}) - \text{MC}(C, i) - \text{MC}(C, j),$$

where $\text{MC}(C, \{i, j\}) = \nu(C \cup \{i, j\}) - \nu(C)$. The *interaction index* of i and j , denoted by $I_{i,j}(N, \nu)$, is a weighted average of such synergy, taken over all coalitions in the game. The absolute value of $I_{i,j}(N, \nu)$ indicates the intensity of the interaction between the two players; greater values indicate greater intensity. In contrast, the sign of $I_{i,j}(N, \nu)$ reflects the nature of the influence that i and j have on one another: $I_{i,j}(N, \nu) < 0$ means they influence each other negatively; $I_{i,j}(N, \nu) > 0$ means they influence each other positively; $I_{i,j}(N, \nu) = 0$ means they either do not influence each other, or their influences cancel out.

By combining a Semivalued with the interaction index, we obtain a *Semivalued interaction index*, defined as follows:

Definition 3 Given a game, (N, ν) , and a discrete probability distribution, $\beta : \{0, \dots, |N| - 1\} \rightarrow [0, 1]$, $\sum_{0 \leq k < |N|} \beta(k) = 1$, the *Semivalued interaction index of a pair of players*, $i, j \in N$, is:

$$I_{i,j}^{\text{SEMI}}(N, \nu) = \sum_{0 \leq k \leq |N|-2} \beta(k) \mathbb{E}[S(X^k, i, j)], \quad (3)$$

where X^k is a coalition drawn uniformly at random from: $\{C : C \subseteq N \setminus \{i, j\} \wedge |C| = k\}$, and $\mathbb{E}[\cdot]$ is the expected-value operator.

The three Semivalue interaction indices that are widely studied in literature are presented in Table 1.

interaction index name	$\beta(l)$
Shapley-value interaction index [37, 17]	$\frac{1}{ V -1}$
Banzhaf-index interaction index [41]	$\frac{\binom{ V -2}{l}}{2^{ V -2}}$
Chaining interaction index [34]	$\frac{2(l+1)}{(n-1)(n-2)}$

Table 1. Values of β for the three main Semivalue interaction indices.

In addition to cooperative game theory, the interaction index has also been studied in various other fields, such as fuzzy systems, aggregation function theory, multi-criteria decision making, statistics and data analysis [33].

Cooperative games with coalition structure: A cooperative game can be viewed and analyzed from the *ex ante* perspective, where the agents have not yet decided on which coalitions to form. Conversely, the game may be analyzed from the *a priori* perspective, where the agents have already formed a specific coalition structure, $CS = \{C_1, \dots, C_m\}$. From this perspective, a *cooperative game with a coalition structure* is a tuple, (N, CS, ν) . Arguably, the most established extension of the Shapley value to such games is the *Owen value* [38]. Before explaining how it works, we need to first introduce the notion of a *quotient game*. In particular, given a cooperative game with a coalition structure, (N, CS, ν) , the corresponding quotient game, (CS, ν^Q) , is a game whose set of players is CS (i.e., every coalition in CS is considered to be a single player), and whose characteristic function is defined as follows:

$$\nu^Q(R) = \nu\left(\bigcup_{r \in R} C_r\right) \text{ for all } R \subseteq M,$$

where $M = \{1, \dots, m\}$ is the set of coalition numbers. For every $R \subseteq M$, we will use Q_R to denote $\bigcup_{r \in R} C_r$. For example, if $CS = \{C_1, C_2, C_3\} : C_1 = \{1, 2\}, C_2 = \{3, 4\}, C_3 = \{5\}$, then $Q_{\{1,3\}} = \{1, 2, 5\}$, and $\nu^Q(\{1, 3\}) = \nu(\{1, 2, 5\})$.

Having presented the quotient game, we are now ready to define the Owen value as follows:

Definition 4 Given a cooperative game with a coalition structure, (N, CS, ν) , the Owen value of a player $i \in C_x \in CS$ is:

$$OV_i(N, CS, \nu) = \sum_{R \subseteq M \setminus \{x\}} \frac{1}{|M| \binom{|M|-1}{|R|}} \sum_{C \subseteq C_x \setminus \{i\}} \frac{1}{|C_x| \binom{|C_x|-1}{|C|}} \text{MC}(Q_R \cup C, i). \quad (4)$$

One generalization of the Owen value that was recently introduced in the literature is *Coalitional Semivalues* [47], defined as follows:

Definition 5 Given a game, (N, CS, ν) , and a discrete probability distribution, $\beta : \{0, \dots, |M| - 1\} \rightarrow [0, 1]$, $\sum_{0 \leq k < |M|} \beta(k) = 1$,

the *Coalitional Semivalue* of a player $i \in C_x \in CS$ is:

$$CSEMI_i(N, CS, \nu) = \sum_{0 \leq k \leq |M| \setminus \{x\}} \beta(k) \sum_{0 \leq l < |C_x|} \alpha(l) \mathbb{E}[\text{MC}(Q_{T^k} \cup X^l, i)], \quad (5)$$

where T^k is a subset drawn from $\{R : R \subseteq M \setminus \{x\} \wedge |R| = k\}$ uniformly at random; X^l is a subset of size l drawn from $\{C : C \subseteq C_x \setminus \{i\} \wedge |C| = l\}$ uniformly at random; $\mathbb{E}[\cdot]$ is the expected-value operator; and $\alpha : \{0, \dots, |C_x| - 1\} \rightarrow [0, 1]$, $\sum_{l=0}^{|C_x|-1} \alpha(l) = 1$.

As shown in Table 2, by adopting the appropriate probability distributions, we obtain the Owen value [38] or any of its modifications proposed in the literature to date, namely: *Owen-Banzhaf value* [39], *symmetric coalitional Banzhaf value* [2], and *symmetric coalitional p-binomial Semivalues* [6].

Solution name	$\beta(k)$	$\alpha(l)$
Owen value [38]	$\frac{1}{ M -1}$	$\frac{1}{ C_x -1}$
Owen-Banzhaf value [39]	$\frac{\binom{ M -1}{k}}{2^{ M -1}}$	$\frac{\binom{ C_x -1}{l}}{2^{ C_x -1}}$
symmetric coalitional Banzhaf value [2]	$\frac{\binom{ M -1}{k}}{2^{ M -1}}$	$\frac{1}{ C_x }$
symmetric coalitional p-binomial Semivalue [6]	$p^k (1-p)^{ M -1-k}$ $p \in [0, 1]$	$\frac{1}{ C_x }$

Table 2. Values of α and β for the Owen value and its various extensions.

Interaction index on games with coalition structure: Finally we are ready to define the *Owen-value interaction index* for cooperative games with a coalition structure. Here, our definition is for the interaction between nodes belonging to different coalitions [53].

Definition 6 Given a cooperative game with a coalition structure, (N, CS, ν) , and two players, $i \in C_x \in CS$ and $j \in C_y \in CS$ such that $C_x \neq C_y$, the Owen-value interaction index between i and j is:

$$I_{i,j}^{OV}(N, CS, \nu) = \sum_{R \subseteq M \setminus \{x,y\}} \frac{1}{(|M|-1) \binom{|M|-2}{|R|}} \sum_{C \subseteq (C_x \cup C_y) \setminus \{i,j\}} \frac{S(Q_R \cup C, i, j)}{(|C_x| + |C_y| - 1) \binom{|C_x| + |C_y| - 2}{|C|}} \quad (6)$$

3 A NEW INTERACTION INDEX FOR NETWORKS

Inspired by the inter-links prediction problem, we construct a new node-similarity measure using three building blocks. The first block is the *interaction index* (to analyze pairs of nodes), the second block is the *Coalitional Semivalue* (to analyze nodes given a community structure), and the third block is the *k-steps group degree centrality* (to quantify the importance of subsets of nodes). To put the three pieces together, our first step is to introduce the following game.

Definition 7 A cooperative game with a coalition structure (played on a graph is a tuple, (G, CS, ν_G) , where G is a graph, CS is a

community structure, and $\nu_G : 2^{|V|} \rightarrow \mathbb{R}$ is a characteristic function defined over the graph G .

We use one such game, where the characteristic function is the k -steps group degree centrality, defined for all $k \in \{1, \dots, |V|\}$ and all $S \subseteq V$ as follows:

$$\nu_D^k(S) = \left| \left\{ v \in V : \min_{u \in S} d(u, v) \leq k \right\} \setminus S \right|.$$

The second step is to combine the Coalitional Semivalue with the interaction index, as shown below:

Definition 8 Given a cooperative game with a coalition structure, (N, CS, ν) , a discrete probability distribution, $\beta : \{0, \dots, |N| - 2\} \rightarrow [0, 1]$, $\sum_{0 \leq k \leq |N| - 2} \beta(k) = 1$, and two players, $i \in C_x \in CS$ and $j \in C_y \in CS$ such that $C_x \neq C_y$, the Coalitional-Semivalue interaction index between i and j is:

$$I_{i,j}^{CSEMI}(N, CS, \nu) = \sum_{0 \leq k \leq |M \setminus \{C_x, C_y\}|} \beta(k) \sum_{0 \leq l \leq |C_x \cup C_y \setminus \{i, j\}|} \alpha(l) \mathbb{E}[S(Q_{T^k} \cup X^l, i, j)], \quad (7)$$

where T^k is a subset of size k drawn from $\{R : R \subseteq M \setminus \{x, y\} \wedge |R| = k\}$ uniformly at random; X^l is a subset of size l drawn from $\{C : C \subseteq C_x \cup C_y \setminus \{i, j\} \wedge |C| = l\}$ uniformly at random; $\mathbb{E}[\cdot]$ is the expected-value operator; $\alpha : \{0, \dots, |C_x \cup C_y \setminus \{i, j\}|\} \rightarrow [0, 1]$, $\sum_{l=0}^{|C_x \cup C_y \setminus \{i, j\}|} \alpha(l) = 1$.

This is a natural extension of Owen-value interaction index that is in line with the definition of Coalitional Semivalues. For instance, by setting $\beta(k) = \frac{1}{|M|-1}$ and $\alpha(l) = \frac{1}{|C_x|-1}$, we obtain the Owen-value interaction index.

Now, we are ready to introduce our new node-similarity measure:

Definition 9 The Coalitional-Semivalue similarity measure between $v \in C_x$ and $u \in C_y$ in graph $G(V, E)$ with community structure CS is defined as:

$$I_{u,v}^{CSEMI}(V, CS, \nu_D^k).$$

Many standard measures evaluate the similarity between two nodes by quantifying the intersection of their spheres of influence. In contrast, the main advantage of our measure is that the intersection is evaluated in the context of the exponential number of subsets of communities and nodes in the network, which may allows us to compute similarity more accurately. One potential drawback of our approach is its potentially-high computational complexity, due to the exponential number of subsets. However, in the following section we develop the closed-form formula for the k -steps Coalitional-Semivalue similarity measure which allows us to compute it in polynomial time.

4 COMPUTATIONAL ANALYSIS

In this section, we circumvent the main potential obstacle that may hamper the application of the Coalitional-Semivalue interaction index—the computational complexity. In more detail, Equation (7) requires iterating over an exponential number of subsets of V . However, building upon a combinatorial and probabilistic analysis, we will develop two polynomial algorithms: one for the Coalitional-Semivalues interaction index, which runs in $O(|V|^3)$ time, and the other is for a special case of this index, namely the Owen-value interaction index, which runs in just $O(|V|)$ time.

To this end, let CSDEGREEII denote the problem of calculating $I_{u,v}^{CSEMI}(V, CS, \nu_D^k)$, where CS is a community structure, ν_D^k is the k -steps degree centrality, and $u, v \in V$. The main theoretical result in this paper is as follows:

Theorem 1 CSDEGREEII is in P.

We note that the above theorem fills a gap in the literature, as highlighted in Table 7 (see Section 7). Before presenting the proof, we first need some additional notation. For every node $v \in V$, let $N_k(v)$ denote the set of “neighbors” reachable from v with at most k steps, and let $deg_k(v)$ denote the number of such nodes. More formally, we have: $N_k(v) = \{u \in V : d(v, u) \leq k \wedge v \neq u\}$ and $deg_k(v) = |N_k(v)|$. We extend this notation to sets of nodes. That is, $N_k(C) = \bigcup_{v \in C} N_k(v) \setminus C$ and $deg_k(C) = |N_k(C)|$. Moreover, for any given node, $v \in C_x \in CS$, we denote the set of adjacent communities as $N_k^{CS}(v) = \{C_y \in CS \setminus C_x : C_y \cap N_k(v) \neq \emptyset\}$, the inter-community degree as $deg_k^{CS}(v) = |N_k^{CS}(v)|$, the set of neighbors within some community $C_y \in CS$ as $N_k^y(v) = N_k(v) \cap C_y$, and the corresponding intra-community degree as $deg_k^y(v) = |N_k^y(v)|$. These can be extended to two communities as follows: $N_k^{y,z}(v) = N_k(v) \cap (C_y \cup C_z)$ and $deg_k^{y,z}(v) = |N_k^{y,z}(v)|$.

In our proof we follow the line of our earlier work [45], where we developed an algorithm to computed the Shapley value-based interaction index was proposed. In this work, we will extend the proof from [45] to take into consideration both the community structure the Owen value-based interaction index, which is much more complex than its Shapley value-based counterpart.

Proof: First of all, let us focus on Equation (7). More specifically, for each pair of nodes $v, u \in V$ such that: $v \in C_i \in CS$ and $u \in C_j \in CS$ and $C_i \neq C_j$, we will show how to compute $\mathbb{E}[S(Q_{T^k} \cup X^l, u, v)]$ —the expected value of their synergy with respect to the random set $Q_{T^k} \cup X^l$. Recall that T^k is drawn uniformly from the set $\{R : R \subseteq M \setminus \{i, j\} \wedge |R| = k\}$, and X^l is drawn uniformly from the set $\{C : C \subseteq C_i \cup C_j \setminus \{u, v\} \wedge |C| = l\}$. Also recall that Q_R denotes $\bigcup_{r \in R} C_r$. Now if we denote $R^{k,l} = Q_{T^k} \cup X^l$, then:

$$\begin{aligned} \mathbb{E}[S(R^{k,l}, u, v)] &= \mathbb{E}[\text{MC}(R^{k,l}, \{u, v\})] - \mathbb{E}[\text{MC}(R^{k,l}, u)] - \mathbb{E}[\text{MC}(R^{k,l}, v)] \\ &= \mathbb{E}[\text{MC}(R^{k,l}, u)] + \mathbb{E}[\text{MC}(R^{k,l}, v)] - \mathbb{E}[\text{MC}(R^{k,l}, u \cap v)] \\ &\quad - \mathbb{E}[\text{MC}(R^{k,l}, u)] - \mathbb{E}[\text{MC}(R^{k,l}, v)] \\ &= - \mathbb{E}[\text{MC}(R^{k,l}, u \cap v)], \end{aligned}$$

where $\text{MC}(R^{k,l}, u \cap v)$ is what we call the “common” contribution of two nodes v and u , which is illustrated in Figure 1, and defined as follows:⁵

$$\begin{aligned} \mathbb{E}[\text{MC}(R^{k,l}, u \cap v)] &= \mathbb{E}[\text{MC}(R^{k,l}, u)] + \mathbb{E}[\text{MC}(R^{k,l}, v)] \\ &\quad - \mathbb{E}[\text{MC}(R^{k,l} \cup \{u\}, v)] - \mathbb{E}[\text{MC}(R^{k,l} \cup \{v\}, u)]. \end{aligned}$$

Now given the function ν_D^k , the pair of nodes $v, u \in V$ can make a positive common contribution to the set of nodes $R^{k,l}$ only through some node from the intersection of their neighborhoods, i.e., some node $n \in N_k(v) \cap N_k(u)$. Intuitively, this happens when such a node n is not under the influence of the set $R^{k,l}$ but is under the influence of $R^{k,l} \cup \{u, v\}$.

⁵ Here, we do not mean to take the intersection of nodes u and v , as this would be incorrect. Instead, for notation convenience, we write $u \cap v$ when referring to the intersection between the contributions of u and v .

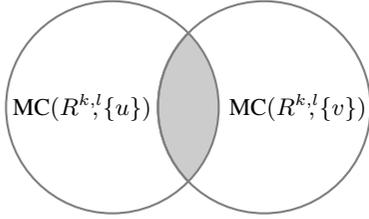


Figure 1. An illustration of $MC(R^{k,l}, u \cap v)$.

We formalize the above observation by introducing the Bernoulli random variable indicating whether nodes v, u make a positive common contribution to the set $R^{k,l}$ through a node $n \in N_k(v) \cap N_k(u)$:

$$\mathbb{E}[B_{k,l,v,u,n}^+] = P[(N_k(n) \cup \{n\}) \cap R^{k,l} = \emptyset]. \quad (8)$$

On the other hand, the pair $v, u \in V$ can make a *negative* common contribution to the set of nodes $R^{k,l}$ through v or u . This happens when either of those two nodes is under the influence of $R^{k,l}$ but not under the influence of $R^{k,l} \cup \{v, u\}$.⁶ Note that when analyzing the common contribution, we only consider nodes in $N_k(v) \cap N_k(u)$. As such, from the negative-contribution perspective, we only consider cases where $v \in N_k(u)$ or $u \in N_k(v)$.

We formalize the above observation by introducing two Bernoulli random variables indicating whether nodes v, u make a negative common contribution to $R^{k,l}$ through the node v or u is defined as:

$$\mathbb{E}[B_{k,l,v}^-] = P[(N_k(v)) \cap R^{k,l} \neq \emptyset], \quad (9)$$

$$\mathbb{E}[B_{k,l,u}^-] = P[(N_k(u)) \cap R^{k,l} \neq \emptyset]. \quad (10)$$

Now, we will develop an exact formula for the equations (8), (9) and (10). We start with a positive contribution. The important observation is that the set $R^{k,l}$ is drawn from the sample space Ω , where $|\Omega| = \binom{|M|-2}{k} \binom{|C_i|+|C_j|-2}{l}$. Having this in mind, we denote $P[(N_k(n) \cup \{n\}) \cap R^{k,l} = \emptyset]$ by P^+ and we get:

$$P^+ = \begin{cases} \frac{\binom{|M|-1-deg_k^{CS}(n)}{k} \binom{|C_i|+|C_j|-deg_k^{i,j}(n)}{l}}{\binom{|M|-2}{k} \binom{|C_i|+|C_j|-2}{l}} & \text{if } n \notin C_i \cup C_j \\ \frac{\binom{|M|-1-deg_k^{CS}(u)}{k} \binom{|C_i|+|C_j|-1-deg_k^{i,j}(n)}{l}}{\binom{|M|-2}{k} \binom{|C_i|+|C_j|-2}{l}} & \text{if } n \in C_i \cup C_j \end{cases} \quad (11)$$

Now if $n \notin C_i \cup C_j$, the above formula has the following combinatorial interpretation. The random set $R^{k,l}$ can contain any community from CS except for C_i and C_j ; there are $|M| - 2$ such communities, where $M = \{1, \dots, m\}$ is the set of community numbers. In order to satisfy the condition $N_k(v) \cap T^k \neq \emptyset$, from all communities in $CS \setminus \{C_i, C_j\}$ we can draw all those that are not in the scope of n . There are $|M| - 2 - deg_k^{CS}(n)$ such communities. However, two additional facts also play an important role: the fact that the community containing n should not be in $R^{k,l}$, and the fact that $R^{k,l}$ can contain any community from CS except for C_i and C_j . Taking this into account, the final number of communities is: $|M| - 1 - deg_k^{CS}(n)$. Thus, the probability of choosing a set $R^{k,l}$ satisfying our condition $N_k(v) \cap T^k \neq \emptyset$ is exactly: $\binom{|M|-1-deg_k^{CS}(n)}{k} / \binom{|M|-2}{k}$.

Next, we show how to satisfy the condition that $N_k(v) \cap X^l \neq \emptyset$.

To this end, from the set $(C_i \cup C_j) \setminus \{v, u\}$ we need to exclude those nodes that are in the scope of n . There are $|C_i \cap C_j| - 2 - deg_k^{i,j}(n)$ such nodes. However, taking into account that $v, u \in N_k^{i,j}(n)$, the probability of choosing a set $R^{k,l}$ satisfying the condition $N_k(v) \cap X^l \neq \emptyset$ is exactly: $\binom{|C_i|+|C_j|-deg_k^{i,j}(n)}{l} / \binom{|C_i|+|C_j|-2}{l}$.

In order to compute the negative contribution, we consider the complementary event: $P^- = 1 - P[(N_k(v)) \cap R^{k,l} = \emptyset]$. Using the same combinatorial argument as the one used to compute P^+ , we get:

$$P^- = 1 - \frac{\binom{|M|-1-deg_k^{CS}(v)}{k} \binom{|C_i|+|C_j|-1-deg_k^{i,j}(v)}{l}}{\binom{|M|-2}{k} \binom{|C_i|+|C_j|-2}{l}}, \quad (12)$$

The formula combining equations (8) and (9) and its analytic form given in equations (11) and (12) is:

$$\begin{aligned} \mathbb{E}[S(R^{k,l}, u, v)] &= \mathbb{E}[B_{k,l,v}^-] + \mathbb{E}[B_{k,l,u}^-] - \sum_{n \in N_k(v) \cap N_k(u)} \mathbb{E}[B_{k,l,v,u,n}^+] \\ &= - \sum_{n \in (N_k(v) \cap N_k(u)) \setminus (C_i \cup C_j)} \left(\frac{\binom{|M|-1-deg_k^{CS}(n)}{k} \binom{|C_i|+|C_j|-deg_k^{i,j}(n)}{l}}{\binom{|M|-2}{k} \binom{|C_i|+|C_j|-2}{l}} \right) \\ &\quad - \sum_{n \in (N_k(v) \cap N_k(u)) \cap (C_i \cup C_j)} \left(\frac{\binom{|M|-1-deg_k^{CS}(n)}{k} \binom{|C_i|+|C_j|-1-deg_k^{i,j}(n)}{l}}{\binom{|M|-2}{k} \binom{|C_i|+|C_j|-2}{l}} \right) \\ &\quad + \sum_{n \in (\{v\} \cap N_k(u)) \cup (\{u\} \cap N_k(v))} \left(1 - \frac{\binom{|M|-1-deg_k^{CS}(n)}{k} \binom{|C_i|+|C_j|-1-deg_k^{i,j}(n)}{l}}{\binom{|M|-2}{k} \binom{|C_i|+|C_j|-2}{l}} \right) \end{aligned} \quad (13)$$

Notice that the sets $N_k^{CS}(u)$, $N_k^{i,j}(u)$ and $N_k(u)$ are easily computable in polynomial time. Based on this, the closed-form formula (13) together with Equation (7) prove that CSDEGREEII is solvable in polynomial time, i.e., it belongs to the class P.

Finally, note that in our proof we omitted the case when u and v belong to the same community. Although, in this case, the reasoning slightly differs from the above, it results in almost the same formula. Nevertheless, such a case is not interesting from the perspective of inter-links recommendation. \square

Now, let us denote by OVDEGREEII the problem of calculating $I_{u,v}^{OV}(V, CS, \nu_D^k, \cdot)$ where CS is a community structure, ν_D^k is the k -steps degree centrality, and $u, v \in V$. Then, the following corollary immediately follows from Theorem 1:

Corollary 1 OVDEGREEII is in P.

Building upon the above theoretical results, we will propose in the next section two algorithms; one solves CSDEGREEII in $O(|V|^3)$ time; the other solves OVDEGREEII in just $O(|V|)$ time, after some preprocessing stage that requires $O(|V|^2)$ time.

5 ALGORITHMS

In this section, we use Equation (13) to develop a polynomial time algorithm for computing the k -steps Coalitional-Semivalue interaction index. In particular, Algorithm 1 computes the k -steps Coalitional-Semivalue interaction index for a given pair of nodes $u, v \in V$ in the graph G . This algorithm is basically an implementation of Equation (7), whereby the expected value operator $\mathbb{E}[S(R^{k,l}, u, v)]$ is computed using Equation (13).

⁶ In some work the definition of $\nu_D^k(C)$ also counts the nodes from C [35]. However, in this paper we follow the convention that the influence of C affects only nodes from outside C .

Algorithm 1: Computing the k-steps Coalitional-Semivalue interaction index

Input: Graph $G(V, E)$, functions β and α , community structure CS , nodes $v, u \in V$ where $v \in C_i \in CS, u \in C_j \in CS$

Data: for $u \in V$:

- $N_k(u)$ —the set of k-neighbors
- $N_k^{CS}(u)$ —the set of adjacent communities
- $N_k^{i,j}(u)$ —the set of adjacent nodes within $C_i \cup C_j$

Output: $I_{u,v}^{CSEMI}$ k-steps coalitional-Semivalue interaction index

```

1  $I_{u,v}^{CSEMI} \leftarrow 0$ ;
2 for  $k \leftarrow 0$  to  $|M| - 2$  do
3   for  $l \leftarrow 0$  to  $|C_i \cup C_j| - 2$  do
4      $S \leftarrow 0$ ;
5     foreach  $n \in (N_k(v) \cap N_k(u)) \setminus (C_i \cup C_j)$  do
6        $S \leftarrow S - \frac{\binom{|M|-1-deg_k^{CS}(n)}{k} \binom{|C_i|+|C_j|-deg_k^{i,j}(n)}{l}}{\binom{|M|-2}{k} \binom{|C_i|+|C_j|-2}{l}}$ 
7     foreach  $n \in (N_k(v) \cap N_k(u)) \cap (C_i \cup C_j)$  do
8        $S \leftarrow S - \frac{\binom{|M|-1-deg_k^{CS}(n)}{k} \binom{|C_i|+|C_j|-1-deg_k^{i,j}(n)}{l}}{\binom{|M|-2}{k} \binom{|C_i|+|C_j|-2}{l}}$ 
9     if  $v \in N_k(u)$  then
10       $S \leftarrow$ 
11       $S + 1 - \frac{\binom{|M|-1-deg_k^{CS}(v)}{k} \binom{|C_i|+|C_j|-1-deg_k^{i,j}(v)}{l}}{\binom{|M|-2}{k} \binom{|C_i|+|C_j|-2}{l}}$ 
12       $S \leftarrow$ 
13       $S + 1 - \frac{\binom{|M|-1-deg_k^{CS}(u)}{k} \binom{|C_i|+|C_j|-1-deg_k^{i,j}(u)}{l}}{\binom{|M|-2}{k} \binom{|C_i|+|C_j|-2}{l}}$ 
14       $I_{u,v}^{CSEMI} \leftarrow I_{u,v}^{CSEMI} + \beta(k)\alpha(l)S$ 

```

It is easy to see that Algorithm 1 runs in $O(|V|^3)$ time. Note that the algorithm requires a preprocessing stage in which the sets $N_k^{CS}(u)$, $N_k^{i,j}(u)$ and $N_k(u)$ are computed. Let us analyze the time required to perform this preprocessing stage. For each node $n \in V$, the sets $N_k^{CS}(u)$ and $N_k(u)$ can be computed using breadth-first search in $O(|V|(|V| + |E|))$ time. Furthermore, we can store all coalition values, i.e., store $\nu(C), \forall C \subseteq N$, using $O(|V|^2)$ space. Next, for the pair of communities C_i and C_j we can compute the set $N_k^{i,j}(n)$ in $O(|V|^2)$ time. As can be seen, compared to the time required to run Algorithm 1, the preprocessing stage takes negligible time.

Although $O(V^3)$ —the time required to run Algorithm 1—is very fast compared to a naive (exponential-time) algorithm, it is still not fast enough to be applied for link prediction in large networks. With this in mind, we now present an even faster algorithm to compute k-steps Owen-value interaction index; see Algorithm 2. Specifically, this algorithm runs in $O(|V|)$ time, and requires the aforementioned preprocessing stage to compute the sets $N_k^{CS}(u)$, $N_k^{i,j}(u)$ and $N_k(u)$. This improvement allows us to compute the similarity between each pair of nodes (not just a single pair) in $O(|V|^3)$ time. Thus, the entire procedure of link prediction also requires $O(|V|^3)$ time.

6 Empirical Evaluation

In this section, we empirically demonstrate the effectiveness of our node-similarity measure in detecting links across communities. Specifically, in our experiments we use the Owen value-based variant

Algorithm 2: Computing the k-steps Owen-value interaction index

Input: Graph $G(V, E)$, community structure CS , nodes $v, u \in V$ where $v \in C_i \in CS, u \in C_j \in CS$

Data: for $u \in V$:

- $N_k(u)$ —the set of k-neighbors
- $N_k^{CS}(u)$ —the set of adjacent communities
- $N_k^{i,j}(u)$ —the set of adjacent nodes within $C_i \cup C_j$

Output: $I_{u,v}^{OV}$ k-steps coalitional-Semivalue interaction index

```

1  $I_{u,v}^{OV} \leftarrow 0$ ;
2 foreach  $n \in (N_k(v) \cap N_k(u)) \setminus (C_i \cup C_j)$  do
3    $I_{u,v}^{OV} \leftarrow I_{u,v}^{OV} - \frac{1}{(deg_k^{CS}(n))(deg_k^{i,j}(n))}$ 
4 foreach  $n \in (N_k(v) \cap N_k(u)) \cap (C_i \cup C_j)$  do
5    $I_{u,v}^{OV} \leftarrow I_{u,v}^{OV} - \frac{1}{(deg_k^{CS}(n))(deg_k^{i,j}(n)-1)}$ 
6 if  $v \in N_k(u)$  then
7    $I_{u,v}^{OV} \leftarrow I_{u,v}^{OV} + 1 - \frac{1}{(deg_k^{CS}(v))(deg_k^{i,j}(v)-1)}$ 
8    $I_{u,v}^{OV} \leftarrow I_{u,v}^{OV} + 1 - \frac{1}{(deg_k^{CS}(u))(deg_k^{i,j}(u)-1)}$ 

```

of our measure, as it can be computed efficiently using Algorithm 2. We compare our measure against six local similarities measures outlined in Table 3; these are arguably the most efficient solutions to the local link-prediction problem in the literature [32, 45].

Similarity Name	Measure
Common Neighbors (CN)	$S_{u,v}^{CN} = N_k(u) \cap N_k(v)$
Salton Index (SI)	$S_{u,v}^{SI} = \frac{N_k(u) \cap N_k(v)}{\sqrt{N_k(u) \times N_k(v)}}$
Jaccard Index (JI)	$S_{u,v}^{JI} = \frac{N_k(u) \cap N_k(v)}{N_k(u) \cup N_k(v)}$
Adamic-Adar Index (AA)	$S_{u,v}^{AA} = \sum_{n \in N_k(u) \cap N_k(v)} \frac{1}{\log N_k(n)}$
Resource Allocation (RA)	$S_{u,v}^{RA} = \sum_{n \in N_k(u) \cap N_k(v)} \frac{1}{N_k(n)}$
Shapley-value interaction index (SV)	$S_{u,v}^{SV} = I_{i,j}^{SV}(V, \nu_D^k)$

Table 3. The six local node-similarity measures used in our experiments.

We evaluate the effectiveness of each node-similarity measure using a standard procedure from the literature on link-prediction. In particular, we compute the similarity of each pair of disconnected nodes, $v, u \in V : (v, u) \notin E$, belonging to two different communities, $C_i, C_j \in CS$ where $v \in C_i, u \in C_j$. After that, links are proposed between the most similar such pairs of nodes. Since in this paper we focus on finding the most accurate predictions based only on local information, we set $k = 1$ in all our experiments. Each such experiment is conducted as follows: given a graph $G = (V, E)$ with a community structure CS , we create a new graph $G' = (V, E')$, which is similar to G and with the same CS but where 10% or 20% of inter-edges are removed at random.⁷ Then, we compute the similarity of each disconnected pair of nodes from different communities

⁷ We also conducted experiments in which 30% and 40% of inter-edges were removed. The effectiveness of all measures was reduced proportionally.

in G' . In order to evaluate the results, we use the *Area Under the Curve* (AUC) measure [32]. The whole process is repeated 100 times and the average AUC is taken.

In more detail, the AUC is computed using the Mann-Whitney U test [22]. To this end, let R be the set of all disconnected pairs of nodes from different communities in G' . This set can be divided into two disjoint sets: the set of “missing” links, denoted by M (i.e., $M = E \setminus E'$), and the set of “non-existing” links, denoted by N (i.e., $N = (V \times V) \setminus E$). Let $\mathbf{n} = |M||N|$ be the number of all comparisons between the missing links and the non-existing links. Furthermore, let \mathbf{n}' be the number of such comparisons in which the missing link is ranked higher than the non-existing link. Finally, let \mathbf{n}'' be the number of such comparisons in which both links are ranked the same. Then, the AUC is computed as follows:

$$AUC = \frac{\mathbf{n}' + \frac{\mathbf{n}''}{2}}{\mathbf{n}}.$$

With this formula, if AUC equals 1 then all missing links are ranked higher than the non-existing links; this is the best possible ranking, where $\mathbf{n}' = \mathbf{n}$ and $\mathbf{n}'' = 0$. On the other extreme, if AUC equals 0 then none of the missing links is ranked higher than, or even the same as, any of the non-existing links; this is the worst possible ranking, where $\mathbf{n}' = 0$ and $\mathbf{n}'' = 0$. A completely random ranking falls between the two extremes, with an expected AUC of 0.5.

We study 8 widely-used real-life networks: Tribes [12], Taro [21], Zachary [54], Terrorists [24], Surfers [30], Polbooks [1], Football [15] and Jazz [16]. Table 4 specifies the sizes of these eight networks, as well as the sizes of the community structures therein.⁸ For each of them, we report the results for the community structure identified by the multilevel community-detection algorithm [51]. We also experimented with other community-detection algorithms, such as *Walktrap* [40], *Fastgreedy* [9], and *Girvan-Newman* [15]; they produced almost the same community structures as the multilevel algorithm. As such, the choice of the community-detection algorithm had a negligible impact on our results.

Network	$ V $	$ E $	$ CS $	Network	$ V $	$ E $	$ CS $
Tribes	16	58	3	Surfers	43	336	2
Taro	22	39	5	Polbooks	105	441	4
Zachary	34	78	4	Football	115	613	10
Terrorists	64	243	5	Jazz	198	2742	4

Table 4. The sizes of the networks and their community structures used in the experiments.

Tables 5 and 6 present the results for our Owen value-based measure, as well the other local link-prediction measures from Table 3. As can be seen, in the experiments where 10% of inter-links were removed (i.e., Table 5), our measure outperforms all the other measures. As for the experiments where 20% of inter-links were removed (i.e., Table 6), our measure also outperforms the other alternatives for all networks except for the *Football* network.

The results in Tables 5 and 6 demonstrate how detecting inter-links locally can be a rather challenging task. For instance, given the *Football* network, all local node-similarity measures are biased; they produce results that are worse than even a completely random classifier whose AUC is expected to be 0.5. Likewise, given the *Taro* network, all measures are either worse, or slightly better than, the

⁸ The datasets for the eight networks were downloaded from the link: <http://www-personal.umich.edu/~mejn/netdata/> as well as the link: <http://konect.uni-koblenz.de/networks/>.

Network	OV	SV	RA	CN	AA	JI	SI
Tribes	0.754	0.438	0.579	0.502	0.581	0.633	0.658
Taro	0.573	0.483	0.394	0.516	0.550	0.571	0.483
Zachary	0.703	0.628	0.684	0.533	0.657	0.573	0.617
Terrorists	0.834	0.774	0.778	0.744	0.795	0.782	0.748
Surfers	0.881	0.731	0.770	0.746	0.765	0.797	0.791
Polbooks	0.806	0.783	0.789	0.743	0.785	0.755	0.772
Football	0.403	0.346	0.365	0.314	0.336	0.373	0.388
Jazz	0.962	0.918	0.928	0.927	0.902	0.919	0.918

Table 5. The area under curve (AUC) for our measure (OV) as well as the six measures from Table 3, given 8 real-life networks in which 10% of inter-links were removed.

Network	OV	SV	RA	CN	AA	JI	SI
Tribes	0.729	0.421	0.584	0.493	0.569	0.622	0.641
Taro	0.525	0.431	0.439	0.358	0.441	0.505	0.510
Zachary	0.644	0.584	0.622	0.519	0.616	0.511	0.548
Terrorists	0.819	0.771	0.782	0.746	0.785	0.759	0.715
Surfers	0.867	0.735	0.766	0.730	0.762	0.766	0.778
Polbooks	0.780	0.739	0.748	0.746	0.747	0.752	0.748
Football	0.351	0.313	0.334	0.300	0.331	0.353	0.365
Jazz	0.966	0.919	0.928	0.905	0.918	0.921	0.924

Table 6. The area under curve (AUC) for our measure (OV) as well as the six measures from Table 3, given 8 real-life networks in which 20% of inter-links were removed.

random one. Nevertheless, for the remaining networks, all measures outperform the random one with very few exceptions.

Finally, we show in Figure 2 how the runtime of our Algorithm 2 grows with the size of the network. Specifically, the dotted line represents the time needed for the preprocessing stage, whereas the solid line represents the time needed for computing the 1-step Owen Value-based interaction index between each pair of nodes in the network. As can be seen, even for 5000 nodes, our algorithm takes less than one minute.

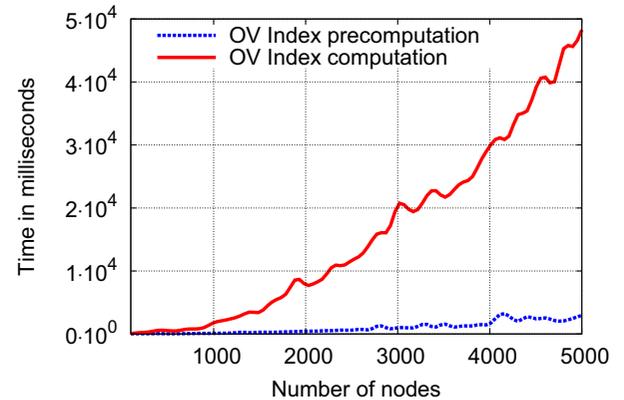


Figure 2. The performance of 1-step Owen value interaction index.

We conclude this section with some negative results. Specifically, when using our measure for *quasi-local* (rather than local) link prediction (i.e., when $k > 1$) the performance of our algorithm drops considerably in terms of AUC for all the networks in our experiments. We believe this comes from the expression $deg_k^{CS}(n)$. In particular, even for a small k , every node in the network can reach many different communities, which can negatively influence the performance. This is because in such a case the differences between the nodes diminish, and nodes become indistinguishable by our measure.

7 RELATED WORK

Our contribution falls at the intersection of (i) positive computational results in cooperative game theory and (ii) efficient link prediction in graph theory. In this section, we briefly discuss both areas of research.

Starting with cooperative game theory, most solution concepts are NP-hard [7]. However, for cooperative games described over networks there is a growing body of literature with various positive results. In particular, when either *group degree* centrality, *group closeness* centrality, or *group betweenness* centrality [13] is used as a characteristic function, it was shown that the Shapley value can be computed in time polynomial in the network size [35, 46]. Moreover, it was proven that the problem of computing any Semivalues—parametrized by any polynomial time computable discrete probability distribution—belongs to the class P for degree and closeness centralities [44], as well as betweenness centrality [46].

Regarding games with a coalition structure, there are positive results about *degree* and *closeness* centralities. In more detail, for both of them we can compute the Owen value and the Coalitional Semivalue in polynomial time [47, 48]. However, it is still an open question whether the same holds of *r betweenness*.

Other positive results can be found on the computation of the interaction index on graphs. More specifically, it was shown that we can compute efficiently the Shapley-value and Semivalue interaction indices with degree centrality [45].

The results that are most closely related to our work are presented in Table 7. The abbreviations SVDEGREE, SDEGREE, OVDEGREE and CSDEGREE stand for computing the Shapley value-, the Semivalue-, the Owen value- and the Coalitional value-based degree centrality, respectively. Furthermore, SVDEGREEII, SDEGREEII, OVDEGREEII and CSDEGREEII are analogous problems related to computing the interaction indices.

Computational result	Algorithm complexity
SVDEGREE is in P	$O(V + E)$ [35]
SDEGREE is in P	$O(V ^2)$ [44]
OVDEGREE is in P	$O(V + E)$ [47]
CSDEGREE is in P	$O(V ^3)$ [47]
SVDEGREEII is in P	$O(V)^*$ [45]
SDEGREEII is in P	$O(V ^2)$ [45]
OVDEGREEII is in P	$O(V)^*$ [this paper]
CSDEGREEII is in P	$O(V ^3)$ [this paper]

(*) some precomputation is required.

Table 7. Computational complexity results for degree centrality and coalitional games played on graphs.

The second body of literature that is strongly related to our work is that on link prediction. Here, we focus on methods based on node-similarity measures [32]. Generally, one can distinguish between three groups of link prediction algorithm: *local*, *quasi-local* and *global*. In more detail, *global algorithms* consider the entire network, which is prohibitive in for large networks. To date, the most efficient algorithm in this group is *Random Walk with Restart* [49], which is based on PageRank [4]. In contrast, *quasi-local algorithms* try to strike a balance between prediction runtime and efficiency. The most effective algorithms here are *Local Random Walk* and *Superposed Random Walk* [31]. Finally, *local algorithms* predict a link between any pair of nodes based solely on the direct neighborhood of those nodes (see Table 3). In practice, these are the only algorithms that can be applied to large networks, e.g., with millions of nodes.

To the best of our knowledge, the two best approaches in this group are: (i) the *Resource Allocation* approach [55], which is inspired by the resource allocation dynamics on complex networks; and (ii) the *Shapley-value interaction index* [45] approach, which is rooted in cooperative game theory.

Some authors have already tried to increase the accuracy of link prediction by taking advantage of the community structure of a network [43, 50]. While they managed to enhance the prediction performance by adding an extra score to nodes from the same community, such an approach method seems to have little value in our application as we are only interested in predicting connections between different communities.

In addition to the methods that are based on node similarity, link prediction can also be carried out based on maximizing likelihood [8, 20], or based on probabilistic models [26, 25]. These methods are computationally complex and are out of scope of this paper.

8 SUMMARY AND FUTURE WORK

In this paper, we proposed a new local node-similarity measure for networks with a community structure. We empirically demonstrated its effectiveness as a solution to the problem of detecting links *between* (rather than *within*) communities. Our measure outperforms other local node-similarity measure from the literature, since it is the first one designed specifically to detect links between heterogeneous nodes, rather than homogeneous ones as is the case with the other measures. Importantly, the Owen value-based variant of our measure can be computed very efficiently; it requires $O(|V|)$ time, after a pre-processing stage that requires $O(|V||E|)$ time. Interestingly, despite the inherent complexity of our measure (which comes from the complexity of the Owen value), link prediction using our algorithm *takes the same time* as the fastest alternative from the literature.

There are several directions for future work. Firstly, while we showed in this paper that the problem OVDEGREEII is in P, it would be interesting to verify whether the problems OVCLOSENESSII and OVBETWEENNESSII are also in P. It would also be interesting to study the similarity measures that correspond to the aforementioned problems, and to evaluate their effectiveness as node-similarity measures for inter-link prediction.

Secondly, since our measure is only restricted to non-overlapping communities, another interesting direction would be to extend our measure to graphs with overlapping communities [27]. Recently, an approach to measure the power of individual nodes in such networks was proposed [48]. In more detail, the authors defined a cooperative game with overlapping coalitions on a graph, and used a game-theoretic concept called the *Configuration value* to compute the power of an individual node. It is an open question whether CVDEGREE and CVDEGREEII are in P, where CVDEGREE stands for Configuration value-based degree centrality, and CVDEGREEII for Configuration-value interaction index.

Finally, it would be worthwhile to introduce a graph-related axiomatization of our similarity measure, following a similar approach to that with which the interaction indices was axiomatized based on concepts from cooperative games.

ACKNOWLEDGEMENTS

Piotr Szczepański was funded by the Polish National Science Centre based on the decision DEC-2013/09/N/ST6/04095. Tomasz Michalak and Michael Wooldridge were supported by the European Research Council under Advanced Grant 291528 (“RACE”).

REFERENCES

- [1] L. A. Adamic and N. Glance, 'The political blogosphere and the 2004 u.s. election: Divided they blog', in *Proceedings of the 3rd International Workshop on Link Discovery*, pp. 36–43, (2005).
- [2] R. Amer, F. Carreras, and J. M. Giménez, 'The modified banzhaf value for games with coalition structure: an axiomatic characterization', *Mathematical Social Sciences*, **43**(1), 45–54, (2002).
- [3] J. F. Banzhaf, 'Weighted Voting Doesn't Work: A Mathematical Analysis', *Rutgers Law Review*, **19**, 317–343, (1965).
- [4] S. Brin and L. Page, 'The anatomy of a large-scale hypertextual web search engine', *Comput. Netw. ISDN Syst.*, **30**(1–7), 107–117, (1998).
- [5] C. V. Cannistraci, G. Alanis-Lobato, and T. Ravasi, 'Minimum curvilinearity to enhance topological prediction of protein interactions by network embedding', *Bioinformatics*, **29**(13), 199–209, (2013).
- [6] F. Carreras and M. Puente, 'Symmetric coalitional binomial semivalues', *Group Decision and Negotiation*, **21**(5), 637–662, (2012).
- [7] G. Chalkiadakis, E. Elkind, and M. Wooldridge, *Computational aspects of cooperative game theory*, Morgan & Claypool Publishers, 2011.
- [8] A. Clauset, C. Moore, and M. E.J. Newman, 'Hierarchical structure and the prediction of missing links in networks', *Nature*, **453**(7191), 98–101, (2008).
- [9] M. E. J. Clauset, A. Newman and C. Moore, 'Finding community structure in very large networks', *Phys. Rev. E*, **70**, 066111, (2004).
- [10] S.F. Crone and D. Soopramanien, 'Predicting customer online shopping adoption—an evaluation of data mining and market modelling approaches', in *Proceedings of the 2005 International Conference on Data Mining*, pp. 215–221, (2005).
- [11] P. Dubey, A. Neyman, and R. J. Weber, 'Value Theory Without Efficiency', *Mathematics of Operations Research*, **6**, 122–128, (1981).
- [12] Kenneth E., 'Cultures of the central highlands', *Southwestern J. of Anthropology*, **10**(1), 1–43, (1954).
- [13] M. G. Everett and S. P. Borgatti, 'The centrality of groups and classes', *Journal of Mathematical Sociology*, **23**(3), 181–201, (1999).
- [14] L.C. Freeman, 'Centrality in social networks: Conceptual clarification', *Social Networks*, **1**(3), 215–239, (1979).
- [15] M. Girvan and M. E. J. Newman, 'Community structure in social and biological networks', in *Proc. of NAS*, **99**(12), 7821–7826, (2002).
- [16] P. Gleiser and L. Danon, 'Community structure in jazz', *Advances in Complex Systems*, **6**, 565, (2003).
- [17] M. Grabisch, 'k-order additive discrete fuzzy measures and their representation', *Fuzzy Sets and Systems*, **92**(2), 167 – 189, (1997).
- [18] M. Grabisch and M. Roubens, 'An axiomatic approach to the concept of interaction among players in cooperative games', *International Journal of Game Theory*, **28**, 547–565, (1999).
- [19] M. Grabisch and M. Roubens, *Probabilistic interactions among players of a cooperative game*, 2000.
- [20] R. Guimerà and M. Sales-Pardo, 'Missing and spurious interactions and the reconstruction of complex networks', *Proceedings of the National Academy of Sciences*, **106**(52), 22073–22078, (2009).
- [21] P. Hage and F. Harary, *Structural Models in Anthropology*, Cambridge University Press.
- [22] J.A. Hanley and B.J. McNeil, 'The meaning and use of the area under a receiver operating characteristic (roc) curve', *Radiology*, **143**(1), 29–36, (1982).
- [23] M. Al Hasan, V. Chaoji, S. Salem, and M. Zaki, 'Link prediction using supervised learning', in *In Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security*, (2006).
- [24] B. Hayes, 'Connecting the dots: Can the tools of graph theory and social-network studies unravel the next big plot?', *American Scientist*, **5**(94), 400–404, (2006).
- [25] D. Heckerman, D. Geiger, and D.M. Chickering, 'Learning bayesian networks: The combination of knowledge and statistical data', *Machine Learning*, **20**, 197–243, (1995).
- [26] D. Heckerman, C. Meek, and D. Koller, 'Probabilistic Models for Relational Data', Technical report, Microsoft Research, (2004).
- [27] A. Lancichinetti and S. Fortunato, 'Community detection algorithms: a comparative analysis', *Physical review E*, **80**(5), 056117, (2009).
- [28] D. Liben-Nowell and J. Kleinberg, 'The link-prediction problem for social networks', *J. Am. Soc. Inf. Sci. Technol.*, **58**(7), 1019–1031, (2007).
- [29] G. Linden, B. Smith, and J. York, 'Amazon.com recommendations: Item-to-item collaborative filtering', *IEEE Internet Computing*, **7**(1), 76–80, (2003).
- [30] F.C. Linton, F.C. Sue, and G. M. Alaina, 'On human social intelligence', *J. of Social and Biological Structures*, **4**(11), 415–425, (1988).
- [31] W. Liu and L. Lü, 'Link prediction based on local random walk', *EPL (Europhysics Letters)*, **89**(5), 58007, (2010).
- [32] L. Lü and T. Zhou, 'Link prediction in complex networks: A survey', *Physica A*, **390**(6), 11501170, (2011).
- [33] J.-L. Marichal and P. Matheron, 'Approximations of lovsz extensions and their induced interaction index', *DAM*, **156**(1), 11–24, (2008).
- [34] J. L. Marichal and M. Roubens, 'The chaining interaction index among players in cooperative games', in *Adv. in Decision Analysis*, volume 4 of *Math.Mod.-TA*, 69–85, (1999).
- [35] T. P. Michalak, K. V. Aadithya, P. L. Szczepański, B. Ravindran, and N. R. Jennings, 'Efficient computation of the shapley value for game-theoretic network centrality', *J. Artif. Intell. Res.*, **46**, 607–650, (2013).
- [36] M. E. J. Newman, 'Analysis of weighted networks', *Phys. Rev. E*, **70**(5), 056131, (2004).
- [37] G. Owen, 'Multilinear extensions of games', *Management Science*, **18**(5-part-2), 64–79, (1972).
- [38] G. Owen, 'Values of games with a priori unions', in *Mathematical economics and game theory: Essays in honor of Oskar Morgenstern*, volume 141 of *LNEMS*, 76–88, Berlin, (1977).
- [39] G. Owen, 'Modification of the Banzhaf-Coleman index for games with a priori unions', in *Power, Voting and Voting Power*, 232–238, (1982).
- [40] Latapy M. Pons, P., 'Computing communities in large networks using random walks', *Journal of Graph Algorithms and Applications*, **10**(2), 191–218, (2006).
- [41] M. Roubens, 'Interaction between criteria and definition of weights in mcdavproblems', in *44th meet. of the Eur. working gr. MCDA*, (1996).
- [42] L. S. Shapley, 'A value for n-person games', in *In Contributions to the Theory of Games, volume II*, 307–317, PUP, (1953).
- [43] S. Soundarajan and J. Hopcroft, 'Using community information to improve the precision of link prediction methods', in *PWWW '12*, pp. 607–608, (2012).
- [44] P. L. Szczepański, M. K. Tarkowski, T. P. Michalak, P. Harrenstein, and M. Wooldridge, 'Efficient computation of semivalues for game-theoretic network centrality', in *AAAI '15*, pp. 461–469, (2015).
- [45] P.L. Szczepański, A.S. Barcz, T.P. Michalak, and T. Rahwan, 'The game-theoretic interaction index on social networks with applications to link prediction and community detection', in *IJCAI '15*, pp. 638–644, (2015).
- [46] P.L. Szczepański, T.P. Michalak, and T. Rahwan, 'Efficient algorithms for a game-theoretic betweenness centrality', *Artificial Intelligence*, **2016**, 39–63, (2016).
- [47] P.L. Szczepański, T.P. Michalak, and M. Wooldridge, 'A centrality measure for networks with community structure based on a generalization of the owen value', in *ECAI '14*, pp. 867–872, (2014).
- [48] M. Tarkowski, P.L. Szczepański, T. Rahwan, T.P. Michalak, and M. Wooldridge, 'Closeness centrality for networks with overlapping community structure', *AAAI '16: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, forthcoming, (2016).
- [49] H. Tong, C. Faloutsos, and J.-Y. Pan, 'Fast random walk with restart and its applications', in *Proceedings of the Sixth International Conference on Data Mining, ICDM '06*, pp. 613–622, (2006).
- [50] J. Valverde-Rebaza and A de Andrade Lopes, 'Structural link prediction using community information on twitter', in *Computational aspects of social networks (CASoN), 2012 Fourth International Conference on*, pp. 132–137. IEEE, (2012).
- [51] Blondel V.D., Guillaume J.-L., Lambiotte R., and Lefebvre E., 'Fast unfolding of communities in large networks', *Journal of Statistical Mechanics: Theory and Experiment*, **2008**(10), P10008, (2008).
- [52] X. Wang and G. Sukthankar, *Social Network Analysis - Community Detection and Evolution*, chapter Link Prediction in Heterogeneous Collaboration Networks, 165–192, 2014.
- [53] L. Xiaoning, L. Shujin, and L. Feng, 'The influence between two players in game with graph restricted communication and a priori unions', in *Control and Decision Conference*, pp. 55–59, (2011).
- [54] W.W. Zachary, 'An information flow model for conflict and fission in small groups', *J. of Anthropological Research*, **33**, 452–473, (1977).
- [55] T. Zhou, L. Lü, and Y.C. Zhang, 'Predicting missing links via local information', *The European Physical Journal B-Condensed Matter and Complex Systems*, **71**(4), 623–630, (2009).

Cluster-Driven Model for Improved Word and Text Embedding

Zhe Zhao and Tao Liu and Bofang Li and Xiaoyong Du^{1, 2}

Abstract. Most of the existing word embedding models only consider the relationships between words and their local contexts (e.g. ten words around the target word). However, information beyond local contexts (global contexts), which reflect the rich semantic meanings of words, are usually ignored. In this paper, we present a general framework for utilizing global information to learn word and text representations. Our models can be easily integrated into existing local word embedding models, and thus introduces global information of varying degrees according to different downstream tasks. Moreover, we view our models in the co-occurrence matrix perspective, based on which a novel weighted term-document matrix is factorized to generate text representations. We conduct a range of experiments to evaluate word and text representations learned by our models. Experimental results show that our models outperform or compete with state-of-the-art models. Source code of the paper is available at <https://github.com/zhezhaoo/cluster-driven>.

1 Introduction

Word embedding models (also known as neural language models) encode syntactic and semantic information of words into low-dimensional real vectors, where words share similar meanings tend to have similar representations. Generating word embedding is one of the most fundamental tasks in the NLP literature. Word embeddings are widely used in tasks such as tagging and text classification, and have been reported to bring significant improvements on those tasks. Most word embedding algorithms are trained by modeling the relationships between target words and their local contexts, which is based on the distributional hypothesis of Harris: *words in similar contexts have similar meanings*. However, global contexts, which usually reflect semantic meanings of target words, are generally ignored by these models. For example, words that often co-occur in the same texts tend to reflect similar topics or sentiment tendencies, even they seldom appear in each others' local contexts.

As far as we know, there is still rare research which utilizes global context for word embedding training besides the following three works. Huang et al. [8] propose GCANLM on the basis of C&W [2], where authors use weighted average of word embeddings to represent texts (global contexts), and the embeddings of words and their corresponding texts are trained to obtain higher scores. However, C&W and GCANLM are slow in computation and are reported to perform relatively poorly on various linguist tasks compared to state-of-the-art methods, such as models in the word2vec toolkit³ [19, 18].

Paragraph Vector(PV), proposed by Le and Mikolov [12], introduces global information into word2vec. PV embeds texts by predicting the words they include, and thus introduces global information into word embedding indirectly, though the aim of PV is training text embedding. Sun et al. [22] demonstrate the superiority of PV (or the variants of PV) on various word-level linguistic tasks. The problem of PV is that, it has to give every text a vector. For large-scale datasets such as Wikipedia, the number of texts is much larger than the vocabulary size, which requires expensive computational resources during the training process. Besides that, none of GCANLM and PV introduce global information of different degrees according to different applications. Intuitively, for tasks like word syntactic analogy, more local information is preferred, while tasks such as sentiment analysis tend to favor global information, where rich semantics are included.

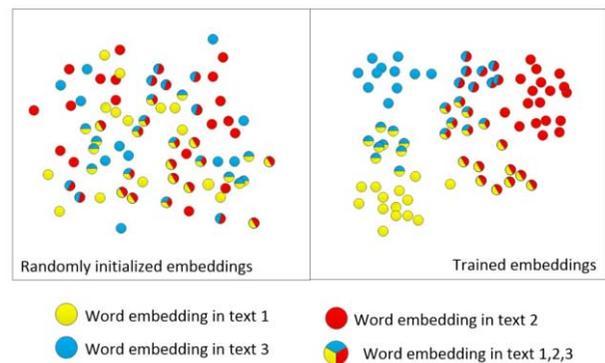


Figure 1. Visualization of forming clusters in two dimensional case.

In this paper, a more general and powerful framework of utilizing global context is proposed for learning improved word and text embedding, namely, the cluster-driven models. The main idea of our models follow the concept of clustering algorithms. The models are trained to make the embedding of words in the same text to form a cluster (as shown in figure 1). Different from other clustering algorithms, in our models, which word belongs to which cluster is preordained and a word may belong to multiple clusters (a word occurs in different texts). Nevertheless, the objectives of our models are the same with other clustering methods: intra-cluster distances are minimized while inter-cluster distances are maximized. As a result, our models extend words' contexts from local windows to the whole texts. Though our model is not based on neural networks, we still call it embedding model since it is trained in an on-line, stochastic fashion. The cluster-driven models can be used standalone, which are able to capture rich semantic information, and can also be inte-

¹ Key laboratory of Data Engineering and Knowledge Engineering, MOE, email: [helloworld][tliu][libofang][duyong]@ruc.edu.cn

² School of Information, Renmin University of China

³ <https://code.google.com/p/word2vec/>

grated into existing word embedding models easily, and hence the degree of utilizing global information can be adapted to the requirements of different applications.

From word embedding to text embedding, considerable attention has been paid to designing various Neural Networks (NNs) to learn complex compositionality of texts, such as word order, sentence structure and even document structure [7]. Word order is taken into consideration in convolutional NNs (CNNs) [10] and recurrent NNs (RNNs) [17, 3]; Recursive NNs (RecNNs) make fully use of syntactic information by constructing neural networks on the basis of parse tree [21]; Recently several works have been proposed to use combination of NNs to model the documents hierarchically [11, 23]. For example, Li [15] uses recursive NN to learn sentence embeddings from word embeddings and use recurrent NN to learn document embeddings from sentence embeddings. Though complex compositionality are learned upon word embedding, these models still don't show significant superiority over bag-of-words models [9]. In this paper, we discover that with richer semantic word embeddings, superior text embeddings, at least for sentiment analysis, can be obtained even by simple strategy such as word embeddings averaging (VecAvg). This paper also provides us better understanding of Paragraph Vector (PV), a very popular method for learning text embeddings. We discover that the superiority of PV comes from the use of global information, rather than the way it trains text embeddings (it trains in prediction manner).

For a thorough comprehension of the cluster-driven models, we analyze it in the co-occurrence matrix perspective. Count-based and embedding methods are two families for generating low dimensional word and text representations. Count-based methods directly utilize co-occurrence statistics and usually obtain dense representations by factorizing co-occurrence matrix [4]. Count-based methods usually served as poor baselines in various linguistic tasks until the works done by Levy and Goldberg [13] and Pennington et al. [20], which demonstrate that count-based models can compete with state-of-the-art word embedding models [14]. In this paper, we extend their works from word representations to text representations. We present count-based counterparts of our cluster-driven models, based on which we factorize a novel weighted matrix of term-document type. Experimental results show this new count-based model can achieve comparable results with state-of-the-art text embedding models, and even outperforms previous approaches on small-scale datasets.

2 Models

2.1 Embedding Models Revisit: Train in Local Manner

Word embedding models can capture the syntactic and semantic information of words from large-scale unlabeled corpus. In contrast to traditional bag-of-words representations, relationship between word embeddings mirrors the syntactic and semantic similarities between two words. For example, words that share similar meanings are close to each other, e.g. 'strong' and 'powerful'. And embedding models can also preserve some interesting linear translation patterns, e.g. $\text{Vec}(\text{'Madrid'}) - \text{Vec}(\text{'Spain'}) + \text{Vec}(\text{'France'}) = \text{Vec}(\text{'Paris'})$.

Most word embedding algorithms are trained by maximizing the log-likelihood of the probability of the target word given its local context [1]:

$$L(\theta_1, \theta_2) = \sum_{i=1}^{|WN|} \log P(w_i | w_i^{\text{context}}) \quad (1)$$

where w_i^{context} denotes the local context of word w_i . $|WN|$ is the number of training words in the whole dataset. Word embeddings and parameters in neural network are respectively denoted by θ_1 and θ_2 . Different word embedding models differ in how they define the conditional probability and how they represent the local contexts.

2.2 Cluster-Driven Models: Train in Global Manner

One obvious drawback of the existing models is that they don't use information beyond local contexts. For capturing global information, two versions of the cluster-driven models are designed: pairwise model and centric model, both of which are trained by making embeddings of words in the same text to form a cluster.

2.2.1 Pairwise Cluster-Driven (PCD)

In pairwise model, word pairs are sampled for distance adjustment according to whether they are in the same text or not. The objective function of the model consists of two components.

Minimizing intra-cluster distances The first component of the objective function is to decrease the distance between the embedding of words in the same texts. A certain number of word pairs are sampled and distances between them are minimized as the following objective:

$$G_1^P(\theta_1) = \sum_{i=1}^{|T|} \sum_{j=1}^{|t_i|} \sum_{k=1}^{|POS|} E_{w_k \sim PT_i(w)} \text{intra_dis}(e_{w_{ij}}, e_{w_k}) \quad (2)$$

where intra_dis is used to measure the distance between two words in the same text. It penalizes the case where embeddings of two words in the same text are far away from each other. $t_i = \{w_{i1}, w_{i2}, \dots, w_{i|t_i|}\}$ denotes i_{th} text and $T = \{t_1, t_2, \dots, t_{|T|}\}$ denotes the whole dataset. e_w denotes the embedding of word w . For each word w_{ij} , $|POS|$ words in the same text are sampled from the distribution $PT_i(w)$ and distances between them are minimized. Intuitively, it is better to shorten the distance between two related words, like 'amazing' and 'amazingly', instead of 'amazing' and 'the'. It is also favorable that the probabilities of words pairs being sampled decline as their distance increases, since very distant word pairs tend to share less relevant information. However, in this paper, $PT_i(w)$ is just the uni-gram distribution of the i_{th} text. We find this simple strategy works pretty well if the number of word pairs sampled is large enough.

Maximizing inter-cluster distances The second component of the objective function is to increase the distance between embeddings of words in different texts. Word pairs are sampled from the whole dataset and the following objective function is maximized:

$$G_2^P(\theta_1) = \sum_{i=1}^{|T|} \sum_{j=1}^{|t_i|} \sum_{k=1}^{|NEG|} E_{w_k \sim P_n(w)} \text{inter_dis}(e_{w_{ij}}, e_{w_k}) \quad (3)$$

where inter_dis is used to measure the distance between two words in different texts. It penalizes the case where the embedding of words in different texts are close to each other. For each training word, $|NEG|$ words are drawn from distribution $P_n(w)$, a uni-gram distribution raised to the n -th power [19].

The final objective function of the model is as follows:

$$G(\theta_1) = G_1^P(\theta_1) - G_2^P(\theta_2) \quad (4)$$

The size of the global contexts is much larger than local contexts. Intuitively, models require much more training time to exploit global contexts. However, in this model, global information can be utilized efficiently and effectively through sampling.

2.2.2 Centric Cluster-Driven (CCD)

In centric model, centroid vector which has the same dimension with word embedding is introduced to denote the center of each cluster. Instead of adjusting distances between the embedding of words directly, we adjust distances between centroid vectors and word embeddings. Like the pairwise case, the objective of the centric model also consists of two components. The first component is to minimize the distances between the centroid vector and the embedding of words in the corresponding text:

$$G_1^C(\theta_1, ct) = \sum_{i=1}^{|T|} \sum_{j=1}^{|t_i|} \text{intra_dis}(e_{w_{ij}}, ct^i) \quad (5)$$

where ct^i denotes the centroid vector of the text t_i . By introducing centroid vectors, we indirectly decrease the distances between all word pairs in the text.

The second component of the objective is to maximize the distances between the centroid vector and the embedding of words in different texts:

$$G_2^C(\theta_1, ct) = \sum_{i=1}^{|T|} \sum_{k=1}^{|NEG| * |t_i|} E_{w_k \sim P_n(w)} \text{inter_dis}(e_{w_k}, ct^i) \quad (6)$$

$|NEG|$ words are drawn from distribution $P_n(w)$ for each training word. Namely, $|NEG| * |t_i|$ words are drawn for text t_i and distances between the centroid vector and the embedding of these sampled words are maximized.

Empirically, the centric model requires less training time to achieve comparable results with the pairwise model. However, the centric model requires much more memory since each text has a unique centroid vector. Besides that, The centric model performs relatively poorly in sentence-level texts. We speculate the reason is that too much noise is introduced when utilizing a vector to represent only a few words.

2.2.3 Distance Measures

Distance/Similarity measures reflect the degree of closeness or separation of two embeddings. They are important for the performance of models. In this paper, different distance measures are used according to whether two words are in the same text or not. Table 1 lists two sets of distance measures. To make sure that global objectives are in the same numeric range with local objectives, we add sigmoid function on distance measures since most objectives of the local embedding models are trained by maximizing the conditional probabilities of target words. When we use the second set of distance measures, the centric model is similar to PV-DBOW, a variant of PV [12]. PV-DBOW can be viewed as a special case of cluster-driven models when only negative sampling is used as softmax.

Table 1. Different sets of distance measures.

	Intra-cluster	Inter-cluster
Measures1	$(\sigma(e_1^T e_2) - 1)^2$	$-(\sigma(e_1^T e_2) - 0)^2$
Measures2	$\log(\frac{1}{\sigma(e_1^T e_2)})$	$\log(\sigma(-e_1^T e_2))$

2.3 Integrated Model

The cluster-driven models can be integrated into existing word embedding models by linearly combining the local and global objective functions:

$$(1 - \lambda)(-L(\theta_1, \theta_2)) + \lambda G(\theta_1) \quad (7)$$

By adjusting λ , we can easily balance the local and global information during the training process. When λ equals to zero, the model is the same with existing local embedding models. More global information is introduced into the model as λ increases. When λ equals to one, only global information is utilized to train word embeddings. In section 5.2, we will demonstrate that the embedding trained in local manner tends to capture syntactic information while the embedding trained in global manner tends to capture semantic information. As a result, we can train word embeddings of different properties according to where embeddings are used. Figure 2 shows the framework of the integrated model.

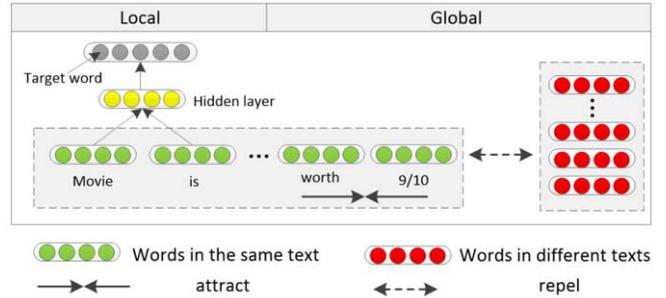


Figure 2. Illustration of the integrated model.

2.4 From Word Embedding to Text Embedding

PV and VecAvg are two approaches for learning text embedding from word embedding in an unsupervised framework [12]. Here, we use PV to refer to the process of learning text embedding by predicting the words it includes. Assumption behind PV is that a good text embedding should be able to predict the words it includes in larger probabilities, while assumption behind VecAvg is that a good text embedding should be similar with the words it includes. They are both essentially bag-of-words models and enjoy the advantages of being efficient and robust. In this paper, we discover that with rich semantic word embedding, neural bag-of-words models like PV and VecAvg can still rival the models that learn complex compositionality upon word embeddings.

3 Theoretical Analysis

To better understand the cluster-driven models, we further explore them in co-occurrence matrices perspective though they do not require to construct co-occurrence matrices at all. Following the theoretical analysis, we factorize the shifted positive pointwise mutual information matrix (SPPMI) of term-document type via singular value decomposition (SVD) to obtain text representations. Count-based models usually serve as poor baselines or are even seldom taken into consideration for generating text representation [16]. However, we discover that when suitable weighted co-occurrence matrix is factorized, count-based models can still achieve comparable results with state-of-the-art models.

3.1 Co-occurrence Matrix Perspective for Cluster-Driven Model

Take CCD for example, we begin by rewriting its objective:

$$\begin{aligned} & \sum_{i=1}^{|T|} \sum_{j=1}^{|\mathcal{t}_i|} \text{intra_dis}(e_{w_{ij}}, ct^i) \\ & - \sum_{i=1}^{|T|} \sum_{k=1}^{|\text{NEG}| * |\mathcal{t}_i|} E_{w_k \sim P_n(w)} \text{inter_dis}(e_{w_k}, c^i) \end{aligned} \quad (8)$$

For specific term-document pair (w_a, t_b) , the objective is:

$$\begin{aligned} & c(w_a, t_b) * \text{intra_dis}(e_{w_a}, ct^b) \\ & - |t_b| * |\text{NEG}| * P_n(w_a) * \text{inter_dis}(e_{w_a}, ct^b) \end{aligned} \quad (9)$$

where $c(w_a, t_b)$ is the number of times word w_a appears in text t_b . From equation 9, we can see more clearly that our model utilizes no more information than term-document co-occurrence matrices and some other basic statistics, such as the length of texts and words distribution. Next, we rewrite the equation by replacing *intra_dis* and *inter_dis* with concrete distance measures:

$$c(w_a, t_b) * (\sigma(e_{w_a}^T ct^b) - 1)^2 \quad (10)$$

$$-|t_b| * |\text{NEG}| * P_n(w_a) * (-\sigma(e_{w_a}^T ct^b) - 0)^2$$

$$\begin{aligned} & c(w_a, t_b) * \log(1/\sigma(e_{w_a}^T ct^b)) \\ & - |t_b| * |\text{NEG}| * P_n(w_a) * \log(\sigma(-e_{w_a}^T ct^b)) \end{aligned} \quad (11)$$

Following the work done by Levy and Goldberg [13], we assume that the objectives of different term-document pairs are independent to each other. Therefore we can directly optimize the objective of each specific pair. Without loss of generality, n is chosen to be 1. We take derivatives of objectives in equation 10 and 11 with respect to $w_a^T ct^b$ and compare them to zero. In both cases, the objectives are optimized when the inner product of specific term-document pairs equals to the shifted pointwise mutual information (SPMI) of them:

$$w_a^T ct^b = \log\left(\frac{c(w_a, t_b)}{|t_b| * P_1(w_a)}\right) - \log(|\text{NEG}|) \quad (12)$$

Therefore, optimizing equation 8 is implicitly factorizing a SPMI matrix of term-document type. For PCD, we can easily prove that it utilizes no more information than term-term matrix and it is implicitly factorizing SPMI matrix of term-term type.

It is worth mentioning that assuming the objectives of different term-document pairs are independent is not realistic, especially in term-document case. A word may occur in many texts and a text always contains multiple words. A word can affect many objectives, so does a text. The independence of the objectives is a hypothesis that is far from the real situation. However, the analysis above inspires us to factorize this matrix to obtain improved text representations.

3.2 Shifted Positive PMI Matrix of Term-document Type Factorization

Shifted PMI matrix can not be directly factorized since it contains too many $-\infty$ ($\log 0$) values, which correspond to the term-document pairs that are never observed in the dataset. A well-known substitution for PMI matrix is positive PMI (PPMI). We factorize shifted positive PMI (SPPMI) matrix of term-document type and it is defined as follows:

$$\max(\text{PMI}(w, t) - \log(|\text{NEG}|), 0) \quad (13)$$

Levy and Goldberg [13] and Levy et al. [14] factorize SPPMI term-term matrix via SVD for acquiring dense word and context vectors. Since all negative values are replaced by zeros, SPPMI term-term matrix lose the information about which term pairs are negatively associated and to what extent.

However, it is not the case for SPPMI matrix of term-document type. We find that PMI term-document matrix usually contains rare negative values besides $-\infty$. Moreover, it is better to assume term-document pairs are uninformative rather than negatively correlated if they are not found in the dataset, because a text only includes hundreds of words, which is small compared to vocabulary size. Viewed from this point, SPPMI is a relatively ideal matrix to be factorized. Experimental results show that text representation obtained by factorizing this novel co-occurrence matrix can compete with or even outperform state-of-the-art baselines.

4 Word Analogy Experiment

4.1 Datasets and Experimental Setup

The word analogy dataset proposed by Mikolov et al. [18] is to evaluate linguistic regularities of word representations. Questions in this dataset are in the form: ‘a is to b as c is to _?’, which are answered by finding the nearest neighbor of $e_a - e_b + e_c$. Training corpus used for word analogy task varies among different published results, and we choose a comparatively widely used corpora Wikipedia2010⁴ as the training data. Pre-processing includes tokenization, lowercasing and substituting number with special character.

Stochastic gradient descent (SGD) is used for objective optimization. We find that two distance measures in table 1 can be used interchangeably as long as they are used with suitable hyper-parameters, such as learning rate and epochs. Here, distance measure 1 is used for PCD and distance measure 2 is used for CCD. Two state-of-the-art local context embedding models, skip-gram (SG) and continues bag-of-words (CBOW), are used as alternatives for integration. The above training protocols are applied to all experiments in this paper.

⁴ <http://nlp.stanford.edu/data/WestburyLab.wikicorp.201004.txt.bz2>

4.2 Integrated Model vs Local Model

As shown in table 2, when the global information is introduced into the models, significant improvements are obtained on semantic analogy questions. Intuitively, global information can hardly provide any information for capturing syntactic regularities. In this sense, global information is the noise and may hurt the models performance in syntactic analogy questions. However, to our surprise, accuracies on syntactic analogy questions do not decline when a certain degree of global information is introduced. Overall, significant improvements on total accuracies are obtained.

To further understand why global information is beneficial for capturing semantic analogy regularities, we analyze some mistakes made by local models. We discover that local models give the wrong answers mainly for the reason that they fail to distinguish words which have similar semantic meanings. Take an analogy question ‘son, daughter, grandfather, ?’ for example. The correct answer is ‘grandmother’, but the local model returns the wrong answer ‘granddaughter’. We notice that when the model is trained in local manner, the embedding of these two words are very close. Local information is not enough to distinguish these two words. However, more information is available when global information is introduced. For example, ‘aged’, ‘life’, ‘maternal’ frequently occur in the global contexts of ‘grandmother’, while they seldom occur in the global contexts of ‘granddaughter’. These different global contexts can help to distinguish the semantics of these two words.

Table 2. Comparison of the local and integrated models. For PCD, λ and $|POS|$ are set to be 0.1 and 5 respectively. For CCD, λ is set to be 0.6. Hyper-parameter settings of the local embedding models follow the word2vec toolkit.

Dim.	Model	Sem.	Syn.	Model	Sem.	Syn.
50	CBOW	55.7	59.9	SG	45.9	50.7
	+PCD	+4.0	-0.6	+PCD	+3.7	+0.3
	+CCD	+3.8	+1.6	+CCD	+4.4	+2.5
100	CBOW	69.5	71.0	SG	62.7	66.0
	+PCD	+3.9	+0.1	+PCD	+3.7	+0.2
	+CCD	+5.1	+1.5	+CCD	+4.1	-0.1

4.3 Comparison of Word Embedding Models

Different state-of-the-art word embedding models are compared in table 3. The corpus size has been shown to be a minor factor compared to the embedding dimensions. Therefore, we group results according to the dimensions. Here, we still list the corpus size for keeping consistent with other researches.

We can observe that CBOW has provided strong baselines on word analogy dataset. By introducing global information upon CBOW, more competitive results are achieved. We can observe that our models perform consistently better than previous state-of-the-art approaches in all dimension settings.

PDC and HDC also introduce global information into word embeddings. The source of superiority of our models to PDC and HDC comes from the choice of λ values, which controls the degrees of global information utilized during the training. Suitable λ can enhance the accuracy in semantic questions significantly without hurting the accuracy in syntactic questions. We also find that different types of word analogy tasks favor different λ , which will be further explored in our future work.

Table 3. Comparison of different word embedding models on word analogy task. The results are grouped according to the dimensions of word embedding. The best methods in each group are underlined and the best in the whole table are also in bold

Model	Dim.	Size	Sem.	Syn.	Tot.
C&W[18]	50	0.66B	9.3	12.3	11.0
GCANLM[18]	50	1B	13.3	11.6	12.3
GLOVE[20]	50	6B	48.5	44.4	46.2
CBOW	50	1B	55.7	59.9	58.3
SG	50	1B	45.9	50.7	48.9
PDC[22]	50	1B	61.2	55.1	57.9
HDC[22]	50	1B	57.8	49.8	53.4
<i>CCD_{CBOW}</i>	50	1B	59.5	61.5	60.7
<i>CCD_{SG}</i>	50	1B	50.3	53.2	51.8
GLOVE[20]	100	1.6B	67.5	54.3	60.3
CBOW	100	1B	69.5	71.0	70.4
SG	100	1B	62.7	66.0	64.7
PDC[22]	100	1B	72.8	68.4	70.4
HDC[22]	100	1B	69.6	64.3	66.7
<i>CCD_{CBOW}</i>	100	1B	<u>74.6</u>	<u>72.5</u>	<u>73.3</u>
<i>CCD_{SG}</i>	100	1B	66.8	65.9	66.3
GLOVE[20]	300	6B	77.4	67.0	71.7
GLOVE[20]	300	42B	81.9	69.3	75.0
CBOW	300	1B	74.6	74.0	74.2
PDC[22]	300	1B	79.6	70.5	74.8
HDC[22]	300	1B	79.7	67.7	73.1
<i>CCD_{CBOW}</i>	300	1B	82.5	75.4	78.1

5 Sentiment Analysis Experiment

5.1 Datasets and Experimental Setup

Four sentiment analysis datasets are used to evaluate the effectiveness of our models. Datasets RT-s and Subj include sentence-level texts while IMDB and RT-2k include document-level texts. Since RT-s and RT-2k datasets only contain limited snippets or documents, additional texts in IMDB dataset are added to them during the training process.

Text embeddings obtained by our models can be regarded as texts features and then fed to logistic regression classifier [6]. 10% of the training set is selected as the validation set to identify optimal hyper-parameters, such as learning rate, $|POS|$ and λ . IMDB dataset has train/test split. The rest three datasets are evaluated by 10-fold cross-validation.

5.2 Words Semantic and Syntactic Relatedness Analysis

We evaluate the quality of word embeddings by judging if the top k nearest neighbors are semantic or syntactic related to the target word. Models are trained on a movie review dataset, IMDB. Both qualitative and quantitative results are presented, which shed some light on the reason why global information is preferred for sentiment analysis tasks.

From Table 4, we can observe the local model tends to return syntactic related words, while the global model tends to return semantic related words. For example, word ‘best’ is the neighbor of word ‘worst’ when the local model is used. Both words are superlative adjectives, but they have opposite sentiment polarities. When trained in global manner, word ‘worst’ has the neighbor word ‘0/10’, which indicates the lowest user rating score in a movie review. Though they have different POS tags, they share exactly the same sentiment tendency.

In addition to just giving several examples and understanding them intuitively, word embedding properties are further analyzed quantita-

tively. Two widely used evaluation criteria in information retrieval literature, average mean precision and DCG@10, are respectively used to evaluate the syntactic and semantic relevance of ranked neighbor lists to the target words. Specifically, 100 target words are sampled in the corpus, and top 30 neighbors of each word are obtained. Whether two words are syntactically related are judged by checking if they have the same POS tags. The semantic relatedness between two words are obtained from the average of judgments from 5 persons. Figure 3 demonstrates that when global information is increasingly introduced into the model, the embeddings reflect more semantic information and are less constrained by syntactic regularities.

Table 4. Illustration of the nearest neighbors of the target words.

Target Word	Neighbors	
	By Local Model	By Global Model
amazing	great, wonderful	10/10, amazingly
worst	dumbest, best	0/10, zero
worthless	talentless, untalented	1/10, lowest

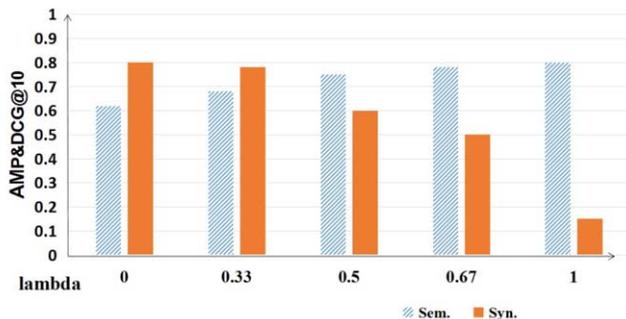


Figure 3. Evaluating the semantic and syntactic information contained in word embedding quantitatively.

5.3 Sentiment Analysis Prefer Global Information

As discussed in Section 2.4, two simple neural bag-of-words methods, PV and VecAvg, are used for generating text embeddings. From Figure 4, we can observe that more global information is preferred for document-level sentiment analysis tasks. Nearly 5 percent improvements are witnessed when global information is introduced. In fact, only about 2 percent improvements are obtained when word order information is taken into consideration in [24]. In this sense global information is of vital importance for document-level sentiment analysis. The performances of PCD and CCD are almost the same on document-level dataset. Therefore, Figure 3 only shows accuracies in the CCD case for the sake of space saving. For sentence-level datasets, introducing global information can not improve accuracy significantly since local windows usually already cover most of the sentences. However, strong results are obtained by using the cluster-driven models standalone, which requires less training time and computational resources.

From Figure 4, we can also observe that PV does not perform better than VecAvg. In contrast to the conclusion from [12], we discover that PV is not superior to VecAvg. In fact, they are both essentially bag-of-words models, where order information is totally ignored.

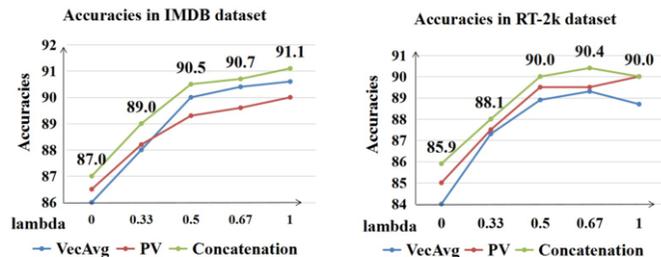


Figure 4. Accuracies on two document level datasets when global information is introduced in different degrees. Concatenation of PV and VecAvg performs better.

Table 5. Results from row 3 and 4 are from Mesnil et al [17]. Their work publishes the source code⁵ and argues that the results provided by Le and Mikolov [12] can not be reproduced. For document-level datasets, integrated models are used while for sentence-level datasets, the cluster-driven models are used standalone. VecAvg is used to generate text embedding from word embedding. CON. (row 9) represents the concatenation of VecAvg and PV. The models are grouped according to how they exploit information in the text. The best methods in each group are underlined and the best in the whole table are also in bold.

Category	Model	RTs	Subj	IMDB	RT2k
bag-of-words	SVM-uni[24]	76.2	90.8	87.0	86.3
	NBSVM-uni[24]	78.1	92.4	88.3	87.8
	PV-DM[17]	76.9	91.7	89.6	88.8
	PV-DBOW[17]	76.1	90.1	89.1	88.7
	DAN-RAND[9]	77.3	-	88.8	-
	DAN[9]	<u>80.3</u>	-	89.4	-
	PCD	78.0	92.4	90.4	89.7
	CCD	75.4	90.9	90.6	90.1
	CON.	78.5	<u>92.6</u>	<u>91.1</u>	90.4
words order	SVM-bi[24]	77.7	91.7	89.2	87.4
	NBSVM-bi[24]	79.4	93.2	91.2	<u>89.5</u>
	NBSVM-tri[17]	-	-	91.9	-
	RNN-LM[17]	-	-	86.6	-
	Ensemble[17]	-	-	92.6	-
	SA-LSTM[3]	-	-	92.8	-
	CNN[10]	81.5	93.6	-	-
complex structure	DCNN[5]	-	-	89.4	-
	RecNN[21]	77.7	-	-	-
	RecNN-RNN[15]	-	-	87.0	-
	WNN[15]	77.8	-	90.2	-
	BENN[15]	77.2	-	<u>91.0</u>	-

5.4 Comparison of Sentiment Analysis Models

In Table 5, our models are compared with state-of-the-art sentiment analysis techniques, which are categorized according to how they exploit information in the text. One of the simplest representations is bag-of-words (BOW), where order information is totally discarded. Though BOW seems to be oversimplified, it still enjoys the advantages of being efficient, robust and concise. Word order is often important for text understanding. Bag-of-ngrams models use n-grams as features to capture words order in short context. CNNs use convolutional filters to extract n-gram information from texts. RNNs model texts sequentially and in theory can capture long-distance patterns in natural languages. Beyond word orders, more complex information such as syntax, relations among sentences is considered to train better text representations. Though information such as order and syntax is important for understanding texts, it always comes at a cost. We surprisingly observe that, even though our models are essentially bag-

⁵ <http://github.com/mesnilgr/iclr15>

of-words models, they can even compete with models which exploit complex information of texts. Since our models ignore word order and syntactic information, they require less training time and computational resources compared to other state-of-the-art approaches.

Our models are also robust and concise. They perform well on both sentence and document level datasets. In contrast, models like CNNs and RecNNs are hard to extend to document-level dataset. Besides that, neural networks or their combinations usually have a large number of hyper-parameters and require careful hyper-parameter tuning. Their performance also closely rely on several sub-tasks, such as pre-trained word embedding and parsing.

5.5 Embedding Models vs. Count-based Models

Almost all the recent works on sentiment analysis take count-based methods as poor baselines. However, work in section 3 inspires us to factorize SPPMI matrix of term-document type. The hyper-parameter includes shifted-constant $|NEG|$ and the threshold for removing low frequency words, which are chosen by validation set. We compare four approaches which utilize exact the same source of information: term-document co-occurrence matrix. As shown in table 6, the novel count-based method can achieve comparable accuracies with state-of-the-art embedding methods such as PV-DBOW and CCD, and is even more robust when dataset is small. We can observe that the performance of embedding models is poor on RT-2k dataset, unless additional unlabeled data is included.

Table 6. Comparison between count-based and embedding methods for sentiment analysis. Results of LSA are from Maas et al. [16]. Results of CCD are different from table 5 since results in table 5 are obtained by using both local and global information. Results of CCD in table 6 only utilize global information, where only term-document matrix information is taken into consideration.

Model	IMDB	RT2k	RT2k+Unlabeled
SPPMI	89.6	89.2	89.7
LSA	84.0	82.8	-
PV-DBOW	89.6	85.4	89.5
CCD	90.5	85.7	90.0

6 Conclusion

In this paper, we introduce the cluster-driven models to exploit global information to learn better word and text embeddings. When the models are used standalone, trained word embeddings can capture rich semantics. The models can also be integrated into existing local embedding models to introduce global information of different degrees. Besides that, analyzing the model in co-occurrence matrix perspective inspires us to factorize SPPMI matrix of term-document type to obtain text representations. From experimental results we can obtain several conclusions:

- Global information enriches the semantic information contained in word embeddings. Improvements are witnessed on all experiments by introducing global information into the models.
- Bag-of-words models can still compete with complex deep neural networks when global information is exploited. We also discover that the superiority of PV comes from the introduction of global information. Training text embedding in prediction manner (PV) is not superior to word embedding average (VecAvg).
- Count-based models are not inferior to embedding models. Strong results on sentiment analysis are achieved by factorizing a novel term-document matrix.

ACKNOWLEDGEMENTS

This work is supported by the National Fundamental Research and Development Program of China (973 Program) under grant 2012CB316205, National Natural Science Foundation of China (61472428, 61003204), the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China No. 14XNLQ06 and Tencent company.

REFERENCES

- [1] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin, 'A neural probabilistic language model', *Journal of Machine Learning Research*, **3**, 1137–1155, (2003).
- [2] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa, 'Natural language processing (almost) from scratch', *Journal of Machine Learning Research*, **12**, 2493–2537, (2011).
- [3] Andrew M. Dai and Quoc V. Le, 'Semi-supervised sequence learning', in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 3079–3087, (2015).
- [4] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman, 'Indexing by latent semantic analysis', *JASIS*, **41**(6), 391–407, (1990).
- [5] Misha Denil, Alban Demiraj, Nal Kalchbrenner, Phil Blunsom, and Nando de Freitas, 'Modelling, visualising and summarising documents with a single convolutional neural network', *CoRR*, **abs/1406.3830**, (2014).
- [6] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin, 'LIBLINEAR: A library for large linear classification', *Journal of Machine Learning Research*, **9**, 1871–1874, (2008).
- [7] Yoav Goldberg, 'A primer on neural network models for natural language processing', *CoRR*, **abs/1510.00726**, (2015).
- [8] Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng, 'Improving word representations via global context and multiple word prototypes', in *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 1: Long Papers*, pp. 873–882, (2012).
- [9] Mohit Iyyer, Varun Manjunatha, Jordan L. Boyd-Graber, and Hal Daumé III, 'Deep unordered composition rivals syntactic methods for text classification', in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pp. 1681–1691, (2015).
- [10] Yoon Kim, 'Convolutional neural networks for sentence classification', in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1746–1751, (2014).
- [11] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao, 'Recurrent convolutional neural networks for text classification', in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pp. 2267–2273, (2015).
- [12] Quoc V. Le and Tomas Mikolov, 'Distributed representations of sentences and documents', in *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pp. 1188–1196, (2014).
- [13] Omer Levy and Yoav Goldberg, 'Neural word embedding as implicit matrix factorization', in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pp. 2177–2185, (2014).
- [14] Omer Levy, Yoav Goldberg, and Ido Dagan, 'Improving distributional similarity with lessons learned from word embeddings', *TACL*, **3**, 211–225, (2015).
- [15] Jiwei Li, 'Feature weight tuning for recursive neural networks', *CoRR*, **abs/1412.3714**, (2014).
- [16] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts, 'Learning word vectors for sentiment analysis', in *The 49th Annual Meeting of the Association for*

- Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pp. 142–150, (2011).
- [17] Grégoire Mesnil, Tomas Mikolov, Marc’ Aurelio Ranzato, and Yoshua Bengio, ‘Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews’, *CoRR*, **abs/1412.5335**, (2014).
- [18] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, ‘Efficient estimation of word representations in vector space’, *CoRR*, **abs/1301.3781**, (2013).
- [19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean, ‘Distributed representations of words and phrases and their compositionality’, in *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pp. 3111–3119, (2013).
- [20] Jeffrey Pennington, Richard Socher, and Christopher D. Manning, ‘Glove: Global vectors for word representation’, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1532–1543, (2014).
- [21] Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning, ‘Semi-supervised recursive autoencoders for predicting sentiment distributions’, in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27-31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 151–161, (2011).
- [22] Fei Sun, Jiafeng Guo, Yanyan Lan, Jun Xu, and Xueqi Cheng, ‘Learning word representations by jointly modeling syntagmatic and paradigmatic relations’, in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pp. 136–145, (2015).
- [23] Duyu Tang, Bing Qin, and Ting Liu, ‘Document modeling with gated recurrent neural network for sentiment classification’, in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pp. 1422–1432, (2015).
- [24] Sida I. Wang and Christopher D. Manning, ‘Baselines and bigrams: Simple, good sentiment and topic classification’, in *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 2: Short Papers*, pp. 90–94, (2012).

Learning Temporal Context for Activity Recognition

Claudio Coppola and Tomáš Krajník and Tom Duckett and Nicola Bellotto¹

Abstract. We investigate how incremental learning of long-term human activity patterns improves the accuracy of activity classification over time. Rather than trying to improve the classification methods themselves, we assume that they can take into account prior probabilities of activities occurring at a particular time. We use the classification results to build temporal models that can provide these priors to the classifiers. As our system gradually learns about typical patterns of human activities, the accuracy of activity classification improves, which results in even more accurate priors. Two datasets collected over several months containing hand-annotated activity in residential and office environments were chosen to evaluate the approach. Several types of temporal models were evaluated for each of these datasets. The results indicate that incremental learning of daily routines leads to a significant improvement in activity classification.

1 Introduction

Automated recognition of human activities is a hot topic of research. It enables a wide range of applications such as security, retail or healthcare, but recently a huge focus has been given to the recognition of the Activities of Daily Living (ADL) due to its potential application in Ambient Assisted Living (AAL). This technology could help to address the predicted shortage of health workers and improve the quality of life of the increasing elderly population in the near future, by assisting people in their daily tasks and identifying potential problems. Furthermore, it could be used also in security applications to detect anomalous situations that could endanger people or property. The introduction of new technologies has made this problem easier to address. In particular, RGB-D sensors together with pose estimation software and smart sensors for the Internet of Things have enabled the possibility of acquiring data for such applications, giving birth to many related datasets [3, 16, 38, 1]. The development of activity recognition is furthermore supported by novel techniques to manage huge quantities of data ('Big Data') and the increased computational power of modern computers, enabling real-time implementations.

The main focus of the recognition models has been the recognition of patterns derived from the data acquired from the sensors. The features used for pattern recognition typically relate to the body movement and the surrounding context, in the case of RGB-D sensors, or by the sensor events in a smart environment. By contrast, in this work we aim to exploit the long-term patterns of recurring activities to improve the performance of activity classification. Prior work showed that the patterns of the spatio-temporal dynamics of the environment can be exploited to improve indoor localization [20] or path planning [12] of a mobile robot in long-term scenarios.

In a similar way this work proposes an approach to calculate prior probabilities of an activity happening at a certain time, which reduces the error rate of a given classification algorithm. We analyse several possible techniques, including a novel approach based on Adaptive Interval Based Models, which delivers continuous improvement to the recognition performance on-the-fly by incrementally performing naïve Bayesian learning. We evaluate our methods on the Aruba Dataset [3], based on the activities of daily living and the Witham Dataset [18], manually annotated from an overhead camera recording in an office environment.

There are two main contributions in this paper: (i) the introduction of a probabilistic formulation to incrementally model temporal and spatial context to improve activity recognition performance of a given classifier, (ii) the introduction of novel probabilistic models of temporal and spatial context, (iii) comparison of different temporal models in order to understand which ones can better represent the temporal structure of daily activities.

The remainder of this paper is organized as follows. Section 2 gives an overview of the state-of-the-art for activity recognition performed with smart sensors and RGB-D cameras and on the use of temporal and spatial models for activity recognition. Section 3 provides a formulation of the activity recognition problem. Section 4 introduces the temporal models used in our experiments. Section 5 explains our method of evaluation for the temporal models. Section 6 reports the results of our experiments, and finally Section 7 presents the conclusion and future work.

2 Related work

Human activity recognition aims to recognize the actions and goals of human agents using a sequence of observations of the agents' actions and the environmental conditions. Tracking and understanding human behaviour through videos is a very important and challenging problem with various useful applications. Activity recognition has originally been performed on RGB video streams with a wide spectra of solutions [15, 30], including a recent approach [14] with unsupervised deep-learning-based hierarchical feature models. This allows to create a system that learns and improves itself by updating the activity models incrementally over time. The development of cheap RGB-D cameras has contributed to the increased focus on this problem, since they allow to reduce the computational requirement for estimating the pose of the human body and the contextual patterns in the scene in real-time. In [10, 11] a probabilistic ensemble of classifiers called a Dynamic Bayesian Mixture Model (DBMM) is proposed to combine different posterior probabilities from a set of classifiers for activity recognition. Wang et al. [39] show a deep structured model built with layered convolutional neural networks. A biologically inspired approach adopting an artificial neural network to combine pose and motion features for action perception is pro-

¹ Lincoln Centre for Autonomous Systems, University of Lincoln, UK
Email: ccoppola@lincoln.ac.uk

posed by [28]. In [6], a simple way to apply qualitative trajectory calculus to model 3D movements of the tracked human body using hidden Markov models (HMMs) is presented. A method for social activity recognition based on proximity of the interacting humans is presented in [5]. Sung et al. [32, 33] perform activity recognition in unstructured environments such as homes and offices with an RGB-D camera. The movement is modelled by transforming the rotation matrix of each joint to the body torso and inferring the activities and sub-activities with a two-layered Maximum Entropy Markov Model (MEMM). A three-level hierarchical discriminative approach is presented in [23]. The activities are decomposed into a lower level representing the pose data, an intermediate level where the poses are combined into simple human actions, and a high level where the actions are spatially and temporally combined into complex human activities. The approach presented in [29] uses HMMs combined with Gaussian Mixture Models (GMM) to model the combination of continuous joint positions over time for activity recognition. In [37], the authors use random occupancy patterns to model activities using context from depth data.

Smart environments allow to mine through the sensor events to classify which activity has happened. Fleury et al. [13] present a dataset with smart sensors for ADL recognition, where the classification is performed using Support Vector Machines (SVM). A mining technique to find the association rules between the activities and their frequent patterns in smart environments is presented in [40]. In [9], the authors use the Back-Propagation algorithm to train a feed-forward Neural Network with features extracted from the motion sensor events. In [8], a method for evaluating the confidence of classification is presented. The method is able to reduce false positives by identifying samples with low confidence that can be further investigated by a human operator. In [4] an activity discovery algorithm is presented which identifies patterns in sensor data with a greedy approach. It searches for a sequence pattern that best compresses the input data; the data is scanned to create initial patterns of length one, which are extended in every loop while minimizing the description of the data.

In [27] analysis of human activities in an office environment is performed using a Layered Hidden Markov Model (LHMM) architecture based on real-time streams of evidence from video, acoustic, and computer interactions. Similarly, a multi-level HMM is presented in [41] for recognising office activities and tracking the users across the rooms. In [26] a solution for office activity recognition is proposed, which handles multiple-user, multiple-area situations, based on an ontological approach, using low-cost, binary and wireless sensors. The idea of exploiting long-term analysis has been presented already by Van Laerhoven et al. [36], using wrist-worn sensors to collect daily activity data to create rhythmic models of the activities. These models are created off-line using a frequentist approach, accumulating the amount of times an annotated activity starts and stops within a certain time interval, which is represented as a bin. In [24] a long-term annotated dataset using many different sensors is introduced. The classification is performed using a binary classifier for each learned activity, collecting features from the sensor data in particular time windows. Daily routines are recognized in [2] from features extracted with a sliding window approach. These are clustered with k -means to calculate their occurrence statistics and store them in a histogram which is classified using a Joint Boosting technique. Suryadevara et al. [34] introduce a wellness determination process to help healthcare providers to assess the performance of the elderly in their daily activities. It verifies the behaviour of elderly people at three different stages (usage of appliances, activity recognition and

forecast levels) in a smart home monitoring environment integrating the spatial and temporal information.

In [7] a model is introduced for long-term monitoring of activities in a smart home. The classification is performed with a Probabilistic Neural Network (PNN), and the daily schedules of activities are then clustered with k -means. The clusters with highest inter-variation are considered as normal and the others as their deviations. Minor et al. [25] present a way of predicting future activity occurrences, with a recurrent predictor, based on the structure of the temporal sequence of the activities. Long-term modelling of indoor environments has been exploited also in other cases. In [19], the authors argue that part of the environment variations exhibit periodicities and represent the environment states by their frequency spectra. The concept of Frequency-based Map Enhancement (FreME_n) was applied to occupancy grids in [22] to achieve compression of the observed environment variations and to landmark-based maps in order to increase robustness of mobile robot localization [20].

In this paper, we proposed a method that can be applied to existing classification algorithms for activity recognition, learning the temporal structure of the classified activities in order to incrementally improve the classification results on-line. In some sense, our approach provides an abstraction for meta-classification that is independent from the particular classification method, i.e. HMM, SVM, etc, and can be combined with any of those, improving their performance. We investigate several possible representations which can be used to model the (prior) occurrence probability of the learned activities.

3 Problem formulation

We formulate the activity classification problem simply as a Bayesian decision making problem. Let us assume that at time t , a person is performing an (unknown) activity from the set of possible activities \mathcal{A} while being observed by a set of sensors. Let some algorithm C processes the sensory readings and classifies that the activity being performed is $o \in \mathcal{A}$. Let us assume that we have experimentally established the performance of C on some representative dataset and thus, we know C 's confusion matrix, i.e. we can characterise the performance of C as a conditional probability distribution $p(o|a)$, where a represents the activity performed. Thus, every time the algorithm C provides us with an observation o , we can establish the posterior distribution $p(a|o, t)$ over the possible activities at time t as:

$$p(a|o, t) = \frac{p(o|a)p(a, t)}{\sum_{b \in \mathcal{A}} p(o|b)p(b, t)}. \quad (1)$$

In our case, we will use a separate spatial/temporal model for each activity. To emphasize that the models are calculated separately, we rewrite the Equation (1) for a single activity a as

$$p_a(o, t) = \frac{p(o|a)p_a(t)}{p(o|a)p_a(t) + p(o|\neg a)(1 - p_a(t))}, \quad (2)$$

where $p_a(t)$ represents the probability of the activity a being performed at time t , i.e. the temporal prior of a . The expression $p_a(t)$ was chosen to emphasize that the temporal models are built independently - it corresponds to $p(a, t)$ in Equation (1).

While most of the research in activity recognition is aimed at the performance of the activity recognition algorithm C , which increases the likelihood of correct activity classification by improving $p(o|a)$ in Equation (2), our work is not concerned with the actual method that is used to determine the activity from the sensory readings. Instead, we focus on the term $p_a(t)$ in (2), which effectively represents

the temporal context of a given activity. We hypothesize that since people tend to perform certain activities on a regular basis, $p_a(t)$ is a (pseudo-)periodic function that can be learned over time and that better knowledge of $p_a(t)$ would positively impact the performance of the classification system represented by Equation (2).

To learn $p_a(t)$, we apply Equation (2) iteratively. Initially, we start with all $p_a(t) = 1/|\mathcal{A}|$, i.e. we assume that the activities occur with the same probability regardless of the time. Whenever an activity is classified by (2), we use the output of (2) to update $p_a(t)$. Then we use the updated $p_a(t)$ in the following classification step.

The key questions that our paper addresses are:

1. Which model should be used to represent the temporal activity context (or prior) $p_a(t)$?
2. How much does the temporal context impact the performance of state-of-the-art classifiers?
3. Can we learn the temporal context even with a weak classifier?

To answer these questions, we tested four different temporal models on two datasets, which contain human activities labelled minute-by-minute over several weeks.

4 Temporal models

In our work, a temporal model of activity a is a function $p_a(t)$, which represents the probability of the activity a occurring at time t . We consider four types of temporal models: Frequency Map Enhancement (FreMEn), which represents cyclic processes by their frequency spectra, Gaussian Mixtures, which are well established in several domains, and naïve and adaptive versions of interval-based models.

4.1 Frequency Map Enhancement

Frequency Map Enhancement (FreMEn) is an emerging technique that improves the efficiency of mobile robots that operate autonomously for long periods of time [20, 12]. The method assumes that states of the robots' operational environments are affected by pseudo-periodic processes, whose influence and periodicity can be obtained through the Fourier transform. Thus, the uncertainty of a given state $s(t)$ is represented as a probabilistic function of time that is a combination of harmonic functions:

$$p(t) = \alpha_0 + \sum_{i=1}^n \alpha_i \cos(\omega_i t + \varphi_i), \quad (3)$$

where the amplitude α_i , phase shift φ_i and frequency ω_i correspond to the most prominent spectral components of the observations of the original state $s(t)$.

In our case, the state $s(t)$ of the FreMEn model is a binary function of time $o_a(t)$ which indicates if the activity a was observed at time t and $p_a(t)$ will be our probabilistic function $p(t)$. To build the FreMEn model, we simply take the results of the past classifications and form a sequence $o_a(t)$ for each activity $a \in \mathcal{A}$. Then, we calculate the Fourier spectrum of each sequence $o_a(t)$, select n of its most prominent (i.e. with highest amplitudes) spectral components and use their amplitudes, periodicities and phase shifts as $(\alpha_i, \omega_i$ and $\varphi_i)$ parameters of the predictive FreMEn model in Equation (3), which is used as a prior for classification in Equation (2). Since the performance of the FreMEn model is affected by the choice of the model order n , we run our experiments with n ranging from 1 to 9 and chose the best performing setting, which was $n = 2$. To speed up calculations, we used the version of FreMEn introduced in [21], which allows for incremental updates.

The main advantage of the FreMEn model is that it naturally represents multiple periodicities that are inferred automatically from the data. However, it poorly represents periodic, but short duration activities, such as teeth brushing or tea making.

4.2 Gaussian Mixture Model

Gaussian Mixture Models, which approximate multi-dimensional functions as weighted sums of Gaussian component densities, are a well-established method that find their applications in numerous fields from Psychology to Astrophysics [35]. A Gaussian Mixture Model of a function $f(t)$ is a weighted sum of m Gaussian functions:

$$f(t) = \frac{1}{\sqrt{2\pi}} \sum_{j=1}^m \frac{w_j}{\sigma_j} e^{-\frac{(t-\mu_j)^2}{2\sigma_j^2}}. \quad (4)$$

The parameters of the GMM components, i.e. the means μ_j , variances σ_j and weights w_k , are typically calculated from the training data by iterative Expectation Maximization (EM) or Maximum A Posteriori (MAP) algorithms. Since the classic GMMs are not meant to represent periodic functions, we simply assume that people perform most of their activities on a daily basis and limit the time domain of GMM-based models to one day. While this assumption is not entirely correct (as activities of weekdays differ from the weekend ones), such a temporal model might still perform better than a 'static' one, where the probability of a given activity is constant in time.

To build the GMM model of $p_a(t)$, we first create a temporal sequence of observations $o_a(t)$ for each activity in the same way as in the FreMEn case. Then, we calculate an initial prior as follows:

$$p'_a(t) = \frac{k}{\tau} \sum_{i=1}^{\lfloor k/\tau \rfloor} o_a(t + (i-1)\tau), \quad (5)$$

where τ is the assumed period (in our case $\tau = 86400$ s), k is the $s(t)$ sequence length, and $\lfloor k/\tau \rfloor$ is a floor operator, that returns the integer part of k/τ . After calculating $p'_a(t)$, we employ the Expectation Maximization algorithm to find the means μ_i , standard deviations σ_i and weights w_i of its Gaussian Mixture approximation:

$$p_a(t) = \frac{1}{\sqrt{2\pi}} \sum_{i=1}^n \frac{w_i}{\sigma_i} e^{-\frac{((t \bmod \tau) - \mu_i)^2}{2\sigma_i^2}}, \quad (6)$$

where τ is the apriori known period of the function $p_a(t)$ and mod is a modulo operator.

The advantages of periodic GMMs are complementary to the advantages of the FreMEn. Periodic GMMs can approximate short-duration activities, but they can represent only one period that has to be known apriori. Similarly to FreMEn, the performance of GMMs depends on the choice of n , which represents the number of Gaussians used in the mixture model. Again, we run our experiments with n ranging from 1 to 9 and chose the best performing setting, which was $n = 3$.

4.3 Interval-based Model

Another temporal model that has been considered partitions the time into disjoint intervals, each with a different prior probability $p_a(t)$. Similarly to the GMM-based models, the partitioning requires that the periodicity τ and model order n (the number of intervals) are chosen apriori. In our interval-based model, $p_a(t)$ is represented by

n values $p'_a(k)$ that denote prior probabilities of a given activity occurring between $\tau m + \tau \frac{k}{n}$ and $\tau m + \tau \frac{k+1}{n}$, where $m \in \mathbb{N}$ and $k \in \{0, 1 \dots n - 1\}$. In the following text, we will refer to the time interval τ/n as the “interval width”. To update or retrieve $p_a(t)$, one has to simply determine the index k of the relevant interval:

$$p_a(t) = p'_a(k) = p'_a(\lfloor (t \bmod \tau) \frac{n}{\tau} \rfloor). \quad (7)$$

Unlike the FreMEn and GMM models, the interval-based model is updated according to Bayes rule in Equation (2). Thus, when a classification is performed at time t , we first calculate k by Equation (7) and then perform the model update by

$$p'_a(k) \leftarrow \frac{p(o|a)p'_a(k)}{\sum_{a \in A} p(o|a)p'_a(k)}. \quad (8)$$

Again, a crucial question here is model granularity (i.e. the interval width that is determined by the number of the represented intervals n). Models with wide intervals cannot represent short-duration activities, whereas models with short intervals require larger amounts of data for training, therefore their learning rate is slow.

4.4 Adaptive Interval Model

To deal with the aforementioned problem, we can store the number of updates performed for each interval $u(k)$ and calculate $p_a(t)$ by aggregating the probabilistic values of neighbouring intervals, so that $p_a(t)$ is based at least on l updates. While the model update remains the same as in the previous case (see Equation (8)) with the only difference is that the value of $u(k)$ is increased by 1, calculating $p_a(t)$ differs. To determine $p_a(t)$, we first calculate the index of the relevant interval k as $\lfloor (t \bmod \tau) \frac{n}{\tau} \rfloor$ (see Equation (7)). We check if the number of updates performed to calculate $p'_a(k)$ is at least l and if not, we include the neighbouring intervals and calculate $p(t)$ as the weighted (by the number of updates) average. This is repeated until the number of measurements used to determine $p_a(t)$ exceeds l . See Algorithm 1 for more details.

Algorithm 1 Adaptive interval prior calculation

```

1: function CALCULATEPRIOR( $t, \tau, n, \mathbf{u}, \mathbf{p}'_a, l$ )
2:    $k \leftarrow \lfloor (t \bmod \tau) \frac{n}{\tau} \rfloor$            ▷ determine interval index
3:    $m \leftarrow u(k)$                        ▷ initialize total number of measurements
4:    $p \leftarrow m p'_a(k)$                    ▷ initialize prior probability
5:   while  $m < l$  do           ▷ num. of measurements must be at least  $l$ 
6:      $p \leftarrow p + p'_a(k+1)u(k+1)$      ▷ add neighbour prior
7:      $p \leftarrow p + p'_a(k-1)u(k-1)$      ▷ add neighbour prior
8:      $m \leftarrow m + u(k+1) + u(k-1)$      ▷ update meas.num.
9:   end while
10:   $p_a(t) \leftarrow p/m$            ▷ the resulting prior is a weighted average
11: end function

```

This “adaptive interval” method calculates $p_a(t)$ over several intervals in the case there is not enough data available, which is equivalent to adjusting the interval width to the number of data gathered. However, one still has to choose the minimal interval width (in our case 60 s), the periodicity (in our case $\tau = 1$ day) and l , which is the minimal number of measurements required to calculate $p_a(t)$. The optimal number of measurements l is subject to investigation in the following sections.

4.5 Modelling the spatial context

Although the main aim of this paper is investigation of long-term temporal models, for the sake of completeness, we included also the evaluation of a spatial model. The use of spatial context is motivated by the fact that certain activities are tied to specific locations, e.g. cooking typically occurs in a kitchen. Similarly to temporal models, we formalise a spatial model of activity a as a function $p_a(l)$, which represents the probability of the activity a performed by a person at location l . The process of using and building a spatial context model is similar to the interval temporal models:

$$p_a(l) \leftarrow \frac{p(o|a)p_a(l)}{\sum_{a \in A} p(o|a)p_a(l)}. \quad (9)$$

The only difference is that the location l is not calculated based on time, but on the position of the person. The combination of spatial and temporal context is considered for an extended version of this work.

4.6 Model overview and evaluation

Each of the aforementioned models has advantages and drawbacks. The main aim of this work is to investigate how these models perform when used as priors for activity recognition. We abstract from the actual algorithm that is used for classification - we simply assume that the classifier can use the priors provided by our spatial and temporal models to estimate which activity is being performed. We assume that if the priors are not provided, the performance of a given classifier depends on its confusion matrix, which represents the conditional probability distribution $p(o|a)$. The primary metric to be investigated is the overall activity recognition error, i.e. the probability that $o \neq a$.

Aruba dataset	Witham dataset
Bed to Toilet	Go Outside
Eating	Reading
Enter Home	Writing
Housekeeping	Watching a video
Leave Home	Cooking
Meal Preparation	Talking
Relax	Sleeping
Resperate	Phoncall
Sleeping	Go to toilet
Wash Dishes	Other
Work	

Table 1. Activities of the Aruba and Witham experiments.

5 Experiments

To evaluate the usefulness of the individual models for activity recognition, we compared their performance on two datasets that cover several weeks of human activity at home and at work.

The first dataset, ‘Aruba’, was collected by the Center for Advanced Studies in Adaptive Systems (CASAS) to support their research concerning smart environments [3]. The Aruba dataset contains ground-truthed activities (Table 1) of a home-bound person in a small apartment for 16 weeks. The second dataset, ‘Witham’,



Figure 1. Aruba dataset - reconstructed layout of the apartment [3].

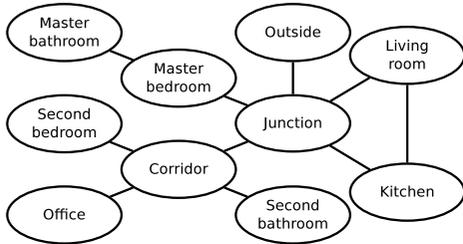


Figure 2. Aruba dataset - topological structure of the apartment.

was gathered at the Lincoln Centre for Autonomous System (L-CAS) as part of the large-scale EU-funded STRANDS project, which aims to enable long-term autonomous operation of intelligent robots in human-populated environments. The Witham dataset, which was gathered for four weeks, contains activities (Table 1) of one of the L-CAS researchers.

Both datasets are freely available as a part of the long-term dataset collection provided by the L-CAS [18, 31]. The entire pipeline that we used for our experiments is open source and is available through the website of the FreMEn temporal modelling method [17].

5.1 Aruba dataset

The Aruba dataset [3] consists of measurements collected by 41 motion, temperature and door closure sensors distributed over a $10 \times 12 \text{ m}^2$, seven-room apartment (see Figure 1) over a period of 16 weeks.

During data collection, the apartment was occupied by a single person who was occasionally visited by other people. While the starting and finishing times of activities are provided with the CASAS dataset, the location of the person is not. Thus, we partitioned the apartment into nine different locations, seven of which represent different rooms and two correspond to corridors, and estimated the person location from the events of the apartment's 33 motion detectors. Thus, the Aruba dataset contains a minute-by-minute timeline of 12 different activities performed at 9 different locations over the course of 16 weeks.

5.2 Witham dataset

The Witham dataset was collected in an open-plan office of the Lincoln Centre for Autonomous Systems (L-CAS). The office consists of a kitchenette, resting area, lounge and 20 working places that are occupied by students and postdoctoral researchers. We installed a ceiling camera that took a snapshot of the office every 10 seconds for 3 weeks, and we hand-annotated activities and locations of one of the researchers over time.

The Witham dataset contains a minute-by-minute timeline of 10 different activities performed at 10 different locations over the course of 3 weeks.

5.3 Evaluation

As mentioned before, we abstract from the internal working of the classifier itself and we simply assume that it can take into account the priors provided by our spatial and temporal models. Thus, we base our evaluation on the fact that we know the conditional probabilities $p(o|a)$ which are represented by the confusion matrix of the evaluated classifier.

The evaluation starts with the prior models being invariant to time (and location) and equal to each other, i.e.

$$p_a(t) = \frac{1}{|\mathcal{A}|}, \quad \forall a \in \mathcal{A}, \quad \forall t \in \mathbb{R}. \quad (10)$$

Then, we retrieve the activity performed at time $t = 0$ from the given dataset and, using the priors initialised by Equation (10) and known $p(o|a)$, we calculate the posterior probabilities $p_a(t|o)$ with the Bayes Equation (2). After that, we simulate the stochastic nature of the activity classification process by running a Monte-Carlo scheme over the probabilities $p_a(t|o)$ and we obtain the simulated classification result $o(t) \in \mathcal{A}$. Then, we update the binary sequences $o_a(t)$ of each activity as follows:

$$\begin{aligned} o_a(t) = 1 &\iff o(t) = a, \\ o_a(t) = 0 &\iff o(t) \neq a. \end{aligned} \quad (11)$$

These sequences are then processed by the models. Then, we increment the time by 60 s and repeat the procedure again. After 1440 iterations, which represent the activity recognition results minute-by-minute for a full day, we compare the ground truth to the results of the simulated activity recognition $o(t)$ and calculate the activity classification error for that particular day. This error is calculated for every day of the available datasets.

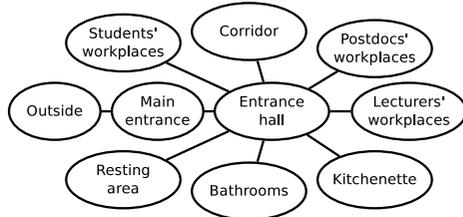


Figure 3. Witham dataset - topological structure of the apartment.

5.3.1 Evaluated classifiers

We evaluated the spatial and temporal models with three different classifiers represented by different distributions $p(o|a)$. The first ‘weak’ classifier has only a 20% probability of correct recognition, i.e. its confusion matrix has 0.2 on the diagonal and the other elements are equal. This corresponds to a high, 80% classification error. The second, ‘good’ classifier has a low, 20% classification error, which means that the diagonal elements of its confusion matrix are equal to 0.8 and the non-diagonal elements are identical.

Finally, we consider a real classifier that was evaluated on the Aruba dataset in [8]. Here, the authors evaluate the performance of a classifier that can indicate lack of evidence to perform an actual classification. This is represented by a special type of observation, called “Irregular”, which constitutes an additional column in their classifier’s confusion matrix. To obtain a square confusion matrix required by our method, the conditional probabilities represented by this additional column are uniformly redistributed across the matrix. The average value of the diagonal elements of the real classifier’s confusion matrix is 85.14% (Figure 4a).

On the Witham dataset, instead, there are no classifiers existing from previous works. To represent the $p(o|a)$ of the real classifier for the Witham dataset, we used a 10×10 submatrix of the real classifier used with the Aruba dataset (Figure 4b).

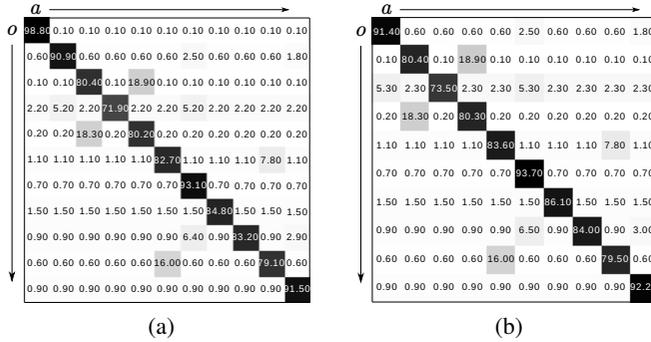


Figure 4. Confusion matrices which characterize the $p(o|a)$ of the ‘real’ classifiers for the Aruba(a) and Witham(b) datasets.

6 Experimental results

Each of the models mentioned in Section 4 depends on a parameter as summarised in Table 2. Here we discuss the sensitivity of these models to the parameter values and how well the models perform on the aforementioned datasets.

Temporal model	Parameter type	Units	Used value
GMM	num. of Gaussians	-	3
FreMEn	num. of periodics	-	2
Interval-based	interval width	minutes	5
Adaptive interv.	num. of samples	-	1000

Table 2. The list parameters for each temporal model which improve the results the most on the datasets.

6.1 Model Parameters

The FreMEn results in Figure 5 show that it can identify periodicities in the observed activities and use them to improve activity classification. Although increasing the FreMEn order does improve the classification performance, the effect is not significant, as shown in Figure 5. The only exception is the static component in the Aruba dataset, since in the case of a weak base-classifier the performance increase does not reach the same magnitude as the higher orders. This suggests that using a FreMEn model of order 3 is sufficient to obtain a good reduction of the error rate.

A similar result was observed using Gaussian Mixture Model based priors. Indeed, as can be seen in Figure 6, the results are fairly stable with respect to the order of the model.

For the Interval Models, the choice of the interval width is important, as shown in Figures 7. In the case of a weak base classifier, an interval width of one hour produced the best results. Furthermore, this choice is the only one improving the same classifier on the Aruba dataset. In all the other cases the sensitivity of the error rate is not very strong.

The Adaptive Interval Models adapt the interval width according to the available quantity of evidence, so the smaller the number of samples the closer the behaviour will be to the atomic unit (1 minute in our case). As shown in Figure 8, the Adaptive Interval Model with a single sample has the same behaviour as the static interval with 1 minute width. In the case of weak classifiers, the number of samples for the adaptation of the intervals does not influence the classification performance, and the same happens with a real base-classifier. In the case of a good classifier (20% error rate) using a higher number of samples improved the model performance.

According to our experiments, the models which are the least sensitive to the variation of classifier and to the parameter choice are the FreMEn and GMM models. Following these results, we will use the best performing cases to compare the models. The parameters used are the ones shown in Table 2.

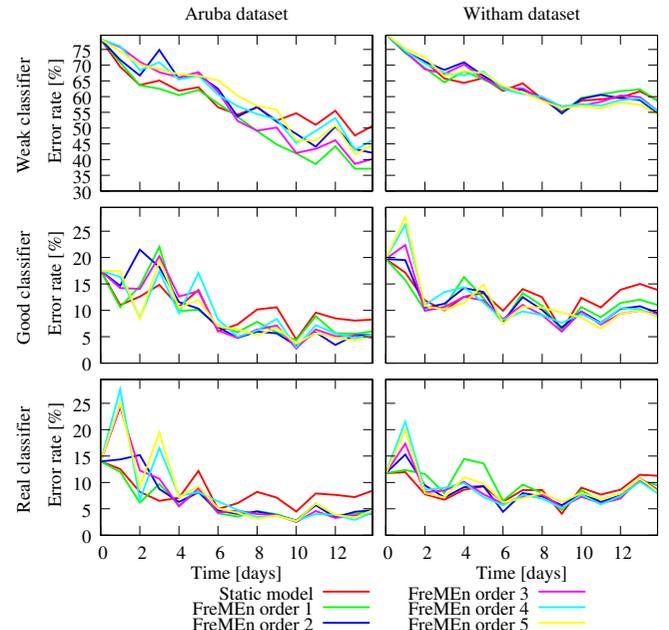


Figure 5. Impact of the number of modelled periodical processes on the FreMEn model. Best viewed in color.

6.2 Model Comparison

Our experiments showed that the use of incrementally learned models for spatial and temporal context can improve the performances of an activity recognition system. In Figure 10, it can be seen that all the temporal models improved the classification results. It is interesting to notice how the Location-based (or spatial) model on the Aruba dataset reduced the error only slightly, while on the Witham dataset it outperformed all the temporal models. This might depend on the fact that the association between activities and locations has

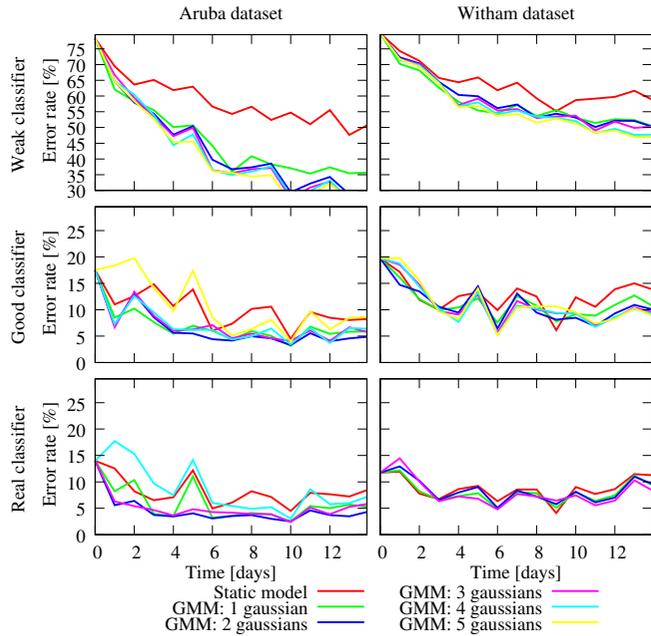


Figure 6. Impact of the number of Gaussians included on the performance of the Gaussian Mixtures. Best viewed in color.

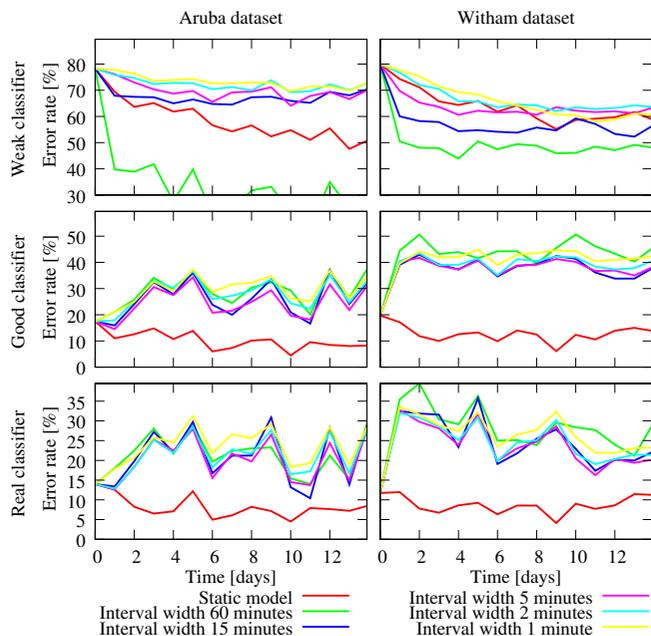


Figure 7. Impact of the interval width on the performance of the Interval models. Best viewed in color.

a higher correlation in an office-like environment rather than in a domestic one. Furthermore, we can observe that the Static component of FreMEn improves, but only slightly compared to the other models, showing the need of having higher frequencies in the weak base-classifiers. Figure 11 shows how the Interval Models tend to fail to represent the temporal context, especially without the adaptive intervals, being unable to improve the results. As in the previous case, the Location-based model works better on the Witham dataset. The remaining models are able to reduce the error rate again. Finally, Figure 12 shows how a realistic base-classifier would benefit from learning of the contextual prior probabilities. The results show that using the right model and parameters, the error rate can be significantly reduced over time, as can be seen in Figures 10, 11 and 12.

To compare the performance of the models, we performed a paired t -test on each pair of models using their error rates for the last 7 days of the experiment with the realistic classifiers. The results of the t -tests are summarized in Figure 9, showing which methods perform significantly better than others at the 95% confidence level.

Overall, the models that produced the most reliable results were the GMM and FreMEn, which had similar performances in reduction of the error rates and stability to the choice of parameters. The only real difference lies in the fact that the GMM starts to reduce the errors right from the beginning, while FreMEn tends to increase the errors, creating pronounced spikes in the error rate during the early days of execution. This effect is caused by the fact that while the GMM is given the information about daily periodicities apriori, FreMEn determines the periodicities by itself, which requires the input data to be at least twice as long as the period that it attempts to detect. The Interval-based Models can actually perform an improvement comparable to the aforementioned models, in the case of a weak classifier (Figure 10), while they appear to worsen performance if the classifier is a strong one (Figures 11, 12). Additional tests indicated that the Interval-based Models improve the performance of classifiers with accuracy lower than 70%, while their use with better classifiers might result in reduction of their performance. This might be

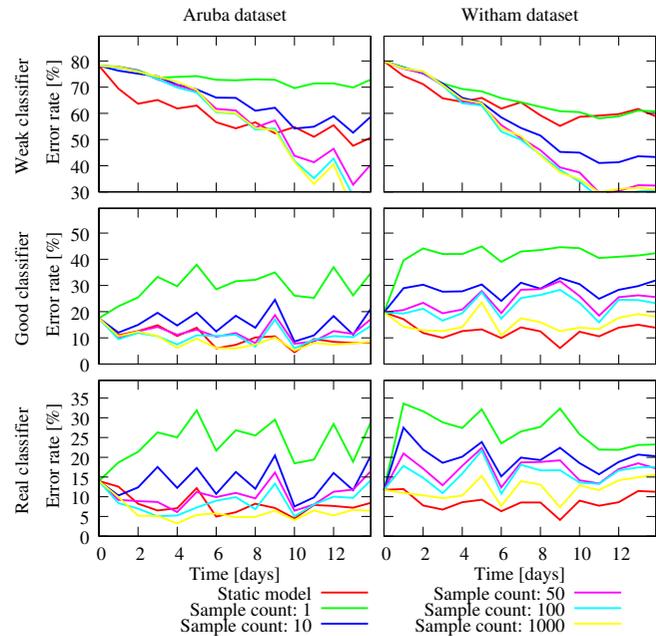


Figure 8. Impact of the number of samples used for prior estimation on the performance of the Adaptive Interval Models. Best viewed in color.

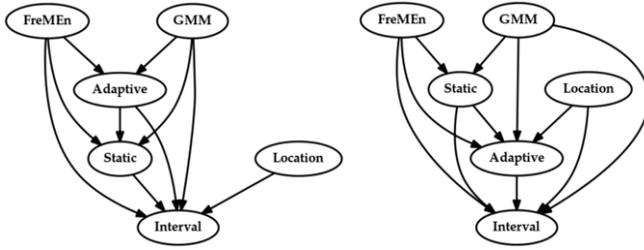


Figure 9. Comparative performance of the examined methods on the Aruba (left) and the Witham (right) dataset. An arrow from A to B indicates that method A has a significantly lower classification error than method B.

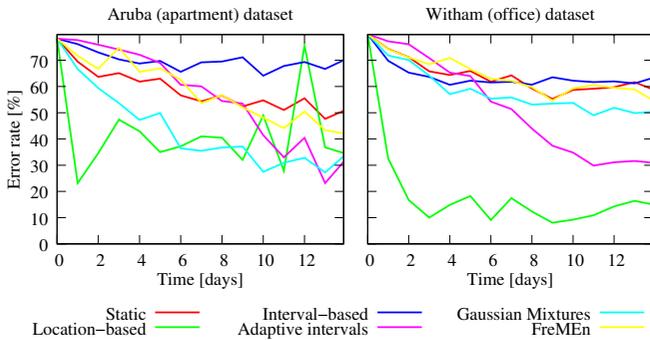


Figure 10. The impact of various spatial and temporal priors on the activity recognition error over time - weak classifier with 80% classification error.

caused by the lack of sufficient evidence during the estimation of the probability priors when the confidence of the classifier is high. The latter can be demonstrated by the fact that the adaptation of the intervals according to the actual evidence improves the model behaviour, reaching performances similar to the GMM and FreMEen.

The Location-based probability priors had discordant results on the two datasets. In the Aruba dataset, it had a negative effect on the error rate of the classification, although it improved when a strong classifier was used. This could mean that the model requires high base accuracy in complex indoor environments, in which the activities do not have a direct association to the place where they occur. On the Witham dataset instead, it did not only improve performance, but also outperformed some of the other temporal prior models. This depends directly on the high association of the activities performed with places in office environments; for example, the activity of writing on the keyboard will always be performed close to the workplace. The effect might be also related to social constraints and office etiquette. For a single-inhabited household, one does not have to consider others, which makes the activity-location constraints a bit weaker.

7 Conclusion

This paper presented a novel approach to activity recognition for indoor environments based on incremental modelling of long-term spatial and temporal context. The presented approach allows to integrate several observations of the same environment in spatial and temporal models that capture the periodic behaviour of the activity occurrences and use this knowledge to construct time and location dependent probability priors to improve the recognition of the activities. In

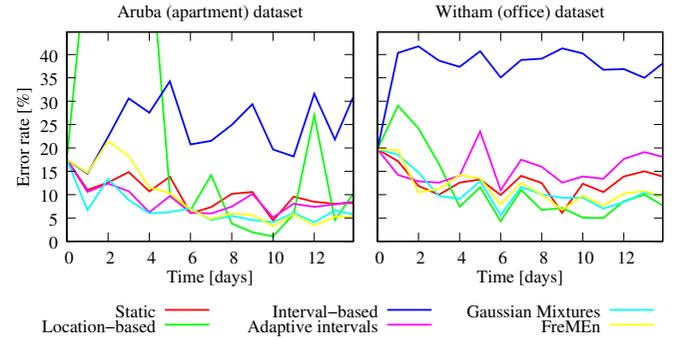


Figure 11. The impact of various spatial and temporal priors on the activity recognition error over time - good classifier with 20% classification error.

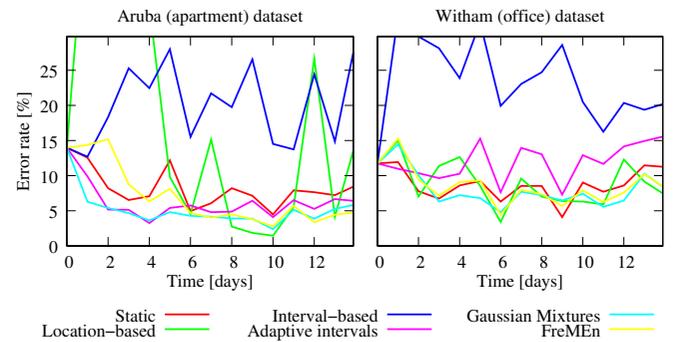


Figure 12. The impact of various spatial and temporal priors on the activity recognition error over time - real classifiers with $\sim 13\%$ classification error.

other words, given the assumption of spatial and temporal structure of the activities, we have tried to learn those patterns to improve the performance of a base classifier with different models. The ability of the models to improve the classification performance through continuous learning was evaluated on two datasets representing home and office environments over a duration of two weeks. The experiments indicated that naïve methods, based on histograms of activity, do not necessarily lead to improvement of the classification rate. On the other hand, more advanced methods reduced the error of activity classification in a significant way. The best performing models were based on the concept of Frequency Map Enhancement (FreMEen), which represents the environment dynamics in the spectral domain, and on periodic Gaussian Mixtures adjusted to model the daily patterns of people behaviour. Both of these temporal models demonstrated the ability to reduce the activity classification error through continuous learning of long-term patterns of human behaviour. The experiments also indicated that the use of spatial context might improve the performance of activity classification as well. Here, the improvement was more significant in the office environment, where the activities are strongly correlated with the location where they occur.

Possible future works will include combination of spatial and temporal models, e.g. by combining FreMEen and Gaussian processes or by applying a different temporal model in each spatial element of the environment. To allow reproduction of the experiments presented and to facilitate the work on the long-term temporal context for activity recognition, we have published the datasets [18] and the evaluation pipeline used in our experiments as open-source code [17].

Acknowledgments

The work was supported by the EU ICT project 600623 ‘STRANDS’ and the EU-H2020 project 643691 ENRICHME.

REFERENCES

- [1] H. Alerndar, H. Ertan, O.D. Incel, and C. Ersoy, ‘Aras human activity datasets in multiple homes with multiple residents’, in *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2013 7th International Conference on*, pp. 232–235, (May 2013).
- [2] Ulf Blanke and Bernt Schiele, ‘Daily routine recognition through activity spotting’, in *Location and Context Awareness*, (2009).
- [3] Diane J Cook, ‘Learning setting-generalized activity models for smart spaces’, *IEEE Intelligent Systems*, **2010**(99), 1, (2010).
- [4] D.J. Cook, N.C. Krishnan, and P. Rashidi, ‘Activity discovery and activity recognition: A new partnership’, *Cybernetics, IEEE Transactions on*, **43**(3), 820–828, (June 2013).
- [5] Claudio Coppola, Diego R. Faria, Urbano Nunes, and Nicola Bellotto, ‘Social activity recognition based on probabilistic merging of skeleton features with proximity priors from rgb-d data’, in *International Conference on Intelligent Robots and Systems (IROS)*, (2016). to appear.
- [6] Claudio Coppola, Oscar Martinez Mozos, Nicola Bellotto, et al., ‘Applying a 3d qualitative trajectory calculus to human action recognition using depth cameras’, in *Proceedings of Intelligent Robots and Systems Workshops (IROSW 2015), 2015 IEEE/RSJ International Conference on*. IEEE, (2015).
- [7] Labiba Gillani Fahad, Arshad Ali, and Muttukrishnan Rajarajan, ‘Long term analysis of daily activities in smart home’, in *Proc. of the European Symp. on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pp. 419–424, (2013).
- [8] Labiba Gillani Fahad, Asifullah Khan, and Muttukrishnan Rajarajan, ‘Activity recognition in smart homes with self verification of assignments’, *Neurocomputing*, **149**, 1286–1298, (2015).
- [9] Hongqing Fang, Lei He, Hao Si, Peng Liu, and Xiaolei Xie, ‘Human activity recognition based on feature selection in smart home using back-propagation algorithm’, *ISA transactions*, **53**(5), 1629–1638, (2014).
- [10] Diego R. Faria, Cristiano Premebida, and Urbano Nunes, ‘A probabilistic approach for human everyday activities recognition using body motion from RGB-D images’, in *IEEE RO-MAN’14*, (2014).
- [11] Diego R. Faria, Mario Vieira, Cristiano Premebida, and Urbano Nunes, ‘Probabilistic human daily activity recognition towards robot-assisted living’, in *IEEE RO-MAN’15: IEEE Int. Symposium on Robot and Human Interactive Communication. Kobe, Japan.*, (2015).
- [12] Jaime Pulido Fentanes, Bruno Lacerda, Tomáš Krajník, Nick Hawes, and Marc Hanheide, ‘Now or later? predicting and maximising success of navigation actions from long-term experience’, in *International Conference on Robotics and Automation (ICRA)*, (2015).
- [13] A. Fleury, N. Noury, and M. Vacher, ‘Introducing knowledge in the process of supervised classification of activities of daily living in health smart homes’, in *IEEE International Conference on e-Health Networking Applications and Services (Healthcom)*, (July 2010).
- [14] Mahmudul Hasan and Amit K Roy-Chowdhury, ‘A continuous learning framework for activity recognition using deep hybrid feature models’, *IEEE Transactions on Multimedia*, **17**(11), 1909–1922, (2015).
- [15] Shian-Ru Ke, Hoang Le Uyen Thuc, Yong-Jin Lee, Jenq-Neng Hwang, Jang-Hee Yoo, and Kyoung-Ho Choi, ‘A review on video-based human activity recognition’, *Computers*, **2**(2), 88–131, (2013).
- [16] Hema S Koppula, Rudhir Gupta, and Ashutosh Saxena, ‘Learning human activities and object affordances from RGB-D videos’, in *International Journal of Robotics Research*, (2012).
- [17] Tomáš Krajník. The frequency map enhancement (FrEMen) project repository. <http://fremen.uk>.
- [18] Tomáš Krajník, Jaime P. Fentanes, João Santos, Christian Dondrup, Marc Hanheide, and Tom Duckett. L-CAS datasets for long-term autonomy of mobile robots. <http://lcas.lincoln.ac.uk/owncloud/shared/datasets/>.
- [19] Tomáš Krajník, Jaime Pulido Fentanes, Grzegorz Cielniak, Christian Dondrup, and Tom Duckett, ‘Spectral analysis for long-term robotic mapping’, in *International Conference on Robotics and Automation (ICRA)*, (2014).
- [20] Tomáš Krajník, Jaime Pulido Fentanes, Oscar M. Mozos, Tom Duckett, Johan Ekekrantz, and Marc Hanheide, ‘Long-term topological localization for service robots in dynamic environments using spectral maps’, in *International Conference on Intelligent Robots and Systems (IROS)*, (2014).
- [21] Tomáš Krajník, Joao Santos, and Tom Duckett, ‘Life-long spatio-temporal exploration of dynamic environments’, in *European Conference on Mobile Robots (ECMR)*, (2015).
- [22] Tomáš Krajník, João Santos, Bianca Seemann, and Tom Duckett, ‘FrOctomap: An efficient spatio-temporal environment representation’, in *Proceedings of Towards Autonomous Robotic Systems (TAROS)*, (2014).
- [23] Ivan Lillo, Alvaro Soto, and Juan Niebles, ‘Discriminative hierarchical modeling of spatio-temporally composable human activities’, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 812–819, (2014).
- [24] Beth Logan, Jennifer Healey, Matthai Philipose, Emmanuel Munguia Tapia, and Stephen Intille, *A long-term evaluation of sensing modalities for activity recognition*, Springer, 2007.
- [25] Bryan Minor, Janardhan Rao Doppa, and Diane J Cook, ‘Data-driven activity prediction: Algorithms, evaluation methodology, and applications’, in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, (2015).
- [26] Tuan Anh Nguyen, Andrea Raspitzu, and Marco Aiello, ‘Ontology-based office activity recognition with applications for energy savings’, *Journal of Ambient Intelligence and Humanized Computing*, **5**(5), 667–681, (2014).
- [27] Nuria Oliver, Eric Horvitz, and Ashutosh Garg, ‘Layered representations for human activity recognition’, in *Multimodal Interfaces, 2002. Proceedings. Fourth IEEE International Conference on*, pp. 3–8. IEEE, (2002).
- [28] GI Parisi, C Weber, and S Wernter, ‘Self-organizing neural integration of pose-motion features for human action recognition’, *Name: Frontiers in Neurobotics*, **9**(3), (2015).
- [29] Lasitha Piyathilaka and Sarath Kodagoda, ‘Human activity recognition for domestic robots’, in *Field and Service Robotics*. Springer, (2015).
- [30] Ronald Poppe, ‘A survey on vision-based human action recognition’, *Image and vision computing*, **28**(6), 976–990, (2010).
- [31] João Santos, Tomáš Krajník, and Jaime P. Fentanes and Tom Duckett, ‘A 3d simulation environment with real dynamics: a tool for benchmarking mobile robot performance in long-term deployments’, in *ICRA Workshop on Artificial Intelligence for Long-term Autonomy*, (2016).
- [32] Jaeyong Sung, Colin Ponce, Bart Selman, and Ashutosh Saxena, ‘Human activity detection from rgbd images.’, *plan, activity, and intent recognition*, **64**, (2011).
- [33] Jaeyong Sung, Colin Ponce, Bart Selman, and Ashutosh Saxena, ‘Unstructured human activity detection from rgbd images’, in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 842–849. IEEE, (2012).
- [34] NK Suryadevara, Subhas C Mukhopadhyay, R Wang, and RK Rayudu, ‘Forecasting the behavior of an elderly using wireless sensors data in a smart home’, *Engineering Applications of Artificial Intelligence*, **26**(10), 2641–2652, (2013).
- [35] D Michael Titterington, Adrian FM Smith, Udi E Makov, et al., *Statistical analysis of finite mixture distributions*, volume 7, Wiley New York, 1985.
- [36] Kristof Van Laerhoven, David Kilian, and Bernt Schiele, ‘Using rhythm awareness in long-term activity recognition’, in *Wearable Computers, 2008. ISWC 2008. 12th IEEE International Symposium on*, pp. 63–66. IEEE, (2008).
- [37] Jiang Wang, Zicheng Liu, Jan Chorowski, Zhuoyuan Chen, and Ying Wu, ‘Robust 3D action recognition with random occupancy patterns’, in *European Conference on Computer Vision (ECCV)*, (2012).
- [38] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan, ‘Learning actionlet ensemble for 3D human action recognition’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **36**(5), 914–927, (2014).
- [39] Keze Wang, Xiaolong Wang, Liang Lin, Meng Wang, and Wangmeng Zuo, ‘3d human activity recognition with reconfigurable convolutional neural networks’, in *Proceedings of the ACM International Conference on Multimedia*, pp. 97–106. ACM, (2014).
- [40] Jiahui Wen, Mingyang Zhong, and Zhiying Wang, ‘Activity recognition with weighted frequent patterns mining in smart environments’, *Expert Systems with Applications*, **42**(17), 6423–6432, (2015).
- [41] Christian Wojek, Kai Nickel, and Rainer Stiefelhagen, ‘Activity recognition and room-level tracking in an office environment’, in *Multisensor Fusion and Integration for Intelligent Systems, 2006 IEEE International Conference on*, pp. 25–30. IEEE, (2006).

Leader-Follower MDP Models with Factored State Space and Many Followers – Followers Abstraction, Structured Dynamics and State Aggregation

Régis Sabbadin¹ and Anne-France Viet²

Abstract. The *Leader-Follower Markov Decision Processes* (LF-MDP) framework extends both Markov Decision Processes (MDP) and Stochastic Games. It provides a model where an agent (the *leader*) can influence a set of other agents (the *followers*) which are playing a stochastic game, by modifying their immediate reward functions, but not their dynamics. It is assumed that all agents act selfishly and try to optimize their own long-term expected reward. Finding equilibrium strategies in a LF-MDP is hard, especially when the joint state space of followers is factored. In this case, it takes exponential time in the number of followers. Our theoretical contribution is threefold. First, we analyze a natural assumption (*substitutability of followers*), which holds in many applications. Under this assumption, we show that a LF-MDP can be solved exactly in polynomial time, when deterministic equilibria exist for all games encountered in the LF-MDP. Second, we show that an additional assumption of *sparsity* of the problem dynamics allows us to decrease the exponent of the polynomial. Finally, we present a *state-aggregation approximation*, which decreases further the exponent and allows us to approximately solve large problems. We empirically validate the LF-MDP approach on a class of realistic animal disease control problems. For problems of this class, we find deterministic equilibria for all games. Using our first two results, we are able to solve the exact LF-MDP problem with 15 followers (compared to 6 or 7 in the original model). Using state-aggregation, problems with up to 50 followers can be solved approximately. The approximation quality is evaluated by comparison with the exact approach on problems with 12 and 15 followers.

1 Introduction

The *Leader-Follower Markov Decision Processes* (LF-MDP) framework [20] is a framework which has been recently proposed to extend both Markov Decision Processes (MDP) [14] and Stochastic Games (SG) [18, 4]. In a LF-MDP, an agent (the *leader*) partially controls the reward functions of several *followers* acting selfishly in a stochastic game, to optimize their long-term expected reward. However, the leader does not influence the dynamics of the stochastic game, which is only governed by the followers' actions. Some recent applications of the LF-MDP framework include management in organizations [21, 13].

Many real-life problems exist where a set of followers act selfishly on a dynamical system in order to maximize their own long-term

profit (in a game-theoretic fashion) while a leader, not acting directly on the system, fixes the rules of the game so that game equilibria favor its own long term objective. The following are intuitive examples of such problems:

- **Carbon tax:** here, the leader fixes a carbon emission tax level (as a modifiable rate of the total carbon emissions), while followers (firms) can take costly measures to decrease their own carbon emission rate. The followers are the only ones to emit carbon, but their rewards/costs are functions of other followers actions (through global carbon emissions) and leader's actions (through taxes). The leader has its own profit function, which depends on total carbon emission as well as total taxes paid by followers.
- **Soccer league:** In a soccer league, clubs (followers) want to maximize both the number of points they score during a whole season (which determines their ranking) and their financial profit. The league (leader) does not own a club, but wishes to maximize its own profit (through taxes on clubs' benefits) and the interest of the championship (which helps generating profit). The league's actions consist in modifying game rules, introducing salary cap, changing tax level, etc. But these do not change directly the state of the system (ranking, points...).
- **Animal health management:** Individual farmers (followers) breed cattle in an area where some disease can spread. Their aim is to maximize their own profit. Control actions (depopulation, treatment) can be applied by followers, but at some cost. The leader can decide on financial incentives to control. These cost him money if followers apply control actions, but the reduction in the disease spread rewards the leader [16].

Even though solution algorithms have been proposed for LF-MDP, based on dynamic programming [16] or reinforcement learning [21], these do not scale to the case where the followers' joint state space is factored, except under very drastic assumptions (no more than two states for each follower in [16]). Even in the case where the state space is not factored, one has to solve multiple instances of n players games, where n is the number of followers, which can only be done in time exponential in n .

In this article, we consider LF-MDP in which n is higher than in usual LF-MDP (> 10) and where the joint state space is a product of followers state spaces. After reviewing the LF-MDP model in Section 2, we show in Section 3 that under some natural assumptions about the followers (substitutability, structured dynamics), we can find equilibrium strategies for the leader and followers in time polynomial in n . Substitutability of followers, in particular, occurs when the transition and reward functions of each follower do not depend on

¹ MIAT, INRA, Toulouse, France, mail: Regis.Sabbadin@toulouse.inra.fr

² BIOEPAR, INRA, Oniris, Nantes, France, email: anne-france.viet@oniris-nantes.fr

the “labeling of other followers”. When this holds, we show that the equilibrium policies of followers which are in the same state are also the same. This suggests that a LF-MDP with substitutable followers can be replaced with smaller LF-MDP, where the number of followers is reduced to the size of the followers’ state space. However, we show that, unfortunately, the solution of the reduced LF-MDP is different from the one of the original LF-MDP in general, except when solution policies are deterministic. Fortunately, our experiments show that this seems to occur very often in practice...

While polynomial in n , time complexity of reduced LF-MDP is still exponential in the size of the state space of each follower, which keeps them hard to solve, especially when n is large (>15). We thus present an approximation of the solution through *state aggregation* which decreases the value of the exponent of n in time complexity. Finally, in Section 5, we present an illustration of the approach based on a realistic problem of coordination of farmers to limit the spread of the Porcine Reproductive and Respiratory Syndrome within a group of farms [11]. It is used, in particular, to empirically validate the quality of approximate policies obtained through state aggregation.

2 The Leader-Follower MDP model

2.1 Definition of the LF-MDP model

2.1.1 States, actions, transitions and rewards

A single leader/multiple followers finite-horizon MDP model [20] is a multiple time steps decision process involving one *leader* and n *followers*. It is defined, in the finite horizon case, as³: $\mathcal{M} = \langle n, \Sigma, A^L, \{A_i^F\}_{i=1..n}, T, r^L, \{r_i^F\}_{i=1..n}, H \rangle$, where:

- Σ is the joint state space of the leader and the followers. It can have a very general form and can be factored, e.g. as $\Sigma = S^L \times S_1^F \times \dots \times S_n^F$.
- $A^L = \{1, \dots, m\}$ is the finite leader action space.
- $A_i^F = \{1, \dots, p_i\}$ is the finite action space of follower i . For sake of notational simplicity, we will consider in this paper that all followers have the same action space $A^F = \{1, \dots, p\}$.
- $T : \Sigma \times (A^F)^n \times \Sigma \rightarrow [0, 1]$ is the joint state transition function. $T(\sigma' | \sigma, \{a_i^F\}_{i=1..n})$ is the probability to transition from state σ to state σ' , when the actions of the followers are set to $a_F = \{a_i^F\}_{i=1..n}$. Note that the leader’s actions do not influence transition probabilities.
- $r^L : \Sigma \times A^L \times (A^F)^n \rightarrow \mathfrak{R}$ is the leader instant reward function.
- $r_i^F : \Sigma \times A^L \times A^F \rightarrow \mathfrak{R}$ is the instant reward function of follower i .
- H is the horizon of the problem.

2.1.2 Policies of the leader and the followers

As usual in finite horizon sequential decision problems, we assume that agents choose their actions at time step t according to non-stationary *policies*, $\delta_t^L, \{\delta_{t,i}^F\}_{i=1..n}$. We will focus on *Markovian, stochastic* policies. $\delta_t^L(a^L | \sigma)$ is the probability that $a^L \in A^L$ is chosen by the leader at time t , given current state $\sigma \in \Sigma$. $\delta_{t,i}^F(a_i^F | \sigma, a^L)$ is the probability that $a_i^F \in A^F$ is chosen by follower i at time t , given current state σ and after having observed the current action a^L of the leader.

Policies are *deterministic*, when $\delta_t^L, \{\delta_{t,i}^F\}_{i=1..n}$ take value in $\{0, 1\}$. In this case, we write $a^L = \delta_t^L(\sigma)$ or $a_i^F = \delta_{t,i}^F(\sigma, a^L)$.

³ Transitions and rewards are considered stationary for the sake of notational simplicity, but the results can be easily extended to the non-stationary case.

2.1.3 Values of policies, equilibrium policies

Let $\Delta = \{\delta_t^L, \{\delta_{t,i}^F\}_{i=1..n}\}_{t=1..H}$ be a given joint policy of the leader and the followers. The *values* Q_Δ^L and $Q_\Delta^{F,i}$ to the leader and the followers are defined as follows, in every joint state and time step:

$$Q_\Delta^L(\sigma, t) = E \left[\sum_{t'=t}^H r_{t'}^L \mid \Delta, \sigma \right], \quad (1)$$

$$Q_\Delta^{F,i}(\sigma, t) = E \left[\sum_{t'=t}^H r_{t',i}^F \mid \Delta, \sigma \right]. \quad (2)$$

Solving a LF-MDP consists in finding an *equilibrium joint policy*, $\Delta^* = \{\delta_t^{L*}, \{\delta_{t,i}^{F*}\}_{i=1..n}\}_{t=1..H}$, for the leader and the followers⁴.

Definition 1 (LF-MDP equilibrium joint policy)

$\Delta^* = \{\delta_t^{L*}, \{\delta_{t,i}^{F*}\}_{i=1..n}\}_{t=1..H}$ is an equilibrium policy if and only if it verifies, $\forall t, \delta_t^L, \{\delta_{t,i}^F\}, \sigma$:

$$Q_{\Delta^*}^L(\sigma, t) \geq Q_{\Delta^* \downarrow \delta_t^L}^L(\sigma, t), \forall \delta_t^L, \quad (3)$$

$$Q_{\Delta^*}^{F,i}(\sigma, t) \geq Q_{\Delta^* \downarrow \delta_{t,i}^F}^{F,i}(\sigma, t), \forall i, \delta_{t,i}^F. \quad (4)$$

$\Delta^* \downarrow \delta_t^L$ (resp. $\Delta^* \downarrow \delta_{t,i}^F$) is the set of policies where the δ_t^{L*} (resp. $\delta_{t,i}^{F*}$) have been replaced with arbitrary policy δ_t^L (resp. $\delta_{t,i}^F$), $\forall t (\forall i)$.

In [20], extending results of [4] from stochastic games to LF-MDP, it was shown that there exist at least one Markovian equilibrium joint policy in which the leader equilibrium policies are deterministic. Such an equilibrium policy can be computed by a *backward induction* type algorithm [14], interleaving Nash equilibria computation steps for the followers and backward induction steps for the leader, at each time step.

2.2 LF-MDP solution algorithm

Let \mathcal{M} be a LF-MDP. An equilibrium joint policy, Δ^* can be computed backward by the following algorithm [16]:

2.2.1 Final time step

At the final time step, H , any follower i applying action $a_i^F \in A^F$ while the state is σ and the leader action is a^L , receives an immediate reward $r_i^F(\sigma, a^L, a_i^F)$, regardless of the other followers’ actions. Therefore, $\delta_{H,i}^{F*}$ is deterministic and:

$$\delta_{H,i}^{F*}(\sigma, a^L) \in \arg \max_{a_i^F \in A^F} r_i^F(\sigma, a^L, a_i^F) \text{ and}$$

$$Q_{\Delta^*}^{F,i}(\sigma, H) = \max_{a_i^F \in A^F} r_i^F(\sigma, a^L, a_i^F), \forall (\sigma, a^L). \quad (5)$$

We define the expected immediate reward of the leader at time $t \in \{1, \dots, H\}$, for a joint followers stochastic policy $\delta_t^F = \{\delta_{t,i}^F\}_{i=1..n}$:

$$r_{\delta_t^F}^L(\sigma, a^L) = \sum_{a^F} \left(\prod_{i=1}^n \delta_{t,i}^F(a_i^F | \sigma, a^L) \right) r^L(\sigma, a^L, a^F). \quad (6)$$

⁴ In the following, we use the terms *equilibrium joint policy* (or *equilibrium policy*, for short) for a solution of a LF-MDP and *Nash equilibrium* for the solution of a normal form game, in order to avoid confusion between the two notions.

Then, for the leader at time step H :

$$\begin{aligned} \delta_H^{L*}(\sigma) &\in \arg \max_{a^L \in \mathcal{A}^L} r_{\delta_H^{L*}}^L(\sigma, a^L), \\ Q_{\Delta^*}^L(\sigma, H) &= \max_{a^L \in \mathcal{A}^L} r_{\delta_H^{L*}}^L(\sigma, a^L), \forall \sigma. \end{aligned} \quad (7)$$

2.2.2 Induction step

A followers' joint equilibrium policy at time step t , given subsequent time steps joint equilibrium policies, is defined inductively as stochastic Nash equilibria⁵ of normal form n -players games ($\forall \sigma, a^L$) [10], where each player's action belongs to A^F .

The game value to player i of joint action a^F in state σ at time t under leader action a^L and assuming that a joint equilibrium policy is applied at subsequent time steps is defined as:

$$\begin{aligned} G_{\sigma, a^L, \Delta^*}^t(i, a^F) &= r_i^F(\sigma, a^L, a_i^F) \\ &+ \sum_{\sigma'} T(\sigma' | \sigma, a^F) Q_{\Delta^*}^{F, i}(\sigma', t+1). \end{aligned} \quad (8)$$

Let $\{\alpha_1^*, \dots, \alpha_n^*\}$ be a solution of the game $G_{\sigma, a^L, \Delta^*}^t$ (α_i^* is a probability distribution over A^F). A followers joint equilibrium policy is given by: $\delta_{i, i}^{F*}(a_{i, i}^F | \sigma, a^L) = \alpha_i^*(a_i^F)$ and

$$Q_{\Delta^*, a^L}^{F, i}(\sigma, t) = \sum_{a^F} \left(\prod_{j=1}^n \alpha_j^*(a_j^F) \right) G_{\sigma, a^L, \Delta^*}^t(i, a^F). \quad (9)$$

Since a followers' Nash equilibrium is determined from action a^L through Equation (9), the leader optimal policies can be computed as the solutions of a non-stationary Markov Decision Process $\langle \Sigma, A^L, \{T_{\delta_t^{F*}}\}, \{r_{\delta_t^{F*}}^L\}_{t=1..H}, H \rangle$, where the $\{r_{\delta_t^{F*}}^L\}_{t=1..H}$ have been defined previously and

$$T_{\delta_t^{F*}}(\sigma' | \sigma, a^L) = \sum_{a^F} \prod_{j=1}^n \delta_{i, j}^{F*}(a_j^F | \sigma, a^L) T(\sigma' | \sigma, a^F). \quad (10)$$

Functions $T_{\delta_t^{F*}}$ and $r_{\delta_t^{F*}}^L$ are determined before being required at each step of the backward induction algorithm.

Optimal policies $\delta_t^{L*}(\sigma)$ and value functions $Q_{\Delta^*}^{L*}(\sigma, t)$ are also computed backward:

$$\begin{aligned} \delta_t^{L*}(\sigma) &\in \arg \max_{a^L \in \mathcal{A}^L} \left\{ r_{\delta_t^{L*}}^L(\sigma, a^L) \right. \\ &+ \left. \sum_{\sigma' \in \Sigma} T_{\delta_t^{F*}}(\sigma' | \sigma, a^L) Q_{\Delta^*}^{L*}(\sigma', t+1) \right\}, \\ Q_{\Delta^*}^{L*}(\sigma, t) &= \max_{a^L \in \mathcal{A}^L} \left\{ r_{\delta_t^{L*}}^L(\sigma, a^L) \right. \\ &+ \left. \sum_{\sigma' \in \Sigma} T_{\delta_t^{F*}}(\sigma' | \sigma, a^L) Q_{\Delta^*}^{L*}(\sigma', t+1) \right\}. \end{aligned} \quad (11)$$

2.3 Computational complexity considerations

The various steps of the generic LF-MDP solution algorithm described above have different time and space complexities.

⁵ Note that since game solutions are involved, there may be more than one Nash equilibrium, leading to different equilibrium values. When solving a LF-MDP, one is usually interested into finding a single such Nash equilibrium.

2.3.1 Step 1: Normal form games generation

To compute a followers Nash equilibrium, we need to build normal form games $G_{\sigma, a^L, \Delta^*}^t$ (Equation 8). Each game has $O(n \times |A^F|^n)$ elements and there are $|\Sigma| \times |A^L|$ games. The time complexity to generate them all is thus $O(n \times |A^F|^n \times |\Sigma| \times |A^L|)$, but we require to store only one such game at a time.

2.3.2 Step 2: Followers policies computation and storage

Followers Nash equilibria are solutions of the games computed above. Storing followers' optimal policies requires space in $O(n \times |\Sigma| \times |A^F| \times |A^L|)$. Finding an (approximate) Nash equilibrium in a game is a hard task⁶ by itself [2].

2.3.3 Step 3: Leader transition and reward computation

Transition tables $T_{\delta_t^{F*}}$ are computed through Equation 10. They require $O(|\Sigma|^2 \times |A^L|)$ space to store and time $O(n \times |A^F|^n \times |\Sigma|^2 \times |A^L|)$ to compute. The reward functions $r_{\delta_t^{F*}}$ require $O(|\Sigma| \times |A^L|)$ space to store and time $O(n \times |A^F|^n \times |\Sigma| \times |A^L|)$ to compute, using Equation 6.

2.3.4 Step 4: Leader dynamic programming step

δ_t^{L*} requires $O(|\Sigma|)$ space to store and $O(|\Sigma|^2 \times |A^L|)$ to compute (Equation 11).

2.3.5 What can we do to decrease space and time complexity?

Given these complexity considerations, one can notice that the time and space complexities of all steps of solving a LF-MDP are at least either exponential in n or at least linear in $|\Sigma|$ (or both). In the case where the joint state of the problem $\sigma \in \Sigma$ is factored⁷, for example when $\Sigma = (S^F)^n$, $|\Sigma|$ is itself exponential in n .

Next, we explore the property of *substitutability of followers* in LF-MDP problems. Under this property, the above steps can be performed, exactly or approximately, at a lower complexity cost. It allows *state abstraction*, a classical property of factored MDP [6, 3, 9]. It also allows *followers abstraction*, i.e. a potential reduction of the *number of players* of all considered games. Since the complexity of solving games (and LF-MDP) is exponential in the number of followers, this may induce an important reduction in time (and space) complexity. Furthermore, in some cases, followers abstraction leads to an exact solution of the LF-MDP.

We will consider two other complexity reduction approaches.

- (i) Structured followers dynamics: We will exploit the *sparsity* of the transition matrix of each follower, to reduce further LF-MDP solution complexity, without adding new approximations.
- (ii) Joint state aggregation: An additional state abstraction approach will allow us to design an approximate LF-MDP solution method that scales to problem with 50-100 followers.

Substitutability and aggregation are particularly legible properties when the leader policy must be expressed in a simple and intelligible way and when followers' states are imperfectly observed. It is

⁶ This problem is PSPACE-complete, where PSPACE is a specific complexity class, "believed" to strictly include P.

⁷ In the most general case, $\Sigma = S^L \times S_1^F \times \dots \times S_n^F$, but we will give up the dependency on S^L to simplify notations.

common in human-based systems (economical or social), where the leader often has to consider aggregate states of “anonymous” followers.

3 Exploiting problem structure to decrease the complexity of solving LF-MDP

3.1 Followers substitutability

3.1.1 State space reduction through substitutability

In Section 2, we considered *global state* $\sigma = (s_1, \dots, s_n)$. but, in many applications, followers are *substitutable*: in the view of the leader, all followers in the same state behave identically.

Definition 2 (Substitutability of followers)

Followers are substitutable⁸ in a LF-MDP \mathcal{M} if and only if:

- $S_i^F = S_j^F$, $A_i^F = A_j^F$ and $r_i^F = r_j^F$, $\forall i, j \in \{1..n\}$ ².
- For any τ and τ_{-i} , permutations of $\{1, \dots, n\}$ where τ_{-i} leaves i at its place ($\tau_{-i}(i) = i$), we have $T(\sigma_\tau | \sigma_\tau, a_\tau^F) = T(\sigma' | \sigma, a^F)$, $r^L(\sigma_\tau, a^L, a_\tau^F) = r^L(\sigma, a^L, a^F)$ and $r_i^F(\sigma_{\tau_{-i}}, a^L, a_i^F) = r_i^F(\sigma, a^L, a_i^F)$.

Example 1 Followers are substitutable when:

- $T(\sigma' | \sigma, a^F) = \prod_{i=1}^n p(s'_i | s_i, f(\sigma), a_i^F)$,
- $r^L(\sigma, a^L, a^F) = \sum_{i=1}^n r_L(s_i, a^L, a_i^F)$ and
- $r^F(\sigma, a^L, a_i^F) = r_F(s_i, a^L, a_i^F)$,

with r_L and r_F rewards functions and provided that function f verifies $f(\sigma) = f(\sigma_\tau)$, $\forall \tau$.

Proposition 1 (Substitutability of optimal policies) If followers are substitutable in a LF-MDP \mathcal{M} , then: $\delta_{t,i}^{L*}(\sigma) = \delta_{t,i}^{L*}(\sigma_\tau)$, $\delta_{t,i}^{F*}(\cdot | \sigma, a^L) = \delta_{t,i}^{F*}(\cdot | \sigma_{\tau_{-i}}, a^L)$, $\forall, \sigma, t, \tau, \tau_{-i}$. The same property holds for $Q_{\Delta^*}^L$ and $Q_{\Delta^*}^{F,i}$ functions.

Sketch of proof: The proof follows the induction. Step H is easy, using the invariance of r^L and r^F from which easily follows the substitutability of $r_{\delta_H^F}$, $\delta_{H,i}^{F*}$, $Q_{\Delta^*}^{F,i}(\cdot, H)$, δ_H^{L*} and $Q_{\Delta^*}^L(\cdot, H)$. The induction step goes as easily, once we have noticed that games $G_{\sigma, a^L, \Delta^*}^t$ are also substitutable. \square

An important consequence of this proposition is that if a LF-MDP is substitutable, then it can be replaced with an equivalent LF-MDP which (reduced) state space, denoted Σ^L is composed of the *equivalence classes* of Σ for the “permutation” relation $\equiv: \sigma \equiv \sigma'$ iff $\exists \tau, \sigma' = \sigma_\tau$.

It is easy to show that any equivalence class can be represented by a *reduced global state*, modeling the number of followers in each of the $k = |S^F|$ states. This reduced global state can thus be represented by the tuple of integers:

$$c = (c_1, \dots, c_k) \in \{0, \dots, n\}^{|S^F|}, \text{ where } \sum_{h=1}^k c_h = n. \quad (12)$$

Σ^L is thus the set of tuples satisfying Equation 12. The size of the reduced state space of the leader is $|\Sigma^L| = \binom{n+k-1}{k-1} = O(n^k)$, instead of k^n for $|\Sigma|$.

The time and space complexities of the steps of the algorithm (Table 1) are now reduced, by replacing every occurrence of $|\Sigma|$, with $|\Sigma^L|$. Step 4 becomes polynomial in n .

⁸ Note that the current definition has links with *stochastic bisimilarity* [6]. However, stochastic bisimilarity only handles factored state space, not factored action space.

3.1.2 Action space reduction through substitutability

Followers’ substitutability implies the substitutability of followers policies $\delta_{t,i}^{F*}$. In the case where followers’ policies are deterministic, then any two followers’ actions are identical when the followers are in the same state. So, the action profiles of the n followers, (a_1, \dots, a_n) , are limited to profiles in which all followers in the same state perform the same action. In this way, a followers’ action profile is of the form $d^F = (d_1^F, \dots, d_k^F) \in (A^F)^k$, leading to decreased joint action space size ($|A^F|^k$ instead of $|A^F|^n$).

However, optimal policies of followers can be stochastic, meaning that two identical policies may lead to different actions choices. Thus, it may be that even under the substitutability assumption, a Nash equilibrium for the followers may give non-zero probabilities to all the $|A^F|^n$ potential profiles. For equilibrium computation, in order to limit the size of the joint action space, we make the approximation that even when followers policies are stochastic, followers in the same state actually implement the same action.

This suggests the following modifications to Equations 8 and 9, which are now used to compute games and followers policies of dimension $k: \forall h = 1, \dots, k$,

$$G_{c, a^L, \Delta^*}^t(h, d^F) = r_h^F(c, a^L, d_h^F) + \sum_{c'} \bar{T}(c' | c, d^F) Q_{\Delta^*}^{F,h}(c', t+1). \quad (13)$$

And, if $\{\alpha_1^*, \dots, \alpha_k^*\}$ is a stochastic Nash equilibrium of the above game, $\delta_{t,h}^{F*}(d_h^F | c, a^L) = \alpha_h^*(d_h^F)$ and

$$Q_{\Delta^*, a^L}^{F,h}(c, t) = \sum_{d^F} \prod_{j=1}^k \alpha_j^*(d_j^F) G_{c, a^L, \Delta^*}^t(h, d^F). \quad (14)$$

We will see in the next subsection how \bar{T} is defined, but first remark that with the above approximation it is assumed that actions for all followers are chosen in the following way : An action $d_h^F \in A^F$ is chosen at random for each $h \in 1, \dots, k$, following distribution $\delta_{t,h}^{F*}(\cdot | c, a^L)$, then all followers j which are in state h apply the same action d_h^F .

When all Nash equilibria are deterministic, this assumption holds, even in the original LF-MDP. In all other cases, the computed equilibria are only approximate. Notice also that when replacing Equations 8 and 9 with Equations 13 and 14, the time complexities of the steps are reduced. For Step 1, for example, it becomes $O(k|A^F|^k|\Sigma^L||A^L|) = O(n^k)$. The space complexity of Step 2 becomes $O(n^k)$ and the games to solve are k -players games so their time complexity is independent on n . In Step 3, Equation 10 can be replaced with: $\forall c, c' \in \Sigma^L, a^L \in A^L$,

$$T_{\delta_{t,i}^{F*}}(c' | c, a^L) = \sum_{d^F} \prod_{h=1}^k \delta_{t,h}^{F*}(d_h^F | c, a^L) \bar{T}(c' | c, d^F). \quad (15)$$

The complexity of Step 3 also depends on the computation of \bar{T} , which we now describe.

3.1.3 Transitions in the reduced model

Making use of the reduction of the state and action spaces, we can compute an aggregate transition function $\bar{T} : \Sigma^L \times (A^F)^k \times \Sigma^L \rightarrow [0, 1]$. $\bar{T}(c' | c, d^F)$ is the probability to transition from any state $\sigma^c \in \Sigma$ “compatible” with c to any state $\sigma^{c'} \in \Sigma$ “compatible”

with c' , when applying an action $a^F = (a_1^F, \dots, a_n^F)$ “compatible” with $d^F = (d_1^F, \dots, d_k^F)$.

To be more precise, \bar{T} is defined as: $\forall c, c', d^F$,

$$\bar{T}(c'|c, d^F) = \sum_{\sigma' \models c'} T(\sigma' | \sigma^c, \dots, \underbrace{d_h^F, \dots, d_h^F}_{c_h}, \dots), \quad (16)$$

where $\sigma^c = (\underbrace{1, \dots, 1}_{c_1}, \dots, \underbrace{k, \dots, k}_{c_k})$ is a full state compatible with c . $\sigma' \models c'$ means that if $\sigma' = (s'_1, \dots, s'_n)$ then, $\forall h = 1, \dots, k$, there are exactly c'_h indices i such that $s'_i = h$.

Proposition 2 \bar{T} defined in Equation 16 is well-defined.

Proof: By well-defined, we mean that $\bar{T}(c'|c, d^F)$ does not depend on the actual choice of σ^c compatible with c . Indeed, this holds thanks to the substitutability of T . \square

The space and time complexities of computing \bar{T} are $O(|\Sigma^L|^2 |A^F|^k)$ and $O(n|\Sigma||\Sigma^L||A^F|^k)$. Indeed, since we have to sum over all σ' compatible with c' and this for all c' to compute \bar{T} , we still have to explore Σ entirely once. Hence, computing \bar{T} is still exponential in n , even though the exponent has been reduced from $2n$ to n .

However, for a fixed c and given follower state h , the c_h followers in state h change their state to a new repartition (c_h^1, \dots, c_h^k) (with $c_h^1 + \dots + c_h^k = c_h$) according to a multinomial distribution $\Gamma_{c_h, d_h^F}^h(c_h^1, \dots, c_h^k)$ of parameters $\{p(h'|h, c, d_h^F)\}_{h'=1..k}$. Furthermore, $\forall c, c', d^F$,

$$\bar{T}(c'|c, d^F) = \sum_{\substack{(c_h^1, \dots, c_h^k), \\ \sum_{h'=1}^k c_h^{h'} = c_h, \\ \sum_{h'=1}^k c_h^{h'} = c'_h}} \prod_{h=1}^k \Gamma_{c_h, d_h^F}^h(c_h^1, \dots, c_h^k).$$

Proposition 3 The time complexity⁹ of computing \bar{T} is $O(n^k |\Sigma^L| |A^F|^k) = O(n^{2k} |A^F|^k)$.

Sketch of proof: For any $c = (c_1, \dots, c_k)$, each c_h can transition to $\binom{c_h+k-1}{k-1}$ k -tuples (c_h^1, \dots, c_h^k) . Therefore, for fixed c and d^F , we need to evaluate only $\sum_h \binom{c_h+k-1}{k-1}$ terms to compute the values $\bar{T}(c'|c, d^F)$, for all feasible c' . The result comes from the fact that $\sum_h \binom{c_h+k-1}{k-1} = O(n^k)$. \square

3.1.4 Approximate reduced LF-MDP: Followers abstraction

The construction of the reduced versions \bar{r}^L and \bar{r}^F is immediate in a LF-MDP with substitutable followers. In the end, we get an approximate LF-MDP $\bar{\mathcal{M}} = \langle k, \Sigma^L, A^L, \prod_{h=1}^k A_h^F, \bar{T}, \bar{r}^L, \{\bar{r}_h^F\}, H \rangle$, which is only exponential in k (and no more in n) to solve. Note the following important proposition:

Proposition 4 In the case where the joint followers equilibrium policies of approximate LF-MDP $\bar{\mathcal{M}}$ are deterministic, they can be used to build deterministic joint equilibrium policies for the original LF-MDP \mathcal{M} . This also holds for the leader equilibrium policies which are always deterministic.

⁹ The complexity results given in the paper are not the tightest possible, for ease of exposition. Here, for example, $\sum_h \binom{c_h+k-1}{k-1} = O(kn^{k-1}) = O(n^k)$.

Sketch of proof: Just note that the only approximation in the process we have described concerns the interpretation of stochastic Nash equilibria. When Nash equilibria are pure, there is a complete equivalence between \mathcal{M} and $\bar{\mathcal{M}}$. \square

In the view of Proposition 4, two facts should be highlighted: (i) Since games may have more than one equilibrium, we should return in priority deterministic equilibria of the games and (ii) Even though the above approach may not be exact in all cases, it is possible to check, after equilibrium policies have been computed in the reduced LF-MDP, whether these provide equilibrium policies in the original LF-MDP. It is enough to check that the games which have been solved for all (t, c, a^L) have a deterministic equilibrium.

3.1.5 Connection with state / action abstraction in MDP and Stochastic Games

The state-space reduction approach that we have described in Section 3.1.1 is very close to state aggregation approaches in MDPs [9, 6, 3, 19] or in stochastic games [17]. We basically identify the case where state aggregation leads to an optimal solution in a LF-MDP (extending both MDP and SG cases). What is especially useful here, is that state aggregation leads to an exponential reduction of the state space.

More original is the form of *followers abstraction* in stochastic games that we propose in Section 3.1.2. State and action spaces abstraction have already been proposed in the field of game theory [5, 8] or in stochastic games [17]. However when action spaces are abstracted, as in [17] (the closest work to ours we have found), only followers action spaces A^F are abstracted¹⁰. In the case we consider, action spaces A^F are already small and need not be abstracted. Instead, we propose to “lump” players together, which is a way to abstract a game where the joint action space is large, due to the number of players and not to the size of individual action spaces. To our knowledge, this is the first proposition in that direction in (stochastic) game theory, or LF-MDP.

3.2 Structured dynamics of followers

Recall that a transition $c \rightarrow c'$ is obtained through the aggregation of n individual changes of states $h \rightarrow h'$. Each transition has probability $p(h'|h, c, d_h^F)$ and is independent of the other transitions, c and d_h^F being given.

We can further decrease complexity when the dynamics is structured, i.e. when a follower in state h can only transition to a few possible states, say at most $N_{Succ} < k$ states.

Proposition 5 When, in LF-MDP $\bar{\mathcal{M}}$, followers in any state h transition to at most N_{Succ} possible states, Computing \bar{T} requires $O(n^{k+N_{Succ}} |A^F|^k)$ time.

Sketch of proof: The number of terms that should be computed for a given pair (c, d^F) is reduced to $\sum_h \binom{c_h+N_{Succ}}{N_{Succ}-1} = O(n^{N_{Succ}})$. \square

4 State aggregation

State aggregation allows us to further decrease the exponent of n in time complexities and suppress it from space complexities, but there is no performance guarantee anymore on the returned policies, and we must resort to experiments to empirically evaluate the merits of this approach (which we will do in the case study section). It consists

¹⁰ The authors considering stochastic games, there is no leader.

Table 1. LF-MDP objects complexity, with and without suggested simplifications/approximations.

	Naive	Substitutability	Aggregation
State space size	$ \Sigma = O(k^n)$	$ \Sigma^L = O(n^k)$	$ \overline{\Sigma^L} = O(K^k)$
Action space size	$ A^F ^n$	$ A^F ^k$	$ A^F ^k$
Joint state transition	$ T = O(k^{2n} A^F ^n)$	$ \overline{T} = O(n^{2k} A^F ^k) \quad \tilde{T} = O(n^{k+N_{Success}} A^F ^k)$	$ \hat{T} = O(K^{2k} A^F ^k)$
Single game size	$ G_{\sigma, a^L, \Delta^*}^t = O(n A^F ^n)$	$ G_{c, a^L, \Delta^*}^t = O(k A^F ^k)$	$ G_{\kappa, a^L, \Delta^*}^t = O(k A^F ^k)$
Nb games / time step	$nb = A^F ^n A^L $	$nb = A^F ^k A^L $	$nb = K^k A^L $

in considering a partition of the set $\{0, \dots, n\}$ into $K + 2$ intervals, where K is an integer dividing N : $I_0 = \{0\}$, $I_{K+1} = \{n\}$ and $I_i = \{\frac{(i-1)n}{K} + 1, \dots, \frac{i.n}{K}\}$, $\forall i = 1, \dots, K$. Then, we define *aggregate states* as tuples of integers $(\kappa_1, \dots, \kappa_k) \in \{0, \dots, K\}^k$. These correspond to every combinations of sets $I_{\kappa_1} \times \dots \times I_{\kappa_k}$, such that there exists a reduced state $c \in \Sigma^L$, $c_h \in I_{\kappa_h}$, $\forall h = 1, \dots, k$. Let $\overline{\Sigma^L}$ denote the set of aggregate states. Obviously, $|\overline{\Sigma^L}| = O(K^k)$, which is now independent of n .

We define two functions, relating reduced and aggregate states:

- $\kappa(c) \in \overline{\Sigma^L}$ is the unique aggregate state which is compatible with the reduced state $c \in \Sigma^L$,
- $\hat{c}(\kappa) \in \Sigma^L$ is a *representative* of the aggregate state $\kappa \in \overline{\Sigma^L}$, defined as the compatible reduced state which components \hat{c}_h are the ‘‘closest possible’’ to the centers of intervals I_{κ_h} . These can be computed, e.g. by solving small size integer linear programs (one for each κ).

Finally, it is possible to generate a LF-MDP on the aggregate state space. The transition function $\hat{T} : \overline{\Sigma^L} \times (A^F)^k \times \overline{\Sigma^L} \rightarrow [0, 1]$ is defined as

$$\hat{T}(\kappa' | \kappa, d^F) = \sum_{c', \kappa(c') = \kappa'} \tilde{T}(c' | \hat{c}(\kappa), d^F). \quad (17)$$

Reward functions \hat{r}^L and \hat{r}_h^F are defined accordingly: $\hat{r}^L(\kappa, a^L, d^F) = \bar{r}^L(\hat{c}(\kappa), a^L, d^F)$ and $\hat{r}_h^F(\kappa, a^L, d_h^F) = \bar{r}_h^L(\hat{c}(\kappa), a^L, d_h^F)$.

Proposition 6 *The time complexity of computing \hat{T} is in $O(|A^F|^k K^k n^k)$.*

The proof is obvious, using the form of Equation 17. \square

Complexity still depends on n , due to the sum over all c' , but all other steps are independent on n . Note that since we are interested in probabilities to transition to aggregate states κ' and not reduced states c' , we could estimate these by simulating the transitions. So doing, the sample size would be a function of $|\overline{\Sigma^L}|$ and not $|\Sigma^L|$, and thus independent on n .

Solution policies of this LF-MDP will assign the same policies to all reduced states compatible with the same aggregate state, and of course to the full states compatible with these reduced states. This may result in a loss of quality of the returned strategies. In Section 5, we illustrate the impact of state aggregation on a disease control problem.

The contributions of this paper to LF-MDP complexity reduction are summarized in Table 1.

5 Case study

In order to demonstrate the practical interest of exploiting substitutability, structured dynamics and state aggregation in LF-MDP, we

will focus on a case study concerning a problem of coordination of farmers to limit the spread of the Porcine Reproductive and Respiratory Syndrome (PRRS) within a group of farms [11].

This problem, as many others in animal health management (or other applications, listed in the introduction) involves some followers and one leader that has the ability to indirectly control their dynamics. The followers each own one herd that may be infected by an endemic disease (PRRS). PRRS is an endemic, non-regulated disease, meaning that treating infected herds is non-compulsory. However, in order to limit the spread of the disease, which impacts pig growth and meat production, farmers associations may propose financial incentives, to limit the cost of treatments to farmers.

This is typically a case where the followers directly influence the dynamics of the system (disease spread) through management, while the leader (the Association) influences their reward functions (through incentives). The farmers are assumed to maximize their own long-term profit, while the association maximizes its own, including both an infection-spread related reward and incentive costs.

5.1 The LF-MDP model of PRRS spread control

5.1.1 The model

We consider a group of n farmers taking decisions for their own herd (followers). PRRS is modeled with a compartmental approach, with five compartments ($k = |S^F| = 5$):

- S : Susceptible (=non-infected).
- S_b : Susceptible with management (biosecurity measure).
- I : Infected by PRRS.
- I_0 : Infected, with control measure starting. Not fully efficient yet
- I_C : Infected, controlled (efficient).

Actions are different in each herd state, but at most two actions (‘do nothing’, ‘manage’) are available in each state.

These govern the transition probabilities from each state¹¹ (see Figure 1):

- S : A S herd becomes I (infected) with probability β_S . Else, either stays S (if action=do nothing) or becomes S_b (action=manage).
- S_b : For a S_b herd, only ‘manage’ action is available. Transition probability to I is reduced to $\nu\beta_S$.
- I : Transitions are deterministic: to I (do nothing) or to I_0 (manage).
- I_0 : Only ‘manage’ action is available: transition to I_C with probability ψ , modeling a stochastic sojourn time in state I_0 .
- I_C : Transitions are deterministic: To I_C (do nothing) or to S (manage=depopulation).

¹¹ Self-transition probabilities are omitted, for sake of readability.

Table 2. Values of parameters in the three sets used for $n = 15$. In addition, $\psi = 0.5$, $\beta = (0, 0, 0.08, 0.06, 0.01)$, $\beta_{out} = 0.005$ and $L^L = red \times L^F$.

Set	ν	L^F	c^F	c^L	<i>perc</i>	<i>red</i>
2001	0.5	(0,0,6,5,4)	(4,1,4,2,101)	(0,3)	0.5	0.75
824	0.73	(0,0,8.76,5.84,2.92)	(8.53,1.46,12.79,2.92,147.46)	(0,4.38)	0.26	0.73
131	0.7	(0,0,4.8,5.6,2.8)	(7.84,1.4,11.76,2.8,101.4)	(0,4.2)	0.7	0.7

Note that while ν and ψ are constants, β_S is a function of the state of all herds. We assume that, the group of herds being tightly linked (geographically, but also by sales and purchases), β_S only depends on the total numbers of herds in each of the 5 states (equation 18). This implies the substitutability of the transition model.

$$\beta_S(c) = \frac{1}{n} \sum_{h=1}^k \beta(h)c(h) + \beta_{out} \quad (18)$$

where the $\beta(h)$ and β_{out} are real-valued parameters.

The reward functions are the following:

$$r^L(\sigma, a^L, a^F) = -c^L(a^L) - \sum_{i=1}^n c^F(s_i)q^L(a^L, a_i^F) - L^L(s_i),$$

$$r^F(\sigma, a^L, a_i^F) = -E_{s_i'} [L^F(s_i')] - \sum_{i=1}^n c^F(s_i)q^F(a^L, a_i^F),$$

where

- c^L is the cost of the leader action,
- $L^X(s_i)$ are the losses of the herd in state s_i for either the farmer if $X = F$ or the leader if $X = L$,
- $c^F(s_i)q^L(a^L, a_i^F)$ is the amount of the action cost of a herd in state s_i paid by the leader,
- $c^F(s_i)q^F(a^L, a_i^F)$ is the amount of the action cost of a s_i herd paid by the farmer.

Where $q^L(a^L, a_i^F)$ and $q^F(a^L, a_i^F)$ are defined by:

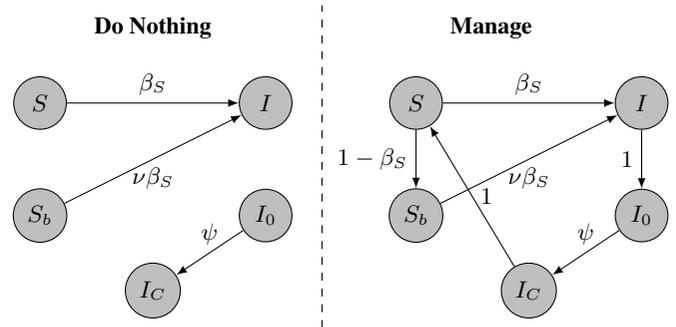
- $q^L(a^L, a_i^F) = perc$ if $a^L = 1$ and $a_i^F = 1$,
- $q^L(a^L, a_i^F) = 0$ else;
- $q^F(a^L, a_i^F) = 1$ if $a^L = 0$ and $a_i^F = 1$,
- $q^F(a^L, a_i^F) = 1 - perc$ if $a^L = 1$ and $a_i^F = 1$,
- $q^F(a^L, a_i^F) = 0$ else.

It can be easily checked that they are also substitutable (note that expectation $E_{s_i'}[\cdot]$ is taken with respect to a substitutable transition function).

5.1.2 The experiments

The comparison with the exact model was run in Matlab for $H = 10$ with $n = 12$, $K \in \{3, 4, 6\}$ and $n = 15$, $K \in \{3, 5\}$. For $n = 12$ and $K = 3$, we generated 1000 sets of parameters values with varying values of costs and losses. For most sets, the leader policy consisted in always doing nothing. For the comparison with the exact model (using the substitutability assumption), we selected 17 scenarios. For comparison with the case $n = 15$, due to the computing time to solve the exact model, we explored only the three sets of parameter values described in Table 2.

We evaluated the impact of the number of classes, K , on various model results at the leader level. As far as the followers policies were concerned, we simply checked whether or not they were deterministic.

**Figure 1.** Transitions between follower's states.

To compare the leader optimal policies computed with the different K (noted δ_K) with the global optimal policy δ^* , we computed 3 indicators :

- *Last_Diff*: The absolute difference between times of the last leader management, between the aggregate and global optimal policies.
- *#Diff*: Number of time steps for which both policies were not equal.
- *Max_Gap*: Maximum (over all time steps) proportion of states for which both policies differ.

Different policies may lead to similar distributions over states at each time step. So, in order to further evaluate our approximation, we compared the distributions over states at time step H obtained when applying δ^* or δ_K . To compute these distributions, we have to choose an initial distribution (time step 1). We considered two initial distributions : (i) Γ_U uniform over all states, and (ii) Γ_E uniform only on “highly infected” states (states where around 40% of followers are in state S or S_b and around 40% are in state I_C). To compare the distributions resulting from δ_K and δ^* , we computed the Bat-tacharyya distances [1] between them, denoted DB_U and DB_E for Γ_U and Γ_E respectively.

As different policies may be equivalent in terms of values, we evaluated the impact of the approximation on the objective function. We computed a distance derived from the root mean square error (RMSE):

$$RMSE_{-x} = \sqrt{\sum_{c \in \Sigma^L} ((V^K(c) - V^*(c))^2 \times \Gamma_x)}.$$

with $x = U$ or $x = E$ and $V^K(\cdot)$ is the value function of policy δ^K .

5.2 Experimental results

5.2.1 Empirical complexity reduction

The decrease in state space size using aggregation of course depends on K and N (Table 3). One should note that state aggregation does not partition the state space into uniform clusters (Fig. 2). We will see in the following section that this does not have a dramatic impact on the performance of the computed policies.

Table 3. Size of the leader states space according to the value of K ($|\Sigma^L|$ if $K = n$ or $|\Sigma^L|$ else; – if not applicable)

n	$K = n$	$K = 3$	4	5	6	10	25
12	1,820	236	361	-	745	-	-
15	3,876	251	-	631	-	-	-
20	10,626	-	496	806	-	10	-
50	316,251	-	-	876	-	6,376	79,101
100	4,598,251	-	-	876	-	6,376	114,526

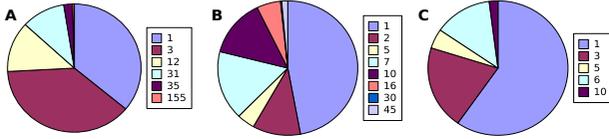


Figure 2. For $n = 12$, repartition of the number of global states per aggregate state for (A) $K=3$, (B) $K=4$ and (C) $K=6$.

5.2.2 Comparison results

Followers’ optimal policies were found deterministic in all parameters sets tested. The leader policies varied when K varied. When considering the time steps where the action ‘manage’ was retained by the leader in at least one state, we had two possible profiles for the leader policy: management only at one time step (for example for parameters set 824) or management in several time steps (for example the sets 2001 and 131). In the tested sets, these profiles were kept when changing K and n . In the three tested configurations, the last management time step was the same for all K , both for $n = 12$ and $n = 15$ ($Last_Diff = 0$). Still, the policies differed in some states when K varied ($\#Diff > 0$). The proportion of states where the policies differed also varied (Max_Gap between 1.5% and 30% for the tested sets).

The impact on the final distributions and expected values varied. For parameters set 824, the impact was null for the distributions and very low for the values. It can be explained by the fact that the policies differed in only one time step. For other parameters sets, the final distributions varied with K , but the variations were low. The impact of the initial distribution (Γ_U or Γ_E) on the variation of expected value with different K was not consistent between different parameters’ sets and number of followers (Fig. 3). Overall, even though this conclusion should be taken with caution, given the small number of configurations tested, it seems that the approximation becomes better when K increases, which seems logical.

6 Concluding remarks and future work

In this article, we have proposed approximation methods to compute solutions to LF-MDP problems. Even though these experiments

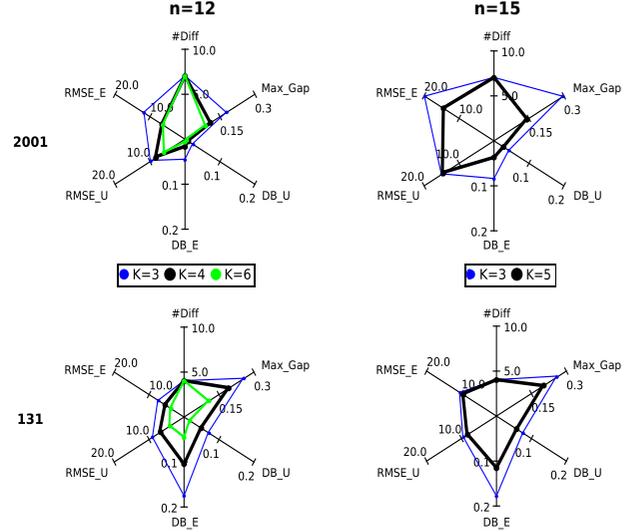


Figure 3. For parameter sets 2001 and 131 (table 2), comparison of the approximated results to the exact ones for various K and n .

have not been reported in the paper, we observed that the suggested approximations permitted to solve approximately problems with up to 100 followers (and $K = 5$). The number of classes for state aggregation were observed in the three parameter sets to be correlated to the approximation quality on the case study whatever n , but this obviously has to be confirmed by further experiments.

Note that, in the proposed approach, the computed equilibria are equilibria in the reduced LF-MDP, not in the original one. However, we have a way to check *a posteriori* (apart in the case of state aggregation), whether all the returned equilibria of games are deterministic. If so, the solution is exact. If not, it is approximate. We leave for further research the question of approximability of LF-MDP equilibrium strategies, which is certainly worth considering and extends that of stochastic Nash equilibrium approximation.

One perspective of this work is to model the heterogeneity of followers, which can be done rather straightforwardly by ‘‘duplicating’’ followers state spaces and modeling followers types by different reward functions. Transitions are allowed between the duplicated state spaces if and only if followers can change their type over time. The new problem is still a LF-MDP, with more states, but potentially more ‘‘structure’’ as well. If followers keep the same ‘‘type’’ all along the problem, no transition is allowed between the duplicated state spaces.

As mentioned in the end of Section 4, the dependency on n of time complexity could be suppressed if *simulation-based* approaches were used to approximate \hat{T} (e.g. Bayesian RL [7]). Bayesian approaches have also been used in the framework of *Partially Observed MDP (POMDP)* [15]. Partial observability of followers states could be considered as well in LF-MDPs, leading to LF-POMDP models, mixing dec-POMDP with Bayesian games, in the line of [12], for example.

Acknowledgments

This work was supported by the French Research Agency, program Investments for the future, project ANR-10-BINF-07 (MIHMES), by the European fund for the regional development FEDER and by the INRA Flagship program SMaCH. We also thank IJCAI and ECAI reviewers.

REFERENCES

- [1] A. Bhattacharyya, 'On a measure of divergence between two statistical populations defined by their probability distributions', *Bulletin of the Calcutta Mathematical Society*, **35**, 99–109, (1943).
- [2] X. Cheng and X. Deng, 'Settling the complexity of 2-player nash equilibrium', in *Proceedings of FOCS*, (2006).
- [3] N. Ferns, P. Panangaden, and D. Precup, 'Metrics for finite Markov decision processes', in *Proceedings of the 20th conference on Uncertainty in Artificial Intelligence*, pp. 162–169. AUAI Press, (2004).
- [4] J. Filar and K. Vrieze, *Competitive Markov Decision Processes*, Springer, 1996.
- [5] A. Gilpin and T. Sandholm, 'Lossless abstraction of imperfect information games', *Journal of the ACM*, **54**(5), (2007).
- [6] R. Givan, T. Dean, and M. Greig, 'Equivalence notions and model minimization in Markov decision processes', *Artificial Intelligence*, **147**(1), 163–223, (2003).
- [7] A. Guez, D. Silver, and P. Dayan, 'Efficient bayes-adaptive reinforcement learning using sample-based search', in *Proceedings of NIPS 2012*, (2012).
- [8] J. Hawkin, R. Holte, and D. Szafron, 'Automated action abstraction of imperfect information extensive-form games', in *National Conference on Artificial Intelligence (AAAI)*, (2011).
- [9] L. Li, T. J. Walsh, and M. L. Littman, 'Towards a unified theory of state abstraction for MDPs.', in *ISAIM*, (2006).
- [10] R.B. Myerson, *Game Theory: Analysis of Conflict*, Harvard University Press, 1997.
- [11] G. Nodelijk, 'Porcine reproductive and respiratory syndrome (prrs) with special reference to clinical aspects and diagnosis: A review', *Veterinary Quarterly*, **24**, 95–100, (2002).
- [12] F. A. Oliehoek, M.T.J. Spaan, J.S. Dibangoye, and C. Amato, 'Heuristic search for identical payoff bayesian games', in *Proc. of AAMAS 2010*, pp. 1115–1122, (2010).
- [13] E. L. Plambeck and S. A. Zenios, 'Performance-based incentives in a dynamic principal-agent model.', *Manufacturing & service operations management*, **2**, 240–263, (2000).
- [14] M. Puterman, *Markov Decision Processes*, John Wiley and Son, 1994.
- [15] S. Ross, J. Pineau, B. Chaib-Draa, and P. Kreitmann, 'A bayesian approach for learning and planning in partially observable Markov decision processes', *Journal of Machine Learning Research*, **12**, 1729–1770, (2011).
- [16] R. Sabbadin and A.F. Viet, 'A tractable leader-follower MDP model for animal disease management', in *27th AAAI Conference on Artificial Intelligence*, (2013).
- [17] T. Sandholm and S. Singh, 'Lossy stochastic game abstraction with bounds', in *EC '12 Proceedings of the 13th ACM Conference on Electronic Commerce*, pp. 880–897, (2012).
- [18] L. S. Shapley, 'Stochastic games', *Proc. Nat. Academy of Science*, **39**, 1095–1100, (1953).
- [19] J. Sorg and S. Singh, 'Transfer via soft homomorphisms', in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, (2009).
- [20] K. Tharakunnel and S. Bhattacharyya, 'Leader-follower semi-Markov decision problems: theoretical framework and approximate solution', in *IEEE international conference on Approximate Dynamic Programming and Reinforcement Learning (ADPRL)*, pp. 111–118, (2007).
- [21] K. Tharakunnel and S. Bhattacharyya, 'Single-leader-multiple-follower games with boundedly rational agents', *Journal of Economic Dynamics and Control*, **33**, 1593–1603, (2009).

CUBE: A CUDA Approach for Bucket Elimination on GPUs

Filippo Bistaffa¹ and Nicola Bombieri² and Alessandro Farinelli³

Abstract. We consider Bucket Elimination (BE), a popular algorithmic framework to solve Constraint Optimisation Problems (COPs). We focus on the parallelisation of the most computationally intensive operations of BE, i.e., join sum and maximisation, which are key ingredients in several close variants of the BE framework (including Belief Propagation on Junction Trees and Distributed COP techniques such as ActionGDL and DPOP). In particular, we propose CUBE, a highly-parallel GPU implementation of such operations, which adopts an efficient memory layout allowing all threads to independently locate their input and output addresses in memory, hence achieving a high computational throughput. We compare CUBE with the most recent GPU implementation of BE. Our results show that CUBE achieves significant speed-ups (up to two orders of magnitude) w.r.t. the counterpart approach, showing a dramatic decrease of the runtime w.r.t. the serial version (i.e., up to $652\times$ faster). More important, such speed-ups increase when the complexity of the problem grows, showing that CUBE correctly exploits the additional degree of parallelism inherent in the problem.

1 INTRODUCTION

Bucket Elimination (BE) [9] is a general algorithmic framework that adopts Dynamic Programming (DP) to incorporate many reasoning techniques. In this paper, we focus on the version of BE that solves Constraint Optimisation Problems (COPs), a general class of problems that can be used to model several optimisation scenarios [8]. BE operates on functions in tabular form by means of two fundamental operations, i.e., *join sum* and *maximisation*, which are the most computationally intensive tasks of the entire algorithm. Such operations are also the key ingredients in several close variants of the BE framework, including Belief Propagation (BP) on Junction Trees [15], and Distributed COP techniques such as ActionGDL [24] and DPOP [20].

Nevertheless, in many large COP instances BE may result in prohibitive computation requirements (both in memory and runtime), as its computational complexity is exponential in the *induced width* of the graph representation of the problem [9]. For this reason, several works in the constrained optimisation literature have tried to deal with this complexity adopting various approaches. On the one hand, Dechter [7] proposed Mini-Bucket Elimination, an approximate version of BE with limited memory requirements and reduced computation. On the other hand, a recent strand of literature [17, 18] has investigated the use of AND/OR search trees, proposing several heuristic approaches and bounding methods to reduce the search space.

In this paper we propose the use of parallel architectures to speed-up the computation associated to COP solution techniques. In particular, in recent years, many computationally intensive applications have successfully employed Graphics Processing Units (GPUs), achieving speed-ups of several orders of magnitude [10]. Parallelisation has also been investigated to speed-up search-based approaches for COP on multi-core CPUs [19], but the application of these techniques to GPUs is difficult for several reasons. On the one hand, general depth-first search is known to be difficult to parallelise [21], especially on highly parallel architectures such as GPUs. Moreover, the use of branch-and-bound may result in heavily unbalanced search trees, requiring complex techniques to balance the workload among the threads [19]. Such techniques are not effective on GPUs, where load balancing is crucial to achieve a high computational throughput.

Against this background, in this paper, we investigate the use of GPUs for BE, motivated by the above discussion, by previous works that successfully parallelised DP on GPUs [13, 6, 23, 5], and by the work of Fioretto et al. [11], who recently proposed a GPU approach for BE. Specifically, we propose CUBE (CUda Bucket Elimination), providing a highly parallel implementation for the join sum and maximisation operations associated to BE. CUBE proposes a novel methodology for the parallelisation of such operations, which is specifically designed to consider two fundamental aspects of the GPU algorithmic design: thread independence and memory management. This allows CUBE to achieve significant speed-ups with respect to previous approaches and specifically to Fioretto et al. [11].

Our work opens future research developments in the field of constraint optimisation as it provides valuable techniques that can help improving the performance, apart from BE itself, in other algorithmic frameworks that adopt join sum and maximisation as subroutines. These operations represent the key ingredients of several solution techniques for COPs that have been proposed to overcome the memory requirements of BE, such as Mini-Bucket Elimination [7] (which adopts the same join sum and maximisation operations discussed in this paper), and the AND/OR search-based approaches proposed by Marinescu and Dechter [17, 18], in which Mini-Bucket heuristics are used to guide the search.

In more detail, this paper advances the state-of-the-art in the following ways:

- We propose a computational model for the join sum and maximisation operations where each thread is completely independent from the others. More specifically, in our approach each thread retrieves its input data and performs the necessary computations avoiding any interaction with the other threads. This allows us to significantly reduce sequential computation, hence fully exploiting the computational capabilities of the GPU.

¹ University of Verona, Italy, email: filippo.bistaffa@univr.it

² University of Verona, Italy, email: nicola.bombieri@univr.it

³ University of Verona, Italy, email: alessandro.farinelli@univr.it

- We avoid unnecessary, expensive memory accesses by proposing a technique that allows threads to locate their input data only on the base of their own ID. We take advantage of the *data reuse* pattern inherent in the join sum and the maximisation operations by caching the input data in the *shared memory*, i.e., the fastest form of memory in the GPU hierarchy [10]. Bandwidth efficiency is also ensured by the high spatial locality inherent in our data representation, achieved through a technique recently proposed by Bistaffa et al. [5] for BP.
- We compare CUBE to the approach proposed by Fioretto et al. [11] on the same experimental settings, i.e., we used the same dataset and the same baseline sequential benchmark (i.e., FRODO [16]). Our results show that CUBE is up to $652\times$ faster than FRODO and that the speed-up obtained by CUBE is up to two orders of magnitude higher than the other GPU approach. In contrast to the approach proposed by Fioretto et al. [11], the speed-ups achieved by CUBE increase when the complexity of the problem grows, thus showing that CUBE correctly exploits the degree of parallelism inherent in the problem. This improvement allows us to compute solutions for COP instances that could not be tackled by previous BE approaches in a reasonable amount of time, showing that our approach is a viable method for real-world problems.

2 BACKGROUND

In this section, we first provide a brief introduction to the BE algorithm (Section 2.1), while Section 2.2 discusses previous works related to BE on GPUs. Section 2.3 describes the main features of GPUs, and Section 2.4 discusses the table memory layout employed by CUBE [5].

2.1 Bucket Elimination

Bucket Elimination (BE) [9] is a general algorithmic framework that adopts DP to incorporate many reasoning techniques. The input of BE is given as a knowledge-base theory encoded by several functions or relations over subsets of variables (e.g., clauses for propositional satisfiability, constraints, or conditional probability matrices for belief networks). In this work, we focus on *Constraint Networks* (CN), following the definitions provided by Dechter [9].

Definition 1 A Constraint Network (CN) consists of a set $X = \{x_1, \dots, x_n\}$ of n variables such that $x_1 \in D_1, \dots, x_n \in D_n$, where D_i represents the domain of the variable x_i , together with a set of m constraints $\{C_1, \dots, C_m\}$.

Definition 2 A constraint C_i is a relation defined on a set $X_i = \{x_{i_1}, \dots, x_{i_h}\}$ of h variables, called the scope of the constraint, such that $X_i \subseteq X$. Such a relation denotes the variables simultaneous legal assignments. Non-legal assignments are denoted as unfeasible. Notice that C_i is a subset of the Cartesian product $D_{i_1} \times \dots \times D_{i_h}$.

In this work we focus on the version of BE that solves COPs, and specifically on Algorithm 1 detailed by Dechter [9]. COPs are a general class of problems, which can be used to model several optimisation scenarios [8].

Definition 3 A Constraint Optimisation Problem is a CN augmented with a set of functions. Let F_1, \dots, F_l be l real-valued functional components defined over the scopes Q_1, \dots, Q_l , $Q_i \subseteq X$, let $\bar{a} = (a_1, \dots, a_n)$ be an assignment of the variables, where $a_i \in D_i$.

The global cost function F is defined by $F(\bar{a}) = \sum_{i=1}^l F_i(\bar{a})$, where $F_i(\bar{a})$ means F_i applied to the assignments in \bar{a} restricted to the scope of F_i . Solving the COP requires to find $\bar{a}^* = (a_1^*, \dots, a_n^*)$, satisfying all the constraints, such that $F(\bar{a}^*) = \max_{\bar{a}} F(\bar{a})$ (or $F(\bar{a}^*) = \min_{\bar{a}} F(\bar{a})$, in case of a minimisation problem).

Algorithm 1 BUCKETELIMINATIONCOP (CN, F_1, \dots, F_l, o)

- 1: Partition $\{C_1, \dots, C_m\}$ and $\{F_1, \dots, F_l\}$ into n buckets according to o
 - 2: **for all** $p \leftarrow n$ down to 1 **do**
 - 3: **for all** C_k, \dots, C_g over scopes X_k, \dots, X_g , and **for all** F_h, \dots, F_j over scopes Q_h, \dots, Q_j , **in bucket** p **do**
 - 4: **if** $x_p = a_p$ **then**
 - 5: $x_p \leftarrow a_p$ in each F_i and C_i
 - 6: Put each F_i and C_i in appropriate bucket
 - 7: **else**
 - 8: $U_p \leftarrow \bigcup_i X_i \quad \{x_p\}$
 - 9: $V_p \leftarrow \bigcup_i Q_i \quad \{x_p\}$
 - 10: $W_p \leftarrow U_p \cup V_p$
 - 11: $C_p \leftarrow \pi_{U_p} (\times_{i=1}^g C_i)$
 - 12: **for all** tuples t over W_p **do**
 - 13: $H_p(t) \leftarrow \Downarrow_{a_p: (t, a_p) \text{ satisfies } \{C_1, \dots, C_g\}} \bigoplus_{i=1}^j F_i(t, a_p)$
 - 14: Place H_p in the latest lower bucket mentioning a variable in W_p , and C_p in the latest lower bucket with a variable in U_p
 - 15: Assign maximising values for the functions in each bucket
 - 16: **Return** $F(\bar{a}^*)$, i.e., the optimal cost computed in the first bucket and \bar{a}^* , i.e., the optimal assignment
-

BE operates on the basis of a *variable ordering* o , which is used to partition the set of functions into n sets B_1, \dots, B_n called *buckets*, each associated to one variable of the COP. In particular, each function F_i is placed in the bucket associated to the last bucket that is associated with a variable in Q_i , i.e., the scope of F_i . Figure 2 shows the buckets corresponding to the example COP in Figure 1, adopting the ordering $o = \langle x_1, x_3, x_2, x_5, x_4, x_6 \rangle$.

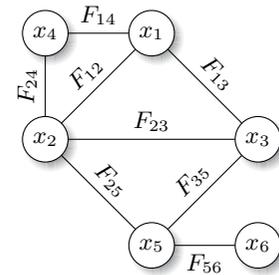


Figure 1: Example COP.

Then, buckets are processed from last to first (top to bottom), by means of two fundamental operations, i.e., *join sum* (denoted as \oplus) and *maximisation* (denoted as \Downarrow). Specifically, all the cost functions in B_p , i.e., the current bucket, are *composed* with the \oplus operation, and the result is the input of a \Downarrow operation. Such operation removes x_p (i.e., the variable associated to B_p) from the table, and produces a new function H_p that does not involve x_p , which is then placed in the last bucket that is associated to a variable appearing in the scope of the new function.

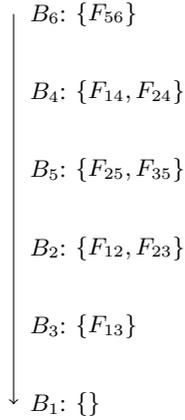


Figure 2: Initial buckets.

Figure 3 shows the execution of BE on the previous example. In particular, if a bucket, say B_4 , contains more than one F_i , such functions are first composed with \oplus and then the corresponding variable (i.e., x_4) is maximised out. In Figure 3, we represent these two subsequent operations by means of the compact notation $\Downarrow\oplus$. In the case of B_4 , the result of $\Downarrow\oplus$ is a function $h_4(x_1, x_2)$ without x_4 , which is placed in B_2 . By operating in such a way, we can guarantee that the resulting function in the first bucket (i.e., $H_3(x_1)$ in Figure 3) contains only the first variable in o , i.e., x_1 , since all the remaining ones have been maximised out during the previous steps. Hence, we compute the optimal assignment for x_1 as the one that maximises $H_3(x_1)$, and propagate such assignment back to the second bucket. Then, we proceed in the same way as before, computing the optimal assignment for the corresponding variable, and propagating the result until all buckets have been processed. Such process terminates when the optimal assignment for all variables has been computed.

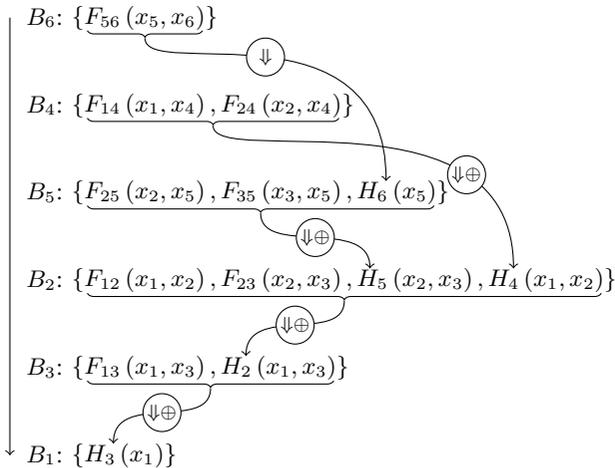


Figure 3: BE execution.

Dechter [9] proves that the computational complexity of the BE algorithm is directly determined by the ordering o .

Proposition 1 *The complexity of BE is time and space exponential in $w^*(o)$, the induced width of the problem given the variable ordering o , i.e., $O(m \cdot k^{w^*(o)})$, where k bounds the domain size and m is the number of constraints.*

As a consequence, it is of utmost importance to adopt a variable ordering o that minimises the induced width $w^*(o)$. Unfortunately, the task of computing such ordering is NP-complete [9], and, for this reason, a greedy procedure (Algorithm 2) [9] is usually adopted to compute a variable ordering of acceptable quality.

Algorithm 2 GREEDYORDERING (CN, $metric(\cdot)$)

- 1: **for all** $k \leftarrow n$ down to 1 **do**
 - 2: $x^* = \arg \min_{x_i \in X} metric(x_i)$
 - 3: $o[k] \leftarrow x^*$
 - 4: Introduce edges in CN between all neighbours of x^*
 - 5: Remove x^* from CN
 - 6: **return** o
-

Notice that Algorithm 2 can be parametrised with different $metric(\cdot)$ functions, that evaluate each node on the basis of different properties. The most commonly used are the *min-degree* heuristic (in which $metric(x_i)$ is the number of neighbours of x_i) and the *min-fill* heuristic (in which $metric(x_i)$ is the number of edges that need to be added to the graph due to the elimination of x_i).

2.2 Related Work

To the best of our knowledge, the only work that specifically focuses on the implementation of the BE algorithm for many-cores architectures is the one by Fioretto et al. [11], in which the authors devise an algorithm to realise the join sum and the maximisation operations (referred as *aggregate* and *project*) on GPUs, by exploiting the high degree of parallelism inherent in these operations.

Although this approach represents a significant contribution to the state-of-the-art, there are some drawbacks that hinder its applicability. First, the indexing of the tables is executed by using a *Minimal Perfect Hash* function [3], i.e., a hash function that maps n keys to n consecutive integers, which can be easily adopted as the indices of such keys. Although minimal perfect hash functions can be used in parallel by different threads to index the input, their construction is inherently sequential, since the index of a key depends on the indices assigned to the previously considered keys [2]. This aspect reduces the efficiency of this approach especially on big instances, as shown by our experiments in Section 4. In contrast, our focus on thread independence and memory management allows us to obtain better speed-ups that increase when growing the size of the instances.

This is possible thanks to the preprocessing technique proposed by Bistaffa et al. [5], which exploits the improved table layout achieved with such preprocessing to implement an efficient GPU version of the base operations of Belief Propagation on Junction Trees (BP on JTs) [15], i.e., *reduction* and *scattering*.

Nonetheless, their approach cannot be directly applied to BE. On the one hand, the join sum operation is fundamentally different from both reduction and scattering, as the reduction corresponds to the maximisation in BE, and scattering computes a completely different output than the join sum. On the other hand, Bistaffa et al. [5] do not adopt the current state-of-the-art technique to implement the reduction operation (i.e., *segmented reduction*) and, hence, their approach can suffer from a reduced computational throughput.

2.3 GPUs

GPUs are designed for compute-intensive, highly parallel computations. These architectures perform particularly well on problems that can be modelled as data-parallel computations where data elements correspond to parallel processing threads, as they are designed on the basis of the Single Instruction Multiple Data (SIMD) model [10]. We program the GPU using the NVIDIA CUDA framework, which requires the definition of particular functions, called *kernels*, executed in parallel by thousands of threads on different inputs. Threads are grouped into thread *blocks*. Threads of the same block share fast forms of storage and synchronisation primitives. Memory plays a crucial role in the design of efficient GPU algorithms. In fact, modern GPUs contain very fast but small-size memories (i.e., registers, cache and *shared memory*), intended to assist high performance computations, stacked above a slower but larger memory (i.e., *global memory*), suitable to hold large amounts of data. Accessing global memory is particularly expensive, and should be reduced as much as possible. To do that, a common practice suggests to exploit data locality, i.e., transferring small portions of frequently used data from global to shared memory and to complete all the computational tasks that use such data before accessing to new one. This allows minimising global memory accesses. Such transfers should be executed in order to have consecutive threads fetching data from consecutive memory addresses, which is denoted as memory *coalescing* (Figure 4). Coalesced accesses are related to the principle of locality of information and they allow the hardware to combine multiple transfers between global and shared memory into a single transaction. In contrast, sparse data results in poor memory performance (Figure 5).

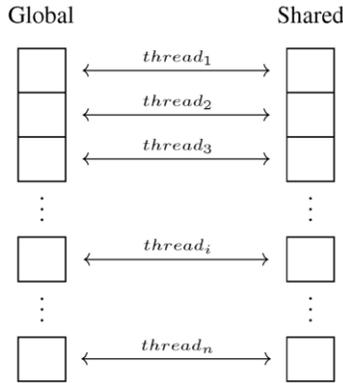


Figure 4: Coalesced accesses.

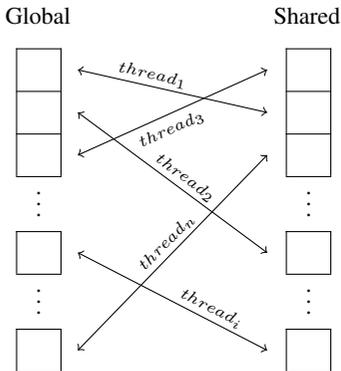


Figure 5: Uncoalesced accesses.

2.4 Preprocessing Tables

BE, as well as BP on JTs, operates on functions in tabular form considering groups of rows having the same assignments of the *shared* variables between two tables. As an example, both tables in Figure 6 contain x_1 , thus BE (and, in particular, the join sum and maximisation operations) will operate on groups having the same value for x_1 (coloured in white and grey). Bistaffa et al. [5] notice that, in general, the arrangement of rows may suffer from poor data locality (i.e., white and grey groups are interleaved in Figure 6), reducing the efficiency of computations associated to BP. Thus, they propose a preprocessing approach for tables to achieve the row arrangement shown in Figure 7. Such arrangement allows optimised memory accesses and it enables tables to be split into smaller chunks (which are now in consecutive memory addresses), so to handle tables that may not fit into the GPU global memory. This table layout enables the GPU to execute *coalesced* loads, grouping several memory accesses and improving the computational throughput (Figure 5).

T_1				T_2			
x_3	x_2	x_1	ϕ^1	x_5	x_4	x_1	ϕ^2
0	0	0	α_1	0	0	0	β_1
0	0	1	α_2	0	0	1	β_2
0	1	0	α_3	0	1	0	β_3
0	1	1	α_4	0	1	1	β_4
0	2	0	α_5	0	2	0	β_5
0	2	1	α_6	0	2	1	β_6
1	0	0	α_7	1	0	0	β_7
1	0	1	α_8	1	0	1	β_8
1	1	0	α_9	1	1	0	β_9
1	1	1	α_{10}	1	1	1	β_{10}
1	2	0	α_{11}	1	2	0	β_{11}
1	2	1	α_{12}	1	2	1	β_{12}

Figure 6: Original tables.

T_1^p				T_2^p			
x_1	x_3	x_2	$p(\phi^1)$	x_1	x_5	x_4	$p(\phi^2)$
0	0	0	α_1	0	0	0	β_1
0	0	1	α_3	0	0	1	β_3
0	0	2	α_5	0	0	2	β_5
0	1	0	α_7	0	1	0	β_7
0	1	1	α_9	0	1	1	β_9
0	1	2	α_{11}	0	1	2	β_{11}
1	0	0	α_2	1	0	0	β_2
1	0	1	α_4	1	0	1	β_4
1	0	2	α_6	1	0	2	β_6
1	1	0	α_8	1	1	0	β_8
1	1	1	α_{10}	1	1	1	β_{10}
1	1	2	α_{12}	1	1	2	β_{12}

Figure 7: Preprocessed tables.

3 BE ON GPUS

This section presents CUBE, a GPU implementation of the joint sum and maximisation operations of BE.

3.1 Join Sum on GPUs

We first discuss the implementation of the join sum operation on GPUs. Such operation, denoted as \oplus , is very similar to the *join* of relational algebra, in which the output table contains one row for each couple of rows of the input tables that have a matching assignment of the shared variables. In the case of the join sum, the value of each row is given by the sum of the values of the corresponding input rows. To better explain how the join sum works, we consider the tables T_1^p and T_2^p in Figure 7. In what follows, we denote as *group* a set of rows that all have the same assignment over the shared variables, or, more intuitively, the same colour.

In order to achieve a full parallelisation of the join sum, we adopt the *gather* paradigm [14], in which each thread is responsible of the computation of exactly one element of the output. Such a paradigm offers many advantages w.r.t. the counterpart approach, i.e., the *scatter*⁴ paradigm, in which each thread is associated to one element of the input and contributes to the computation of many output elements. In fact, scatter-based algorithms have a reduced degree of parallelism since they often require atomic primitives (which inherently serialise parts of the computation) to avoid having multiple threads concurrently operating on the same output. As previously discussed, only the array ϕ is stored in memory, since we assume that tables are *complete*⁵ and, hence, it is not necessary to store the variable assignment part. Therefore, we only discuss on how we compute the array ϕ of the output table $T_i \oplus T_j$, denoted as ϕ_{\oplus} . We map one GPU thread t to each element of ϕ_{\oplus} , denoted as $\phi_{\oplus}[t]$.⁶

Our main goal is that each thread should be capable of computing the indices of its input rows in T_1^p and T_2^p in a closed form only on the base of its own ID t , with the aim of avoiding unnecessary memory accesses to the input data. To achieve this, we now introduce some background concepts needed to explain our indexing approach. First, notice that the number of rows in each group is equal to the number of all the possible assignments of the non-shared variables in the scope of the table, i.e., the product of the domain sizes of such variables. In particular, each group in T_1^p consists of 6 rows, as $|D_2| \cdot |D_3| = 6$, and the same applies to T_2^p , i.e., $|D_4| \cdot |D_5| = 6$. Since the join sum operation associates each of these 6 rows in T_1^p to each of the 6 matching rows in T_2^p , the corresponding group in the output table will contain $|D_2| \cdot |D_3| \cdot |D_4| \cdot |D_5| = 36$ rows. In general, it is easy to verify that, if $X_i = \{x_{i_1}, \dots, x_{i_n}\}$ and $X_j = \{x_{j_1}, \dots, x_{j_k}\}$ are the scopes of the input tables T_i and T_j , the output table $T_i \oplus T_j$ (where the \oplus operator represents the join sum) contains a number of rows equal to

$$\left(\prod_{x_a \in X_i \cap X_j} |D_a| \right) \cdot \underbrace{\left(\prod_{x_b \in X_i} |D_b| \right) \cdot \left(\prod_{x_c \in X_j} |D_c| \right)}_{\text{rows}(X_i, X_j)}. \quad (1)$$

For convenience, we define the function *rows* to denote the number of rows in each group of the output table induced by the scopes X_i and X_j . Formally, $\text{rows} : 2^X \times 2^X \rightarrow \mathbb{N}$, where 2^X denotes the powerset of X .

⁴ Even if this paradigm shares the same name with the *scattering* phase of BP, it refers to a completely different concept.

⁵ A table T_i with the scope X_i is *complete* if it contains all the possible assignments over the domains of the variables in X_i . We represent *unfeasible* rows as ∞ values.

⁶ We adopt the *zero-based* convention, i.e., arrays start at index 0.

Algorithm 3 JOINSUMGPU(t, X_i, X_j)

- 1: $g \leftarrow \lfloor \frac{t}{\text{rows}(X_i, X_j)} \rfloor$ {Output group t belongs to}
 - 2: $idx \leftarrow t \bmod \text{rows}(X_i, X_j)$ {ID of t within g }
 - 3: $\#_i \leftarrow \prod_{x_b \in X_i} |D_b|$ {# of rows associated to g in T_i }
 - 4: $\#_j \leftarrow \prod_{x_c \in X_j} |D_c|$ {# of rows associated to g in T_j }
 - 5: $\gamma \leftarrow g \cdot \#_i + \lfloor \frac{idx}{\#_j} \rfloor$ {Input row in T_i }
 - 6: $\delta \leftarrow g \cdot \#_j + idx \bmod \#_j$ {Input row in T_j }
 - 7: $\phi_{\oplus}[t] \leftarrow \phi_i[\gamma] + \phi_j[\delta]$ {Compute and store output}
-

Algorithm 3 summarises the approach we propose to compute the join sum of two tables T_i and T_j , which is executed in parallel by each thread to index the input tables and to compute each row of the output table. As a first step, each thread t identifies which group it belongs to (Line 1), by dividing its index t for the number of rows in each output group, i.e., $\text{rows}(X_i, X_j)$. Specifically, t operates within the g^{th} group. Furthermore, t computes its index idx relative to the first row of its group in Line 2.

Then, to compute the indices γ and δ of its two input rows, t first calculates $\#_i$ and $\#_j$, representing the number of rows of each group in T_i and T_j respectively, by multiplying the sizes of the domains of the non-shared variables in each table (Lines 3 and 4).

A further inspection of Lines 5 and 6 reveals how Algorithm 3 organises the *rows* (X_i, X_j) elements of the g^{th} output group among the corresponding GPU threads. It associates the first $\#_j$ rows of such group to the first row of the g^{th} group in T_i , and each of these threads is then associated to each of the $\#_j$ rows of the g^{th} group in T_j . This pattern is then repeated for the second row of the g^{th} group in T_i , and so on for all the $\#_i$ rows of the g^{th} group in T_i (Figure 8). The offsets $g \cdot \#_i$ and $g \cdot \#_j$ ensure the selection of the g^{th} group in T_i and T_j , as they represent the total number of rows in the g groups that precede the g^{th} one in each input table.

Example 1 For a better understanding, we show how Algorithm 3 computes the row at index 59 of $T_1^p \oplus T_2^p$. Such a row would be computed by the thread $t = 59$, associated to the index $idx = 23$ of the output group $g = 1$, i.e., the grey one. In fact, as introduced earlier in this section, $\text{rows}(X_1, X_2) = 36$. It is easy to verify that $\#_i = \#_j = 6$. Then, t computes the indices of its input rows in T_1^p and T_2^p , i.e., $\gamma = 6 + 3 = 9$ and $\delta = 6 + 5 = 11$. Hence, $t = 59$ computes the element at index 23 of the output grey group, i.e., the one associated to the line at index 3 of the grey group in T_1^p and the last line of the grey group in T_2^p , as represented by γ and δ .

Note that the only input required by each thread executing Algorithm 3 is its own ID t , since X_i and X_j are equal and known in advance by all threads. t does not determine *which* operations are executed (as they are equal for all threads), but only *where* the input data is located. For these reasons, Algorithm 3 fits perfectly the SIMD model adopted by GPU architectures. In addition, Algorithm 3 does not contain any branching instruction, which would cause a phenomenon called *divergence*, which reduces the degree of parallelism by forcing the serialisation of threads executing different branches of the program [12], hence limiting its computational throughput.

Finally, Algorithm 3 relies on a *data reuse* pattern, as each row of T_i is the input of $\#_j$ output elements and, symmetrically, each row of T_j is the input of $\#_i$ output elements. We avoid expensive accesses to the GPU global memory⁷ by first transferring each coloured group

⁷ Global memory, in which the data is initially stored, is the slowest type of memory of the GPU hierarchy [10].

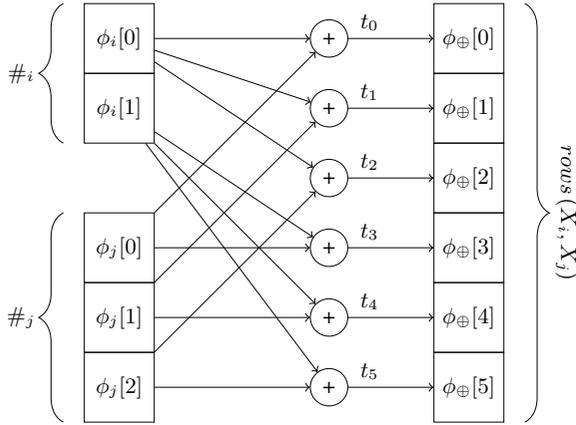


Figure 8: Join sum output computation.

to the shared memory, which allows threads to fetch data roughly $100\times$ faster [10]. Notice that the use of the shared memory is possible only because we represent the input data with the table layout discussed in Section 2.4, in which coloured groups are in small, contiguous chunks of memory. Since GPUs only have tens of KB of shared memory available, it is not possible to achieve the same benefits with the original tables (Figure 6), which should be transferred *in toto*, possibly exceeding the hardware capabilities of the GPU.

For the same reason, CUBE is capable of processing tables that are larger than the GPU global memory. In fact, our approach exploits the proposed table layout by splitting large tables into manageable chunks that can be processed independently. Specifically, this division is achieved by computing the maximum number of kernels, namely max_s , which can execute at the same time without exceeding the memory capabilities of the device. In our implementation, max_s is dynamically determined at runtime as the maximum number of kernels whose total amount of input and output data can be stored into global memory. We also take into account the space constraints deriving from the use of shared memory (see Section 2.3), by enforcing that single coloured chunks of data can fit in such memory.

Figure 9 shows an example in which the input is processed in three different pieces by the kernels K_1 , K_2 and K_3 . Notice that the independence among such computations can be exploited by enabling a *pipelined* execution model, in which each kernel K_i starts processing as soon as its input chunk has been transferred to the GPU. Nonetheless, consumer NVIDIA GPUs have only one channel that can be used for data transfers, preventing the parallelisation of the transfers from the host to the device with the ones from the device to the host. However, more advanced GPUs (e.g., NVIDIA Tesla) feature an additional transfer channel, enabling a full pipeline (Figure 10).

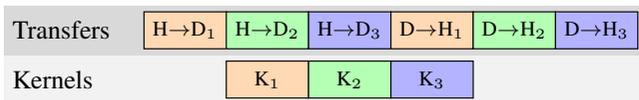


Figure 9: Pipeline with one transfer channel.

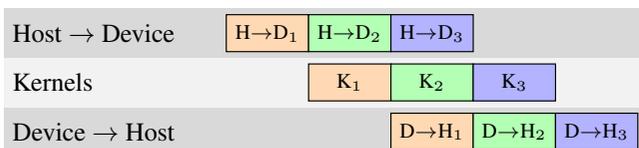


Figure 10: Pipeline with two transfer channel.

3.2 Maximisation on GPUs

Maximisation can be seen as a particular case of the relational algebra *project* operation. In the case of BE, maximisation operates by removing the variable associated to the current bucket from the input table T_i . As a consequence, the resulting table contains D_p copies of each unique assignment of the variables in its scope, i.e., $X_i \setminus \{x_p\}$. Maximisation then maps each unique assignment to the maximum of the D_p values mentioned above. For example, if we want to compute the maximisation of T_1 (Figure 6) by removing x_3 , we first obtain the table shown in Figure 11, in which each unique variable assignment is highlighted with a different colour (here $D_p = D_3 = 2$). The final output is computed as shown in Figure 12. Figures 11 and 12 highlight the high degree of parallelisation inherent in the maximisation operation, as each coloured group can be processed independently from the others. The maximisation of the D_p values within each coloured group can be realised with a *reduction* operation, which can be efficiently implemented on the GPU by means of a well-known parallel algorithm [10].

x_2	x_1	ϕ^1
0	0	α_1
0	1	α_2
1	0	α_3
1	1	α_4
2	0	α_5
2	1	α_6
0	0	α_7
0	1	α_8
1	0	α_9
1	1	α_{10}
2	0	α_{11}
2	1	α_{12}

Figure 11: T_1 without x_3 .

x_2	x_1	$m(\phi^1)$
0	0	$\max(\alpha_1, \alpha_7)$
0	1	$\max(\alpha_2, \alpha_8)$
1	0	$\max(\alpha_3, \alpha_9)$
1	1	$\max(\alpha_4, \alpha_{10})$
2	0	$\max(\alpha_5, \alpha_{11})$
2	1	$\max(\alpha_6, \alpha_{12})$

Figure 12: Maximisation output.

Nonetheless, Figure 11 also highlights the poor data locality of this table layout (similar to the one in Figure 6), which causes the same issues discussed in Section 2.4. To overcome these problems, we preprocess the input table to achieve the row arrangement shown in Figure 13. In particular, we aim at placing each coloured group in consecutive memory locations, so to achieve a better data locality and improve the efficiency of the maximisation operation. This is equivalent to moving x_p to the last column, and is implemented in CUBE with Bistaffa et al.'s technique by considering as *shared* all the variables in the scope of the table minus x_p .

This table layout enables an efficient GPU algorithm to compute the final output of the maximisation operation, i.e., $m(\phi_1)$ in the above example. In general, the array ϕ of the output table can be

x_2	x_1	x_3	$p(\phi^1)$
0	0	0	α_1
0	0	1	α_7
0	1	0	α_2
0	1	1	α_8
1	0	0	α_3
1	0	1	α_9
1	1	0	α_4
1	1	1	α_{10}
2	0	0	α_5
2	0	1	α_{11}
2	1	0	α_6
2	1	1	α_{12}

Figure 13: T_1 after the preprocessing.

computed with a *segmented reduction* algorithm [22], a well-known GPU primitive that differs from the standard reduction in that the latter operates on the entire set of input elements (e.g., it computes the maximum over the entire length of the input array), while the former operates on several fractions of the input data, i.e., the coloured groups in our case. The use of the segmented reduction allows an improved computational throughput w.r.t. the approach proposed by Bistaffa et al. [5], in which the authors implement the same operation by manually devising a series of small standard reductions that can lead to a low GPU utilisation when the coloured segments are small.

Finally, we further increase the efficiency of CUBE with two improvements. On the one hand, we avoid unnecessary data transfers between the host and the device when computing the maximisation operation. In particular, since the BE algorithm always applies the maximisation operation on the result of the join sum operation, we can avoid to transfer the join sum result (produced on the GPU memory) from the GPU to the CPU and directly run the maximisation on the GPU, hence saving two data transfers. On the other hand, if the tables are particularly small, we execute both the join sum and the maximisation on the CPU, since the overhead of the transfers to the GPU would hinder the benefits of parallelisation.

4 EMPIRICAL EVALUATION

The main goals of the empirical analysis are: i) to evaluate the parallel speed-up that CUBE achieves w.r.t. a sequential version of BE, ii) to compare CUBE against the most recent approach to parallelise BE on GPU, i.e., the work by Fioretto et al. [11], and iii) to evaluate the scalability of our approach w.r.t. the size of the problem.

Following Fioretto et al. [11], we considered 3 different CN topologies: i) *random networks* with a graph density of 0.3, ii) *scale-free networks* generated with the Barabási-Albert model [1] using $m = 2$, and iii) *2-dimensional square grid networks*, in which internal nodes are connected to four neighbours, while nodes on the edges (resp. corners) are connected to two (resp. three) neighbours. Each function F_i is generated using uniformly distributed random integer values in $[0, 100]$ and the constraint tightness (i.e., ratio of entries in such tables different from $-\infty$) is set to 0.5 for all experiments. Domain size is 5 for all experiments. We compared both GPU approaches with FRODO [16], a standard sequential COP solver also adopted by Fioretto et al. as baseline benchmark. In particular, within FRODO we employ the DPOP algorithm [20].

To ensure a fair comparison, we run all the algorithms on the same instances and adopting the same variable ordering, i.e., the one produced by FRODO. We consider the entire execution time for all the algorithms, including data transfers.⁸ All our experiments are run on a machine with a 3.10GHz processor, 16 GB of memory and a NVIDIA Tesla K40. CUBE is implemented in CUDA.⁹ For Fioretto et al.’s approach we use the authors’ implementation.

4.1 Experimental Results

Figures 14–16 show the speed-up of both GPU approaches w.r.t. FRODO when increasing the number of variables in the CN. Each data point in the plots represents the average over 20 random instances of the ratio between the runtime required by the GPU approach and FRODO’s runtime.

The results show that CUBE allows a dramatic runtime reduction w.r.t. to FRODO, by computing the solution at least one order of magnitude faster than the sequential approach in every experiment. In particular, CUBE is, on average, $530\times$ faster than FRODO when considering the biggest instances in our experiments (i.e., random networks with $n \geq 30$ and scale-free and grid networks with $n \geq 70$), by reaching a maximum speed-up of $652\times$.

More important, such speed-ups increase when the complexity of the problem grows, thus confirming the scalability of CUBE, which correctly exploits the additional degree of parallelism inherent in the problem. In contrast, the speed-up of the approach by Fioretto et al. decreases when the size of the problem increases.

Finally, the results show that the speed-up saturates after a certain number of variables (25 for random networks, 70 for scale-free networks, and 36 for grid networks). This saturation happens when the GPU reaches a full occupancy and it runs the maximum number of threads (i.e., 30720 for our GPU model). After that, the hardware forces blocks of threads to run sequentially, hence limiting the maximum speed-up.

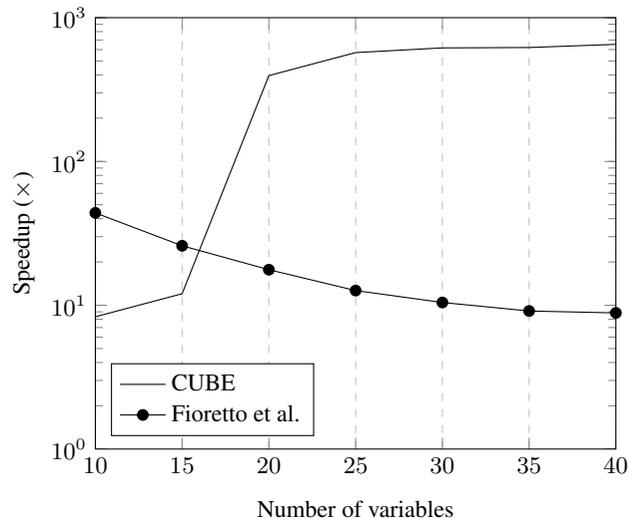


Figure 14: Speed-up on random networks.

⁸ We measured that, on average, data transfers take approximately 20% of the entire CUBE runtime.

⁹ Available at <https://github.com/filippobistaffa/CUBE>.

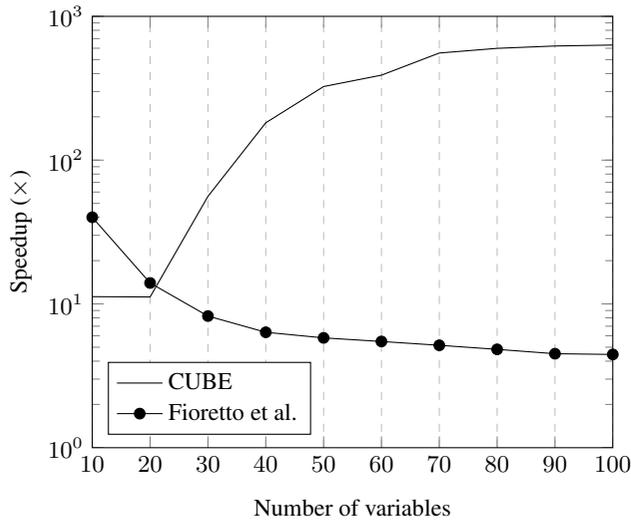


Figure 15: Speed-up on scale-free networks.

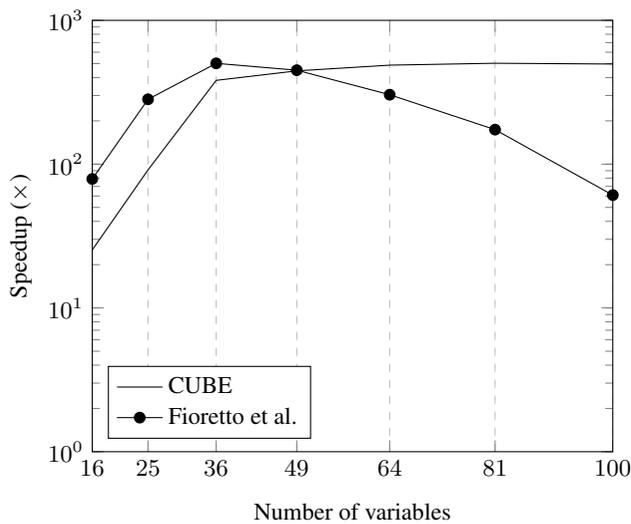


Figure 16: Speed-up on grid networks.

5 CONCLUSIONS

This paper proposes CUBE (CUda for Bucket Elimination), a high-throughput GPU implementation of the BE algorithm. Our experimental results show that CUBE outperforms the most recent GPU implementation of BE, by achieving parallel speed-ups up to two orders of magnitude higher. More important, the speed-ups achieved by CUBE increase when the complexity of the problem grows, allowing us to solve problems that could not be tackled by the sequential BE implementation in a reasonable amount of time.

Future work will aim at validating our approach on real-world COP instances, such as pedigree haplotyping problems [19] and satellite management problems [4]. We also plan to integrate our GPU techniques in other algorithmic frameworks, such as Mini-Bucket Elimination [7] (which adopts the same join sum and maximisation operations discussed in this paper), and the AND/OR search-based approaches proposed by Marinescu and Dechter [17, 18], in which Mini-Bucket heuristics are used to guide the search.

REFERENCES

- [1] Réka Albert and Albert-László Barabási, ‘Statistical mechanics of complex networks’, *Reviews of modern physics*, **74**(1), 47, (2002).
- [2] Dan Anthony Feliciano Alcantara, *Efficient hash tables on the GPU*, University of California at Davis, 2011.
- [3] Ricardo Baeza-Yates and Patricio V. Poblete, ‘Algorithms and theory of computation handbook’, chapter Searching, CRC Press, (2010).
- [4] Eric Bensana, Michel Lemaitre, and Gerard Verfaillie, ‘Earth observation satellite management’, *Constraints*, **4**(3), 293–299, (1999).
- [5] Filippo Bistaffa, Alessandro Farinelli, and Nicola Bombieri, ‘Optimising memory management for belief propagation in junction trees using GPGPUs’, in *IEEE International Conference on Parallel and Distributed Systems*, pp. 526–533, (2014).
- [6] Sebastián Dormido Canto, Ángel P. de Madrid, and Sebastián Dormido Bencomo, ‘Parallel dynamic programming on clusters of workstations’, *Parallel and Distributed Systems, IEEE Transactions on*, **16**(9), 785–798, (2005).
- [7] Rina Dechter, ‘Mini-buckets: A general scheme for generating approximations in automated reasoning’, in *International Joint Conference on Artificial Intelligence*, pp. 1297–1303, (1997).
- [8] Rina Dechter, ‘Bucket elimination: A unifying framework for reasoning’, *Artificial Intelligence*, **113**(1–2), 41–85, (1999).
- [9] Rina Dechter, *Constraint processing*, Morgan Kaufmann, 2003.
- [10] Rob Farber, *CUDA Application Design and Development*, Elsevier, 2012.
- [11] Ferdinando Fioretto, Tiej Le, Enrico Pontelli, William Yeoh, and Tran-Cao Son, ‘Exploiting GPUs in solving (distributed) constraint optimization problems with dynamic programming’, in *Principles and Practice of Constraint Programming*, 121–139, Springer, (2015).
- [12] Tianyi David Han and Tarek S Abdelrahman, ‘Reducing branch divergence in GPU programs’, in *ACM GPGPUs Workshop*, (2011).
- [13] Stephen Huang, Hongfei Liu, and Venkatraman Viswanathan, ‘Parallel dynamic programming’, *Parallel and Distributed Systems, IEEE Transactions on*, **5**(3), 326–328, (1994).
- [14] Vipin Kumar, Ananth Grama, Anshul Gupta, and George Karypis, *Introduction to parallel computing: design and analysis of algorithms*, Benjamin/Cummings Publishing Company, 1994.
- [15] Steffen L Lauritzen and David J Spiegelhalter, ‘Local computations with probabilities on graphical structures and their application to expert systems’, *Journal of the Royal Statistical Society*, 157–224, (1988).
- [16] Thomas Léauté, Brammert Ottens, and Radoslaw Szymanek, ‘FRODO 2.0: An open-source framework for distributed constraint optimization’, in *IJCAI DCR Workshop*, pp. 160–164, (2009).
- [17] Radu Marinescu and Rina Dechter, ‘Dynamic orderings for AND/OR branch-and-bound search in graphical models’, *Frontiers in Artificial Intelligence and Applications*, **141**, 138, (2006).
- [18] Radu Marinescu and Rina Dechter, ‘Best-first AND/OR search for graphical models’, in *AAAI Conference on Artificial Intelligence*, pp. 1171–1176, (2007).
- [19] Lars Otten and Rina Dechter, ‘A case study in complexity estimation: Towards parallel branch-and-bound over graphical models’, in *Conference on Uncertainty in Artificial Intelligence*, pp. 665–674, (2012).
- [20] Adrian Petcu, *A Class of Algorithms for Distributed Constraint Optimization*, Phd. thesis no. 3942, Swiss Federal Institute of Technology (EPFL), 2007.
- [21] John H Reif, ‘Depth-first search is inherently sequential’, *Information Processing Letters*, **20**(5), 229–234, (1985).
- [22] Shubhabrata Sengupta, Mark Harris, Yao Zhang, and John D Owens, ‘Scan primitives for GPU computing’, in *Graphics hardware*, pp. 97–106, (2007).
- [23] Guangming Tan, Ninghui Sun, and Guang R Gao, ‘Improving performance of dynamic programming via parallelism and locality on multicore architectures’, *Parallel and Distributed Systems, IEEE Transactions on*, **20**(2), 261–274, (2009).
- [24] Meritxell Vinyals, Juan A Rodriguez-Aguilar, and Jesús Cerquides, ‘Constructing a unifying theory of dynamic programming DCOP algorithms via the generalized distributive law’, *Autonomous Agents and Multi-Agent Systems*, 439–464, (2011).

Managing Energy Markets in Future Smart Grids Using Bilateral Contracts

Cailli re Romain, Aknine Samir¹ and Nongailard Antoine² and Sarvapal D. Ramchurn³

Abstract. Future smart grids will empower home owners to buy energy from real-time markets, coalesce into energy cooperatives, and sell energy they generate from their local renewable energy sources. Such interactions by large numbers of small prosumers (that both consume and produce) will engender potentially unpredictable fluctuations in energy prices which could be detrimental to all actors in the system. Hence, in this paper, we propose negotiation mechanisms to orchestrate such interactions as well as pricing mechanisms to help stabilise energy prices on multiple time scales. We then prove 1) that our solution guarantees that, while prices fluctuations can be constrained, 2) that it is individually rational for agents to join energy cooperatives and 3) that the negotiation mechanisms we employ result in pareto-optimal solutions.

1 INTRODUCTION

Future smart grids aim to allow the seamless integration of distributed renewable energy (wind or solar) to provide clean and renewable energy. Moreover, as smart meters are deployed as part of smart grid initiatives, home owners will be able to participate in energy markets to, not only buy and store energy, but also shift their consumption according to real-time prices as well as sell the surplus energy they generate from their local energy sources, acting as *prosumers* [8]. Crucially, with smarter communication technologies and home energy management systems, prosumers will be able to form collectives to have a greater say in energy markets.

The smart grid will therefore engender an influx of such new and smaller actors into the energy markets trading alongside larger existing players energy producers that manage large energy sources (nuclear or gas). Experience from existing energy wholesale markets and commodity stock markets, indicate that, in contrast to markets with a few large suppliers, prices in these open markets will tend to fluctuate unpredictably whenever imbalances exist between demand and supply (see figure 1). This may lead to speculation in the market, which exacerbates the situation. In such circumstances, the complexity of coping with fluctuating prices will make it even more difficult for consumers to save money and manage their energy. Moreover, the difficulty of predicting demand and supply could lead to dangerous imbalances that could cause blackouts. Previous work has investigated allocating demand according to supply using specific pricing signals [11, 7], which incentives consumers to shift their needs to times where supply is high. Moreover, they have proposed the use of batteries and the exchange of energy [3, 14] to buy and stock when

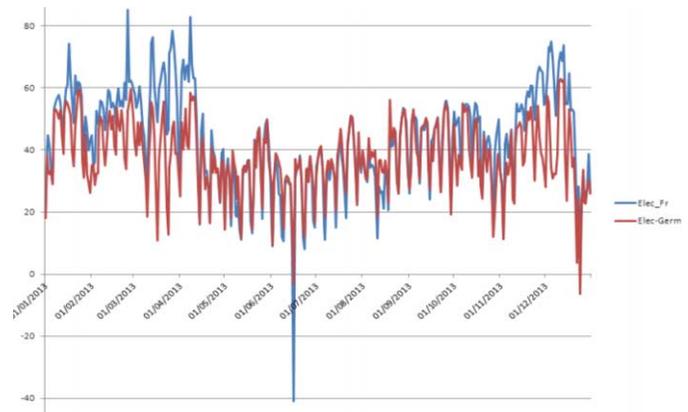


Figure 1. The evolution of the spot price. The red (resp. blue) line represents the price on the German (resp. French) market in 2013 (€/MWh)

energy is cheap and consume or store when energy is expensive. Finally, some authors propose to gather consumers into cooperatives [2] or coalitions [12] to benefit from cheaper prices in the forward market and to buy less as possible on the spot market. Unfortunately, these different works assume that the grid is always able to provide energy, if the other sources (renewable generators or batteries) cannot.

In this paper, we study the problem of forming such cooperatives without such assumptions and provide negotiation-based solutions to the settlement of contracts between prosumers. We consider cooperatives because they significantly improve the buying power of small prosumers. To help manage such cooperatives, we assume that they are set up and led by individual aggregators that buy from selected providers to supply energy at the cheapest rate to the cooperative. Similar to [12], aggregators do so using predictions of energy consumption provided by individual prosumers. Moreover, to ensure that prosumers are incentivised to predict their behaviours accurately, the aggregator penalises any deviations from such predictions. These penalties reflect charges they would incur from the provider should they over or under consume. However, in so doing, a key challenge they face is that individual consumers may want to consume more to avoid being penalised by the aggregator (and in turn the producer). Moreover, despite the formation of large cooperatives, there is no guarantee that prices will stabilise in the long run.

Against this background, we propose an approach that uses pricing signals while attempting to control the high price volatility when there is (or when agents forecast and speculate) an imbalance between supply and demand in the energy market. To address such issues, we propose a model based on bilateral contracts that constrain

¹ Univ. Lyon LIRIS, UMR 5205, F-69622, Lyon, France, email: surname.name@univ-lyon1.fr

² Univ. Lille, CNRS, Centrale Lille, UMR 9189 - CRISTAL (SMAC), F-59000 Lille, France, email: antoine.nongailard@univ-lille1.fr

³ School of Electronics and Computer Science, University of Southampton, Southampton, UK, e-mail : sdr1@ecs.soton.ac.uk

the retail price as well as the demand. In a similar way to existing energy trading protocols (e.g., in the UK or Netherlands), our framework breaks down the creation of energy supply contracts according to three different timescales: yearly, daily, and hourly. By so doing, contracts account for different levels of perceived uncertainty in demand on these timescales. Given this scenario, this paper advances the state of the art in the following ways:

1. We propose a new mechanism to manage energy markets through the use of constraints on energy prices and demand for energy . We prove that the negotiation of the annual contract (the first of the three levels), between providers and aggregators, where these constraints are set, leads to pareto-optimal solutions.
2. We propose a novel pricing scheme that we apply at the two lower levels (i.e., daily and hourly). Our pricing scheme incentivises prosumers to (i) make predictions to know the amount of energy they need during the different parts of the day ahead allowing them to optimise their costs (by shifting the use of their appliances, using their batteries, etc...) according to the price signal and (ii) avoid deviating from their predicted hourly consumption. We prove the monotonicity of the price with respect to consumption (i.e. agents are not incentivised to consume in order to avoid under-consumption penalties) and provide a closed-form formula to compute expected payments for any prosumer in a cooperative.
3. We develop an algorithm based on mathematical programming, that allows agents to minimise their cost given the price signal they receive and the limits of consumption that they are committed to holding. We prove that, in our mechanism, prosumer agents who use this algorithm, are encouraged to join a cooperative.

The rest of the paper is structured as follows. Section 2 presents the state of the art. Section 3 gives the model of each type of agents and Section 4 gives their behaviour. Section 5 shows the negotiation mechanism. Section 6 analyses the different properties of the mechanism, Section 7 describes the experiments and we conclude in section 8.

2 RELATED WORK

To tackle the problem of peak demand, several works proposed incentive tariff schemes which aim to balance supply and demand. In *Demand-Response* models [4], suppliers change their tariffs when the demand is high, encouraging consumers to reduce their consumption. Several mechanisms based on *Demand-Response* have been proposed: *Time-of-use* (TOU) in which price of energy is high at peak times and low at off-peak times (typically after 11pm). *Critical peak pricing* [6] imposes a much higher price at peak times compared to tariffs with TOU. *Real Time pricing* (RTP) [10] varies the price across the day in line with supply and demand at individual time points. Recently, a new tariff scheme has been proposed by [13], called *Prediction-of-use* (POU). In this model, consumers give a forecast of their baseline consumption against which they are given a price for predicted consumption. If they deviate from this forecast based consumption, they will be penalised. This tariff scheme incentivises consumers to forecast their consumption accurately. In our model, we combine TOU with POU schemes to both account for varying levels of demand and supply during the day and to obtain accurate predictions of consumption from prosumers.

Other works study the formation of prosumer (or consumer) agent coalitions or cooperatives with the aim to reduce energy costs and increase storage efficiency. [2] presents a scheme for electricity consumption shifting. Agents participating in the scheme are motivated

to form cooperatives, in order to reduce their electricity bills via lower group prices granted for sizable consumption shifting from high to low demand time intervals. Even though this work uses agents cooperatives to shift peak load, it doesn't focus on price volatility. [11] proposes a new multiagent coordination algorithm to shape the energy consumption of the cooperative. To coordinate individual consumers, they introduce a virtual signal sent by a central coordinator, to induce consumers to shift demand. They show that their algorithm is scalable with respect to the number of agent, but they use a two-level threshold rate, which is a high constraint to deal with the price volatility. In our work, the price has an upper and lower bound which are negotiated according to agents' preferences. Among these, we note the work of [12] that proposes a coalition formation mechanism in which agents set up a coalition which simulates a *Virtual Energy Consumer* (VEC). The VEC buys an important amount of energy on the forward market where prices are fixed and quite cheap. If the VEC needs more, it buys more on the spot market, where prices are higher. Being in a cooperative, agents buy more energy on the forward market (compared to the spot market) and compensate for their deviation within the cooperative. However, this work focuses on stable coalition formation rather than controlling price volatility. In [9], authors study the POU scheme and analyse the case of efficient buyer groups. They study the structure groups should take to buy in an efficient way in POU context. In contrast, in our model, one level of our architecture uses the TOU approach to negotiate a baseline price. This price is taken as reference to use POU scheme.

3 MODEL OF AGENTS

In this work, we consider three sets of agents. \mathcal{A}_U denotes the set of prosumer agents, \mathcal{A}_A denotes the set of aggregator agents and \mathcal{A}_F denotes the set of provider agents. We provide the model of each type of agents. In the following sections, we describe the model of prosumer agents which may own a storage capacity, a generator and smart meters allowing them to manage their demand to fit the production signal. As well as prosumers, producers can estimate their production (with smart meters technologies) and produce a signal that foster the best behaviour of their customers, i.e. stalling the level of demand on the production. In our model, the ability of the prosumer/consumer to fit the production signal leads to the control of the price volatility. Next, we assume that a day is divided into a set \mathcal{T} of hourly slots such that each agent needs to decide its behaviour for each slot.

3.1 Model of prosumer agents

Let ps_u be an agent, representing a prosumer u , belonging to a cooperative. At each slot, ps_u requests for an amount of energy q_{ps} , from the cooperative, to support its needs. ps_u is committed itself to respect limits of requested energy with Q_{ps}^{min} (resp. Q_{ps}^{max}), the minimum (resp. maximum) energy requested by ps_u from the cooperative $\forall t \in \mathcal{T}$. b_{ps}^t is the energy needed by ps_u at slot t . b_{ps} is different from q_{ps} as it only takes into account the appliance consumptions but not the possible production or use of storage energy owned by the prosumer. s_{ps} is the storage capacity of ps_u . soc_{ps}^- represents the state of charge and soc_{ps}^+ represents the remaining storage capacity, i.e. $soc_{ps}^- + soc_{ps}^+ = s_{ps}$. p_{ps} represents the energy produced by the prosumer's renewable generator, $p_{ps} \leq p_{ps}^{max}$, with p_{ps}^{max} the maximum production capacity of the generator. σ_{ps}^b is the forecast mean error of ps_u consumption. The deviation

between the forecast of ps_u 's needs and its real needs follows a normal law $\mathcal{N}(b_{ps}, \sigma_{ps}^b)$ (see [5] for explanation). σ_{ps}^p is the forecast mean error of ps_u production. The deviation between the forecast of the prosumer's production and its real production follows the normal law $\mathcal{N}(p_{ps}, \sigma_{ps}^p)$. ps_u can always sell its over-supply to its cooperative, even if its predictions are not accurate. We next prove in Property 3 that it is interesting for a prosumer to belong in a cooperative.

3.2 Model of provider agents

Each provider agent $pv_f \in \mathcal{A}_F$ has a production capacity $p_{pv} \in [0; p_{pv}^{MAX}]$ where p_{pv}^{MAX} is its maximum production capacity. We assume that each pv_f has an optimal production capacity p_{pv}^{OPT} which allows it to maximise its profit, with $0 < p_{pv}^{OPT} < p_{pv}^{MAX}$. The tariff proposed by the provider is minimal when the demand is equal to p_{pv}^{OPT} and increases as the demand deviates from it. A_{pv} is the annual subscription cost of pv_f and pen_{pv} represents its penalty costs. Moreover, each ps_f has a maximal tariff tr_{pv}^{MAX} above which nobody is willing to buy its energy. The properties 1 and 2 show that p_{pv}^{OPT} and tr_{pv}^{MAX} lead to bounded tariffs.

3.3 Model of aggregator agents

Let ag_a be an aggregator agent, with Q_{ag}^{MIN} (resp. Q_{ag}^{MAX}), the minimum (resp. maximum) amount of energy requested by the cooperative from its providers $\forall t \in \mathcal{T}$. If at slot t the energy q_{ag}^t requested by ag_a is higher than Q_{ag}^{MAX} , this involves that some prosumers request more than they can. In so doing, the prosumers who over-consume have to contract with providers to account for their over-consumption. Property 4 shows that in this way the cooperative is not penalised. σ_{ag}^b is the forecast mean error of the cooperative consumption. $Coop_{ag}$ represents the set of agents managed by the aggregator and \mathcal{P}_{ag} represents the set of providers it contracts with.

4 BEHAVIOUR OF AGENTS

In this section, we describe the behaviour of each type of agents. First, the prosumer behaviour is modelled by a linear program that minimises its day energy cost, given its needs, production, storage and shifting capacities and the price signal, for each slot. Second, we propose a tariff computation formula used by the provider to set the price according to the requested energy and its production. Third, we introduce the algorithm of the aggregator behaviour that computes the best amount to request for the annual contract negotiations with several providers and distributes the penalty of the cooperative among its prosumers.

4.1 Prosumer agents behaviour

The optimisation of ps_u cost is held on three time scales: (i) at the annual contract level, ps_u announces its limits Q_{ps}^{min} and Q_{ps}^{max} which make him pay the cheapest subscription cost and lower the maximum tariff allowing the cooperative to satisfy its needs, (ii) at the daily contract level, ps_u announces the vector $\langle q_{ps}^t, \dots, q_{ps}^{t+n} \rangle$ which minimises its cost taking into account its storage capacities, its production, its possible shifting and the price signal, (iii) at the hourly contract level, ps_u should not deviate from q_{ps}^t to avoid penalties. Every day, prosumer minimises equation (2) subject to constraints $\{c_0, \dots, c_{13}\}$:

$$\min \sum_{t=1}^{24} tr^t \cdot q^t \quad (2)$$

With tr^t , the tariff at slot t and q^t the energy requested at slot t by the agent.

$$c_0 : q^t = b_{ps}^t - p_{ps}^t + qs_{ps}^{+,t} - qs_{ps}^{-,t} + ef_{ps}^{+,t} - ef_{ps}^{-,t}$$

$$c_1 : q^t \geq b_{ps}^t - p_{ps}^t - be_{ps}^t - soc_{ps}^{-,t}$$

This guarantees that the requested energy at slot t is enough to support the prosumer's need according to the available energy in its battery, its production and shifting, with $be_{ps}^t \in [0, b_{ps}^t]$ the part of b_{ps}^t which is shiftable at slot t . $ef_{ps}^{+,t}$ represents shifting that can be made in advance, i.e. increase the demand at slot t and $ef_{ps}^{-,t} \leq be_{ps}^t$ shifting that can be done later, i.e. decrease the demand at slot t . $qs_{ps}^{+,t}$ is the amount of energy stored in the battery at slot t and $qs_{ps}^{-,t}$ the energy extracted from the battery at slot t .

$$c_2 : q^t \leq Q_{ps}^{max}$$

$$c_3 : q^t \geq Q_{ps}^{min}$$

The above two constraints guarantee that the requested energy at slot t respects the upper and lower limits.

$$c_4 : soc_{ps}^{-,t} \geq qs_{ps}^{-,t} - qs_{ps}^{+,t}$$

This constraint guarantees that the extracted energy from the battery at slot t is lower than the remaining energy in the battery.

$$c_5 : soc_{ps}^{+,t} \geq qs_{ps}^{+,t} - qs_{ps}^{-,t}$$

This constraint guarantees that the remaining storage capacity is higher than the amount of energy stored at slot t .

$$c_6 : soc_{ps}^{-,1} = soc_{init_{ps}}$$

$$c_7 : soc_{ps}^{+,1} = s_{ps} - soc_{init_{ps}}$$

The above two constraints initialize the program with the energy available (and remaining capacity) in the battery at the first slot.

$$c_8 : soc_{ps}^{-,t} + soc_{ps}^{+,t} = s_{ps}$$

$$c_9 : soc_{ps}^{+,t+1} = soc_{ps}^{+,t} - qs_{ps}^{+,t} + qs_{ps}^{-,t}$$

$$c_{10} : soc_{ps}^{-,t+1} = soc_{ps}^{-,t} + qs_{ps}^{+,t} - qs_{ps}^{-,t}$$

These three constraints guarantee the battery integrity.

$$c_{11} : soc_{ps}^{-,24} = soc_{ps}$$

This guarantees that a specific level of energy, soc_{ps} , will remain at the end of the day.

$$c_{12} : \sum_{t \in \mathcal{T}} ef_{ps}^{+,t} = \sum_{t \in \mathcal{T}} ef_{ps}^{-,t}$$

$$c_{13} : \sum_{t \in \mathcal{T}} ef_{ps}^{+,t} \leq \sum_{t \in \mathcal{T}} be_{ps}^t$$

The above two constraints guarantee the management of the shifting. This linear program allows prosumers to benefit from their battery and their shifting possibilities to adjust their consumption according to the price signal and take advantage of the lower price during the day.

Prosumer agent states : (i) Each prosumer is initialised with parameters Q_{ps}^{min} , Q_{ps}^{max} , s_{ps} , p_{ps}^{max} , σ_{ps}^b , σ_{ps}^p and sends its profile to its aggregator. (ii) At the end of each day, ps computes soc_{ps} , solves (2) and sends the resulted schedule of hourly demand to the aggregator. (iii) When aggregator returns the price signal, ps solves (2) again and sends back the new schedule.

4.2 Provider agent behaviour

The provider uses a function $\mathcal{F} : \mathcal{Q} \rightarrow \mathcal{T}r$ where \mathcal{Q} is an amount of energy and $\mathcal{T}r$ a tariff.

$\mathcal{F}_{pv}(q_{pv}^{tot}) = tr_{pv}^{OPT} + tr_{ag}^{max} (1 - e^{\frac{(-|p_{pv}^{OPT} - q_{pv}^{tot}|)}{tr_{ag}^{max}}})$ (3)
 tr_{pv}^{OPT} is the tariff proposed by the provider when $p_{pv}^{OPT} = q_{pv}^{tot}$ and $tr_{ag}^{max} \in [0, tr_{pv}^{MAX}]$ is the coefficient negotiated in the annual contract. $q_{pv}^{tot} = \sum_{ag \in \mathcal{A}_A} q_{ag}$, with q_{ag} the amount of energy requested by the aggregator ag_a who has an annual contract with pv_f . Designed in this way, the tariff function allows having a monotonous and continuous increasing behaviour between tr_{pv}^{OPT} and $tr_{pv}^{OPT} + tr_{ag}^{max}$ and incentivised prosumers to adjust their consumption to the production.

Property 1: Under the hypothesis that the provider computes the tariffs according to (3), the maximal tariff he will apply to a cooperative is less or equal to $tr_{pv}^{OPT} + tr_{ag}^{max}$.

Proof. To prove that, we study the evolution of the price when the difference between the supply and demand goes toward infinity. We will show that $\lim_{|p_{pv}^{OPT} - q_{pv}^{tot}| \rightarrow +\infty} \mathcal{F}_{pv}(q_{pv}^{tot}) = tr_{pv}^{OPT} + tr_{ag}^{max}$. First,

$$\begin{aligned} & \lim_{|p_{pv}^{OPT} - q_{pv}^{tot}| \rightarrow +\infty} tr_{ag} = \\ & \lim_{|p_{pv}^{OPT} - q_{pv}^{tot}| \rightarrow +\infty} tr_{pv}^{OPT} + tr_{ag}^{max} \left(1 - e^{\left(\frac{-|p_{pv}^{OPT} - q_{pv}^{tot}|}{tr_{ag}^{max}}\right)}\right) \\ & e^{-x} \rightarrow 0 \text{ when } x \rightarrow +\infty \text{ so, by substitution, we get:} \\ & \lim_{|p_{pv}^{OPT} - q_{pv}^{tot}| \rightarrow +\infty} tr_{pv}^{OPT} + tr_{ag}^{max} \left(1 - e^{\left(\frac{-|p_{pv}^{OPT} - q_{pv}^{tot}|}{tr_{ag}^{max}}\right)}\right) = \\ & \lim_{|p_{pv}^{OPT} - q_{pv}^{tot}| \rightarrow +\infty} tr_{pv}^{OPT} + tr_{ag}^{max} (1 - 0) \\ \text{As a result } & \lim_{|p_{pv}^{OPT} - q_{pv}^{tot}| \rightarrow +\infty} \mathcal{F}_{pv}(q_{pv}^{tot}) = tr_{pv}^{OPT} + tr_{ag}^{max}. \quad \square \end{aligned}$$

The first property guarantees that supply cannot rationally rise without an increase of the demand, because the lower the prices, the less profitable is the mechanism.

Property 2: Under the hypothesis that the provider computes the tariffs according to (3), the minimal tariff it will apply to a cooperative is tr_{pv}^{OPT} .

Proof. The tariff is minimal if the total amount of requested energy is equal to the optimal production of the provider, i.e. $p_{pv}^{OPT} = q_{pv}^{tot}$. To prove that, we study the level of the price when the supply is equal to the demand.

$$\text{if } p_{pv}^{OPT} = q_{pv}^{tot}, \text{ we have } e^{\left(\frac{-|p_{pv}^{OPT} - q_{pv}^{tot}|}{tr_{ag}^{max}}\right)} = e^{\left(\frac{0}{tr_{ag}^{max}}\right)} = 1 \text{ and } tr_{pv}^{OPT} + tr_{ag}^{max} \cdot (1 - 1) = tr_{pv}^{OPT} \quad \square$$

The second property guarantees that prices are lower when all produced energy is requested.

4.3 Aggregator agent behaviour

An aggregator agent manages supply and demand within a cooperative. This agent contracts with providers to meet the demand of the cooperative (the sum of the demands of the agents in the cooperative) allowing to benefit from competition between providers to decrease the energy price⁴. We differentiate three kinds of contracts: (i) the annual contract sets the maximum tariff applicable by a provider and the maximal and minimal demands requested by the cooperative at each slot, (ii) the daily contract fixes a set of hourly contracts a day ahead, (iii) the hourly contract matches an amount of energy and a tariff at a given slot. POU tariff scheme is applied on hourly contracts taking as baseline negotiated earlier in the daily contract. Indeed, some prosumers may under-consume while others may over-consume. Individually, each agent will pay penalties if it under-consumes or over-consumes. However, inside the cooperative, under consumption of some prosumer will balance over-consumption of others and vice versa. As shown in section 4.4, it allows cancelling or reducing agent penalties and limiting the need for agents to request the market to sell the energy they under-consume or to buy the energy they over-consume, thus, limiting the volatility in the energy market.

⁴ We suppose the financial profit brought by these multiple subscriptions should be higher than the involved costs.

The aggregator will spread the need of the cooperative between several contracts negotiated with potential suppliers, using the following linear program. This linear program computes the upper and lower bound to negotiate with each potential provider with the purpose of minimising the cost over the year, considering the subscription cost and the penalty cost of each provider.

$$\min \sum_{pv \in \mathcal{P}_{ag}} A_{pv} \cdot x_{ag}^{pv} + pen_{pv} \cdot \sigma_{ag} \cdot \mathcal{H} \cdot y_{ag}^{pv} \quad (5)$$

With $x_{ag}^{pv} = Q_{ag,pv}^{max}$ the maximal amount of energy requested by the aggregator agent to provider pv , $y_{ag}^{pv} = Q_{ag,pv}^{max} - Q_{ag,pv}^{min}$ the width of the energy band of the aggregated demand of the cooperative to the provider pv , \mathcal{H} is the number of hours in a year and σ_{ag} the mean deviation consumption of the cooperative.

s.t. $\sum_{pv \in \mathcal{P}_{ag}} x_{ag}^{pv} = Q_{ag}^{MAX}$, with Q_{ag}^{MAX} the upper bound of the aggregated demand of the cooperative.

The sum of the maximal amount of energy in ag_a 's contract must allow providing Q_{ag}^{MAX} to the cooperative.

$$\text{s.t. } \sum_{pv \in \mathcal{P}_{ag}} x_{ag}^{pv} - y_{ag}^{pv} = Q_{ag}^{MIN}$$

This sum guarantees that the deviation sum $\sum_{pv \in \mathcal{P}_{ag}} Q_{ag,pv}^{max} - Q_{ag,pv}^{min} = Q_{ag}^{MAX} - Q_{ag}^{MIN}$ compensates the total deviation of the cooperative.

$$\text{s.t. } \forall pv \in \mathcal{P}_{ag}, x_{ag}^{pv} \geq y_{ag}^{pv}$$

This inequality is equivalent to $Q_{ag,pv}^{max} \geq Q_{ag,pv}^{max} - Q_{ag,pv}^{min}$. It allows not having $Q_{ag,pv}^{max} < Q_{ag,pv}^{min}$. With $Q_{ag,pv}^{max}$ (resp. $Q_{ag,pv}^{min}$) the maximum (resp. minimum) energy requested by ag_a from pv .

4.3.1 Prosumers production tariffication

We consider the case where a prosumer agent ps_u produces more than its needs i.e. $p_{ps}^t > b_{ps}^t$. It first meets its needs, then it will sell the oversupply to other members of the cooperative, knowing that the sale price is lower than the providers' prices. The agent which belongs to the cooperative makes sure that it sells its energy. In return, the cooperative can benefit from cheaper energy than the providers one. The remuneration of ps_u , computed by ag_a , can be formulated as:

$$tr_{ps}^t = \mathcal{F}_{ag}(q_{ag}^t + p_{ps}^{pred,t}) \cdot (1 - e^{-|p_{ps}^{pred,t} - p_{ps}^{real,t}|}) \quad (4)$$

where tr_{ps}^t is the sale tariff of ps_u at slot t . $\mathcal{F}_{ag}(q_{ag}^t + p_{ps}^t)$ is the mean tariff applied by the set of providers \mathcal{P}_{ag} to ag_a for demand $q_{ag}^t + p_{ps}^{pred,t}$, with q_{ag}^t the cooperative demand and $p_{ps}^{pred,t}$ the forecast production of ps_u . This tariff is only effective if the agent produces exactly what it forecasts. If not, the tariff is multiplied by $(1 - e^{-|p_{ps}^{pred,t} - p_{ps}^{real,t}|})$. The more the deviation between the forecast and the real production is high, the less ps_u will be rewarded, because the deviation involves possible penalties for the cooperative. The formulation of (4) incentives prosumers to forecast accurately their production.

Property 3: Under the hypothesis that ag_a negotiates with several providers, and considering an agent ps_u , who forecasts that it will produce an oversupply p_{ps}^{pred} at slot t , agents in the cooperative find it preferable to consume the locally produced energy, i.e. consume the oversupply of ps_u , each time it's possible.

Proof. As $(1 - e^{-|p_{ps}^{pred,t} - p_{ps}^{real,t}|}) \in [0, 1]$, we have the inequality $\mathcal{F}_{ag}(q_{ag}^t + p_{ps}^t) \cdot (1 - e^{-|p_{ps}^{pred,t} - p_{ps}^{real,t}|}) \leq \mathcal{F}_{ag}(q_{ag}^t + p_{ps}^t) \quad \square$

The third property guarantees that agents will exchange energy between them before resorting to the grid, involving less exchanges on the market.

4.3.2 Prosumers and providers interactions

When ps_u announces its consumption limits, it commits to the cooperative to not deviate from these limits at each slot. If ps_u doesn't respect this limit commitment and the amount request by the aggregator become higher than the upper bound negotiated in the annual contract, it has to directly pass a contract with a provider for each slot where the limits are not respected.

Property 4: The agents belonging to a cooperative are not penalised by an agent who consumes more than its higher limit at one slot, i.e. $q_{ps}^{real} > Q_{ps}^{max}$ involves $q_{ag} > Q_{ag}^{MAX}$, since it will contract for its over demand (demand $> Q_{ps}^{max}$ without available balancing into the cooperative).

Proof. Let $q_{ag/\{ps\}} < Q_{ag/\{ps\}}^{MAX}$ and $q_{ps}^{real} > Q_{ps}^{max}$ the deviation of the agent such that $q_{ag/\{ps\}} + q_{ps}^{real} > Q_{ag/\{ps\}}^{max} + Q_{ps}^{max}$. ps_u will pass a contract directly with a provider for the quantity $q_{ps}^{real} - Q_{ps}^{max}$. So the quantity bought by the cooperative becomes $q_{ag/\{ps\}} + q_{ps}^{real} - (q_{ps}^{real} - Q_{ps}^{max}) = q_{ag/\{ps\}} + Q_{ps}^{max}$ which is less than $Q_{ag/\{ps\}}^{MAX} + Q_{ps}^{max}$ as $q_{ag/\{ps\}} < Q_{ag/\{ps\}}^{MAX}$. \square

Property 4 guarantees an incentive for prosumers to respect their commitment towards the cooperative.

4.4 Penalty distribution

Before introducing the computation formula of penalty distribution, we denote: $\Delta q_{ag} = \sum_{ps \in Coop} q_{ps}^{pred} - q_{ps}^{real}$: the deviation consumption of the cooperative. If $\Delta q_{ag} > 0$ (resp. $\Delta q_{ag} < 0$), the cooperative under-consumes (resp. over-consumes). Let $\Delta q_{ps} = |q_{ps}^{pred,t} - q_{ps}^{real,t}|$: the deviation consumption of ps_u , $Pen_a = \sum_{pv \in P_{ag}} Pen_{pv}$: the penalty of the cooperative, pen_{ps} : the penalty of ps_u , $C^+ = \sum_{ps \in Coop} f(q_{ps}^{pred}, q_{ps}^{real}, dif_{ag})$ with:

$$f(x, y, z) = \begin{cases} 1 & \text{if } z > 0 \text{ and } x > y \\ & \text{or } z < 0 \text{ and } x < y \\ 0 & \text{else} \end{cases}$$

The function f returns 1 if ps_u contributes to penalise the cooperative, i.e. it over-consumes (resp. under-consumes) when the cooperative over-consumes (resp. under-consumes). The sum enumerates the agents who penalise the cooperative $Q^- = \sum_{ps \in Coop} g(q_{ps}^{pred,t}, q_{ps}^{real,t}, dif_{ag}) \cdot \Delta q_{ps}$ with:

$$g(x, y, z) + f(x, y, z) = 1$$

The function g returns 1 when ps_u contributes to decrease the deviation of the cooperative, i.e. it over-consumes (resp. under-consumes) when the cooperative under-consumes (resp. over-consumes). The product $g(\cdot) \cdot \Delta q_{ps}$ computes the amount of energy which was under-consumed (resp. over-consumed) when the cooperative over-consumed (resp. under-consumed). The penalties distribution formula applied by the aggregator agent to the prosumers of the cooperative is then:

$$pen_{ps} = \begin{cases} Pen_{ag} \cdot \frac{\Delta q_{ps} - \frac{Q^-}{C^+}}{\sum \Delta q_{ag}} & \text{if } f(\cdot)=1 \\ 0 & \text{else} \end{cases}$$

The fraction $\frac{Q^-}{C^+} (= \frac{\sum_{ps \in Coop} g(q_{ps}^{pred,t}, q_{ps}^{real,t}, dif_{ag}) \cdot \Delta q_{ps}}{\sum_{ps \in Coop} f(q_{ps}^{pred,t}, q_{ps}^{real,t}, dif_{ag})})$ corresponds to the compensation of the cooperative which is spread between all the agents in a fair way. Now that the aggregator knows the amount it will request for each supplier, it will negotiate with each of them, to fix the tariff and the lower bound of the contracts. The following section provides the description of the interactions between

the agents. The algorithm 1 gives a global view of aggregators' behaviour detailed in section 5.

Algorithm 1: Algorithm of an aggregator ag_a

```

1 if all the prosumer profile are received then
2    $Q_{ag}^{MIN} \leftarrow \sum_{ps \in Coop} Q_{ps}^{min}$ ,  $Q_{ag}^{MAX} \leftarrow \sum_{ps \in Coop} Q_{ps}^{max}$ ,
    $\sigma_{ag} \leftarrow \sqrt{\sum_{ps \in Coop} (\sigma_{ps}^b)^2}$  and solve (5);
3   for  $pv \in P_{ag}$  do
4      $tr_{pv}^{max} \leftarrow 0$ ,  $Q_{pv,ag}^{min} \leftarrow Q_{pv,ag}^{min}$ ,  $Q_{pv,ag}^{max} \leftarrow Q_{pv,ag}^{max}$  (5);
5      $ag_a$  submits the proposal to  $pv_f$ ;
6 if  $ag_a$  receives a proposal (Annual-Contract:  $\mathcal{X}_{ag}, \mathcal{X}_{pv}, id$ ) then
7    $ag_a$  computes  $\mathcal{U}_{ag}$  and  $\mathcal{U}_{pv}$ ;
8   if  $\mathcal{U}_{ag} \geq \mathcal{U}_{pv}$  then
9      $ag_a$  accepts (Annual-Contract:  $\mathcal{X}_{ag}, \mathcal{X}_{pv}, id$ );
10    for  $ps \in Coop$  do
11       $\Delta_{ps} = Q_{ps}^{max} - Q_{ps}^{min}$ ,  $\Delta_{ag} = \sum_{ps \in Coop} \Delta_{ps}$ ;
12    else
13       $ag_a$  computes  $Q_{pv}^{max'}$ ,  $Q_{pv}^{min'}$ ,  $tr_{pv}^{max'}$  such that
         $Z_{ag} > Z_{pv}$ ;
14       $\mathcal{X}_{ag} \leftarrow \{Q_{pv}^{max'}, Q_{pv}^{min'}\}$ ,  $\mathcal{X}_{pv} \leftarrow \{tr_{pv}^{max'}\}$ ;
15       $ag_a$  proposes the new
        Annual-Contract( $\mathcal{X}_{ag}, \mathcal{X}_{pv}, id$ );
16 if the plannings of all the prosumers are received then
17   for  $ps \in Coop$ ,  $t \in \mathcal{T}$  do
18      $schedule[t] = schedule[t] + q_t^{pv}$ ;
19   if there is no difference in the plannings then
20      $ag_a$  accepts the daily contract;
21   else
22      $ag_a$  sends the new Planning;
23 if  $ag_a$  receives all the tariffs from its providers then
24    $ag_a$  sends the price signal to the prosumers;
25 if end of slot then
26    $Pen_{ag} = \sum_{pv \in P} pen_{pv} * |q_{ag}^{real} - q_{ag}^{pred}|$ ;
27   for  $ps \in Coop$  do
28     if  $f(ps) == 1$  then
29        $pen_{ps} \leftarrow Pen_{ag} \cdot \frac{\Delta q_{ps} - \frac{Q^-}{C^+}}{\sum \Delta q_{Coop}}$ ;
30        $ag_a$  sends  $pen_u$  to  $ps_u$ ;

```

5 NEGOTIATION MECHANISM

In this section, we present the negotiation mechanism, used for contracts over the three time scales, and formalise the different types of contracts handle in each scale. At the first level, providers and prosumers agree on tariff and demand constraints, that lower levels have to satisfy. The goal of the first level is to guarantee to the provider a bounded demand, while it guarantees to prosumers a tariff range. The bounded demand allows to easily know what would be the demand in the future, thus, decreasing the speculation possibilities. Formally, an annual contract is a triplet: $\{Q^{min}, Q^{max}, tr^{max}\}$. At the second level, providers and aggregators contract on a daily contract, formalised by a vector of n hourly contracts: $\langle H^t, H^{t+1}, \dots, H^{t+n} \rangle$. An hourly contract is a triple $\langle q_{ag}, tr_{pv}, t \rangle$ with q_{ag} the forecast demand selling at tr_{pv} at slot t .

5.1 Negotiation of the annual contract

Over the period set out in the contracts, the aggregator will spread the demand of the cooperative among several providers. The maximum amount requested on each provider results from (5). Hence, aggregator and providers will negotiate. They will do concessions on the coefficient tr_{max} and the lower bound using MCP⁵ (there are no concessions on Q_{max} since the cooperative has to be sure to get enough energy at each slot). In the MCP, agents submit round by round proposals making a concession at each new proposal. An agreement is reached when, $u_{ag}(x_{ag}) \leq u_{pv}(x_{pv})$, or $u_{pv}(x_{pv}) \leq u_{ag}(x_{pv})$, with x_{ag} the proposal of the aggregator agent, x_{pv} the proposal of the provider agent, u_{ag} the aggregator utility function and u_{pv} the provider utility function. The Zeuthen strategy indicates the agent which has to make a concession during the next round by calculating the Zeuthen index, $Z_i = \frac{u_i(x_i) - u_i(x_j)}{u_i(x_i)}$. The agent with the lower Z_i has to make a concession. To negotiate the annual contract, aggregators ag_a and providers pv_f use the following utility functions:

- $u_{ag} = 2 \cdot Q_{ag,pv}^{max} - Q_{ag,pv}^{min} - tr_{ag}^{max}$
- $u_{pv} = Q_{ag,pv}^{min} + tr_{ag}^{max}$

The utility of aggregators is high if the negotiated energy band is high, i.e. $Q_{ag,pv}^{max} - Q_{ag,pv}^{min}$ is high and when tr_{ag}^{max} is low. Moreover, the utility is high if $Q_{ag,pv}^{max}$ is high, allowing a bigger amount of energy for the cooperative. The utility of the provider is high when $Q_{ag,pv}^{min}$ is high, i.e. when the negotiated energy band is narrow (as $Q_{ag,pv}^{max}$ is constant the abilities to predict the demand is facilitated) and when the tariff is high. To guarantee the pareto-optimality of negotiated solution with the MCP, both utility functions have to be symmetric, i.e. if $u_i(\mathcal{X}_1) = u_i(\mathcal{X}_2)$ then $u_j(\mathcal{X}_1) = u_j(\mathcal{X}_2)$. So, let $u_i(x_1) = 2 \cdot Q_{x_1}^{max} - Q_{x_1}^{min} - tr_{ag}^{x_1}$ and $u_i(x_2) = 2 \cdot Q_{x_2}^{max} - Q_{x_2}^{min} - tr_{ag}^{x_2}$, with $u_i(x_1) = u_i(x_2)$. Then $u_j(x_1) = Q_{x_1}^{min} + tr_{ag}^{x_1}$ and $u_j(x_2) = Q_{x_2}^{min} + tr_{ag}^{x_2}$. If $u_i(x_1) = u_i(x_2)$ then $2 \cdot Q_{x_1}^{max} - Q_{x_1}^{min} - tr_{ag}^{x_1} = 2 \cdot Q_{x_2}^{max} - Q_{x_2}^{min} - tr_{ag}^{x_2}$. As $Q_{ag,pv}^{max}$ are constant during the negotiations we have $-Q_{x_1}^{min} - tr_{ag}^{x_1} = -Q_{x_2}^{min} - tr_{ag}^{x_2}$ so $Q_{x_1}^{min} + tr_{ag}^{x_1} = Q_{x_2}^{min} + tr_{ag}^{x_2}$ and $u_j(x_1) = u_j(x_2)$ if $u_i(x_1) = u_i(x_2)$. Thus, both utility functions are symmetric.

The concession strategy is the following one for the provider:

- $tr_{t+1}^{max} = tr_t^{max} \cdot \frac{Q_{max} - Q_{t+1}^{min}}{Q_{max} - Q_t^{min}}$, with $tr_{t+1}^{max} < tr_t^{max} \forall t$
- $Q_{t+1}^{min} = Q_t^{min} \cdot \frac{Q_{max} - Q_{t+1}^{min}}{Q_{max} - Q_t^{min}}$, with $Q_{t+1}^{min} < Q_t^{min} \forall t$

We can see that the concessions on tr_t^{max} and Q_t^{min} are monotonic and decreasing. The concession strategy for aggregator are the following one:

- $tr_{t+1}^{max} = tr_t^{max} + \sigma_a^b$, with $tr_{t+1}^{max} > tr_t^{max} \forall t$
- $Q_{t+1}^{min} = (Q_t^{min} + 1) \cdot \frac{Q_{max}}{\sigma_b + pen_f \cdot Q_{max}}$, with $Q_{t+1}^{min} > Q_t^{min} \forall t$

We can see that the concessions on tr_t^{max} and Q_t^{min} are monotonic and increasing. Since (i) the utility functions are symmetric and follow a monotonic behaviour (according to the concession functions), (ii) the use of Zeuthen index, (iii) the MCP context, then the negotiation will converge on a pareto-optimal solution. Moreover, in a context where a cooperative can negotiate with several providers, there are no incentive to lie on the price for the provider. Indeed, if he proposes a higher price, the cooperative will negotiate with other providers and if he proposes a lower price he will do less benefit.

⁵ For these negotiations, agents adopte, in the first instance, the MCP and the Zeuthen strategy. We choose to apply the MCP but our model allows to use other protocols.

5.1.1 Annual contract steps

(i) Each prosumer agent sends its consumption profile to the aggregator. (ii) The aggregator computes the cooperative profile and the distribution of the demands between the providers. It submits the annual contract proposals. (iii) The provider accepts or counters proposals. (iv) The negotiation between the agents continues following the MCP protocol.

5.2 Negotiation of the daily contract

The negotiation of the daily contract follows the following steps:

1. Each prosumer agent sends its consumption profile of the day to the aggregator $\langle q_{ps}^t, q_{ps}^{t+1}, \dots, q_{ps}^{t+n} \rangle$.
2. The aggregator computes the profile of the cooperative $\langle \sum_{ps \in Coop} q_{ps}^t, \sum_{ps \in Coop} q_{ps}^{t+1}, \dots, \sum_{ps \in Coop} q_{ps}^{t+n} \rangle$ and requests providers.
3. Providers send their tariffs $\langle tr_{pv}^t, tr_{pv}^{t+1}, \dots, tr_{pv}^{t+n} \rangle$.
4. Aggregators transfer pricing signal as proposed in [7, 14, 11]. Prosumers can reschedule their planning to reduce their bill.
5. Go to step 2 if there are reschedules.

5.3 The hourly contract

There are no negotiations for the hourly contract. Agents pay for their consumption with penalties according to their deviation. We will show in the next section that, according to their expected payment, prosumers have an incentive to participate in a cooperative.

6 THEORETICAL ANALYSIS

This section studies three properties of the model. First, we show that prosumers reduce their cost when they consume less, even if they have to pay penalties. Then, we give the expected bill of an agent who is a member of a cooperative. We take up this property to show that, in our model, prosumers pay fewer penalties in a cooperative. Finally, we prove the volatility minimisation.

6.1 Monotonicity with respect to consumption

A prosumer may not be incited to consume unnecessarily in order to avoid paying penalties. The perceived benefit by an agent who doesn't consume has to be higher than any penalty: $q_{ps}^t \cdot tr_{ag}^t > q_{ps}^{t'} \cdot tr_{ag}^t + pen_{pv} \cdot (|q_{ps}^t - q_{ps}^{t'}|)$. Let q_{ps}^t be the energy requested and $q_{ps}^{t'}$ the energy consumed:

$$\begin{aligned} q_{ps}^t \cdot tr_{ag}^t &> q_{ps}^{t'} \cdot tr_{ag}^t + pen_{pv} \cdot (q_{ps}^t - q_{ps}^{t'}) \\ q_{ps}^t \cdot tr_{ag}^t &> q_{ps}^{t'} \cdot tr_{ag}^t + pen_{pv} \cdot q_{ps}^t - pen_{pv} \cdot q_{ps}^{t'} \\ q_{ps}^t \cdot tr_{ag}^t - pen_{pv} \cdot q_{ps}^t &> q_{ps}^{t'} \cdot tr_{ag}^t - pen_{pv} \cdot q_{ps}^{t'} \\ q_{ps}^t \cdot (tr_{ag}^t - pen_{pv}) &> q_{ps}^{t'} \cdot (tr_{ag}^t - pen_{pv}) \\ q_{ps}^t &> q_{ps}^{t'} \text{ if } (tr_{ag}^t - pen_{pv}) > 0 \end{aligned}$$

The first inequality shows that our system does not incite agents to consume to avoid under consumption penalties subject to the condition $tr_{ag}^t - pen_{pv} > 0$; i.e. the tariff is higher than the penalties. This shows that the property coming from POU is kept in our case, thus agents are not incited to request energy on the grid to avoid penalties.

6.2 Expected payment

We can formulate the expected penalty paid by an agent by:

$$pen_{ps} = pen_{ag}(\sigma_{ps}^b - \frac{\sqrt{\sum_{ps \in Coop} (\sigma_{ps}^b)^2}}{C^+})$$

with $\sqrt{\sum_{ps \in Coop} (\sigma_{ps}^b)^2}$ the mean deviation of the consumption of the set of agents who limit the deviation, with $\sum_{ps \in Coop} f(x, y, z) = C^+$, the number of agents in this set. Let x_{ag} be the variable representing the deviation of the cooperative, x_{ps} the variable representing the deviation of ps_u , μ_{ag} the mean consumption of the cooperative and μ_{ps} the mean consumption of the prosumer. The prosumer will pay a penalty only in two of these cases: $P(x_{ag} > \mu_{ag}) \cdot P(x_{ps} > \mu_{ps}) + P(x_{ag} < \mu_{ag}) \cdot P(x_{ps} < \mu_{ps}) = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}$. The expected penalty of an agent is:

$$pen_{ps} = \frac{1}{2} \cdot pen_{ag}(\sigma_{ps}^b - \frac{\sqrt{\sum_{ps \in Coop} (\sigma_{ps}^b)^2}}{C^+})$$

The expected payment of an agent becomes :

$$\sum_{t=0}^n tr_{ag}^t \cdot q_{ps}^t + \frac{1}{2} \cdot pen_{ag}(\sigma_{ps}^b - \frac{\sqrt{\sum_{ps \in Coop} (\sigma_{ps}^b)^2}}{C^+})$$

6.3 Individual rationality

We will show that an agent has interest to be a member of a cooperative, by demonstrating that its expected payment is lower when the agent is in a cooperative. Hence, we demonstrate the following inequality:

$$\sum_{t=0}^n tr_{ag}^t \cdot q_{ps}^t + \frac{1}{2} \cdot pen_{ag}(\sigma_{ps}^b - \frac{\sqrt{\sum_{ps \in Coop} (\sigma_{ps}^b)^2}}{\sum_{ps \in Coop} f(x_{ps}, y_{ps}, z)}) < \sum_{t=0}^n tr_{ag}^t \cdot q_{ps}^t + pen_{ps} \cdot \sigma_{ps}^b \quad (1)$$

Suppose that $pen_{ag} = pen_{ps}$, in this case the inequality becomes:

$$\begin{aligned} \frac{1}{2} \cdot (\sigma_{ps}^b - \frac{\sqrt{\sum_{ps \in Coop} (\sigma_{ps}^b)^2}}{\sum_{ps \in Coop} f(x_{ps}, y_{ps}, z)}) &< \sigma_{ps}^b \\ \frac{\sigma_{ps}^b}{2} - \frac{\sqrt{\sum_{ps \in Coop} (\sigma_{ps}^b)^2}}{2 \cdot \sum_{ps \in Coop} f(x_{ps}, y_{ps}, z)} &< \sigma_{ps}^b \\ - \frac{\sqrt{\sum_{ps \in Coop} (\sigma_{ps}^b)^2}}{2 \cdot \sum_{ps \in Coop} f(x_{ps}, y_{ps}, z)} &< \frac{\sigma_{ps}^b}{2} \\ - \frac{\sqrt{\sum_{ps \in Coop} (\sigma_{ps}^b)^2}}{\sum_{ps \in Coop} f(x_{ps}, y_{ps}, z)} &< \sigma_{ps}^b \end{aligned}$$

$\sqrt{\sum_{ps \in Coop} (\sigma_{ps}^b)^2} \geq 0$ and $\sigma_{ps}^b \geq 0$ by definition of standard deviation. $\sum_{ps \in Coop} f(x_{ps}, y_{ps}, z) \geq 1$ because $f(x_{ps}, y_{ps}, z) = 1$ for prosumer agent ps_u . The negative sign in front of the ration of two positive numbers guarantees the inequality. Joining the third property encourage agents to exchange energy in the cooperative before requesting the grid.

6.4 Volatility minimisation

According to function (3), the tariff applied by a provider pv_f is constant and equal to tr^{opt} if its requested amount of energy $q_{pv}^{tot} = p_{pv}^{opt}$. Thus, in case of any pricing signal that encourages consumers to request an amount of energy equal to the optimal production of producers, we have the sum $\sum_{i=1}^{24} \sqrt{(q_{pv}^{tot} - p_{pv}^{opt})^2}$ which is minimising.

This leads to $\sum_{i=1}^{24} \sqrt{(tr_{pv}^i - \overline{tr_{pv}})^2} \rightarrow 0$, with $\overline{tr_{pv}}$ the mean tariff, due to (3). The volatility is the standard deviation of the price. As $\sum_{i=1}^{24} \sqrt{(tr_{pv}^i - \overline{tr_{pv}})^2}$ is the standard deviation of the price we can say that the volatility is going toward 0.

7 EMPIRICAL EVALUATION

In this section, we first describe the initialization of our data. Then, we present and discuss the results of our evaluation. Our goal is to show the price evolution according to the tariff scheme we propose. Thus, we look at the tariff evolution according to the demand evolution (shifting and use of storage capacity to store) and the level of the penalties inside a cooperative which may add volatility in the final price paid by the prosumer.

7.1 Experimental setup

We begin by considering a scenario where each agent is in a cooperative and has a generator, a storage capacity, some loads, some shifting possibilities, some forecast capacities and expected consumption limits.

Consumption limits : Q^{max} (resp. Q^{min}) is drawn randomly in $[Q^{min}, 2 \cdot Q^{min}]$ (resp. $[6.5, 11.5]$) as in [11].

Generator : Each agent is equipped with a generator with a maximal production drawn randomly in $[1, 6]$ 6 KW is the maximum delivery power for suitable domestic⁶ wind turbine or solar panel.

Storage capacity : Each agent is equipped with a storage capacity which is drawn randomly in $[0; 6, 4]$, the 6.4 KWh correspond to the powerwall storage capacity⁷. At the beginning of the simulation the storage capacity starts with an amount of energy which is drawn randomly in $[0, s_{ps}]$.

Loads : At each slot, the forecast consumption of a consumer is drawn randomly in $[0, Q^{max}]$. Then the real consumption is drawn randomly following the law $\mathcal{N}(q^{pred}, \sigma^b)$, so the real consumption can be higher than Q^{max} .

Forecast capacities : σ^b is randomly drawn between 3.59 and 17.9 percent of Q^{max} , 3.59 represents poor predictors and 17.9 good predictors in [13]. σ^p is randomly drawn between 0 and 5 percent of P^{max} .

Shifting possibilities : For each slot, the shifting possibility of the slot is drawn randomly between 0 and 50 percent of the forecast consumption of the slot.

Number of agents : We begin by testing the model with one cooperative. The number of agents in the cooperative is set to 10.

LP Solver : For the different linear programs we use the glpk solver [1].

7.2 Empirical results

Figure 3 shows the deviation harmonization effect. We can observe that the deviation is lower when the agents are in a cooperative than when they are alone. The harmonization effect leads to a reduction of 63% of the deviation, i.e. the under-consumption (or over-consumption) of some agents is compensated at 63% of the over-consumption (or under-consumption) of the others. Thus, this abates the level of penalties, and so doing, the volatility of the rate paid by the prosumers. The mechanism uses the combination of storage capacities and shifting possibilities. The use of this combination allows to shift the demand of the cooperative towards slots where tariffs are low. As figure 2a shows at slot 8, we can observe a peak of demand from the cooperative. At the same slot on figure 2b, we can observe that this peak of demand is in part due to a shift of demand, agents move their shiftable needs and store energy on this slot. On figure 2c, we can note that at slot 8, tariff is not moving so much far from the

⁶ <http://www.energysavingtrust.org.uk/domestic/wind-turbines>

⁷ <http://www.teslamotors.com/powerwall>

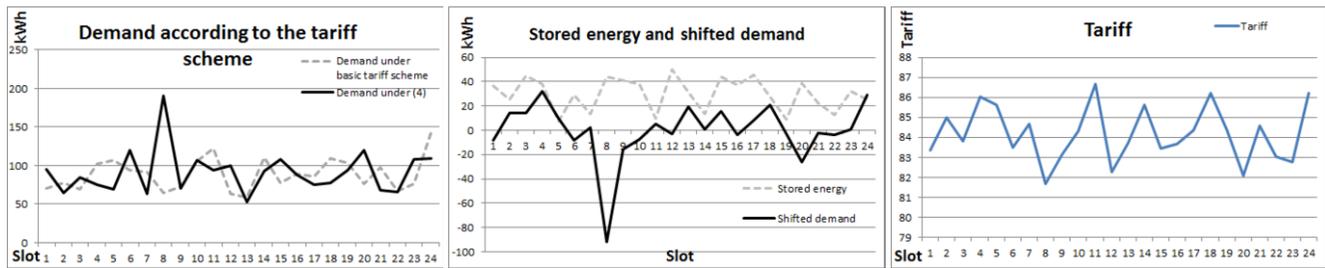


Figure 2. Evolution of the demand of a cooperative and the associated prices

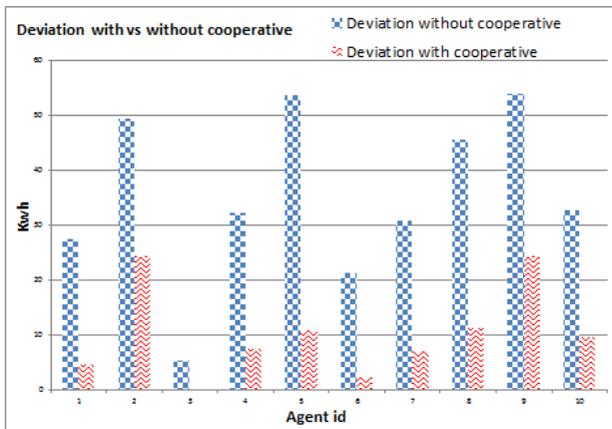


Figure 3. The harmonization effect on consumer deviation

mean price (81.7 for a mean of 84.17). On a more global vision, Figure 2a and 2b show quite high flows in the cooperative management of energy. The demand of the cooperative moves between 50 and 200 kWh/slot during a day. On the shifting side, they evolve from 30 to -90 kWh/slot in the day (-90 denotes that the consumption is scheduled earlier in the day). Against that, figure 2c shows that the range of price is quite low, moving between 81.7 and 86.7 per slot. The volatility of the price in the day is 1.92 (standard deviation) with a mean of 84.17. For comparison, in 2014, a report⁸ outlines a volatility of 11.51 between the beginning of 2009 and the end of 2012.

8 CONCLUSION

In this paper, we attack the problem of designing a sustainable mechanism to limit the pricing volatility in smart grids energy market where lot of agents are willing to buy and sell energy. The mechanism we present extends on three time scales; at the first level negotiations provide pareto-optimal solutions where energy bands and tariff bands are defined. We introduce a new tariff scheme which takes place at the two lowest levels to incentive agents to make and respect their forecast consumption a day ahead. We propose the algorithms of prosumer agents to minimise their cost in the context of bounded prices and demand. We show that our mechanism keeps some properties of POU tariff scheme (individual rationality and monotonicity on price w.r.t the consumption) and reduces the price volatility. Finally, we present an experiment evaluation that shows, in the case of our parameters, that the price progresses in a narrow window. This work is developed in collaboration with an industrial partner in building construction. Further works will focus on *island of prosumers*

(in a building for example) which can be autonomous and disconnected from the grid (in some slot). The goal is to reduce or lower the provider's need and incentives the exchange between prosumer or between islands of prosumers.

REFERENCES

- [1] <http://www.gnu.org/software/glpk/glpk.html>.
- [2] Charilaos Akasiadis and Georgios Chalkiadakis, 'Agent cooperatives for effective power consumption shifting', in *AAAI*, (2013).
- [3] Muddasser Alam, Sarvapali D Ramchurn, and Alex Rogers, 'Cooperative energy exchange for the efficient use of energy and resources in remote communities', in *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pp. 731–738. International Foundation for Autonomous Agents and Multiagent Systems, (2013).
- [4] Mohamed H Albadi and EF El-Saadany, 'A summary of demand response in electricity markets', *Electric Power Systems Research*, **78**(11), 1989–1996, (2008).
- [5] M. Hazewinkel, 'Theory of errors', *Encyclopedia of mathematics*, (2001).
- [6] Karen Herter, 'Residential implementation of critical-peak pricing of electricity', *Energy Policy*, **35**(4), 2121 – 2130, (2007).
- [7] Sarvapali D Ramchurn, Perukrishnen Vytelingum, Alex Rogers, and Nicholas R Jennings, 'Agent-based homeostatic control for green energy in the smart grid', *ACM Transactions on Intelligent Systems and Technology (TIST)*, **2**(4), 35, (2011).
- [8] Sarvapali D Ramchurn, Perukrishnen Vytelingum, Alex Rogers, and Nicholas R Jennings, 'Putting the 'smarts' into the smart grid: a grand challenge for artificial intelligence', *Communications of the ACM*, **55**(4), 86–97, (2012).
- [9] Valentin Robu, Meritxell Vinyals, Alex Rogers, and Nicholas R Jennings, 'Efficient buyer groups for prediction-of-use electricity tariffs', in *AAAI*, (2014).
- [10] P. Samadi, A.-H. Mohsenian-Rad, R. Schober, V.W.S. Wong, and J. Jatskevich, 'Optimal real-time pricing algorithm based on utility maximization for smart grid', in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pp. 415–420, (2010).
- [11] Andreas Veit, Ying Xu, Ronghuo Zheng, Nilanjan Chakraborty, and Katia P Sycara, 'Multiagent coordination for energy consumption scheduling in consumer cooperatives.', in *AAAI*, (2013).
- [12] Meritxell Vinyals, Filippo Bistaffa, Alessandro Farinelli, and Alex Rogers, 'Stable coalition formation among energy consumers in the smart grid', in *Proceedings of the 3rd International Workshop on Agent Technologies for Energy Systems (ATES 2012)*, (2012).
- [13] Meritxell Vinyals, Valentin Robu, Alex Rogers, and Nicholas R. Jennings, *Prediction-of-use games: a cooperative game theory approach to sustainable energy tariffs*, 829–836, ACM, 2014.
- [14] Thomas Voice, Perukrishnen Vytelingum, Sarvapali Ramchurn, Alex Rogers, and Nick Jennings, 'Decentralised control of micro-storage in the smart grid', (2011).

⁸ <http://www.wec-france.org/DocumentsPDF/RECHERCHE/79rapportfinal.pdf>

A Rational Account of Classical Logic Argumentation for Real-World Agents

M. D’Agostino and S.Modgil¹

Abstract.

Classical logic based argumentation (*CIAR*) characterises single agent non-monotonic reasoning and enables distributed non-monotonic reasoning amongst agents in dialogues. However, features of *CIAR* that have been shown sufficient to ensure satisfaction of rationality postulates, preclude their use by resource bounded agents reasoning individually, or dialectically in real-world dialogue. This paper provides a new formalisation of *CIAR* that is both suitable for such uses and satisfies the rationality postulates. We illustrate by providing a rational dialectical characterisation of Brewka’s non-monotonic Preferred Subtheories defined under the assumption of restricted inferential capabilities.

1 Introduction

Context. In Dung’s seminal theory of argumentation [13], arguments are built from a possibly inconsistent knowledge base \mathcal{B} . Attacks between arguments are defined, and preferences over arguments can then be used to decide whether one argument successfully attacks (defeats) another [1, 18]. The graph of arguments and defeats is then evaluated, based on the intuitive principle that an argument is justified if all its defeaters are themselves defeated by justified arguments. \mathcal{B} ’s argumentation defined consequences are then the justified arguments’ conclusions, and have been shown to correspond to the consequence relations of a number of non-monotonic logics. For example, classical logic arguments [15] are pairs (Δ, α) built from a base \mathcal{B} of classical wff, where the premises Δ are a *consistent* subset of \mathcal{B} that classically entail the conclusion α , and no proper subset of Δ entails α . An argument X attacks Y if X ’s conclusion negates one of Y ’s premises. [2, 18] show that given preferences over arguments defined on the basis of a total ordering on \mathcal{B} , the argumentation defined consequences correspond to the non-monotonic consequences from \mathcal{B} defined by Preferred Subtheories (*PS*) [4].

Argumentation’s dialectical characterisation of non-monotonic consequence, and the intuitive, familiar nature of the evaluative principles, accounts for its widely advocated benefits in enabling individual agent reasoning, and distributed (‘dialogical’) reasoning amongst computational and/or human agents [19]. However, features of classical logic instantiations of Dung graphs (*CIAR*) posited to ensure satisfaction of rationality postulates [5, 6], preclude its use by resource-bounded agents reasoning dialectically², either as individuals or in real-world dialogues. Firstly, the consistency and subset minimality checks on arguments’ premises incur prohibitive computational ex-

pense. Moreover, the inconsistency of arguments’ premises are in real-world argumentation established dialectically, by showing that an interlocutor contradicts herself. On the other hand, the consistency check ensures satisfaction of the *non contamination* postulates [6]. Secondly, exclusively targeting attacks at an argument’s premises leads to the so called ‘foreign commitment problem’ whereby an agent is forced to commit to the premises of his interlocutor when attacking his interlocutor’s arguments [16]. However, allowing attacks on the conclusions of arguments results in violation of the *consistency* postulates [5]. Thirdly, consistency may also be violated unless one assumes that a Dung graph includes *all* arguments defined by a base \mathcal{B} . However, this further precludes the use of *CIAR* by resource-bounded agents.

Contributions This paper proposes a new account of *CIAR* that is suitable for resource bounded agents reasoning individually and in real-world dialogues, and is provably rational. We review background in Section 2, and then Section 3 presents our first contribution. We propose a new dialectical ontology for *CIAR* arguments that distinguishes amongst premises assumed true, and those assumed true ‘for the sake of argument’. Agents are therefore not forced to commit to the premises of their interlocutors despite the fact that attacks are targeted at premises. We also accommodate the use of *CIAR* by resource bounded agents, by not requiring consistency or subset minimality checks on arguments’ premises, and, subject to intuitive assumptions on available resources for constructing arguments, we allow for instantiation of Dung graphs by subsets of arguments defined by a base. Our formalisation also accommodates the real-world move whereby the mutual inconsistency of arguments’ premises is demonstrated dialectically. We then provide an account of Preferred Subtheories that assumes limited inferential resources, and show that the defined non-monotonic consequence relation corresponds to the argumentation defined consequences obtained by our dialectical formalisation of *CIAR*. Section 4 presents our second contribution. We show that our approach satisfies key results that hold for Dung’s theory³ despite our conservative adaptation of Dung’s evaluative principles. We also show that despite satisfaction of the above desiderata for real-world applications of *CIAR*, the consistency and closure postulates [5], as well as the *non contamination* postulates [6], are satisfied. Section 5 concludes by discussing related and future work.

2 Background

We review classical logic instantiations of Dung graphs (*CIAR*) [15, 18] that study [5]’s rationality postulates. We assume the propositional language \mathcal{L} consisting of atoms \perp, a, b, c, \dots with the

¹ University of Milan, email: marcello.dagostino@unimi.it, and King’s College London, email: sanjay.modgil@kcl.ac.uk

² By ‘dialectic’ we mean ‘a method of examining and discussing opposing ideas in order to find the truth’ (www.merriam-webster.com).

³ We will refer the reader to [11] where space limitations preclude full details of proofs in this paper.

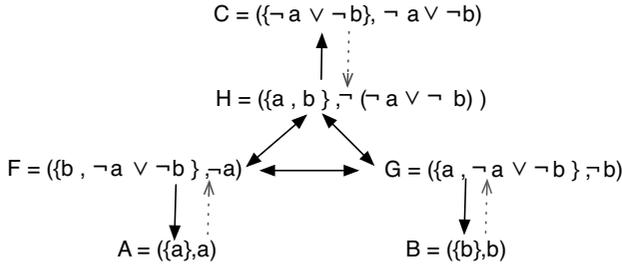


Figure 1. Attacks on premises are solid arrows. Dotted arrows are additional attacks if attacks can target conclusions.

usual connectives and definition of classical *wff*. Lower case and upper case Greek letters (as well as the symbol \mathcal{B}) respectively refer to arbitrary classical *wff* and finite sets of classical *wff*. We assume the complement function:

$$\bar{\phi} = \psi \text{ if } \phi \text{ is of the form } \neg\psi; \text{ else } \bar{\phi} = \neg\phi$$

and let $Cn(\Delta)$ denote $\{\alpha \mid \Delta \vdash \alpha\}$, where \vdash is the classical consequence relation. The arguments \mathcal{A} defined by a base \mathcal{B} of classical propositional *wff*, are pairs (Δ, α) where $\Delta \subseteq \mathcal{B}$, and: 1) the premises Δ are consistent; 2) $\Delta \vdash \alpha$; 3) no strict subset of Δ satisfies 2.

[15] study *CIAR* assuming variously defined notions of attacks. [18] additionally assume a strict argument preference relation $\prec \subseteq \mathcal{A} \times \mathcal{A}$ ($X \prec Y$ denotes Y strictly preferred to X), and define attacks and defeats as follows. For any $X = (\Delta, \alpha), Y = (\Gamma, \beta) \in \mathcal{A}$:

- X attacks Y (denoted $(X, Y) \in \mathcal{C}$) if $\alpha = \bar{\gamma}$ for some $\gamma \in \Gamma$, in which case X is said to attack Y on γ (or on $(\{\gamma\}, \gamma)$).
- X defeats Y (denoted $(X, Y) \in \mathcal{D}$) if X attacks Y on γ , and $X \not\prec (\{\gamma\}, \gamma)$ ($X \Rightarrow Y$ ($X \not\Rightarrow Y$) denotes that X does (not) defeat Y).

[18] study well known preference relations over arguments that are defined by an ordering \leq over the formulae in \mathcal{B} (where $<$ and \sim are defined in the usual way). In particular, for any $X = (\Delta, \alpha), Y = (\Gamma, \beta)$:

$$X \prec_{El} Y \text{ if } \exists \delta \in \Delta, \forall \gamma \in \Gamma: \delta < \gamma \quad (\textit{Elitist preference})$$

A Dung argumentation framework (*AF*) is then a tuple $(\mathcal{A}, \mathcal{D})$. For any $E \subseteq \mathcal{A}$, $X \in \mathcal{A}$ is *acceptable* w.r.t. (i.e., *defended* by) E if $\forall Y$ s.t. Y defeats X , $\exists Z \in E$ s.t. Z defeats Y . The extensions (sets of justified arguments) can then be defined under Dung's semantics [13]:

Definition 1 Let $(\mathcal{A}, \mathcal{D})$ be a *AF*. Then $E \subseteq \mathcal{A}$ is conflict free if $\forall X, Y \in E: X \not\Rightarrow Y$. For any conflict free $E \subseteq \mathcal{A}$:

E is said to be an extension that is: *admissible* if every $X \in E$ is acceptable w.r.t. E ; *complete* if admissible and every $X \in \mathcal{A}$ that is acceptable w.r.t. E , is in E ; *grounded* if E is the minimal (under set inclusion) complete extension; *preferred* if E is a maximal (under set inclusion) complete extension; *stable* if every $Y \notin E$ is defeated by an argument in E .

A correspondence then holds between the stable extensions of a *AF* $(\mathcal{A}, \mathcal{D})$ defined by (\mathcal{B}, \leq) (where \leq is total, and $\mathcal{A}, \mathcal{C}, \prec_{El}, \mathcal{D}$ are defined as above), and the widely studied Preferred Subtheories (*PS*) non-monotonic consequence relations defined over (\mathcal{B}, \leq) [4].

Definition 2 Let $(\mathcal{B}_1, \dots, \mathcal{B}_n)$ be the stratification of (\mathcal{B}, \leq) such that $\alpha \in \mathcal{B}_i, \beta \in \mathcal{B}_j, i < j$ iff $\beta < \alpha$. A preferred subtheory (*ps*) Σ is a set $\Sigma_1 \cup \dots \cup \Sigma_n$ such that for $i = 1 \dots n$, $\Sigma_1 \cup \dots \cup \Sigma_i$ is a \subseteq -maximal consistent subset of $\mathcal{B}_1 \cup \dots \cup \mathcal{B}_i$.

Intuitively, a *ps* is obtained by taking a \subseteq -maximal consistent subset of \mathcal{B}_1 , extending this with a \subseteq -maximal consistent subset of \mathcal{B}_2 , and so on. For example, $\Sigma = \{a, \neg a \vee \neg b\}$ and $\Sigma' = \{a, b\}$ are the *ps* of $(\mathcal{B}_1 = \{a\}, \mathcal{B}_2 = \{\neg a \vee \neg b, b\})$. [18] then show that:

Σ is a preferred subtheory of (\mathcal{B}, \leq) iff E is a stable extension of the *AF* defined by (\mathcal{B}, \leq) , where $\Delta \subseteq \Sigma$ iff $(\Delta, \alpha) \in E$.

One then obtains a correspondence between the *PS* non-monotonic consequences and the argumentation defined consequences. These can be defined either sceptically:

$$\{\alpha \mid \forall \Sigma : \Sigma \vdash \alpha\} = \{\alpha \mid \forall E : \exists (\Delta, \alpha) \in E\}$$

or credulously, which involves selecting the classical consequences of a single *ps*, equivalently the conclusions of arguments in a single stable extension. For example, Figure 1 shows some of the arguments and attacks (represented as solid arrows) defined by $(\{a, b, \neg a \vee \neg b\}, b \sim \neg a \vee \neg b < a)$. The ordering determines that $F \prec_{El} A$, hence $F \not\Rightarrow A, F \not\Rightarrow H, F \not\Rightarrow G$, and the remaining attacks succeed as defeats. Then $\{A, C, G\}$ and $\{A, H, B\}$ are, respectively, subsets of the two stable extensions E and E' , and α is a conclusion of an argument in E iff $\alpha \in Cn(\{a, \neg a \vee \neg b\})$, α is a conclusion of an argument in E' iff $\alpha \in Cn(\{a, b\})$.

A number of features of *CIAR* ensure that rationality postulates for argumentation are satisfied.

Firstly, the consistency check on arguments' premises ensures satisfaction of the *non contamination* postulates [6]. Essentially, these postulates state that arguments built from syntactically disjoint subsets of \mathcal{B} should not impact on each other's justification status. To illustrate, suppose $\mathcal{B} = \{p, \neg p, s\}$, and suppose we allow the 'inconsistent argument' $Y = (\{p, \neg p\}, \neg s)$ which attacks $X = (\{s\}, s)$. Then (assuming $\prec = \emptyset$) there is a complete extension (the grounded extension) that does not include X ⁴. However, intuitively the status of X should not be affected by arguments built from the syntactically disjoint $\{p, \neg p\}$.

Secondly, exclusively targeting attacks on arguments' premises ensures satisfaction of the consistency postulates [5]. To see why, observe that if attacks on conclusions are additionally permitted (as illustrated by the dotted attacks in Figure 1), then (assuming $\prec = \emptyset$), one obtains an additional stable extension containing $\{A, B, C\}$, whose conclusions are mutually inconsistent.

Thirdly, in *CIAR* it is tacitly assumed that all arguments defined by \mathcal{B} are included in the *AF* for evaluation. In particular:

$$\text{if } (\Gamma, \alpha) \in \mathcal{A}, \text{ then } \forall \gamma \in \Gamma, (\Gamma \setminus \{\gamma\} \cup \{\bar{\alpha}\}, \bar{\gamma}) \in \mathcal{A} \quad (\textit{contraposition})$$

is posited as sufficient for satisfaction of the consistency postulates. To illustrate, suppose in Figure 1's example, that the *AF* only includes F and A , and let $F \prec A$. Then neither A or F defeat each other, and both are contained in a complete extension E that violates consistency. However, assuming contraposition the *AF* must additionally include G and H . Moreover, [18] also show that if $F \prec A$, and \prec satisfies properties that are deemed 'reasonable', then it must be that either $G \not\Rightarrow B$ or $H \not\Rightarrow C$, and so either G defeats F on B , or H defeats F on C . But then E cannot be complete. To see why, if

⁴ In general, all arguments in an *AF* will be attacked by arguments built from inconsistent premises (given the *ex-falso* principle), and it is well known that \emptyset is the grounded extension of an *AF* that contains no un-attacked arguments.

either G or H defeats F , then by assumption of E being complete, there must be an argument in E that defends F by defeating G or H . But then any such argument must also defeat A or F , and so E would not be conflict free. Hence, contraposition and reasonable preference relations are shown to guarantee satisfaction of consistency (note that [18] show that the *Elitist* preference is reasonable).

3 Dialectical Classical Logic Argumentation

3.1 Motivation

Apart from its intuitive characterisation of single agent reasoning, a key advantage of argumentation [19] is that it provides a formal basis for dialogue amongst computational and/or human agents [17]. Given argumentative characterisations of non-monotonic consequence relations (e.g. Logic Programming [13], Prioritised Default Logic [23] and Preferred Subtheories [18]), such dialogues effectively enable distributed non-monotonic reasoning amongst communicating agents. Agents submit arguments⁵ in order to establish acceptance of the initial claim (a belief or decision option). Intuitively, the agent advocating the initial claim, attempts to build an admissible extension that includes an argument concluding the claim. In these dialogues, a base \mathcal{B}_p is incrementally defined by the agents' 'public commitments'; that is, the contents of exchanged locutions (rather than assuming a given initial base in the case of single agent reasoning), and agents can construct arguments from premises in their private bases *and* the incrementally defined \mathcal{B}_p .

However, we argue that the three features of *CIAR* shown to ensure satisfaction of rationality postulates (as discussed in the previous section): 1) preclude use of *CIAR* by resource bounded agents reasoning individually or in dialogues, and; 2) preclude modelling features of dialectical reasoning that are ubiquitous in real-world dialogue.

Firstly, the tacit assumption that an *AF* is instantiated by all arguments defined by a base \mathcal{B} , is clearly not feasible for resource bounded agents, given that deciding whether $\Delta \vdash \alpha$ is in general NP-hard (hence most likely intractable). One would thus want to *identify as 'undemanding' a set of assumptions as possible on available resources for constructing arguments, such that rationality is preserved when AFs are not instantiated by all definable arguments (D1)*.

In particular one would want to relax the contraposition condition. To illustrate, suppose arguments are classical *Intelim* natural deduction (*I-ND*) proofs [9]. *I-ND* allows parameterisation of proofs by the depth of nesting of discharged assumptions, such that step-wise increments in depth define a hierarchy of tractable inference relations, and each depth bounded system can be used to reflect the assumed inferential capabilities of real-world agents. Now, suppose $Ag1$ submits arguments whose premises include the inconsistent $\Pi = \{p, p \rightarrow \neg q, p \rightarrow q\}$. $Ag2$ can respectively attack these premises with $A = (\{p \rightarrow q, p \rightarrow \neg q\}, \neg p)$ or $B = (\{p, p \rightarrow q\}, \neg(p \rightarrow \neg q))$, or $C = (\{p, p \rightarrow \neg q\}, \neg(p \rightarrow q))$. Assuming \prec is reasonable, one such attack must be a defeat. Hence $Ag1$ must defend itself by submitting an argument that defeats A or B or C . But then this argument must defeat one of $Ag1$'s own arguments on a premise in Π , and so $Ag1$ cannot construct an admissible set containing the arguments with premises Π . However, suppose neither of the attacks by B and C succeed, so that $Ag2$ must defeat with A . But then it may be that $Ag2$ has insufficient resources to construct A (indeed, in *I-ND*, constructing A requires greater nesting of discharged assumptions than B or C) and so $Ag1$ may be able to construct an

admissible extension containing arguments with mutually inconsistent premises.

Furthermore, the computational non-viability of *CIAR* is further exacerbated by the checks on arguments' premises. Checking for consistency is of course as computationally demanding as deciding $\Delta \vdash \alpha$ ⁶. Moreover, the subset minimality check implies that for every constructed argument (Δ, α) , one must in the worst case check that $\forall \Delta' \subset \Delta, \Delta' \not\vdash_c \alpha$. Hence, for resource bounded agents one would want *a rational account of CIAR that does not require checking for consistency or subset minimality of premises (D2)*.

Moreover, in real-world dialogues, the inconsistency of arguments' premises is typically established *dialectically*, via the well known Socratic move of demonstrating that an opponent's argument(s) rests on inconsistent premises [7, 21]. Also, in real-world dialogues one wants to avoid the anomaly of an agent being forced to commit to the premises of his interlocutor (known as the 'foreign commitment problem' [16]), which arises due to restricting attacks to targeting premises. To illustrate, consider an agent $Ag1$ submitting A in Figure 1. $Ag2$ counters with F . $Ag1$ cannot now counter F with A , but rather has to publically commit to a premise of his opponent (either b or $\neg a \vee \neg b$), by defending A with either H or G , and so having to possibly defend these premises from challenges by other agents. Hence, one would want *an account of CIAR that accommodates the dialectical demonstration that arguments' premises are inconsistent (D3) and avoids the foreign commitment problem (D4)*.

3.2 Defining Dialectical Argumentation

We now formalise an account of *CIAR* that satisfies the desiderata **D1** – **D4**. Our starting point is the observation that when interlocutors construct arguments, they typically distinguish their own premises that they accept as true, from the premises that their opponent commits to and that they want to criticise: "on the basis of the premises I regard to be true, and supposing for the sake of argument what you regard to be true, then I can show some conclusion that contradicts one of your premises". This pattern is pervasive in real argumentation practice, and motivates the following definition of arguments in which we also drop the consistency and subset minimality checks. Attacks are then targeted only at premises and *not* suppositions. Also, arguments may now conclude \perp , and these can target any premise. However, letting $\text{atoms}(\mathcal{B})$ denote the set of propositional atoms in \mathcal{B} , we henceforth assume finite bases \mathcal{B} such that $\perp \notin \text{atoms}(\mathcal{B})$ (i.e., \perp is reserved as a notational device to express that an inconsistency has been reached in the course of constructing an argument).

Definition 3 A dialectical argument X defined by \mathcal{B} is a triple (Δ, Γ, α) such that $(\Delta \cup \Gamma) \subseteq \mathcal{B}$ and $\alpha \in \text{Cn}(\Delta \cup \Gamma)$.

We say that Δ , Γ and α are, respectively, X 's premises, suppositions and conclusion. We let $\text{prem}(X) = \Delta$, $\text{supp}(X) = \Gamma$, and generalise this notation to sets of arguments in the obvious way.

Also, if $\text{Cn}(\Delta \cup \Gamma) = \mathcal{L}$ then X is said to be *inconsistent*; else X is *consistent*. Finally, if $\text{supp}(X) = \emptyset$ then X is said to be *unconditional*; else X is *conditional*.

Let \mathcal{A} be the dialectical arguments defined by \mathcal{B} . Then:

$\mathcal{C} = \{(X, Y) \mid X, Y \in \mathcal{A}, X = (\Delta, \Gamma, \phi)(\phi = \alpha \text{ or } \phi = \perp), Y = (\Pi, \Sigma, \psi) \text{ and if } \phi = \alpha \text{ then } \bar{\alpha} \in \Pi\}$.

If $\phi = \alpha$, X is said to attack Y on premise $\bar{\alpha}$; equivalently, on the

⁵ Arguments may be defined implicitly, e.g., $(\{q, q \rightarrow p\}, p)$ obtained by claiming q and (responding to a 'why' locution) asserting 'since $q, q \rightarrow p$ '.

⁶ Consistency checking is computationally hard not just for isolated or artificially constructed examples, but also in typical cases, as shown in [8].

argument $Y' = (\{\bar{\alpha}\}, \emptyset, \bar{\alpha})$. If $\phi = \perp$, X attacks Y on any $\beta \in \Pi$ (any $Y' = (\{\beta\}, \emptyset, \beta)$).

In dialogues, agents can suppose the truth of premises in their interlocutors' argument(s), when attacking their interlocutors' arguments. This motivates the following notion of *dialectical attacks*:

Definition 4 Let $S \subseteq \mathcal{A}$. Then $X \in \mathcal{A}$ *dialectically attacks* $Y \in \mathcal{A}$, with respect to S (denoted $X \rightarrow_S Y$) iff $(X, Y) \in \mathcal{C}$ and $\text{supp}(X) \subseteq \text{prem}(S)$.

Example 1 The dialectical arguments \mathcal{A} defined by $\mathcal{B} = \{a, b, \neg a \vee \neg b\}$ include:

$A_1 = (\{a\}, \emptyset, a)$	$B_1 = (\{b\}, \emptyset, b)$
$C_1 = (\{\neg a \vee \neg b\}, \emptyset, \neg a \vee \neg b)$	$F_1 = (\{b, \neg a \vee \neg b\}, \emptyset, \neg a)$
$G_1 = (\{a, \neg a \vee \neg b\}, \emptyset, \neg b)$	$H_1 = (\{a, b\}, \emptyset, \neg(\neg a \vee \neg b))$
$F_2 = (\{b\}, \{\neg a \vee \neg b\}, \neg a)$	$G_2 = (\{a\}, \{\neg a \vee \neg b\}, \neg b)$
$H_2 = (\{a\}, \{b\}, \neg(\neg a \vee \neg b))$	$I_1 = (\{a, b, \neg a \vee \neg b\}, \emptyset, \perp)$
$I_2 = (\emptyset, \{a, b, \neg a \vee \neg b\}, \perp)$	

Notice that G_1 and G_2 are epistemically distinguished by the partitioning of premises and suppositions, but are 'logically equivalent'⁷. In what follows, we refer to preference relations that are *invariant modulo logical equivalence*.

Definition 5 Let $X = (\Delta, \Gamma, \alpha)$. Then:

- $[X] = \{X' = (\Delta', \Gamma', \alpha) \mid \Delta' \cup \Gamma' = \Delta \cup \Gamma\}$.
- $\forall Y, Z \in [X]$ we say that Y and Z are logically equivalent.
- $\prec \subseteq \mathcal{A} \times \mathcal{A}$ is invariant modulo logical equivalence (*imle*) if $Y \prec X$ implies $\forall X' \in [X], \forall Y' \in [Y] : Y' \prec X'$

We now define dialectical defeat, acceptability and extensions under Dung's semantics. Any defeating argument Y challenging the acceptability of X w.r.t. E , can suppose the truth of premises in arguments in E , and any defense against Y can suppose the truth of premises in Y :

Definition 6 Let $(\mathcal{A}, \mathcal{C}, \prec)$ be a Dialectical Classical Framework (*DCF*) where \mathcal{A}, \mathcal{C} are defined as in Definition 3, and \prec is *imle*.

- $X \in \mathcal{A}$ *defeats* $Y \in \mathcal{A}$, with respect to $S \subseteq \mathcal{A}$ (denoted $X \Rightarrow_S Y$) iff $X \rightarrow_S Y$ on Y' , and $X \not\prec Y'$.
- Let $E \subseteq \mathcal{A}$. Then $X \in \mathcal{A}$ is *acceptable* w.r.t. E iff $\forall Y \in \mathcal{A}$ s.t. $Y \Rightarrow_{E \cup \{X\}} X$, $\exists Z \in E$ s.t. $Z \Rightarrow_{\{Y\}} Y$.

Conflict free sets and extensions of *DCF*s are defined as in Definition 1, where $E \subseteq \mathcal{A}$ is now conflict free if for no $X, Y \in E$, $X \Rightarrow_{E \cup \{Y\}} Y$, and an extension is *stable* if $\forall Y \notin E, \exists X \in E$ s.t. $X \Rightarrow_{\{Y\}} Y$. The argumentation defined consequences are then the conclusions of *unconditional arguments*⁸ in extensions of a *DCF*.

Example 2 (Example 1 cont.) Suppose $\{A_1, G_1, G_2, C_1\} \subseteq E$, $F_1 \not\prec A_1$. Then:

$F_1 \Rightarrow_{E \cup \{A_1\}} A_1$. Also, $F_2 \Rightarrow_{E \cup \{A_1\}} A_1$ since $\{\neg a \vee \neg b\} \subseteq \text{prem}(E \cup \{A_1\})$ and \prec is *imle*.

G_1 attacks F_1 and F_2 on b (on B_1), and $G_2 \rightarrow_{\{F_1\}} F_1$ on B_1 , but $G_2 \not\rightarrow_{\{F_2\}} F_2$ since $\text{supp}(G_2) \not\subseteq \text{prem}(F_2)$.

Suppose $G_1 \not\prec B_1$. Hence, $G_1 \Rightarrow_{\{F_1\}} F_1$, $G_1 \Rightarrow_{\{F_2\}} F_2$, and since \prec is *imle*, $G_2 \Rightarrow_{\{F_1\}} F_1$.

⁷ In the sense that they are identical proofs distinguished only by the distinction between premises and suppositions. This is a stronger notion of equivalence than that which would apply to $(\{a, b\}, \emptyset, a \wedge b)$ and $(\{a \wedge b\}, \emptyset, a \wedge b)$.

⁸ Since their conclusions are based only on premises assumed true, and not premises supposed true for the sake of argument.

Continuing with this example, suppose the Elitist preference \prec_{El} defined by an ordering \leq on \mathcal{B} . We illustrate how the desiderata **D1** – **D4** are satisfied. We will formally show satisfaction of the rationality postulates in Section 4

(D4) Suppose an admissible extension E_1 containing A_1 , such that $F_1 \Rightarrow_E A_1$ (when a defeat is on $X \in E$ we will index the defeat with E rather than $E \cup \{X\}$). Now, rather than defending A_1 with G_1 , it suffices to include G_2 in E_1 in order to defeat F_1 . G_2 does not include as a premise (and so does not imply commitment to and the potential need to defend) the opponent's premise $\neg a \vee \neg b$.

(D2) Note that any argument in E_1 will be attacked (on any premise) by the inconsistent I_1 . However, I_2 , which has empty premises and so cannot be attacked (and is therefore said to be *unassailable*), is trivially acceptable w.r.t. any set of arguments, and so can be included in E_1 . I_2 attacks I_1 (since $\text{supp}(I_2) \subseteq \text{prem}(I_1)$) on each of I_1 's premises (i.e., on A_1, B_1 and C_1), and at least one of these attacks must succeed as a defeat. To suppose otherwise would mean that $I_2 \prec_{El} A_1, I_2 \prec_{El} B_1$ and $I_2 \prec_{El} C_1$. But it is easy to verify that these preferences hold only if we assume $\alpha < \alpha$ for some $\alpha \in \{a, b, \neg a \vee \neg b\}$, contradicting the irreflexivity of $<$. This illustrates how *non-contamination* is satisfied, despite arguments with inconsistent premises. Recall the example base $\{p, \neg p, s\}$ in Section 3.1. Then $X = (\{s\}, \emptyset, s)$ is in every complete extension E , since even though $I = (\{p, \neg p\}, \emptyset, \neg s)$ may defeat X , any such E will include the unassailable $(\emptyset, \{p, \neg p\}, \perp)$ which must (by the same reasoning as above) defeat I on p or $\neg p$, and so defend X .

(D3) Furthermore, we can now formalise the dialectical move whereby one shows that an interlocutor contradicts himself. Recall Section 3.1 and the arguments with inconsistent premises $p, p \rightarrow \neg q, p \rightarrow q$. Any E that includes these arguments cannot be admissible since given $I = (\emptyset, \{p, p \rightarrow \neg q, p \rightarrow q\}, \perp)$, then $I \rightarrow_E X$ for any $X \in E$, and (reasoning as above) either $I \Rightarrow_E (\{p\}, \emptyset, p)$ or $I \Rightarrow_E (\{p \rightarrow \neg q\}, \emptyset, p \rightarrow \neg q)$ or $I \Rightarrow_E (\{p \rightarrow q\}, \emptyset, p \rightarrow q)$. Since I is unassailable, no argument in E can defend against I .

(D1) The above illustrates that consistency is preserved, despite not having to assume all arguments defined under contraposition. We now define the notion of a *partially instantiated DCF* (*pDCF*), which makes a relatively undemanding (in terms of the required resources) set of assumptions as to the arguments that must be included in a *DCF* and that suffice to guarantee satisfaction of the rationality postulates. One can thus, for example, assume instantiation by a finite subset of the arguments defined by a base, and so accommodate uses of argumentation by real-world agents with limited resources. Before defining *pDCF*s, we introduce the following required notation:

Notation 3 $\mathcal{B} \parallel \mathcal{B}'$ denotes $\text{atoms}(\mathcal{B}) \cap \text{atoms}(\mathcal{B}') = \emptyset$ (\mathcal{B} and \mathcal{B}' are said to be *syntactically disjoint*). Also, $\mathcal{B}|_{At} = \{\alpha \in \mathcal{B} \mid \text{atoms}(\{\alpha\}) \subseteq At\}$ (e.g., $\{\neg a \vee \neg b, c \wedge a\}|_{\{a, b\}} = \{\neg a \vee \neg b\}$).

Definition 7 $(\mathcal{A}, \mathcal{C}, \prec)$ is a *partially instantiated DCF* (*pDCF*) if \mathcal{A} is any subset of the set of all arguments defined by a base \mathcal{B} , such that:

- P1 $\forall \alpha \in \mathcal{B} : (\{\alpha\}, \emptyset, \alpha) \in \mathcal{A}$
- P2 If $X \in \mathcal{A}$ then $\forall X' \in [X] : X' \in \mathcal{A}$
- P3 If $(\Delta_1, \Gamma_1, \alpha) \in \mathcal{A}, (\Delta_2, \Gamma_2, \bar{\alpha}) \in \mathcal{A}$, then $(\Delta_1 \cup \Delta_2, \Gamma_1 \cup \Gamma_2, \perp) \in \mathcal{A}$.
- P4 If $(\Delta \cup \Gamma, \emptyset, \alpha) \in \mathcal{A}$ and $\Delta \parallel \Gamma \cup \{\alpha\}$, then either $(\Delta, \emptyset, \perp) \in \mathcal{A}$ or $(\Gamma, \emptyset, \alpha) \in \mathcal{A}$.

P1 is self-explanatory. P2 expresses that given some X , additional resources are not required to assume construction of logically equivalent arguments (since these differ only in terms of the epistemic distinction between premises and suppositions). P3 is key for showing

consistency. It expresses that given arguments with conflicting conclusions, then resources suffice to combine their premises and suppositions to yield inconsistent arguments. To illustrate, in Example 1, suppose we only assume construction of the conflicting $A1$ and $F1$, and $F1 \prec A1$. By P3 and P2, we have the unassailable I_2 which must (reasoning as described earlier) defeat $F1$ or $A1$. Hence no admissible extension can include the conflicting $F1$ and $A1$, despite the absence of arguments defined under contraposition.

Finally, P4 is required to show satisfaction of the non-contamination postulates. To elaborate, since standard accounts of *CIAR* make no reference to specific proof theories for constructing arguments, they employ subset minimality as a somewhat 'blunt instrument' for ensuring that premises are relevant to deriving the argument's conclusion⁹. However, in practice agents clearly do not check for subset minimality. Rather, the proof theoretic means by which one entails a conclusion from premises, may ensure to varying degrees, the relevance of the premises for deriving the conclusion. Now, let us identify a notion of relevance that in principle can be satisfied by specific proof theories. Observe that by the properties of classical logic, if $\Delta \cup \Gamma \vdash \alpha$, $\Delta \parallel \Gamma \cup \{\alpha\}$, and α is not a tautology, then either α is provable from Γ (in which case Δ is redundant) or α must be provable from the inconsistent Δ by the explosivity of classical logic (in which case Γ is redundant). Of course, if α is a tautology, then $\Gamma \vdash \alpha$. Indeed, the *I-ND* natural deduction proof theory of [9] allows for a notion of proof that does not make use of syntactically disjoint premises; thus irrelevant proofs of this kind cannot be constructed (see [10, Definition 15, Theorem 9]). However, for proof theories that do allow such proofs, P4 simply states that if resources suffice to construct an argument that redundantly uses premises, then resources suffice to construct their non-redundant versions¹⁰.

We now show that Dialectical *CIAR* characterises Preferred Subtheories, where the latter is now defined under the assumption that resources may not suffice to infer all classical consequences from a base.

Definition 8 Let $\vdash_r \subseteq \vdash$ be any resource bounded classical consequence relation, such that: 1) for any Δ , if $\beta \in \Delta$ then $\Delta \vdash_r \beta$; 2) if $\Delta \vdash_r \alpha$ and $\Delta \vdash_r \neg\alpha$ then $\Delta \vdash_r \perp$.

We say Δ is *r*-inconsistent iff $\Delta \vdash_r \perp$; *r*-consistent otherwise. A *r*-preferred subtheory of (\mathcal{B}, \leq) is then defined as in Definition 2, with '*r*-consistent' substituting for 'consistent'.

The following uses the notation $\text{Args}(\Sigma) = \{X \mid \text{prem}(X) \subseteq \Sigma\}$.

Theorem 4 Let $(\mathcal{A}, \mathcal{C}, \prec_{EI})$ be a pDCF defined by (\mathcal{B}, \leq) , such that $(\Delta, \Gamma, \alpha) \in \mathcal{A}$ iff $\Delta \cup \Gamma \vdash_r \alpha$. Then:

1) Σ is a *r*-preferred subtheory of (\mathcal{B}, \leq) implies $E = \text{Args}(\Sigma)$ is a stable extension of $(\mathcal{A}, \mathcal{C}, \prec_{EI})$.

2) E is a stable extension of $(\mathcal{A}, \mathcal{C}, \prec_{EI})$ implies $\Sigma = \bigcup_{X \in E} \text{Prem}(X)$ is a *r*-preferred subtheory of \mathcal{B} .

PROOF.

Proof of 1): Suppose for contradiction that E is not conflict free. Then $X, Y \in E$, $X = (\Delta, \Gamma, \phi)$ ($\phi = \perp$ or β), $X \Rightarrow_E Y$ on

⁹ Clearly arguments may not be subset minimal and yet use all the premises to derive a conclusion, e.g., two applications of modus ponens deriving q from p , $p \rightarrow q$, $p \rightarrow ((p \rightarrow q) \rightarrow q)$.

¹⁰ For example, consider r provable from $(\Gamma = \{p, p \rightarrow r\}) \cup (\Delta = \{q\})$. Assuming natural deduction rules, one could by $\wedge_{\mathcal{I}}$ obtain $p \wedge q$, then by $\wedge_{\mathcal{E}}$, p , and then by $\rightarrow_{\mathcal{E}}$, r from p and $p \rightarrow r$. Clearly, such a proof, which redundantly makes use of q , implies sufficient resources for a proof of r from Γ (by a single application of $\rightarrow_{\mathcal{E}}$).

$\bar{\beta} \in \text{prem}(Y)$. We have $\Sigma \vdash_r \bar{\beta}$. Since $\Gamma \subseteq \text{prem}(Y) \subseteq \text{prem}(E)$, then $\Sigma \vdash_r \perp$ or β . Either case contradicts Σ is *r*-consistent.

Suppose $Y \in \mathcal{A} \setminus E$. Hence $\exists \gamma \in \text{prem}(Y)$, $\gamma \notin \Sigma$. We show $\exists X \in E$, $X \Rightarrow_{\{Y\}} Y$. By construction, $\Sigma = \Sigma_1 \cup \dots \cup \Sigma_n$ such that for $i = 1 \dots n$, $\Sigma_1 \cup \dots \cup \Sigma_i$ is a maximal *r*-consistent subset of $\mathcal{B}_1, \dots, \mathcal{B}_i$. Hence, suppose $\gamma \in \mathcal{B}_j$ for some $j = 1 \dots n$. Then $\Sigma_1 \cup \dots \cup \Sigma_j \cup \{\gamma\} \vdash_r \perp$. Hence $X = (\Delta, \{\gamma\}, \perp) \in \text{Args}(\Sigma_1 \cup \dots \cup \Sigma_j) \subseteq E$ s.t. $X \rightarrow_{\{Y\}} Y$. Since $\gamma \in \mathcal{B}_j$, and $\Delta \subseteq \bigcup_{k=1}^j \mathcal{B}_k$, $X \not\prec_{EI} (\{\gamma\}, \emptyset, \gamma)$. Hence $X \Rightarrow_{\{Y\}} Y$.

Proof of 2): Suppose for contradiction that $\Sigma = \bigcup_{X \in E} \text{Prem}(X)$ is not *r*-consistent (i.e., $\Sigma \vdash_r \perp$). Then $Z = (\emptyset, \Sigma, \perp) \in \mathcal{A}$. By properties of \prec_{EI} (see Section 3.2) $\exists \alpha \in \Sigma$, $Z \not\prec (\{\alpha\}, \emptyset, \alpha)$. Hence $\exists B \in E$, $Z \Rightarrow_E B$ on α . No argument in E can defeat the unassailable Z , contradicting E is stable.

Suppose for contradiction that Σ is not \subseteq -maximal *r*-consistent. Let $\Sigma_1, \dots, \Sigma_n$ partition Σ s.t. for $i = 1 \dots n$, Σ_i is a (possibly empty) subset of \mathcal{B}_i . Then, for some i , for $k = 1 \dots i-1$, $\Sigma_1, \dots, \Sigma_k$ is a \subseteq -maximal *r*-consistent subset of $\mathcal{B}_1, \dots, \mathcal{B}_{i-1}$, and $\exists \alpha \in \mathcal{B}_i$ s.t.:

i) $\alpha \notin \Sigma_i$ ii) $\Sigma_1 \cup \dots \cup \Sigma_{i-1} \cup \Sigma_i \cup \{\alpha\} \not\vdash_r \perp$.

Given i), $\exists Y = (\{\alpha\}, \emptyset, \alpha) \in \mathcal{A}$, $Y \notin E$. Since E is stable, $\exists X \in E$, $X \Rightarrow_{\{Y\}} Y$, hence $X \not\prec_{EI} Y$. Consider two cases:

• Suppose X concludes \perp . It cannot be that $\text{supp}(X) = \emptyset$, since this would imply $\text{prem}(X) \vdash \perp$, contradicting the *r*-consistency of Σ . Hence $X = (\Delta, \{\alpha\}, \perp)$.

• Suppose X concludes $\bar{\alpha}$, $\text{prem}(X) = \Delta$, $\text{supp}(X) = \emptyset$ or $\{\alpha\}$. By P3 and P2 (Def.7), $\exists X' = (\Delta, \{\alpha\}, \perp)$. Since X and X' have the same premises, and E is complete, then (by [11, Lemma 14]) $X' \in E$. Since $X \not\prec_{EI} Y$ then $\forall \beta \in \Delta$, $\beta \not\prec \alpha$, and so $X' \not\prec_{EI} Y$ and $X' \Rightarrow_{\{Y\}} Y$.

Given ii), it must be that $\exists \beta \in \Delta$, s.t. $\beta \in E_j$, $j > i$. But then $X \prec_{EI} Y$, respectively $X' \prec_{EI} Y$, contradicting $X \not\prec_{EI} Y$, respectively $X' \not\prec_{EI} Y$.

QED

As in Section 2, this result establishes a correspondence between the *PS* and argumentation consequence relations, where the latter are conclusions of *unconditional* arguments in stable extensions.

4 Properties and Postulates

4.1 Dung's Fundamental Lemma and Monotonicity of the Characteristic Function

We now study two key properties of *AF*'s [13] as they apply to pDCFs. Firstly, the Fundamental Lemma (*FL*) states that:

if X, X' are acceptable w.r.t. an admissible E , then $E \cup \{X\}$ is admissible and X' is acceptable w.r.t. $E \cup \{X\}$.

Secondly, an *AF*'s characteristic function \mathcal{F} is defined as:

$\mathcal{F}(S) = \{X \mid X \text{ is acceptable w.r.t. } S\}$ where $S \subseteq \mathcal{A}$

Hence, the fixed points of \mathcal{F} are an *AF*'s complete extensions. Then \mathcal{F} is shown to be monotonic: $E \subseteq E'$ implies $\mathcal{F}(E) \subseteq \mathcal{F}(E')$.

For pDCFs, the *FL* and monotonicity of a pDCF's characteristic function cannot straightforwardly be shown, since proofs of these properties rely on the fact that attacks and defeats on any argument X is fixed and independent of the premises in a given set E . However, we can show similar properties for 'epistemically closed' sets E that enjoy the following property:

if $W = (\Pi, \Sigma, \beta) \in E$, then for any $\Sigma' \subseteq \Sigma$ such that $\Sigma' \subseteq \text{prem}(E)$, E also includes $W' = (\Pi \cup \Sigma', \Sigma \setminus \Sigma', \beta)$.

Epistemically closed sets are so named, as commitment to premises Σ' in E implies commitment to the logically equivalent W' .

Definition 9 Let $Cl_{ec}(E) = E \cup \{W' \mid W \in E, W' \in [W], \text{prem}(W) \subseteq \text{prem}(W'), \text{prem}(W') \subseteq \text{prem}(E)\}$. Then E is *epistemically closed* (*ec*) if $E = Cl_{ec}(E)$.

Proofs of the following two propositions are shown in [11, Lemma 19] and [11, Lemma 23] respectively .

Proposition 5 Let X, X' be acceptable w.r.t. an admissible extension E of a $pDCF$ $(\mathcal{A}, \mathcal{C}, \prec)$. Then:

1. $Cl_{ec}(E \cup \{X\})$ is admissible.
2. X' is acceptable w.r.t. $Cl_{ec}(E \cup \{X\})$.

Proposition 6 Let E, E' be two *ec* admissible extensions of $(\mathcal{A}, \mathcal{C}, \prec)$ such that $E \subseteq E'$. Then $\mathcal{F}(E) \subseteq \mathcal{F}(E')$.

We sketch a key step in the proof of Proposition 5 that illustrates the importance of assuming epistemically closed sets.

Suppose X acceptable w.r.t. an admissible E where $Y \in E$. Inclusion of X in E may mean $Z \Rightarrow_{E \cup \{X\}} Y$, but $Z \not\Rightarrow_E Y$, since:

$$Z = (\Delta, \Gamma, \phi), \Phi \subseteq \Gamma \text{ and } \Phi \subseteq \text{prem}(X), \Phi \not\subseteq \text{prem}(E).$$

Since $Z \not\Rightarrow_E Y$, we cannot assume that the admissibility of E implies some Q in E (and hence $E \cup \{X\}$) defeating Z . Hence, we cannot immediately assume that Y is acceptable w.r.t. $E \cup \{X\}$, and so $E \cup \{X\}$ is admissible.

However we can show that there is an argument in $Cl_{ec}(E \cup \{X\})$ that defeats Z . Consider the following line of reasoning:

- $Z' \Rightarrow_E Y$ where $Z' = (\Delta \cup \Phi, \Gamma \setminus \Phi, \phi)$
- Hence $\exists W = (\Pi, \Sigma, \beta) \in E, W \Rightarrow_{\{Z'\}} Z'$
- Note: $\Sigma \subseteq \Delta \cup \Phi$ and $\Pi \cup \Phi \subseteq \text{prem}(E \cup \{X\})$ (1)

Consider two cases:

a) Suppose W defeats Z' on $\alpha \in \Phi$.

We have the logically equivalent $W' = (\Sigma \cap \Delta, \Pi \cup (\Sigma \cap \Phi), \beta)$.

Given (1), $W' \Rightarrow_{E \cup \{X\}} X$. Since X is acceptable w.r.t. E , $\exists Q \in E$ s.t. $Q \Rightarrow_{\{W'\}} W'$. Since $\text{prem}(W') \subseteq \text{prem}(Z)$, then $Q \Rightarrow_{\{Z\}} Z$.

b) Suppose W defeats Z' on a premise in Δ . We have $W' = (\Pi \cup (\Sigma \cap \Phi), \Sigma \cap \Delta, \beta)$, $W' \Rightarrow_{\{Z\}} Z$. Given (1) and the assumption that $E \cup \{X\}$ is epistemically closed, then $W' \in E \cup \{X\}$.

Proposition 5 suffices to prove a key result implied by the *FL*:

Proposition 7 Every admissible extension of a $pDCF$ is a subset of a preferred extension.

See [11, Proposition 22] for proof of the above. Proposition 6, together with the fact that every fixed point of \mathcal{F} is epistemically closed, facilitates proof of a key result following from the monotonicity of \mathcal{F} (the proof of which is shown in [11, Proposition 25]):

Proposition 8 The characteristic function \mathcal{F} of a $pDCF$ has a unique least fixed point (the grounded extension).

4.2 Rationality Postulates

We now show that the rationality postulates in [5] and [6] are satisfied, under some intuitive assumptions on preference relations.

Recall that in Example 1, no admissible extension contains the conflicting A_1 and F_1 since the unassailable I_2 must defeat either

A_1 , or F_1 on B_1 , or F_1 on C_1 . To suppose otherwise implies $I_2 \prec A_1, I_2 \prec B_1$, and $I_2 \prec C_1$. But such a preference relation would be incoherent as one would be rejecting the dialectical demonstration that A_1 and F_1 make use of mutually inconsistent premises, and effectively prefers arguments built from inconsistent premises. Indeed, in general, a strict preference $Y \prec X$, where $Y = (\Delta, \Gamma, \phi)$ attacks $X = (\{\alpha\}, \emptyset, \alpha)$, can be interpreted as:

from amongst the inconsistent $\Delta \cup \Gamma \cup \{\alpha\}$, one preferentially accepts arguments constructed from α and rejects arguments constructed from $\Delta \cup \Gamma$.

Hence $I_1 \prec A_1, I_1 \prec B_1$, and $I_1 \prec C_1$ collectively indicate preferentially accepting arguments built from the inconsistent $\{a, b, \neg a \vee \neg b\}$ and rejecting arguments built from $\{a, b, \neg a \vee \neg b\}$. Contradiction.

Given the above interpretation of $Y \prec X$ it should follow that $Y' \prec X$, where $Y' = (\Delta, \Gamma \cup \{\alpha\}, \phi)$ ($\phi = \bar{\alpha}$ or $\phi = \perp$). Hence, if $F_1 \prec A_1$ then $(\{b, \neg a \vee \neg b\}, \{a\}, \neg a) \not\prec A_1$ would be incoherent. Similarly, if $(\{b, \neg a \vee \neg b\}, \{a\}, \neg a) \prec A_1$ then $F_1 \not\prec A_1$ would be incoherent. We therefore assume that preference relations satisfy the following properties:

Definition 10 Let $(\mathcal{A}, \mathcal{C}, \prec)$ be a $pDCF$. Then \prec is dialectically coherent iff:

- $\forall (\emptyset, \Delta, \perp) \in \mathcal{A}: \exists \alpha \in \Delta$ such that $(\emptyset, \Delta, \perp) \not\prec (\{\alpha\}, \emptyset, \alpha)$. (Pref1)
- $\forall X = (\{\alpha\}, \emptyset, \alpha), Y = (\Delta, \Gamma, \phi), Y' = (\Delta, \Gamma \cup \{\alpha\}, \phi)$ ($\phi = \bar{\alpha}$ or $\phi = \perp$): $Y \prec X$ iff $Y' \prec X$. (Pref2)

We prove [5]'s closure and consistency rationality postulates for $pDCF$ s, under the assumption that \prec is dialectically coherent (note, one can straightforwardly show that the *Elitist* \prec_{E1} is dialectically coherent). [5] state these postulates with reference to complete extensions. Also, recall (Section 3 and footnote 8), that the argumentation based consequences are the conclusions of *unconditional* arguments. Hence we define:

$$\text{conc}(E) = \{\phi \mid (\Delta, \emptyset, \phi) \in E\}.$$

[5]'s postulates are stated with respect to a general framework for argumentation logics that integrate deductive and defeasible reasoning, so that arguments are trees whose links denote application of strict and defeasible inference rules, and sub-arguments correspond to sub-trees. [18] extend the framework to accommodate *CIAR*, in which arguments are constructed from premises (the leaf nodes) that entail a conclusion (root node), via application of a single strict inference rule encoding the classical entailment. Hence, for *CIAR*, an argument X is a tree of depth 1, whose leaves are the 'elementary' arguments associated with the premises of X , and are X 's sub-arguments. Therefore, we have the following formulation of the *sub-argument postulate* which [5] state as: if X is in a complete extension E , then all sub-arguments of X are in E :

Theorem 9 [Sub-argument Closure] Let E be a complete extension of a $pDCF$ $(\mathcal{A}, \mathcal{C}, \prec)$, and $X \in E$. Then for all $\alpha \in \text{prem}(X)$: $(\{\alpha\}, \emptyset, \alpha) \in E$.

PROOF. By *PI* (Definition 7), $X' = (\{\alpha\}, \emptyset, \alpha) \in \mathcal{A}$. If $Y \Rightarrow_{E \cup \{X'\}} X'$, then $Y \Rightarrow_{E \cup \{X\}} X$ (on X'). Since E is complete, X is acceptable w.r.t. E and so $\exists Z \in E$ s.t. $Z \Rightarrow_{\{Y\}} Y$. Hence X' is acceptable w.r.t. E , and since E is complete, $X' \in E$. QED

We now prove that *direct consistency* holds more generally for *admissible*, and not just complete, extensions.

Theorem 10 [Direct Consistency] *Let E be an admissible extension of a $pDCF$ $(\mathcal{A}, \mathcal{C}, \prec)$. Then $\forall \alpha, \beta \in \text{conc}(E)$, $\alpha \neq \perp$ and $\alpha \neq \beta$.*

PROOF. Suppose for contradiction that E contains $X = (\Delta, \emptyset, \alpha)$ and $Y = (\Gamma, \emptyset, \beta)$, and 1) $\alpha = \perp$, or 2) $\alpha = \beta$. Suppose 1) is the case. By P2, $Z = (\emptyset, \Delta, \perp) \in \mathcal{A}$. Suppose 2) is the case. By P3, $\exists Z' = (\Delta \cup \Gamma, \emptyset, \perp) \in \mathcal{A}$. By P2, $Z = (\emptyset, \Delta \cup \Gamma, \perp) \in \mathcal{A}$. In either case, $\text{supp}(Z) \subseteq \text{prem}(E)$. Hence $\forall \beta \in \text{supp}(Z)$, $\exists W \in E$ s.t. $Z \rightarrow_E W$ on $(\{\beta\}, \emptyset, \beta)$. By Pref1 (Definition 10), at least one such attack succeeds as a defeat. Since the unassailable Z cannot itself be defeated by an argument in E , then this contradicts $W \in E$ is acceptable w.r.t. E . QED

[5]'s closure under strict rules postulate states that if $\text{conc}(E) \vdash \alpha$, then there is an argument X in E that concludes α . This postulate is stated for $pDCF$ s, under the assumption that resources suffice to construct such an X . We refrain from mentioning [5]'s *indirect consistency* postulate as this immediately follows from direct consistency and closure under strict rules. Note however that (together with P2 and P3) the proofs of direct consistency and closure indicate that if resources suffice to recognise inconsistency in a set of premises, either through use of these premises in constructing an argument concluding \perp , or arguments with conflicting conclusions, then this suffices to ensure satisfaction of the rationality postulates.

Theorem 11 [Closure under Strict Rules] *Let E be a complete extension of a $pDCF$ $(\mathcal{A}, \mathcal{C}, \prec)$, $E' \subseteq E$, and $\text{conc}(E') \vdash \alpha$. Suppose there exists a $X = (\Delta, \emptyset, \phi) \in \mathcal{A}$ such that $\Delta = \text{prem}(E')$. Then $X \in E$.*

PROOF. Suppose $Y \Rightarrow_{E \cup \{X\}} X$ on some $X' = (\{\alpha\}, \emptyset, \alpha)$. $\text{prem}(X) \subseteq \text{prem}(E)$ implies $\text{supp}(Y) \subseteq \text{prem}(E)$. Hence since $\alpha \in \text{prem}(E)$, $\exists X'' \in E$ s.t. $Y \Rightarrow_{E \cup \{X''\}} X''$ on X' . Since E is complete $\exists Z \in E$ s.t. $Z \Rightarrow_{\{Y\}} Y$. Hence X is acceptable w.r.t. E . Since E is complete, $X \in E$. QED

Finally, the contamination postulates – *non-interference* and *crash resistance* [6] – essentially state that the conclusions of arguments in complete extensions of an AF defined by \mathcal{B}_1 are preserved when unioning some \mathcal{B}_2 with \mathcal{B}_1 , such that the propositional atoms in \mathcal{B}_2 are disjoint from those in \mathcal{B}_1 . However, [6] does not account for the use of preferences. For $pDCF$ s, we also need to refer to the preferences over arguments defined by the union of \mathcal{B}_1 and \mathcal{B}_2 , such that the preference relations over arguments defined for each of \mathcal{B}_1 and \mathcal{B}_2 are preserved. In what follows, we define a composition operator for $pDCF$ s, and assume the same resources are available for constructing arguments from \mathcal{B}_1 , \mathcal{B}_2 and $\mathcal{B} = \mathcal{B}_1 \cup \mathcal{B}_2$, so that for \mathcal{A} defined by \mathcal{B} , $(\mathcal{A}_1 \cup \mathcal{A}_2) \subseteq \mathcal{A}$. Also, if resources suffice to construct a ‘tautological’ argument $X = (\emptyset, \emptyset, \alpha)$ from \mathcal{B}_1 (\mathcal{B}_2), then X can also be constructed from \mathcal{B}_2 (\mathcal{B}_1) and \mathcal{B} .

Definition 11 Let $(\mathcal{A}_1, \mathcal{C}_1, \prec_1)$ be defined by \mathcal{B}_1 , $(\mathcal{A}_2, \mathcal{C}_2, \prec_2)$ defined by \mathcal{B}_2 . Then $(\mathcal{A}, \mathcal{C}, \prec) = (\mathcal{A}_1, \mathcal{C}_1, \prec_1) \oplus (\mathcal{A}_2, \mathcal{C}_2, \prec_2)$, iff:

1. $\mathcal{A}_1 \cup \mathcal{A}_2 \subseteq \mathcal{A}$ (it is obvious to see that $(\mathcal{C}_1 \cup \mathcal{C}_2) \subseteq \mathcal{C}$).
2. $\forall X = (\emptyset, \emptyset, \alpha) : X \in \mathcal{A}_1$ iff $X \in \mathcal{A}_2$ iff $X \in \mathcal{A}$.
3. \prec is any preference ordering such that:

- $\forall X_1, Y_1 \in \mathcal{A}_1 : (X_1, Y_1) \in \prec_1$ iff $(X_1, Y_1) \in \prec$

- $\forall X_2, Y_2 \in \mathcal{A}_2 : (X_2, Y_2) \in \prec_2$ iff $(X_2, Y_2) \in \prec$

In Section 3.2 we informally described how the contaminating effect of inconsistent arguments is avoided in our approach. However, contamination may also arise as a result of dropping the subset minimality check on arguments.

Example 12 Let $\mathcal{B}_1 = \{p, \neg p\}$ and $\mathcal{B}_2 = \{s\}$, and :

- $\mathcal{A}_1 =$
 $\{X_1 = (\{p\}, \emptyset, p), X'_1 = (\emptyset, \{p\}, p), Y_1 = (\{p\}, \{\neg p\}, \perp),$
 $X_2 = (\{\neg p\}, \emptyset, \neg p), X'_2 = (\emptyset, \{\neg p\}, \neg p), Y_2 = (\{\neg p\}, \{p\}, \perp),$
 $Z = (\{\neg p, p\}, \emptyset, \perp), U = (\emptyset, \{\neg p, p\}, \perp)\}.$

Suppose also that $X_2 \prec_1 X_1$. Then $X_2 \not\Rightarrow_{E_1} X_1$, and $Y_2 \not\Rightarrow_{E_1} X_1$ (since by Pref2 $Y_2 \prec_1 X_1$, and $Z \not\Rightarrow_{E_1} X_1$ ($Z \prec_1 X_1$ since \prec_1 is *inle*). $E_1 = \{X_1, X'_1, X'_2, Y_1, U\}$ is the single complete (grounded and preferred) extension.

$E_2 = \{X_2, X'_2, Y_2, U\}$ is *not* admissible, since X_2 and Y_2 are both defeated by X_1 and Y_1 , and neither defeats can be defended.

- $\mathcal{A}_2 = \{S = (\{s\}, \emptyset, s), S' = (\emptyset, \{s\}, s)\}$, and $\prec_2 = \emptyset$.
- $(\mathcal{A}, \mathcal{C}, \prec) = (\mathcal{A}_1, \mathcal{C}_1, \prec_1) \oplus (\mathcal{A}_2, \mathcal{C}_2, \prec_2)$, where $\mathcal{A} = \mathcal{A}_1 \cup \mathcal{A}_2 \cup \{C = (\{\neg p, s\}, \emptyset, \neg p), Z' = (\{\neg p, s, p\}, \emptyset, \perp)\}$ and their logically equivalent arguments, and $\prec = \prec_1^{11}$. Now, we obtain two preferred extensions:

E'_1 that is a superset of E_1 and contains S and S' ;

E'_2 that is a superset of E_2 and contains S, S' and C .

We obtain the additional E'_2 because X_2 and Y_2 are now defended by C , since $C \not\prec X_1$ and so $C \Rightarrow_{\{X_1\}} X_1$. Furthermore, the grounded extension E now contains S (recall that $U \in E$ will defend against Z 's (and Z' 's) defeat on S) but not P . Hence, contamination has taken place since adding the syntactically disjoint s has changed the credulously and sceptically defined consequences of $(\mathcal{A}_1, \mathcal{C}_1, \prec_1)$.

The problem here is that by adding premise s to X_2 to obtain C , X_2 has been strengthened, since (given $\prec = \prec_1$) $X_2 \prec X_1$ but $C \not\prec X_1$. However, the strengthening of X_2 is clearly counter-intuitive, since s is an irrelevant premise in C . Thus we would expect that $C \prec X_1$. Given this latter preference, we then obtain that E'_1 is the single complete (grounded and preferred) extension of $(\mathcal{A}, \mathcal{C}, \prec)$.

Hence, preference relations must satisfy the following property in order to prevent contamination

Definition 12 Let $(\mathcal{A}, \mathcal{C}, \prec)$ be a $pDCF$. Then \prec is *relevance coherent* iff $\forall X, Y, Y'$ such that

$Y = (\Gamma, \emptyset, \alpha), Y' = (\Delta \cup \Gamma, \emptyset, \alpha)$, and $\Delta \parallel \Gamma \cup \{\alpha\}$ (Δ syntactically disjoint from $\Gamma \cup \{\alpha\}$): if $Y \prec X$ then $Y' \prec X$.

Of course, relevance coherence is trivially satisfied by proof theories that preclude construction of arguments with syntactically disjoint premises [10]. Note also that one can straightforwardly show that the *Elitist* \prec_{El} is relevance coherent.

The following results assume $pDCF$ s whose preference relations are dialectically and relevance coherent. In [6, 22], the *crash resistance* and *non-interference* postulates are formulated w.r.t. the ‘consequences’ of an AF . We analogously define the consequences of $pDCF$ s, and state satisfaction of the postulates for the complete (and hence grounded, preferred and stable) semantics (recall that $\text{conc}(E)$ denotes the conclusions of unconditional arguments).

¹¹ Note that all three $pDCF$ s satisfy P1 – P4 in Definition 7.

Definition 13 Let $(\mathcal{A}, \mathcal{C}, \prec)$ be a $pDCF$. Then $Cn((\mathcal{A}, \mathcal{C}, \preceq)) = \{\text{conc}(E_1), \dots, \text{conc}(E_n)\}$ where E_1, \dots, E_n are the complete extensions of $(\mathcal{A}, \mathcal{C}, \prec)$.

Non-interference states that the consequences of a $pDCF$ defined by a base \mathcal{B}_1 , restricted to the atoms in \mathcal{B}_1 , remain unchanged in the $pDCF$ defined by the union of \mathcal{B}_1 and a syntactically disjoint \mathcal{B}_2 .

Theorem 13 [Non Interference] Let $\mathcal{B}_1 \parallel \mathcal{B}_2$, $(\mathcal{A}, \mathcal{C}, \prec) = (\mathcal{A}_1, \mathcal{C}_1, \preceq_1) \oplus (\mathcal{A}_2, \mathcal{C}_2, \prec_2)$. Then:

$$Cn((\mathcal{A}_1, \mathcal{C}_1, \preceq_1)|_{\text{atoms}(\mathcal{B}_1)}) = Cn((\mathcal{A}, \mathcal{C}, \preceq)|_{\text{atoms}(\mathcal{B}_1)})^{12}.$$

PROOF. See [11, Theorem 37].

QED

Referring to Example 12, $Cn((\mathcal{A}_1, \mathcal{C}_1, \prec_1)|_{\text{atoms}(\mathcal{B}_1)}) = \{\{p\}\}$. If \prec is not relevance coherent then $Cn((\mathcal{A}, \mathcal{C}, \prec)|_{\text{atoms}(\mathcal{B}_1)}) = \{\{p\}, \{\neg p\}\}$. However, assuming relevance coherence, then $C \prec X_1, C \not\#_{\{X_1\}} X_1, E'_2$ is not a complete extension of $(\mathcal{A}, \mathcal{C}, \preceq)$, and so $Cn((\mathcal{A}, \mathcal{C}, \prec)|_{\text{atoms}(\mathcal{B}_1)}) = \{\{p\}\}$.

Definition 14 A base \mathcal{B}_1 is said to be contaminating iff there exists a $(\mathcal{A}_1, \mathcal{C}_1, \prec_1)$ defined by \mathcal{B}_1 , such that for any \mathcal{B}_2 and $(\mathcal{A}_2, \mathcal{C}_2, \prec_2)$ defined by \mathcal{B}_2 , where $\mathcal{B}_1 \parallel \mathcal{B}_2$: $Cn((\mathcal{A}_1, \mathcal{C}_1, \prec_1)) = Cn((\mathcal{A}, \mathcal{C}, \prec))$, where $(\mathcal{A}, \mathcal{C}, \prec) = (\mathcal{A}_1, \mathcal{C}_1, \prec_1) \oplus (\mathcal{A}_2, \mathcal{C}_2, \prec_2)$.

Theorem 14 [Crash Resistance] There does not exist a contaminating base \mathcal{B} .

PROOF. See [11, Theorem 39].

QED

Referring to Example 12, $Cn((\mathcal{A}_1, \mathcal{C}_1, \prec_1)) = \{\{p\}\} \neq Cn((\mathcal{A}, \mathcal{C}, \prec)) = \{\{p, s\}\}$.

5 Conclusions

This paper has argued that features of propositional classical logic instantiations of AF s ($CIAR$) that suffice to ensure satisfaction of rationality postulates, preclude uses of argument characteristic of real-world dialectical reasoning by resource bounded agents. Our solution has been to provide an account of $CIAR$ in which the ontology of classical logic arguments explicitly distinguishes between an argument's premises assumed true, and those supposed true for the sake of argument. In so doing, we obviate the need for checking consistency and subset minimality of premises, and identify an intuitive set of assumptions on the available resources for constructing arguments for inclusion in a framework, and show that the resulting formalism satisfies the closure, consistency, non-interference and crash resistance postulates. We thus provide a rational account of $CIAR$ that is suitable for use by resource bounded agents. Our account also avoids the foreign commitment problem, and formalises the real-world use of argument in dialectically demonstrating that an agent's premises are inconsistent. We have shown that key properties of Dung's theory are preserved, and we provide an argumentative characterisation of the Preferred Subtheories non-monotonic logic, under the assumption that agents have limited inferential capabilities.

[14] also identify requirements for practical applications of argumentation. They stipulate that the computational cost of validating the legitimacy of a constructed argument should be at most polynomial (in the size of the arguments), and whether an argument attacks another should be at most linear (in the size of the argument's conclusion). Both are satisfied by our approach (the former trivially since we drop checks on premises). [14] also argue that an argument's premises should be relevant to its conclusion. Although we

drop the subset minimality check, we suggest that the issue of relevance should be addressed by the specific proof theoretic means for constructing arguments.

Pragmatic considerations also motivate dropping consistency and subset minimality checks on arguments in [3]. Arguments are Gentzen style sequents and arguments with inconsistent premises are attacked by sequents with empty antecedents. In this work, the distinction between premises and suppositions, and the use of preferences are not considered. The postulates in [5] are not studied and neither is there consideration of argumentation under resource bounds. Finally, [7] also distinguish between premises and suppositions, but in a restricted logical setting (arguments are constructed from literals and defeasible rules). This work studies only the grounded semantics, does not consider preferences or investigate satisfaction of the rationality postulates.

A number of works show satisfaction of the non-interference and crash resistance postulates for argumentation formalisms that integrate deductive and defeasible reasoning. [6] show that logic programming and Default Logic instantiations of Dung frameworks satisfy these postulates under the *semi-stable* semantics. In [22], arguments are built from a set of *classically consistent* propositional formulae, and defeasible and strict inference rules. [22] do not consider the use of preferences, and show satisfaction of the postulates under the assumption that inconsistent arguments (identified as those whose contained premises together with the conclusions of defeasible rules are classically inconsistent) are excluded from the argumentation framework. Finally, [12] define a version of the $ASPIC^+$ framework [18] in which the strict inference rules encode inference in [20]'s paraconsistent logic. The focus of [12] is on showing satisfaction of the closure and consistency postulates, and satisfaction of non-interference and crash resistance is not formally shown (the authors state that satisfaction of these postulates can be taken for granted given the absence of the *Ex Falso* principle). Finally, we have identified that contamination may result if one does not implement the subset minimality check on an argument's premises. While we drop the subset minimality check, we identify a notion of relevance that is more readily addressed by classical proof theories. For proof theories that do not exclude construction of arguments making use of irrelevant premises, we show that contamination is avoided if preference relations do not strengthen arguments upon addition of irrelevant premises (one such preference relation being the widely used Elitist preference). This result is closely related to a result shown in [18], which states that the argumentation defined consequences of a framework remain unchanged if one additionally includes non-subset minimal arguments, provided that they are not stronger than their subset minimal counterparts.

With regard to future research directions, we recognise that while we accommodate agents whose resources are bounded with respect to the *construction* of arguments, we need to investigate the complexity of computing semantics (i.e., *evaluation* of arguments) given our dialectical definition of attacks (defeats) on arguments. Finally, we are currently extending our dialectical formulation of arguments and acceptability to the $ASPIC^+$ framework [18]. $ASPIC^+$ is a general framework for structured argumentation that accommodates arguments built from strict inference rules that encode the inference relations of deductive logics, as well as defeasible inference rules. We thus aim to provide a general account of structured argumentation for use by real-world resource bounded agents.

Acknowledgements: We thank the reviewers whose comments helped improve this paper.

¹² Recall Notation 3

REFERENCES

- [1] L. Amgoud and C. Cayrol, 'A reasoning model based on the production of acceptable arguments', *Annals of Mathematics and Artificial Intelligence*, **34**(1-3), 197–215, (2002).
- [2] L. Amgoud and S. Vesic, 'Handling inconsistency with preference-based argumentation', in *Scalable Uncertainty Management: 4th International Conference, SUM 2010*, pp. 56–69. Springer, (2010).
- [3] O. Arieli and C. Straßer, 'Sequent-based logical argumentation', *Argument and Computation*, **6**(1), 73–99, (2015).
- [4] G. Brewka, 'Preferred subtheories: An extended logical framework for default reasoning', in *International Joint Conference on Artificial Intelligence*, pp. 1043–1048, (1989).
- [5] M. Caminada and L. Amgoud, 'On the evaluation of argumentation formalisms', *Artificial Intelligence*, **171**(5-6), 286–310, (2007).
- [6] M. Caminada, W. Carnielli, and P. Dunne, 'Semi-stable semantics', *Logic and Computation*, **22**(5), 1207–1254, (2012).
- [7] Martin Caminada, 'Dialogues and HY-arguments', in *Non-Monotonic Reasoning*, pp. 94–99, (2004).
- [8] V. Chvátal and E. Szemerédi, 'Many hard examples for resolution', *Journal of the ACM*.
- [9] M. D'Agostino, 'An informational view of classical logic', *Theoretical Computer Science*, **606**, 79–97, (2015).
- [10] M. D'Agostino, D. Gabbay, and S. Modgil, 'Normal proofs and non-contamination in classical natural deduction', *Technical Report*, https://dl.dropboxusercontent.com/u/5626429/CND_TR.pdf, (2016).
- [11] M. D'Agostino and S. Modgil, 'Classical logic, argumentation and dialectic: Technical report', *Technical Report*, www.dcs.kcl.ac.uk/staff/smodgil/ECAITechnicalReport.pdf, (2016).
- [12] D. Grooters and H. Prakken, 'Combining paraconsistent logic with argumentation', in *Computational Models of Argument. Proceedings of COMMA 2014*, pp. 301–312. IOS Press, (2014).
- [13] P. M. Dung, 'On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n -person games', *Artificial Intelligence*, **77**, 321–357, (1995).
- [14] P.M. Dung, F. Toni, and P. Mancarella, 'Some design guidelines for practical argumentation systems', in *Proc. Conference on Computational Models of Argument: COMMA 2010*, pp. 183–194, (2010).
- [15] N. Gorogiannis and A. Hunter, 'Instantiating abstract argumentation with classical logic arguments: Postulates and properties', *Artificial Intelligence*, **175**(910), 1479 – 1497, (2011).
- [16] M. Caminada, S. Modgil, and N. Oren, 'Preferences and unrestricted rebut', in *Computational Models of Argument: Proceedings of COMMA 2014*, pp. 209–220. IOS Press, (2014).
- [17] P. McBurney and S. Parsons, 'Chapter 13: Dialogue games for agent argumentation', in *Argumentation in AI*, 261–280, Springer, (2009).
- [18] S. Modgil and H. Prakken, 'A general account of argumentation and preferences', *Artificial Intelligence*, **195**(0), 361 – 397, (2013).
- [19] S. Modgil, F. Toni, F. Bex, I. Bratko, C. Chesñevar, W. Dvořák, M.A. Falappa, X. Fan, S. Gaggl, A.J. García, M.P. González, T. Gordon, J. Leite, M. Možina, C. Reed, G. Simari, S. Szeider, P. Torroni, and S. Woltran, 'Chapter 21: The added value of argumentation', in *Agreement Technologies*, ed., S. Ossowski, 357–403, Springer, (2013).
- [20] N. Rescher and R. Manor, 'On inference from inconsistent premises', *Journal of Theory and Decision*, **1**, 179–219, (1970).
- [21] G. Vlastos, 'The socratic elenchus', *The Journal of Philosophy*, **79**(11), 711–714, (1982).
- [22] Y. Wu and M. Podlaskowski, 'Implementing crash-resistance and non-interference in logic-based argumentation', *Journal of Logic and Computation*, **25**, 303–333, (2015).
- [23] A.P. Young, S. Modgil, and O. Rodrigues, 'Prioritised default logic as rational argumentations', in *To appear in: Proc. International Joint Conference on Autonomous Agents and Multi-Agents Systems (AAMAS'2016)*, (2016).

Two Dimensional Uncertainty in Persuadee Modelling in Argumentation

Anthony Hunter¹

Abstract. When attempting to persuade an agent to believe (or disbelieve) an argument, it can be advantageous for the persuader to have a model of the persuadee. Models have been proposed for taking account of what arguments the persuadee believes and these can be used in a strategy for persuasion. However, there can be uncertainty as to the accuracy of such models. To address this issue, this paper introduces a two-dimensional model that accounts for the uncertainty of belief by a persuadee and for the confidence in that uncertainty evaluation. This gives a better modeling for using lotteries so that the outcomes involve statements about what the user believes/disbelieves, and the confidence value is the degree to which the user does indeed hold those outcomes (and this is a more refined and more natural modeling than found in [19]). This framework is also extended with a modelling of the risk of disengagement by the persuadee.

1 INTRODUCTION

Computational models of argument can potentially be used for systems to persuade users to change their behaviour (e.g. to eat less, to exercise more, to use less electricity, to vote, etc) [17]. However, most proposals for dialogical argumentation focus on protocols (e.g. [26, 27, 12, 7]) with strategies being under-developed. See [35] for a review of strategies in multi-agent argumentation.

There are some proposals for using probability theory in dialogical argumentation: A probabilistic model of the opponent is used for selection of moves by an agent based on what it believes the other agent is aware of [31]; The history of previous dialogues is used to predict the arguments that an opponent might put forward [13]; A probabilistic finite state machine can represent the possible moves that each agent can make in each state [18], and generalized to POMDPs when there is uncertainty about what an opponent is aware of [14]. However, none of these use the beliefs of the persuadee or use asymmetric dialogues where only the persuader presents arguments (a requirement when the persuader is a software agent and it is not possible for it to understand natural language arguments from the persuadee). In [4], a probabilistic model of beliefs of the persuadee is used by the persuader to choose beliefs to present, but there is no consideration of update of the model resulting from dialogue, of confidence in the model, of persuasion outcomes involving statements about belief, of expected utility, or of risk of disengagement (which are issues we consider here).

There is a recent proposal for asymmetric persuasion dialogues with a general definition for probabilistic user models, and a general definition for updating user models in terms of mass redistributions

[19]. In this paper, that proposal is generalized by introducing a two-dimensional notion of uncertainty with multiple user models and a measure of confidence in them. We will use a logical language to represent and reason with the beliefs in the user models and the confidence in them. This enables a more accurate modelling of expected utility than in [19] since belief statements in the logical language are outcomes in the utility analysis. This is extended with a modelling of risk of disengagement by the persuadee (a key problem when a dialogue is too long) and use this for selecting optimal dialogues.

2 PRELIMINARIES

This paper is based on abstract argumentation [10]. The dialogues concern an argument graph G without self-attacks where $\text{Args}(G)$ is the set of arguments in G , and $\text{Attacks}(G)$ is the set of attack relations in G .

A **system** (the *persuader* running as an app) has a dialogue with a **user** (the *persuadee* using the app) to persuade him/her to believe (or disbelieve) some combination of arguments (e.g. about doing more exercise) as explained in Section 4. The system is aware of all the arguments in the argument graph G whereas the user is not necessarily aware of all the arguments in G .

A **dialogue** is a sequence of moves $D = [m_1, \dots, m_k]$. Equivalently, we can use D as a function with an index position i to return the move at that index (i.e. $D(i) = m_i$). A **protocol** specifies what moves should/can follow each move in a dialogue.

In this paper, we consider one protocol as an illustration. The only moves are posit of an argument A by the system, denoted $A!$, or termination by the system, denoted \oplus , or by the user, denoted \otimes . Once terminated, no further moves are possible. An example of untermi-nated dialogue is $[A!, C!, D!, A!, C!, D!, A!]$, of a system-terminated dialogue is $[A!, C!, \oplus]$, and of a user-terminated dialogue is $[A!, C!, \otimes]$.

3 PROBABILISTIC USER MODELS

We will use the epistemic approach to probabilistic argumentation [34, 16, 21, 2].

Definition 1. A **mass distribution** P over $\text{Args}(G)$ is such that $\sum_{X \subseteq \text{Args}(G)} P(X) = 1$. Let $\text{Dist}(G)$ be the set of mass distributions over G . The **probability of an argument** A is $P(A) = \sum_{X \subseteq \text{Args}(G) \text{ s.t. } A \in X} P(X)$.

For a mass distribution P , and $A \in \text{Args}(G)$, $P(A)$ is the belief that an agent has in A (i.e. the degree to which the agent believes the premises and the conclusion drawn from those premises). When $P(A) > 0.5$, then the agent believes the argument to some degree,

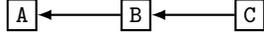
¹ Department of Computer Science, University College London, London, UK

whereas when $P(A) \leq 0.5$, then the agent disbelieves the argument to some degree.

The following constraint ensures that the mass distribution respects the structure of the graph, without forcing an unattacked argument to be believed [16].

Definition 2. A mass distribution P is **rational** for G iff $\forall (A, B) \in \text{Attacks}(G)$, if $P(A) > 0.5$, then $P(B) \leq 0.5$.

Example 1. Consider the following argument graph. Mass distribution $P_1(A) = 0.6$, $P_1(B) = 0.9$, and $P_1(C) = 0.9$ is not rational, whereas $P_2(A) = 0.6$, $P_2(B) = 0.3$, and $P_2(C) = 0.9$ is rational, and $P_3(A) = 0$, $P_3(B) = 1$, and $P_3(C) = 0.3$ is rational.



The system (the persuader) uses a mass distribution P as a model of the user (the persuadee), and it can update the model at each stage of the dialogue (see Section 7). This is useful for asymmetric dialogues where the user is not allowed to posit arguments. So the only way the user can treat arguments that s/he does not accept is by disbelieving them (and the system aims to reflect this in the user model). In contrast, in symmetric dialogues, the user may be allowed to posit counterarguments to an argument that s/he does not accept.

4 PERSUASION OBJECTIVES

An **objective** is a Boolean combination of arguments. If $A \in \text{Arg}(G)$, then A is a positive literal, and $\neg A$ is a negative literal. Let $\text{AFormulae}(G)$ denote all the formulae that can be formed from the arguments in G using \wedge , \vee , and \neg as connectives in the usual way.

Informally, an objective is positive or negative from the point of view of the persuader. If it is positive (respectively negative), then the persuader wants the objective to be satisfied (respectively not satisfied) by the arguments believed by the persuader. We consider how to specify whether an objective is positive or negative in Section 9.

In order to formalize the satisfaction of objectives, we treat each subset of $\text{Args}(G)$ as a model (i.e. a possible world).

Definition 3. The **satisfaction relation**, denoted \models , is defined as follows where $X \subseteq \text{Args}(G)$, $A \in \text{Args}(G)$, and $\alpha, \beta \in \text{AFormulae}(G)$: (1) $X \models A$ when $A \in X$; (2) $X \models \alpha \wedge \beta$ iff $X \models \alpha$ and $X \models \beta$; (3) $X \models \alpha \vee \beta$ iff $X \models \alpha$ or $X \models \beta$; and (4) $X \models \neg \alpha$ iff $X \not\models \alpha$.

Essentially \models is a classical satisfaction relation. So if α is a classical tautology, then $X \models \alpha$ for all $X \subseteq \text{Args}(G)$, and if α is a classical contradiction, then $X \not\models \alpha$ for all $X \subseteq \text{Args}(G)$. For $\alpha \in \text{AFormulae}(G)$, let $\text{Models}(\alpha) = \{X \subseteq \text{Args}(G) \mid X \models \alpha\}$. For each graph G , we assume an ordering over the arguments $\langle A_1, \dots, A_n \rangle$ so that we can encode each model by a binary number: For a model X , if the i th argument is in X , then the i th digit is 1, otherwise it is 0. E.g. for $\langle A, B, C \rangle$, the model $\{A, C\}$ is represented by 101.

According to the user model, the probability of an objective ϕ is the sum of the probability of each model satisfying the objective.

Definition 4. For $P \in \text{Dist}(G)$, the **probability of objective** $\phi \in \text{AFormulae}(G)$ is $P(\phi) = \sum_{X \in \text{Models}(\phi)} P(X)$.

Suppose $\alpha \in \text{AFormulae}(G)$ and P is a mass distribution. If α is a contradiction of classical logic, then $P(\alpha) = 0$, and if α is a tautology of classical logic, then $P(\alpha) = 1$. Also, if $\{\alpha\} \vdash \beta$, then $P(\alpha) \leq P(\beta)$, and if $\neg(\alpha \wedge \beta)$ is a classical tautology, then $P(\alpha \vee \beta) = P(\alpha) + P(\beta)$.

5 BELIEF STATEMENTS

We use statements (defined next) involving a mass distribution applied to an objective as atoms in a language. These represent the belief a persuadee has in an objective.

Definition 5. A **belief statement** is of the form $P(\alpha)\#x$ where $\alpha \in \text{AFormulae}(G)$ is an objective, $\# \in \{=, \geq, \leq, >, <\}$, and $x \in [0, 1]$. A **belief formula** is a Boolean combination of belief statements (i.e. if ϕ is a belief statement, then it is a belief formula, and if ϕ and ψ are belief formulae, then each of $\phi \wedge \psi$, $\phi \vee \psi$ and $\neg \phi$ is a belief formula). Let $\text{BFormulae}(G)$ be the set of belief formulae.

Example 2. For $A, B \in \text{Args}(G)$, $(P(A \wedge B) > 0.9) \vee (P(\neg A \wedge \neg B) < 0.5)$ is an example of a belief formula.

We assume equivalences, denoted \equiv , between belief formulae: (1) $P(\alpha) \geq x \equiv (P(\alpha) = x) \vee (P(\alpha) > x)$, (2) $P(\alpha) \leq x \equiv (P(\alpha) = x) \vee (P(\alpha) < x)$, (3) $P(\alpha) \neq x \equiv \neg(P(\alpha) = x)$, (4) $P(\alpha) \not> x \equiv \neg(P(\alpha) > x)$, and (5) $P(\alpha) \not< x \equiv \neg(P(\alpha) < x)$.

Definition 6. The **satisfying distributions** for a belief statement $P(\alpha)\#x$ is $\text{Sat}(P(\alpha)\#x) = \{P' \in \text{Dist}(G) \mid P'(\alpha)\#x\}$, where $\# \in \{=, \geq, \leq, >, <\}$. The set of satisfying distributions for a belief formula is as follows where ϕ and ψ are belief formulae: (1) $\text{Sat}(\phi \wedge \psi) = \text{Sat}(\phi) \cap \text{Sat}(\psi)$; (2) $\text{Sat}(\phi \vee \psi) = \text{Sat}(\phi) \cup \text{Sat}(\psi)$; and (3) $\text{Sat}(\neg \phi) = \text{Sat}(\top) \setminus \text{Sat}(\phi)$.

Example 3. For $\langle A, B \rangle$, if $P_1(11) = 1$ and $P_2(00) = 1$, then $P_1, P_2 \in \text{Sat}(P(A \wedge B) = 1) \vee P(\neg A \wedge \neg B) = 1)$. For $\langle C \rangle$, if $P_3(1) = 0.5$ and $P_4(1) = 0.6$, then $P_3 \notin \text{Sat}(P(C) > 0.5)$ and $P_4 \in \text{Sat}(P(C) > 0.5)$.

Proposition 1. (1) For $x \in (0, 1]$, $\text{Sat}(P(\perp) = x) = \emptyset$. (2) $\text{Sat}(P(\top) = 1) = \text{Dist}(G)$. (3) For any objective α , $\text{Sat}(P(\alpha) \leq 1) = \text{Dist}(G)$ and $\text{Sat}(P(\alpha) \geq 0) = \text{Dist}(G)$. (4) When $x \neq y$, $\text{Sat}(P(\alpha) = x \wedge P(\alpha) = y) = \emptyset$. (5) When $\vdash \alpha \leftrightarrow \beta$, $\text{Sat}(P(\alpha) = x) = \text{Sat}(P(\beta) = x)$.

Definition 7. $\phi, \psi \in \text{BFormulae}(G)$ are **disjoint** iff $\text{Sat}(\phi) \cap \text{Sat}(\psi) = \emptyset$.

Example 4. Each pair of statements is disjoint: (1) $P(A) = 0.5, P(A) = 0.7$; (2) $P(A) \geq 0.6, P(A) < 0.5$; (3) $P(A) > 0.5, P(\neg A) > 0.7$; and (4) $P(A) = 0.3, P(A \wedge B) = 0.7$.

Definition 8. $\{\phi_1, \dots, \phi_k\} \subseteq \text{BFormulae}(G)$ are **exhaustive** iff $\text{Sat}(\phi_1) \cup \dots \cup \text{Sat}(\phi_k) = \text{Dist}(G)$.

Example 5. The set of belief formulae $\{P(A) > 0.8, P(A) \leq 0.8 \wedge P(A) > 0.6, P(A) \leq 0.6\}$ is exhaustive.

Proposition 2. Let $S \subset \text{BFormulae}(G)$ and let $\phi, \phi' \in \text{BFormulae}(G)$. If $S \cup \{\phi\}$ is exhaustive and pairwise disjoint, and $\text{Sat}(\phi) = \text{Sat}(\phi')$, then $S \cup \{\phi'\}$ is exhaustive and pairwise disjoint.

As we see in Section 9, belief formulae can represent desired/undesired outcomes of a dialogue. The system may want to persuade the user to believe α to some degree (i.e. it is a positive objective). For instance, the system may want the user to believe α above a threshold of 0.9 (i.e. $P(\alpha) > 0.9$). Or it may want to persuade the user to disbelieve α and so α is a negative objective (e.g. $P(\alpha) \leq 0.5$). So the aim of the dialogue is to change the belief/disbelieve in an objective depending on whether it is a positive or negative objective.

6 CONFIDENCE IN BELIEF FORMULAE

The confidence distribution is a probability distribution over mass distributions. It gives the probability that a given user model is the correct representation of the user's beliefs.

Definition 9. A confidence distribution is $Pr : \text{Dist}(G) \rightarrow [0, 1]$ s.t. $\sum_{P \in \text{Dist}(G)} Pr(P) = 1$. For a belief formula ϕ , a **formula confidence** is $Pr(\phi) = \sum_{P \in \text{Sat}(\phi)} Pr(P)$.

For instance, the formula confidence $Pr(P(A) = 0.7) > 0.5$ means that the persuader has at least 0.5 confidence in the persuadee belief in A being 0.7.

Example 6. For $\langle A, B \rangle$, consider P_1 , P_2 , and P_3 , defined below. Some examples of confidence are: $Pr(P(A) = 1) = 1/2$, $Pr(P(A) \geq 1/2) = 1$, $Pr(P(A) = 1/2) = 1/2$, $Pr(P(B) = 0) = 1/4$, $Pr(P(\neg B) = 1) = 1/4$, $Pr(P(\neg B) = 1/2) = 1/4$, and $Pr(P(A \wedge B) \geq 1/4) = 3/4$.

	$Pr(P_1) = 1/2$	$Pr(P_2) = 1/4$	$Pr(P_3) = 1/4$
11	1	0	1/4
10	0	1/2	1/4
01	0	0	1/4
00	0	1/2	1/4

Clearly, for all Pr , $Pr(P(\perp) = 0) = 1$, $Pr(P(\perp) = 1) = 0$, $Pr(P(\top) = 1) = 1$, and $Pr(P(\top) = 0) = 0$.

Proposition 3. For objectives α , and β , and $x, y, z \in [0, 1]$, formula confidence satisfies: (1) $Pr(P(\alpha) \geq x) > z$ if $Pr(P(\alpha) \geq y) > z$ and $y \geq x$; (2) $Pr(P(\alpha) \geq x) \geq Pr(P(\alpha) \geq y)$ when $y \geq x$; (3) $Pr(P(\alpha) \geq x) \geq Pr(P(\beta) \geq x)$ where $\{\beta\} \vdash \alpha$; (4) $Pr(P(\alpha) \geq x) = Pr(P(\neg\alpha) \leq (1-x))$; (5) $Pr(P(\alpha) \geq 0.5 \vee P(\beta) \geq 0.5) = 1$ where $\{\beta\} \vdash \neg\alpha$; (6) $Pr(P(\alpha) \geq x) = Pr(P(\alpha) = x) + Pr(P(\alpha) > x)$; and (7) $Pr(P(\alpha) < x) + Pr(P(\alpha) = x) + Pr(P(\alpha) > x) = 1$.

If there is positive confidence that the attacker (respectively attackee) is believed, then there is positive confidence that the attackee (respectively attacker) is not believed.

Proposition 4. Let Pr be s.t. if $Pr(P') > 0$, then P' is rational. For all $(A, B) \in \text{Attacks}(G)$,

1. if $Pr(P(A) > 0.5) > 0.5$, then $Pr(P(B) \leq 0.5) > 0.5$.
2. if $Pr(P(B) > 0.5) > 0.5$, then $Pr(P(A) \leq 0.5) > 0.5$.

The following results ensure that we can use belief formulae as outcomes in a lottery (Section 10).

Proposition 5. If $\{\phi_1, \dots, \phi_n\} \subseteq \text{BFormulae}(G)$ is exhaustive, then $Pr(\phi_1 \vee \dots \vee \phi_n) = 1$.

Proposition 6. If $\phi, \psi \in \text{BFormulae}(G)$ are disjoint, then $Pr(\phi \vee \psi) = Pr(\phi) + Pr(\psi)$.

We can treat atoms in $\text{BFormulae}(G)$ as atoms in a classical propositional language, thereby use \vdash as the classical propositional consequence relation.

Proposition 7. Let $\phi, \psi \in \text{BFormulae}(G)$. If $\{\phi\} \vdash \psi$, then $Pr(\phi) \leq Pr(\psi)$.

As the number of different mass distributions with non-zero confidence increases, the confidence in some belief formulae will fall.

Proposition 8. Let Pr^1 and Pr^2 be confidence distributions, and let $\text{Dom}(Pr) = \{P \mid Pr(P) > 0\}$. If $\text{Dom}(Pr^1) \subseteq \text{Dom}(Pr^2)$, then there is a $\phi \in \text{BFormulae}(G)$ such that $Pr^1(\phi) \geq Pr^2(\phi)$.

The confidence value is important for two reasons. First, it gives a better modeling for using lotteries so that the outcomes involve statements about what the user believes/disbelieves, and the confidence value is the degree to which the user does indeed hold those outcomes (and this is a more refined and more natural modeling than found in [19]). Second, it allows for uncertainty about the user to be better managed. If we are sure we know what the user believes, then have one probability distribution, whereas for example, if we are not sure about the user we have, we may have multiple distributions.

So we will treat belief statements as outcomes in a lottery (for calculating expected utility), and use a confidence distribution to give the probability that we get that outcome.

7 UPDATING USER MODELS

This section reviews some proposals in [19]. To update a user model during a dialogue, a mass redistribution function takes a mass distribution and returns a revised mass distribution. Possibilities for this include probabilistic conditioning. However, in this paper, we use an alternative defined next for redistributing mass from models (i.e. possible worlds) not satisfying α to models satisfying α .

Definition 10. [19] Let $\alpha \in \text{AFormulae}(G)$ be a literal, let P be a mass distribution, and let $k \in [0, 1]$. A **refinement function**, denoted $H_\alpha^k(P)$, returns the mass distribution P' as follows where $X \in \text{Models}(G)$

$$P'(X) = \begin{cases} P(X) + (k \times P(h_\alpha(X))) & \text{if } X \models \alpha \\ (1-k) \times P(X) & \text{if } X \not\models \alpha \end{cases}$$

and where $h_\alpha(X) = X \setminus \{A\}$ when α is of the form A and $h_\alpha(X) = X \cup \{A\}$ when α is of the form $\neg A$.

The above function is called refinement because it refines the mass distribution using an update. See Table 1 for examples of redistribution using the refinement function. In the above definition, h_α returns the model closest to X but with α no longer satisfied. If $k = 1$, then all the mass is transferred from the models not satisfying α to models satisfying α . If $k < 1$, then only a proportion is transferred. This gives flexibility to model update in different kinds of user. For instance, if we want to model a user that when conceding an argument is believable, s/he does not fully believe the argument, we can use $k < 1$ to update the model so that the argument is not fully believed in the model.

Table 1. Examples of mass redistribution

AB	P	$H_A^1(P)$	$H_{\neg A}^1(P)$	$H_A^{0.75}(P)$	$H_B^1(P)$
11	0.6	0.7	0.0	0.675	0.8
10	0.2	0.3	0.0	0.275	0.0
01	0.1	0.0	0.7	0.025	0.2
00	0.1	0.0	0.3	0.025	0.0

Given a mass distribution P , representing a user's beliefs at the current state of the dialogue, we want to update the model depending on the move made. For this, we consider the notion of an update method $\sigma(P_{i-1}, D(i)) = P_i$ which generates a mass distribution P_i

from P_{i-1} based on the move $D(i)$. Each method σ is defined as a rule with a condition (based on the move, the current mass distribution, and the graph), and a consequent that specifies the redistribution.

To illustrate, the trusting method (below) raises the belief in a posit, and lowers the belief in attackers and attackees.

Definition 11. [19] For step i in the dialogue, the **trusting method** generates P_i from P_{i-1} as follows, where $\Phi = \{-C \mid (A, C) \in \text{Attacks}(G) \text{ or } (C, A) \in \text{Attacks}(G)\}$.

$$\text{If } D(i) = A!, \text{ then } P_i = H_{\Phi}^1(H_A^1(P_{i-1})).$$

Example 7. For $\langle A, B \rangle$, consider the argument graph in Example 1 with dialogue $[A!, \oplus]$ and the trusting method. Let the initial mass be $P_0(011) = 0.3$, $P_0(010) = 0.2$, $P_0(001) = 0.3$, and $P_0(000) = 0.2$. After move $A!$, $P_1(101) = 0.6$, and $P_1(100) = 0.4$.

The strict method (defined next) only allows a posit to update the belief in the posit when there is no attacker of the posit that is believed.

Definition 12. [19] For step i in the dialogue, the **strict method** generates P_i from P_{i-1} as follows, where $\Phi = \{-C \mid (A, C) \in \text{Attacks}(G)\}$.

$$\begin{aligned} \text{If } D(i) = A!, \\ \text{and for all } (B, A) \in \text{Attacks}(G), P_{i-1}(B) \leq 0.5, \\ \text{then } P_i = H_{\Phi}^1(H_A^1(P_{i-1})), \text{ else } P_i = P_{i-1} \end{aligned}$$

Example 8. For $\langle A, B, C \rangle$, consider the graph in Example 1 with dialogue $[A!, C!, A!, \oplus]$ and the strict method. Let the initial mass be $P_0(111) = 0.2$, $P_0(110) = 0.3$, $P_0(011) = 0.3$, and $P_0(010) = 0.2$. After the first $A!$, $P_1(111) = 0.2$, $P_1(110) = 0.3$, $P_1(011) = 0.3$, and $P_1(010) = 0.2$. After $C!$, $P_2(101) = 0.5$, and $P_2(001) = 0.5$. After the second $A!$, $P_3(101) = 1$.

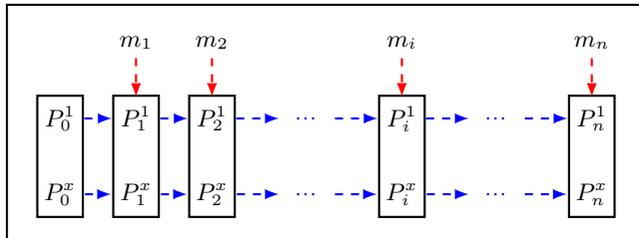


Figure 1. Schematic of the update of the 2D model where $D = [m_1, \dots, m_n]$ and $\mathcal{P}_0 = \langle P_0^1, \dots, P_0^x \rangle$. At the end of the dialogue, the user models are $\mathcal{P}_n = \langle P_n^1, \dots, P_n^x \rangle$.

See [19] for further update methods and for more discussion of how they are used. These are only illustrative of updates methods. With a wider range of moves, a wider range of update methods can be considered. For instance, with moves to get information from the user, further update methods can be defined.

8 2D MODELS

We combine user models (i.e. a mass distribution representing the persuadee beliefs) with the confidence distribution.

Definition 13. A **2D model** is a tuple (\mathcal{P}, Pr) where \mathcal{P} is a tuple $\langle P^1, \dots, P^x \rangle$ s.t. each P^i in \mathcal{P} is a mass distribution and Pr is a confidence distribution s.t. $\sum_{i=1}^x Pr(P^i) = 1$.

So each P^i in the tuple denotes a mass distribution modelling the user. We may have different ones because we are unsure which is correct, though some may be identical.

The most certain 2D model is when the mass distributions are identical (i.e. for all $P^i, P^j \in \mathcal{P}$, if $Pr(P^i) > 0$ and $Pr(P^j) > 0$, then $P^i \equiv P^j$).

At the other extreme, the least certain 2D model is when there is a $P \in \mathcal{P}$ for each $X \subseteq \text{Args}(G)$ such that $P(X) = 1$, and where $Pr(P) = 1/k$ s.t. $k = 2^n$ and $|\text{Arg}(G)| = n$.

In the following definition for updating the 2D model, we can use a different update method for each user model to mimic different ways a user might update his/her beliefs.

Definition 14. Let $(\mathcal{P}_{i-1}, Pr_{i-1})$ is a 2D model where $\mathcal{P}_{i-1} = \langle P_{i-1}^1, \dots, P_{i-1}^x \rangle$, let m_i be a move, and let σ^j be the update method for user model P^j . The **2D model update** \mathcal{P}_{i+1} is $\langle \sigma^1(P_{i-1}^1, m_i), \dots, \sigma^x(P_{i-1}^x, m_i) \rangle$ where for all $P^j \in \mathcal{P}_i$, $Pr_i(P^j) = Pr_{i-1}(P_{i-1}^j)$.

So for each step i of the dialogue, the above definition updates \mathcal{P}_{i-1} to give \mathcal{P}_i . This is represented schematically in Figure 1. For a dialogue D with n moves, and initial 2D model (\mathcal{P}_0, Pr_0) s.t. $\mathcal{P}_0 = \langle P_0^1, \dots, P_0^x \rangle$, we use the function $\text{Update}(\mathcal{P}_0, Pr_0, D) = (\mathcal{P}_n, Pr_n)$ to denote the iterative application of the above definition starting with m_1 , then m_2 , and so on, until m_n .

Example 9. For $\langle A, B, C \rangle$, consider the argument graph in Example 1 with dialogue $D = [C!, A!, \oplus]$. Let $\mathcal{P}_0 = \langle P_0^1, P_0^2 \rangle$ where $P_0^1(m) = 1/8$ for all models, and $P_0^2(011) = 1/2$ and $P_0^2(010) = 1/2$. Also let σ^1 be strict update and σ^2 be trusting update. For move $m_1 = C!$, $\mathcal{P}_1 = \langle P_1^1, P_1^2 \rangle$ where $P_1^1(101) = 1/2$ and $P_1^1(001) = 1/2$, and $P_1^2(001) = 1$. Then for move $m_2 = A!$, $\mathcal{P}_2 = \langle P_2^1, P_2^2 \rangle$ where $P_2^1(101) = 1$, and $P_2^2(101) = 1$. So $\text{Update}(\mathcal{P}_0, Pr_0, D) = (\mathcal{P}_2, Pr_2)$.

For some update functions, e.g. the trusting method, and some belief formulae, we can always construct a dialogue that will result in total confidence in the formulae, and so the mass distributions in \mathcal{P}_0 become more similar. For instance, the following result shows that for a conflictfree set of arguments, each argument can be posited in a dialogue, and the trusting update method ensures that they are all believed.

Proposition 9. For a 2D model (\mathcal{P}_0, Pr_0) , and belief statement π of the form $P(\alpha) \geq 0.5$, where α is a conjunction of arguments, if σ is the trusting update method for all $P \in \mathcal{P}$, and there are no conjuncts A, B in α such that $(A, B) \in \text{Attacks}(G)$, then there is a dialogue $D = [m_1, \dots, m_n]$ s.t. $Pr_n(\pi) = 1$, where $\text{Update}(\mathcal{P}_0, Pr_0, D) = (\mathcal{P}_n, Pr_n)$.

To recap, the 2D model allows us to use multiple user models and multiple update methods to represent the persuadee.

9 UTILITY CONSTRAINTS

The objectives introduced in Section 4 represent what the persuader wants the persuadee to believe or disbelieve.

Definition 15. An **objective tuple** is a pair (Q^+, Q^-) where $Q^+ \subseteq \text{AFormulae}(G)$ and $Q^- \subseteq \text{AFormulae}(G)$ such that $Q^+ \cap Q^- = \emptyset$. We refer to Q^+ as the set of positive objectives and Q^- as the set of negative objectives.

Example 10. An example of an objective tuple is $(\{A\}, \{B\})$ where A is “You are doing little exercise, and so you should do a brisk 30min walk everyday” and B is “Sugar-loaded sports drinks are advertised for sports people, and therefore they are healthy”.

So a positive objective is an objective that the system wants the user to believe and a negative objective is an objective that the system wants the user to disbelieve. Therefore, for an objective tuple (Q^+, Q^-) , outcomes are belief statements as tabulated below. Hence, for a positive objective α , belief in α is a positive outcome, and disbelief in α is a negative outcome. For example, for a positive objective, α , $P(\alpha) > 0.9$ is a positive outcome and $P(\alpha) < 0.4$ is a negative outcome. Similarly, for a negative objective α , belief in α is a negative outcome, and disbelief in α is a positive outcome.

objective	x	belief statement as outcome
α is +ve	$x \in (0.5, 1]$	$P(\alpha) > x$ is +ve outcome.
α is -ve	$x \in (0.5, 1]$	$P(\alpha) > x$ is -ve outcome.
α is +ve	$x \in [0, 0.5]$	$P(\alpha) \leq x$ is -ve outcome.
α is -ve	$x \in [0, 0.5]$	$P(\alpha) \leq x$ is +ve outcome.

We can generalize to arbitrary formulae in $BFormulae(G)$ as follows: If ϕ and ψ are +ve (respectively -ve) outcomes, then $\phi \wedge \psi$ and $\phi \vee \psi$ are +ve (respectively -ve) outcomes. And if ϕ is a +ve (respectively -ve) outcome, then $\neg\phi$ is a -ve (respectively +ve) outcome.

Definition 16. A persuasion utility function, denoted U , for an objective tuple (Q^+, Q^-) is an assignment from $BFormulae(G)$ to \mathbb{R} such that: (1) If ϕ is a +ve outcome, then $U(\phi) > 0$; (2) If ϕ is a -ve outcome, then $U(\phi) < 0$.

Example 11. Continuing Example 10, we can choose the outcomes and U such that $U(P(A) > 0.9) = 10$, $U(P(A) > 0.5 \wedge P(A) \leq 0.9) = 8$, $U(P(A) \leq 0.5) = -10$, $U(P(B) > 0.5) = 5$, and $U(P(B) \leq 0.5) = -5$.

Note, if a formulae is neither +ve nor -ve, it is not necessarily of zero utility. For example, let ϕ be +ve, and ψ be -ve, then $U(\phi \wedge \psi)$ might be greater than 0 if ϕ is more important than ψ , or less than 0 if ψ is more important than ϕ .

Definition 17. A persuasion utility function U for (Q^+, Q^-) is **sensible** iff U satisfies the following conditions.

1. If $x > y$, and α is a +ve (resp. -ve) objective, then $U(P(\alpha) \geq x) \geq U(P(\alpha) \geq y)$ (resp. $U(P(\alpha) \geq x) \leq U(P(\alpha) \geq y)$).
2. If $\{\alpha\} \vdash \beta$, and α, β are +ve (resp. -ve) objectives, then $U(P(\alpha) \geq x) \geq U(P(\beta) \geq x)$ (resp. $U(P(\alpha) \geq x) \leq U(P(\beta) \geq x)$).
3. If $\{\phi\} \vdash \psi$, and ϕ, ψ are +ve (resp. -ve) outcomes, then $U(\phi) \geq U(\psi)$ (resp. $U(\phi) \leq U(\psi)$).

This definition provides intuitive constraints on the persuasion utility function. Condition 1 ensures increased (resp. decreased) belief in a +ve (resp. -ve) objective has increased (resp. decreased) utility; Condition 2 ensures belief in an inferentially stronger +ve (resp. -ve) objective has increased (resp. decreased) utility; and Condition 3 ensures an inferentially stronger +ve (resp. -ve) outcome has increased (resp. decreased) utility.

Proposition 10. If (Q^+, Q^-) is an objective tuple, then there is a persuasion utility function U for (Q^+, Q^-) such that U is sensible.

So if the positive and negative objectives are disjoint, then we are guaranteed to identify a persuasion utility function that is sensible (in the sense of Definition 17).

10 EXPECTED UTILITY

A lottery with possible outcomes o_1, \dots, o_n that are pairwise disjoint and exhaustive (i.e. exactly one of them is guaranteed to occur), that occur with probabilities p_1, \dots, p_n respectively, is written as $[p_1, o_1; \dots; p_n, o_n]$. For a utility function U , the expected utility of a lottery L is $\sum_{i=1}^n p_i \times U(o_i)$. We harness this notion of a lottery as follows.

Definition 18. Let D be a dialogue, let $S = \{\phi_1, \dots, \phi_k\}$ be a set of disjoint and exhaustive outcomes (i.e. belief formulae), let (P_0, Pr_0) be the initial 2D model, let $Update(P_0, Pr_0, D) = (P_n, Pr_n)$, and let U be a utility function. The **lottery** for Pr, U, S is $Lot(Pr, U, S) =$

$$[Pr(\phi_1), \phi_1; \dots; Pr(\phi_k), \phi_k]$$

Then the **expected utility** for Pr, U, S is $EU(Pr, U, S) =$

$$(Pr(\phi_1) \times U(\phi_1)) + \dots + (Pr(\phi_k) \times U(\phi_k))$$

Example 12. For $\langle A, B \rangle$, let $P_n^1(11) = 1$, $P_n^2(11) = 0.6$, $P_n^2(01) = 0.4$, and $P_n^3(01) = 1$, with $Pr(P_n^1) = 0.5$, $Pr(P_n^2) = 0.3$, and $Pr(P_n^3) = 0.2$. Let the objective tuple be $(\{A\}, \emptyset)$. Hence, ϕ_1 is a positive outcome, ϕ_2 is neither a positive nor negative outcome, and ϕ_3 is a negative outcome. So using the values for Pr and U in the table, the expected utility is 4.5.

ϕ	$Pr(\phi)$	$U(\phi)$
$\phi_1 = P(A) > 0.9$	0.5	10
$\phi_2 = (P(A) \leq 0.9) \wedge (P(A) > 0.5)$	0.3	5
$\phi_3 = P(A) \leq 0.5$	0.2	-10

Using the 2D model, we can determine the optimal dialogues for a lottery as follows.

Definition 19. A dialogue D is **optimal** w.r.t. the initial 2D model (P_0, Pr_0) , utility function U , and $Update(P_0, Pr_0, D) = (P_n, Pr_n)$, when $EU(Pr_n, U, S)$ is maximized.

In the following example, we show how we can choose between dialogues using a 2D model.

Example 13. Consider the following argument graph with the objective tuple $(\{A \vee C\}, \emptyset)$.



Let $\mathcal{P}_0 = \langle P_0^1, P_0^2 \rangle$ be defined as follows, and assume we use the strict update method. Note, we give the probability for each argument rather than each model to save space.

	A	B	C	D
P_0^1	0	1	0	0
P_0^2	0	0	0	1

The updated mass distributions are given below for dialogue $D_1 = [A!, \oplus]$ (left) and for dialogue $D_2 = [C!, \oplus]$ (right).

	A	B	C	D
P_0^1	0	1	0	0
P_0^2	1	0	0	1

	A	B	C	D
P_0^1	0	1	1	0
P_0^2	0	0	0	1

Assume $Pr(P_n^1) = 2/3$ and $Pr(P_n^2) = 1/3$. and outcomes $\phi_1 = P(A \vee C) > 0.5$ and $\phi_2 = P(A \vee C) \leq 0.5$ s.t. $U(\phi_1) = 5$ and $U(\phi_2) = -5$. So for dialogue D_1 , expected utility is $(1/3 \times 5) + (2/3 \times -5) = -5/3$, and for dialogue D_2 , expected utility is $(2/3 \times 5) + (1/3 \times -5) = 5/3$.

We could select a longer dialogue $D_3 = [A!, C!, \oplus]$ giving the following updated mass distributions, and with expected utility $(1 \times 5) + (0 \times -5) = 5$.

	A	B	C	D
P_1	0	1	1	0
P_2	1	0	0	1

For the shorter dialogues, D_2 is better than D_1 . However, D_3 is better than both D_2 and D_1 , but D_3 is longer.

At one extreme, if the 2D model only contains one user model, then the outcome is known with certainty (i.e there is complete confidence in the belief statement).

Proposition 11. If $[Pr(\phi_1), \phi_1; \dots; Pr(\phi_k), \phi_k]$ is a lottery, and $|\mathcal{P}| = 1$, then there is a $\phi_i \in \{\phi_1, \dots, \phi_k\}$ s.t. $Pr(\phi_i) = 1$, and for all $\phi_j \in \{\phi_1, \dots, \phi_k\} \setminus \{\phi_i\}$, $Pr(\phi_j) = 0$.

Example 14. Consider the disjoint and exhaustive outcomes $P(A \vee B) = 1$, $P(\neg A \wedge \neg B) = 1$, and $P(A \vee B) < 1 \wedge P(\neg A \wedge \neg B) < 1$. Let $\mathcal{P} = \{P'\}$ and so $Pr(P') = 1$. Let $P'(11) = 1$. Hence, $Pr(P(A \vee B) = 1) = 1$.

At the other extreme, there are various situations that give rise to a uniform distribution over the outcomes. We consider the following which reflects the ignorance when there are multiple mass distributions with no agreement.

Proposition 12. Let $[Pr(\phi_1), \phi_1; \dots; Pr(\phi_n), \phi_n]$ be a lottery where Pr is a uniform distribution over \mathcal{P} . Also for each $P \in \mathcal{P}$, there is a $X \subseteq \text{Args}(G)$ s.t. $P(X) = 1$, and for each $X \subseteq \text{Args}(G)$, there is a $P \in \mathcal{P}$ s.t. $P(X) = 1$. If there is an $x > 0$ s.t. for each $\phi_i \in \{\phi_1, \dots, \phi_n\}$, $|\text{Sat}(\phi_i)| = x$, then for each $\phi_i, \phi_j \in \{\phi_1, \dots, \phi_n\}$, $Pr(\phi_i) = Pr(\phi_j)$.

Example 15. For $\langle A \rangle$, let $\mathcal{P} = \{P_1, P_2\}$ where $P_1(1) = 1$ and $P_2(0) = 1$. Let $Pr(P_1) = 1/2$ and $Pr(P_2) = 1/2$. Consider the outcomes $P(A) > 0.5$ and $P(\neg A) > 0.5$. Hence, $Pr(P(A) > 0.5) = 1/2$ and $Pr(P(\neg A) > 0.5) = 1/2$

Between these extremes, the 2D model can be valuable in identifying the optimal dialogues. Note, that normally we do not envisage that the 2D model will contain many mass distributions. Furthermore, we will focus on update methods that are computationally efficient. Hence, we envisage the approach is computationally viable.

11 MODELLING DISENGAGEMENT

For every user-terminated dialogue D , there is a probability that the user of the app will disengage before the end of the dialogue (e.g. through loss of interest). This can rise as the length of the dialogue increases. We assume a **stay-in probability**, denoted q , which for step i in the dialogue is the probability that the user will remain engaged for the next step $i + 1$. We assume no disengagement after the ultimate posit.

Definition 20. Let $D = [m_1, \dots, m_n]$ be a system-terminated dialogue with stay-in probability q . If $n > 2$, the **probability of engagement** is $prob_{engage} = q^{n-2}$ and the **probability of disengagement**

is $prob_{disengage} = \sum_{i=1}^{n-2} q^{(i-1)} \times (1 - q)$. If $n = 1$ or $n = 2$, then $prob_{engage}$ is 1, and $prob_{disengage}$ is 0.

Example 16. Consider the dialogue in Figure 2 with the stay-in probability being 0.9. So $prob_{engage}$ is $0.9 \times 0.9 = 0.81$ and $prob_{disengage}$ is $0.1 + (0.9 \times 0.1) = 0.19$.

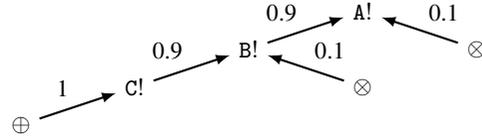


Figure 2. For dialogue $[A!, B!, C!, \oplus]$, each node is a move. The left branch is the system-terminated dialogue, and each branch that ends in \otimes is a user-terminated dialogue. Each arc in the tree is labelled with the probability of engagement (leftwards) or disengagement (rightwards).

Proposition 13. For a system-terminated dialogue D , and a stay-in probability q , $prob_{engage} + prob_{disengage} = 1$.

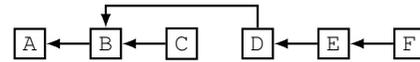
Since disengagement is often a clear event in a dialogue, obtaining a stay-in probability can be obtained from analyzing previous dialogues for a class of users.

Given the probability of engagement $Prob_{engage}$ and a lottery $[Pr(\phi_1), \phi_1; \dots; Pr(\phi_n), \phi_n]$, we form a revised lottery as specified in the following result where \otimes denotes the outcome of disengagement..

Proposition 14. If $[Pr(\phi_1), \phi_1; \dots; Pr(\phi_n), \phi_n]$ is a lottery, and $Prob_{engage} \in [0, 1]$, then the following is a lottery where for each outcome ϕ_i , $Pr^*(\phi_i)$ is $Pr(\phi_i) \times Prob_{engage}$.

$$[Pr^*(\phi_1), \phi_1; \dots; Pr^*(\phi_n), \phi_n; 1 - Prob_{engage}, \otimes]$$

Example 17. For $\langle A, \dots, F \rangle$, consider the graph with $\mathcal{P} = \{P_0\}$ where $P_0(010010) = 1$. So $P_0(B) = 1$ and $P_0(E) = 1$.



Let A be a +ve objective, and let $P(A) \geq 0.9$ and $P(A) < 0.9$ be outcomes in the lottery. So $D_1 = [C!, A!, \oplus]$ and $D_2 = [F!, D!, A!, \oplus]$ are dialogues that terminate with $Pr(P(A) \geq 0.9) = 1$ according to the strict update method. Let the stay-in probability be $3/4$. So $Prob_{engage} = 3/4$ for D_1 and $Prob_{engage} = 9/16$ for D_2 . Hence, the revised lottery has for D_1 , $Pr^*(P(A) \geq 0.9) = 3/4$, $Pr^*(P(A) < 0.9) = 0$, and $Pr^*(\otimes) = 1/4$, and for D_2 , $Pr^*(P(A) \geq 0.9) = 9/16$, $Pr^*(P(A) < 0.9) = 0$, and $Pr^*(\otimes) = 7/16$. So for this stay-in probability, the optimal dialogue is D_1 .

	A	B	C	D	E	F
P_0	0	1	0	0	1	0
P_n for D_1	1	0	1	0	0	0
P_n for D_2	1	0	0	1	0	1

Shorter dialogues can be preferable (as above). In general, we trade a decrease in expected utility for a decrease in risk of disengagement (e.g. for Example 13 whether D_2 or D_3 is optimal would depend on the stay-in probability).

12 DISCUSSION

This paper provides the following contributions (which are potentially important features for using argumentation in software for changing behaviour): (1) A 2D model of uncertainty giving predictions of the beliefs of the persuadee, and of the confidence in those predictions; (2) A framework for updating the 2D model through dialogues; and (3) Shown how the 2D model can be used to optimize choice of moves while taking into account the risk of disengagement.

For this, the epistemic approach to probabilistic argumentation has been used. This contrasts with the constellations approach (e.g. [11, 23, 15]) which is concerned with the uncertainty about the structure of the graph rather than belief in arguments.

The proposal in this paper relies on 2D models. This can be generated by querying the user, or by learning from previous interactions with the user or similar users. Some recent studies indicate the potential viability of an empirical approach [30, 9, 33].

Utility theory has been considered previously in argumentation (for example [29, 32, 24, 25]) though none of these represent the uncertainty of moves made by each agent in argumentation. There is an approach using expected utility where outcomes are specified as particular arguments being included or excluded from extensions [20], but it is based on the constellations approach (as opposed to the epistemic approach), and there is no consideration of updates to the model. Outcomes from asymmetric dialogues have also been considered in [5], but that work focuses on whether it is guaranteed, possible, or impossible to present a winning coalition of arguments with respect to grounded semantics, and there is no consideration of uncertainty.

There is increasing interest in formalizing the notion of the strength of an argument, with a number of proposals (e.g. [3, 8, 24, 22, 1, 6, 28]). It would be interesting to investigate the pros and cons of using these conceptualizations of strength of an argument instead of epistemic probabilities in this framework. Nonetheless, some clear advantages of the epistemic approach are the clear semantics for the evaluation of the arguments, the ease with which epistemic approach can be used in a lottery, and the possibility to obtain the probabilities by analysing statistical data concerning the behaviour agents.

The work in this paper goes beyond [19]: (1) to better model lotteries so that outcomes involve statements about user beliefs, and the confidence value is the degree to which the user does indeed hold those outcomes which is a more refined and natural modeling than [19]; (2) to allow for uncertainty about the user to be handled, and so if we are sure we know the users beliefs, then we have one distribution, whereas if we are unsure about kind of user, we have multiple distributions; and (3) to model the risk of disengagement which is a practical issue that significantly affects the usability of any argumentation approach for behavior change.

Our current research is directed at generating probability distributions for user models. We are exploring the use of queries to the user where the user can express belief in individual arguments (such as strongly agree, agree, neither agree nor disagree, etc which are then mapped to the [0,1] interval). If we do this for some arguments, we can attempt to guess the belief in remaining arguments. We are also exploring how classes of user might believe/disbelieve certain arguments. So by knowing beliefs in arguments for some members of the class, and by having criteria for assigning individuals to a class with some probability, we may construct the 2D model for a user. We envisage that by surveying representative samples of individuals, we can obtain useful 2D models. We aim to develop similar methods to those used in [30, 9, 33]. We plan to undertake empirical evaluation

of the approach in apps to persuade users to change their behaviour with respect to some aspect of their lifestyle (e.g. to eat less, to drink less alcohol, to drive more safely, to recycle more, etc). We see the theoretical developments in this paper being viable and valuable for the prototype system that we are implementing.

ACKNOWLEDGEMENTS

This research was partly funded by EPSRC grant EP/N008294/1 for the Framework for Computational Persuasion project.

REFERENCES

- [1] L. Amgoud and J. Ben-Naim, 'Axiomatic foundations of acceptability semantics', in *Proceedings of KR'16*, pp. 2–11, (2016).
- [2] P. Baroni, M. Giacomin, and P. Vici, 'On rationality conditions for epistemic probabilities in abstract argumentation', in *Computational Models of Argument (COMMA'14)*, pp. 121–132, (2014).
- [3] Ph. Besnard and A. Hunter, 'A logic-based theory of deductive arguments', *Artificial Intelligence*, **128**(1-2), 203–235, (2001).
- [4] E. Black, A. Coles, and S. Bernardini, 'Automated planning of simple persuasion dialogues', in *Computational Logic in Multi-agent Systems (CLIMA'14)*, volume 8624 of *LNCSS*, pp. 87–104. Springer, (2014).
- [5] E. Black and A. Hunter, 'Reasons and options for updating an opponent model in persuasion dialogues', in *Theory and Applications of Formal Argumentation (TFAFA'15)*, (2015).
- [6] E. Bonzon, J. Delobelle, S. Konieczny, and N. Maudet, 'A comparative study of ranking-based semantics for abstract argumentation', in *Proceedings of AAAI'16*, pp. 914–920, (2016).
- [7] M. Caminada and M. Podlaszewski, 'Grounded semantics as persuasion dialogue', in *Computational Models of Argument (COMMA'12)*, pp. 478–485, (2012).
- [8] C. Cayrol and M. Lagasque-Schiex, 'Graduality in argumentation', *Journal of Artificial Intelligence Research*, **23**, 245–297, (2005).
- [9] F. Cerutti, N. Tintarev, and N. Oren, 'Formal arguments, preferences, and natural language interfaces to human: An empirical evaluation', in *Proceedings of ECAI*, pp. 207–212, (2014).
- [10] P. Dung, 'On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and n-person games', *Artificial Intelligence*, **77**, 321–357, (1995).
- [11] P. Dung and P. Thang, 'Towards (probabilistic) argumentation for jury-based dispute resolution', in *Computational Models of Argument (COMMA'10)*, pp. 171–182. IOS Press, (2010).
- [12] X. Fan and F. Toni, 'Assumption-based argumentation dialogues', in *Proceedings of IJCAI'11*, pp. 198–203, (2011).
- [13] C. Hadjinikolis, Y. Siantos, S. Modgil, E. Black, and P. McBurney, 'Opponent modelling in persuasion dialogues', in *Proceedings of IJCAI*, pp. 164–170, (2013).
- [14] E. Hadoux, A. Beynier, N. Maudet, P. Weng, and A. Hunter, 'Optimization of probabilistic argumentation with markov decision models', in *Proceedings of IJCAI'15*, (2015).
- [15] A. Hunter, 'Some foundations for probabilistic argumentation', in *Computational Models of Argument (COMMA'12)*, pp. 117–128, (2012).
- [16] A. Hunter, 'A probabilistic approach to modelling uncertain logical arguments', *International Journal of Approximate Reasoning*, **54**(1), 47–81, (2013).
- [17] A. Hunter, 'Opportunities for argument-centric persuasion in behaviour change', in *Logics in Artificial Intelligence (JELIA'14)*, volume 8761 of *LNCSS*, pp. 48–61. Springer, (2014).
- [18] A. Hunter, 'Probabilistic strategies in dialogical argumentation', in *Scalable Uncertainty Management (SUM'14)*, volume 8720 of *LNCSS*, pp. 190–202. Springer, (2014).
- [19] A. Hunter, 'Modelling the persuadee in asymmetric argumentation dialogues for persuasion', in *Proceedings of IJCAI 2015*, (2015).
- [20] A. Hunter and M. Thimm, 'Probabilistic argument graphs for argumentation lotteries', in *Computational Models of Argument*, pp. 313–324, (2014).
- [21] A. Hunter and M. Thimm, 'Probabilistic argumentation with incomplete information', in *Proceedings of ECAI*, pp. 1033–1034, (2014).
- [22] J. Leite and J. Martins, 'Social abstract argumentation', in *Proceedings of IJCAI'11*, (2011).

- [23] H. Li, N. Oren, and T. Norman, 'Probabilistic argumentation frameworks', in *Theory and Applications of Formal Argumentation (TAAFA'11)*, pp. 1–16, (2011).
- [24] P. Matt and F. Toni, 'A game-theoretic measure of argument strength for abstract argumentation', in *Logics in Artificial Intelligence (JELIA'08)*, volume 5293 of *LNCS*, pp. 285–297, (2008).
- [25] N. Oren and T. Norman, 'Arguing using opponent models', in *Argumentation in Multi-agent Systems*, volume 6057 of *LNCS*, pp. 160–174, (2009).
- [26] H. Prakken, 'Coherence and flexibility in dialogue games for argumentation', *Journal of Logic and Computation*, **15**(6), 1009–1040, (2005).
- [27] H. Prakken, 'Formal systems for persuasion dialogue', *Knowledge Engineering Review*, **21**(2), 163–188, (2006).
- [28] A. Rago, F. Toni, M. Aurisicchio, and P. Baroni, 'Discontinuity-free decision support with quantitative argumentation debates', in *Proceedings of KR'16*, pp. 63–73, (2016).
- [29] I. Rahwan and K. Larson, 'Pareto optimality in abstract argumentation', in *Proceedings of AAAI 2008*, pp. 150–155. AAAI Press, (2008).
- [30] I. Rahwan, M. Madakkatel, J. Bonnefon, R. Awan, and S. Abdallah, 'Behavioural experiments for assessing the abstract argumentation semantics of reinstatement', *Cognitive Science*, **34**(8), 14831502, (2010).
- [31] T. Rienstra, M. Thimm, and N. Oren, 'Opponent models with uncertainty for strategic argumentation', in *Proceedings of IJCAI'13*, pp. 332–338. IJCAI/AAAI, (2013).
- [32] R. Riveret, H. Prakken, A. Rotolo, and G. Sartor, 'Heuristics in argumentation: A game theory investigation', in *Computational Models of Argument (COMMA 2008)*, pp. 324–335. IOS Press, (2008).
- [33] A. Rosenfeld and S. Kraus, 'Providing arguments in discussions based on the prediction of human argumentative behavior', in *Proceedings of AAAI'15*, (2015).
- [34] M. Thimm, 'A probabilistic semantics for abstract argumentation', in *Proceedings of ECAI'12*, pp. 750–755, (2012).
- [35] M. Thimm, 'Strategic argumentation in multi-agent systems', *Kunstsichliche Intelligenz*, (2014).

A Data Driven Similarity Measure and Example Mapping Function for General, Unlabelled Data Sets

Damien Lejeune and Kurt Driessens¹

Abstract. Deep networks such as autoencoders and deep belief nets are able to construct alternative, and often informative, representations of unlabeled data by searching for (hidden) structure and correlations between the features chosen to represent the data and combining them into new features that allow sparse representations of the data. These representations have been chosen to often increase the accuracy of further classification or regression accuracy when compared to the original, often human chosen representations. In this work, we attempt an investigation of the relation between such discovered representations found using related but differently represented sets of examples. To this end, we combine the cross-domain comparison capabilities of unsupervised manifold alignment with the unsupervised feature construction of deep belief nets, resulting in an example mapping function that allows re-encoding examples from any source to any target task. Using the t-Distributed Stochastic Neighbour Embedding technique to map translated and real examples to a lower dimensional space, we employ KL-divergence to define a dissimilarity measure between data sets enabling us to measure found representation similarities between domains.

1 Introduction

While raw data is abundant, the difficulty with using this data, and according to the authors' one of the biggest challenges in the current big data hype, is the lack of any structured way of representing the data, leading to many different, human chosen, but unmatching representations of similar or related, but most of the time not identical² data. Examples of this are numerous, ranging from data stemming from medical questionnaires, where almost never the same questions are asked, but the topics are often similar, over gene transcription data where old style microarray data and more recent RNAseq measurements exist over a pool of intersecting but not identical gene sets to control oriented data where samples of system behaviour of a number of control problems exist, but almost never match in the chosen representation.

The subfield of machine learning in which this problem is tackled is known under the names transfer learning, inductive transfer and domain adaptation. The idea behind transfer learning is to enhance learning performance on a task by employing, i.e., re-using data, experience, and/or solutions from different but related tasks that were solved earlier. Existing work on transfer learning and in domain adaptation for supervised tasks [18, 9, 11] mainly focuses on the shift of the probability distributions observed between different tasks and how to correct for those, but not on the issue of different

representations. Also in reinforcement learning, the usual drawback of the tabula rasa approach when confronting new tasks has lead to a flurry of research on transfer learning [28]. Historically, defining the relation between the old task and the new, e.g. feature mapping, goal mapping, translating the model or policy to match the representations was handled by a human expert [29].

More recent work proposes autonomous transfer methods that aim at deriving the inter task mapping from learning examples of the two tasks automatically [27, 8, 7, 5]. These work by studying, or having an algorithm analyse the internal structure of the examples in the data set and trying to exploit the similarities between data sets in such structure. For example, in reinforcement learning, the dynamics of the task to be solved can be observed from sample interactions with the environment, consisting of the state the agent was in, the action that was selected and the state the action lead to. Analysing these dynamics and mapping the samples from both tasks into a joint feature space can give an indication of how the two tasks are related, i.e. where and how they shared dynamics and control response. The same idea applies to standard supervised learning tasks such as classification or regression, where the structure arises from the fact that, in principle, examples do not uniformly cover the entire example space as defined by the human chosen representational format. Existing work that takes this approach has used sparse coding [8] and manifold alignment [32] to define the joint space.

The contribution of this work is twofold, as we introduce:

- (i) a **data driven difference measure** for comparing data sets
- (ii) an **automatically derived inter-task mapping** that can be used to compare any two data sets and thus learning tasks, whether they are supervised, unsupervised or reinforcement learning.

The approach takes the shape of a pipeline employing well studied and tested techniques. The pipeline relies on deep belief nets to generate expressive features for both data repositories and on unsupervised manifold alignment to find the best mapping between these features. Applying one deep belief net to generate hidden feature activations, a forward and backwards projection into the alignment space and the other deep belief net to reconstruct the visible node activations from the projected hidden node activations, it becomes possible to translate learning examples from a source task into examples for the target task. By comparing the embedding of the original and the translated examples in a low dimensional projection built using the t-Distributed Stochastic Neighbour Embedding, we define a difference measure based on the Kullback-Leibler divergence between the two example distributions. Experiments show that the pipeline is able to autonomously find meaningful analogies between data-sets that match human intuition.

The pipeline draws on the power of already developed techniques

¹ Maastricht University, email: kurt.driessens@maastrichtuniversity.nl

² With identical, we refer to the representation and domain of the data, not the examples measured.

and applies and combines them in a new problem domain. The combination of the techniques adds little to no complexity to these techniques except for the way in which they are combined. No additional parameters to set or tune are introduced and no expert knowledge on any of the involved tasks is required. The exact set-up of the pipeline will be further discussed in section 3.

The rest of the paper is structured as follows: we introduce the necessary concepts and discuss related work in section 2. Section 4 demonstrates the capabilities of the pipeline using data sets from a number of well known classification tasks and reinforcement learning benchmarks to show that the discovered relations and similarities have a striking resemblance to human intuition.

2 Preliminaries

In this section, we introduce the problem we are trying to address with this work, we present related work and introduce the different concepts we will use later on. Given the similarity possible application of our technique to transfer learning and the similarity of the settings, we will adopt the vocabulary of the transfer learning domain and talk about **source** and **target** data sets/tasks to make referencing the two different domain easier.

2.1 Autonomous Transfer Learning

Transfer learning [30, 28] is concerned with the re-use of data, or experience in the case of reinforcement learning, or the learned models and policies from a previously learned source task, to improve learning performance in a new, sometimes more complex target task. **Shallow** transfer is concerned with the transfer of information when the learning examples from the source and the target task share their feature space. However, since these features are usually chosen by a human expert, or simply chosen by their availability for a learning task, they often differ between tasks, leading to what is known as **deep** transfer. This is the setting under which we operate in this work. When reusing previous experience from a source task S with feature representation ψ_S in a new target task T with different features ψ_T , the two representations need to be related. This relation often takes the form of an inter-task mapping $\mathcal{T} : \psi_S \rightarrow \psi_T$ that relates the expert chosen features to each other.

When a set of source tasks (S_1, S_2, \dots, S_n) is available to transfer from, or if the task is to build an optimised learning path through a number of related learning tasks, a task similarity measure \mathcal{M} can be used to select which source task to transfer from, or which task to select next. Such a measure will map any combination of two learning tasks to \mathbb{R} , giving a score to each combination according to their similarity.

Related Work

Given the difficulties with its tabula rasa learning premise, transfer learning for reinforcement learning tasks has received quite a bit of attention in the past years [28]. Where historically the required inter-task mapping was defined by domain experts [29] more recent work has focussed on completely autonomous transfer. Bou Ammar et al. [8] construct a joint space for the source and task domain samples through sparse coding [21]. In this joint space, samples from the source and the target domain are paired using a Euclidean distance and used as learning examples in a supervised learning task that attempts to model the inter-task mapping. The problem with this approach is that in the commonly high dimensional joint space, the

Euclidean distance carries little information. In follow up work [7], Bou Ammar et al. use a three way restricted Boltzmann machine to learn the inter-task mapping. This setup relies on a complex training phase based on mini-batch learning and re-learning that randomly pairs samples from the two domains and relies on the reconstruction error of the Boltzmann machine to select which pairs match best. This makes training the machine quite involved, as many repetitions of randomly paired samples are necessary.

For supervised learning tasks, unsupervised manifold alignment has been used to relate the learning examples from the source and target tasks [32]. Very recently, Bou Ammar et al. [2] applied this manifold alignment technique to policy gradient reinforcement learning. Since this technique is so closely related to ours, in fact, the unsupervised manifold alignment is a part of our pipeline, we will compare the inter-task mapping generated by both techniques in our experimental section.

On top of the inter-task mapping, our approach also gives rise to a domain similarity measure. Of existing work on domain similarity measures, the most related to ours in the work by Bou Ammar et al. on RBDist [6] that bases its similarity measure on the reconstruction error of a single restricted Boltzmann machine and the follow up work on DRBDist [4] using a combination of Deep Belief Nets a bit like ours. The main difference is that DRBDist uses an identity mapping between the top layers of two DBNs. This not only places a restriction on the number of nodes of the top layer of the DBNs, i.e. the number of nodes must be equal, it also assumes a ordered pairwise similarity between the features learned by both networks. Additionally, since the similarity measure relies on the reconstruction error, it is very sensitive to the convergence results of the DBN learning phase. In fact, when trying to reproduce the results from this work, we were unable to reach the same results.

2.2 Deep Belief Networks

A restricted Boltzmann machine (RBM) is a neural network (NN) consisting of two layers (one visual and one hidden) of stochastic nodes that can be trained to represent a probability distribution over data points [19]. Each neuron in a layer is connected to all neurons in the other layer forming a fully connected bipartite graph. An RBM is trained using gradient descent where the gradient is not directly computed but estimated by the contrastive divergence algorithm [17]. A deep belief network (DBN) is a stack of RBMs trained in a layer-wise setup. Increasing the number of layers is done to improve the fit of the probability distribution and can be shown to extract, in an unsupervised way, higher level features from a dataset.

While a single hidden layer RBM is able to capture basic features, stacking multiple layers on top of each other allows learning higher level features. Such a stack of RBMs, which itself forms a NN, is called a Deep Belief Network (DBN). Each layer is trained in a greedy, layer wise fashion. This forces each additional hidden layer to learn a new feature representation to encode the layer below and makes this technique well suited for automated feature extraction. It has been successfully applied to domains like handwritten digit recognition [25], speech recognition [15] and Atari games [23], among other things [3].

2.3 Manifold Alignment

Finding feature correspondences in a source and a target task is important when assessing the similarity between them. Chang

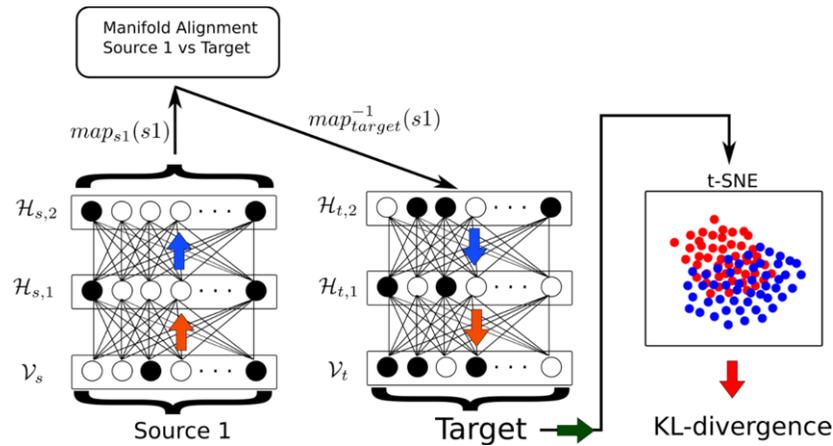


Figure 1. Pipeline using DBNs, manifold alignment and t-SNE.

Wang [32] proposed a method to align two datasets of arbitrary dimensions by finding these correspondences in an unsupervised way. The alignment is performed in a new space into which the data can be mapped. The algorithm also provides inverse mappings which allow to reconstruct data points belonging to a source task into a target task’s original space. It then follows that the quality of this reconstruction will depend on the quality of the alignment and therefore will depend on the similarity between the two datasets.

The alignment and the corresponding transformation is achieved by first computing intra-similarities between the points in the same dataset. Then, the local geometries in both domains are represented by the distances of the k -nearest neighbours for each point. Intuitively, if the two sets have the same local geometries, up to affine transformations, the sets can be perfectly aligned.

Having this local information, the mappings for both domains can be found by minimising a cost function that forces points exhibiting similar geometries to be mapped closely together in the common space, and at the same time enforce the points which are close to others in their respective space to also be close in the common space.

2.4 t-Distributed Stochastic Neighbour Embedding

Measuring similarities between data points or data sets in high dimensional spaces is complex and not always meaningful. t-Distributed stochastic neighbour embedding or t-SNE [31] is a dimensionality reduction method that aims to maintain the same data distribution from the original high dimensional (HD) space in a low dimensional (LD) one. Since the technique originates in visualisation, LD space is usually two dimensional.

The mapping done in such a way that close points in the HD space should also be close into the LD space. In other words, t-SNE finds, in an unsupervised way, a mapping $\mathbb{R}^n \rightarrow \mathbb{R}^2$ that conserves the local geometries of the original dataset. To do so, it computes a joint probability distribution for observing a point x_j around a point x_i using a multivariate Gaussian distribution centred at x_i . The algorithm also finds a proper variance for each distribution at x_i in order to model the local geometries by taking the density around that point into consideration. This probability is at the core of the method as the LD space should reflect the same probability distribution as the HD one and therefore model the same similarities. To achieve this, a gradient descent search is performed on the Kullback-Leibler divergence

with the aim at minimising the difference between the distribution in \mathbb{R}^n and the one in \mathbb{R}^2 . The embedding in the low dimensional space comes at the price of not conserving the global geometries of the original dataset. Points far away in the HD space will be even farther away in the LD.

3 Data Driven Similarity Construction

We want our approach to match and find similarities between the two domains completely autonomously, so we define a set-up that computes a similarity measure between any two previously unseen (and possibly unlabelled) domains and constructs a mapping between the two domains that allows re-encoding samples from one domain into samples of the other. The similarity measure will indicate the degree to which the two domains match and will quantify the quality of the re-encoding.

3.1 Working Constraints

Having the inter-task mapping and the domain similarity measure constructed completely unsupervised leads to the following constraints:

- (i) No prior information about the domains must be required besides data samples.
- (ii) No requirements can be placed on the dimensionality of the different domains.

Our approach consists of a pipeline that makes use of three core methods to construct both the inter-task mapping and the similarity measure: (i) deep belief networks [16], (ii) manifold alignment [32] without correspondence and (iii) t-SNE [31].

3.2 The pipeline

Our pipelined approach, illustrated by Figure 1, goes as follows:

- (i) For each of the datasets, a separate DBN is used to extract high level features. Each dataset consists of (unlabelled) samples from one domain. This step is expected to result in a better representation of the data for each domain and to help increase the comparativeness by working on underlying characteristics instead of the

low level (human chosen) features of the raw data. No restrictions are placed on the number of nodes used in any layer or on the number of layers itself, so the DBN can be optimised to match the domain it is used for, independent of the rest of the pipeline and according to the experience of the user.

- (ii) The manifold alignment uses samples from two domains, re-encoded into the feature space extracted by the DBNs, and computes a mapping to a space where the samples from the source and the target domains can be compared. As mentioned in Section 2.3, this method is able to work with data originating from spaces with different dimensionalities, to get rid of affine transformations and to provide an inverse mapping that allows the data from one domain to be transformed into the other domain. The quality of this transformation is influenced by the quality of the alignment which in turn is dependent of the similarity between the two datasets. A good alignment will transform the source's data points into a set of data points that matches the original target distribution.
- (iii) In order to obtain a meaningful similarity measure, t-SNE reduces the original, usually high dimensional, space into a two dimensional one. Then, the Kullback-Leibler divergence (KL-divergence) is computed in t-SNE's space between the original target distribution and the distribution of the reconstructed source samples. This KL-divergence represents the measure between the source and target datasets.

The training of the DBNs is done for each domain to be measured, but since this step is independent of any other domains involved, it only has to be executed once for each domain. Subsequently, the manifold alignment is trained with the source's and target's last hidden layers activations for each pair of domains that we want to compare. The last phase is to train the t-SNE algorithm in order to reduce the dimensionality of the target's space. With the aim to use the KL-divergence over points laying on the plane generated by t-SNE, probability distributions for the target's original and reconstructed datasets have to be estimated. In the experiments reported below, we used kernel density estimators at 150×150 equidistant points.

4 Experiments

In this section we empirically evaluate the pipeline presented above. Since there is no ground truth when comparing different datasets with respect to the inter domain mapping and the computed similarity measure, we've selected domains that can be matched using human intuition, as well as domains from reinforcement learning where related work generated a base for comparison.

To compare with previous work and to illustrate the importance of each step in the pipeline we first use the MNIST dataset of written characters [20]. Since this dataset is composed of easy to interpret images, this data allows us to clearly show how the approach works and what the influence is of each step. To allow comparisons on this data set, we treat the samples of each digit as a separate domain, allowing us to visually show the results of the re-encoding to compare the similarity measure to human intuition.

To illustrate the use of the pipeline on data stemming from spaces with different dimensionality, we then apply our approach to accomplish transfer between the pen-digits dataset [1] and the MNIST dataset. This setup again allows us to show qualitative results of the re-construction. Additionally, we present some quantitative results of the influence our inter-task mapping method could have on classifier accuracy through example transfer.

We end with a set of experiments on reinforcement learning benchmarks which allows us to compare to previously published data

driven similarity measures [4].

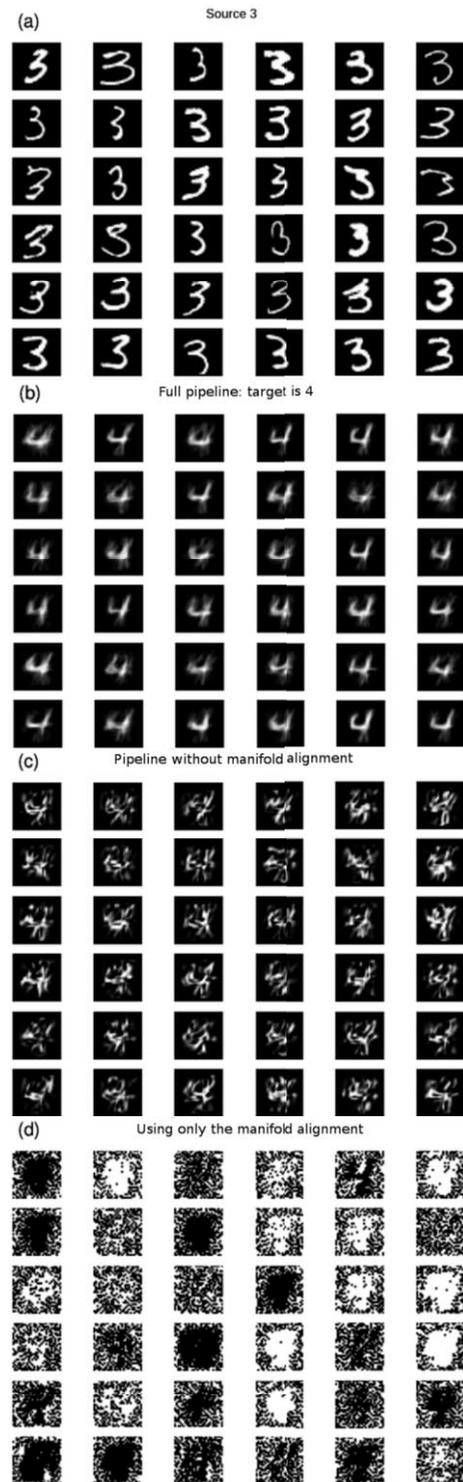


Figure 2. Experiment using 3 as source and 4 as target. (a) samples used as source (b) shows the reconstruction of with the full pipeline. (c) the reconstruction without manifold alignment. (d) the reconstruction without DBNs

4.1 MNIST

The MNIST [20] dataset holds images of centered handwritten digits, each composed of 28×28 grayscale pixels. The goal of this experiment is to measure the similarity between different digits in MNIST. For example, it aims at testing if the digit 0 is closer to 8 than it is to 4. Multiple parameters had to be set up and have been kept fixed for all experiments in this section. Given that the pipeline is independent of the used DBN architecture, we decided to follow literature and composed the DBN of two hidden layers: the first one having 300 units and the second one having 100 units. Each NN was trained to model the distribution of one digit, using on average 6000 examples per digit. The manifold alignment used a parameter $\mu = 1$ and was trained on 1000 sets of activations (i.e. samples) per dataset. Once the data is reconstructed by the target DBN, it is used to compute the t-SNE scatter plot. t-SNE initialised the LD space embedding using a PCA with 2 components and used a maximum of 1000 iterations for the optimisation.

Figure 2b shows the results when using the full pipeline, taking samples of the digit 3 as source and digit 4 as the target. The reconstructed digits are a bit more blurry than the originals, but the samples do cover both the open version of the digit 4 as well as the closed top version. The t-SNE scatter plot in Figure 3a shows that the reconstructed samples are centred inside the distribution of the original samples. This means that less variance will be observed from the reconstructed samples. The dissimilarity measure for this experiment was 11.61.

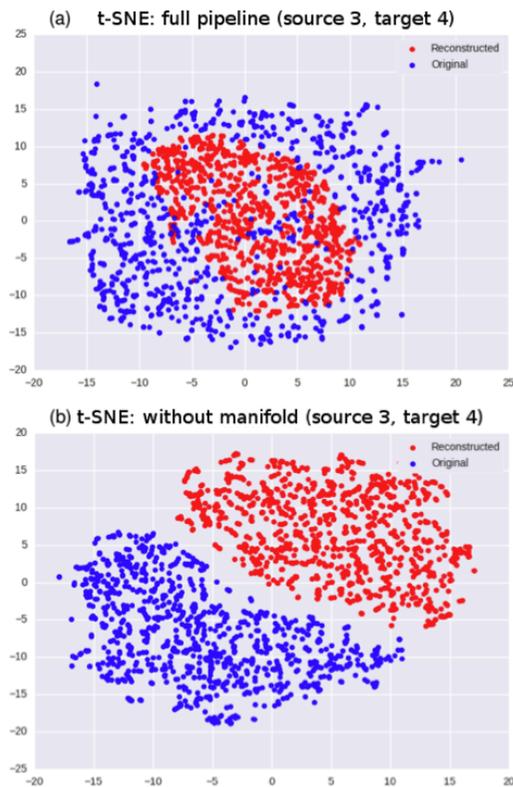


Figure 3. Examples of t-SNE for the experiment using 3 as source and 4 as target. (a) t-SNE associated to Figure 2b of the experiment with the full pipeline. (b) t-SNE associated to Figure 2c when the manifold alignment has been removed from the pipeline

Table 1 shows all dissimilarity values computed by our measure. For example 8 is easier to reconstruct from a 3 than a 4 which, from a human perspective, is explained by the loops and shape of 3 being similar to 8. Because of the lack of a ground truth, the table also includes the summarised results of an online survey we held to estimate human intuition about the similarity of written digits. In case two digits received approximately the same number of votes, both are listed. The survey was filled in by approximately 100 participants. One can observe that our approach agrees with the survey results for a number of cases, but surely not all.

4.1.1 Removing the DBNs

As discussed in the related work section, recent work by Bou Ammar et al. [2] constructs an inter-task mapping using only the manifold alignment part of our setup. In Figure 2d we depict the results obtained by attempting to transform a 3 digit into a 4 using the manifold alignment on the raw MNIST data. Comparing the results with those of Figure 2b shows the amount of noise generated by this approach.

4.1.2 Removing the manifold alignment

Figure 2c shows the experiment again, but removing the manifold alignment step, a setup briefly mentioned in [4]. Since both DBNs have the same architecture, the activations of the source’s last hidden layer can be mapped with an identity function to those of the target. It has to be noted that there is no theoretical foundation for the use of an identity mapping. The weights of the DBN are initialised randomly in order to break symmetry [14, p. 173] which leads to a random allocation of the high level features to the nodes, even when the DBN is trained multiple times on the same data. This experiment empirically demonstrates the necessity of using a method invariant under affine transformations. The results given by t-SNE in Figure 3b show that two distant distributions have been found. The KL-divergence for these distributions is 57.17, much higher than the value from the experiment using the full pipeline.

The results presented above demonstrate the necessity and advantages of each step in the pipeline. Each stage is required to solve a part of the problem described.

4.2 Pendigits

The pen-digits dataset [1] represents handwritten digits captured using a graphic tablet. Each instance is composed of 8 pairs of x, y coordinates taken along the path of the digit as depicted in Figure 4. The following experiment compares this sequential digit representation to the images of MNIST. While the domain of the two datasets is hand-written digits, the representation of the data is very different: first of all, the pen-digit samples have only 16 dimensions and represents coordinates over time, while MNIST uses 784 pixels intensities. While in this case, there seems to be a ground truth for the similarity measure to discover, without any background information about the two data-sets, they are very difficult, if not impossible for humans to match.

The pen-digit DBNs use two hidden layers with sizes of 60 units for the middle and 80 units for the last layer. The MNIST DBNs as well as the manifold alignment and t-SNE were configured as before.

Figure 4 shows the original pen-digit data for the digit 8 mapped on a two dimensional field, the reconstructed MNIST like images

Source\Target	0	1	2	3	4	5	6	7	8	9
0	$\ll 1$	49.67	12.38	3.47	10.15	12.70	9.03	15.91	6.75	11.18
1	14.19	$\ll 1$	10.96	10.27	10.99	12.50	10.03	7.16	7.74	13.20
2	13.98	46.41	$\ll 1$	12.36	11.78	8.68	14.35	16.64	4.94	9.83
3	11.24	39.71	10.90	$\ll 1$	11.61	8.48	13.14	16.91	3.58	8.08
4	13.58	32.64	11.14	7.80	$\ll 1$	11.40	12.71	10.12	4.40	9.04
5	13.39	56.18	9.50	5.45	10.80	$\ll 1$	15.84	14.43	5.42	15.16
6	11.78	41.37	10.27	6.21	8.22	11.73	$\ll 1$	15.66	5.23	13.61
7	14.38	47.07	8.61	7.19	6.07	12.23	12.22	$\ll 1$	9.69	8.34
8	14.14	44.05	9.30	6.95	12.56	10.44	16.73	17.46	$\ll 1$	15.98
9	12.99	41.39	8.37	5.07	9.30	13.56	13.81	9.36	6.29	$\ll 1$
2nd most similar	3	4	9	0	7	3	0	1	3	3
3rd most similar	6	3	7	9	6	2	1	9	4	4
human choices	6	7	3	8;2	9	6	0;5	1	3	4

Table 1. Similarity measure values obtained when comparing different MNIST digits. The bottom row represents the most similar digit according to a humans obtained with an online survey.

generated and the t-SNE plot of the resulting probability distributions. Again, it can be observed that although the reconstructions are a bit more blurry than the originals, a variety of recognisable 8 digits is produced.

We show the full dissimilarity matrix for all digits in Table 2. Surprisingly, most of the time, the pen-digits and their corresponding MNIST digit turn out to be the most similar by a huge margin when compared to others. For digit 3, there is a clear mismatch. For digits 1 and 2, the difference between the first and second closest digit is so small that it should be considered ambiguous. Given that t-SNE involves stochasticity in the mapping, two runs of the algorithm could give slightly different results [31] pointing to a different most similar digit. Nonetheless, even then, the matching digits remain among the topmost similar.

4.3 A Transfer Learning Scenario: Mapping Pendigit to MNIST

While not the primary aim of our data similarity measure, the inter-task mapping generated by our approach gives rise to a simple transfer learning scenario, as examples from one domain can be transformed into additional learning examples in another domain. To make this approach useful however, one would need to have both sufficient data to learn a deep belief network modelling the target data while at the same time, too few examples of the target set to be able to learn a good classification algorithm. Although these two seem to contradict each other, unsupervised learning of the DBN, followed by a fine tuning stage using a limited amount of data might be possible.

We tested the performance of such a transfer scenario using our automatically generated inter-task mapping. We performed a simple classification experiment that aims at identifying the digit 8 in a full set of digits, i.e. classifying 8's versus all other digits. For this we compared using a standard MNIST dataset with a 10% share of the dataset for each digit, with datasets that included an extra share of re-constructed 8's. We believe that this is a good analogy to how this technique could be used in practice, by generating a number of extra learning examples through the re-construction of a number of available source task examples. All experiments were performed using 10-fold cross validation, using the SMO support vector machine implementation in WEKA [13]. The results are displayed in Table 3.

They show a contribution made by the re-encoded learning examples. Unfortunately, we were unable to measure a correlation between the similarity measure and the amount of improvement in this setting. We assume the re-encodings of the examples are too close to measure a difference, enhanced by the fact that the re-encodings often appear at the center of the originals data distribution.

We would also like to emphasise that the experiment described in Table 3 is closely related to the problem described in domain adaptation [18, 9, 11]. Techniques used in domain adaptation attempt at using source data related to target data used to train a classifier in order to improve its accuracy [11]. The pipeline presented in this work could and should be investigated further in conjunction with the problems addressed by domain adaptation.

4.4 Comparing Markov Decision Processes

Next, we test our approach on three standard reinforcement learning benchmarks that have been previously compared [7]: (1) inverted pendulum, where the goal is to swing up and balance a pole with an underpowered motor, (2) cart pole, where the goal is to balance a pole hinged to a cart by pulling the cart back and forth and (3) mountain car, where the goal is to drive an under-powered car up a hill by building up momentum.

The datasets for these tasks were generated by uniformly sampling the environment for a state s , picking the action a that maximises a Q-function learned by SARSA and adding the state s' that the chosen action led to to build $\langle s, a, s' \rangle$ triplets that sample the domain's transition function. For each task 5000 samples were generated. Computing the similarities between the different domains resulted in Table 4 and match the ranking found in [4].

The table highlights the similarities between IP \rightarrow MC and CP \rightarrow MC and the fact that the similarity measure is not symmetric. MC does not seem to present a close similarity to IP nor CP. We believe this is caused by the relative simplicity of the learned policy of MC. The t-SNE space for the CP \rightarrow MC transfer case is shown in Figure 5. It can be observed that the samples from MC are divided into, what we believe are, separate pathways. The samples re-constructed from CP do not follow these pathways, but represent a decent spread over the example space. In the reverse transfer case (MC \rightarrow CP), this is not the case and all samples re-encoded from MC are grouped together instead.

Pen\MNIST	0	1	2	3	4	5	6	7	8	9
0	22.42	68.09	87.47	71.72	81.99	63.04	70.19	83.23	72.56	77.77
1	57.56	45.06	59.04	62.30	57.06	64.08	55.95	60.24	51.51	62.28
2	68.76	55.05	45.15	63.49	60.09	51.46	60.44	51.13	63.08	58.88
3	66.01	54.63	81.02	51.42	63.86	32.06	81.85	61.56	64.90	65.14
4	56.88	61.22	55.08	58.84	17.40	48.34	44.61	50.52	64.18	29.66
5	43.86	46.51	49.61	31.60	46.95	18.01	49.10	44.59	31.57	37.20
6	70.12	60.91	65.05	73.87	69.00	71.35	24.17	63.41	73.54	62.02
7	61.41	56.61	53.34	50.77	47.28	57.02	63.48	14.02	58.72	46.40
8	61.43	43.34	44.75	42.24	45.75	24.45	58.16	46.80	8.31	48.55
9	73.64	66.28	65.04	62.55	48.07	53.82	61.94	44.28	56.28	12.71
Most similar	0	8,1*	8,2*	5	4	5	6	7	8	9

Table 2. Table of dissimilarities between the pen-digits as source and the MNIST as target. Asterisks highlight ambiguities in the similarity between digits.

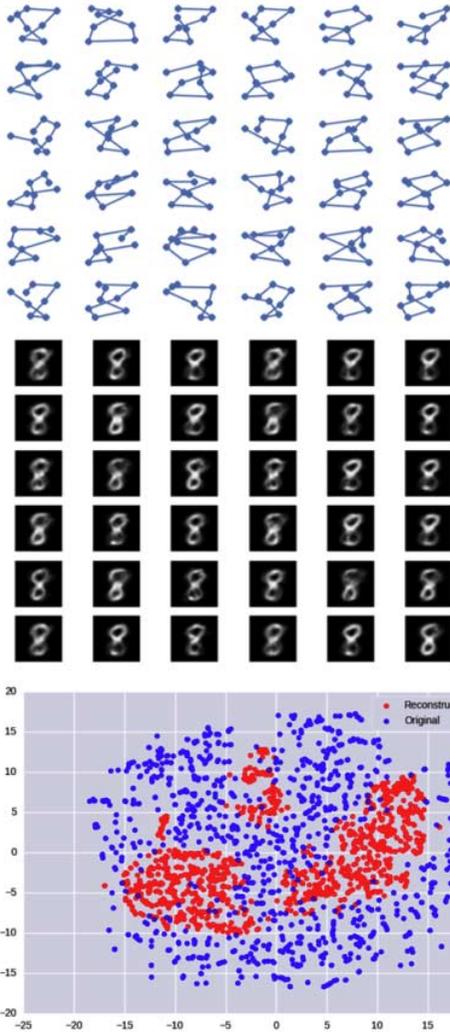


Figure 4. Pen-digit experiment. The top image shows samples from the pen-digit dataset. Only the dots are taken into consideration by the DBNs, the lines represent the sequential nature of the measurements. The middle image shows the corresponding MNIST reconstruction. The bottom image is the t-SNE plot.

Source	Dissimilarity Value	ROC area for class 8
none	n/a	0.848
pen digit 5	31.57	0.92
pen digit 6	73.54	0.92
pen digit 8	8.31	0.92

Table 3. Table of dissimilarities between the pen-digits as source and the MNIST as target. Asterisks highlight ambiguities in the similarity between digits.

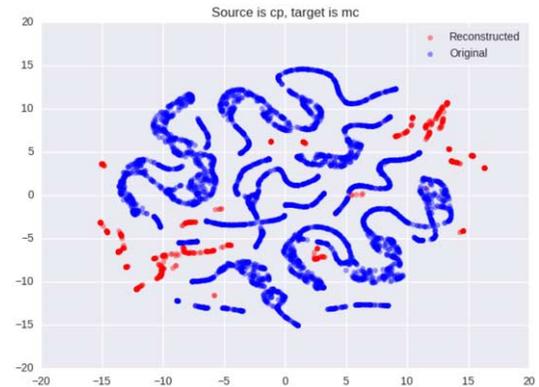


Figure 5. t-SNE space for transferring CP samples to MC.

5 Conclusions and Future Work

We presented a fully autonomous and data driven technique for computing (i) an inter-task mapping and (ii) dissimilarity measure for unrestricted data sets. While domain specific data driven similarity measures exists, for example in the image analysis field [26], it is our belief that we are the first to present a technique that is able to trans-

Source\Target	IP	CP	MC
IP	0.76	20.84	4.23
CP	18.35	5.19	7.05
MC	20.12	21.70	0.002

Table 4. The dissimilarity values between all reinforcement learning tasks.

late learning examples between highly dissimilar domains with such high correlation to human intuition. We showed in the experimental section that the presented technique can be applied to supervised, unsupervised and reinforcement learning tasks. We showed a number of experiments that both quantitatively and qualitatively illustrate the power of the presented technique as well as the necessity and contribution of each part of the involved pipeline.

In future work we would like to approach even more challenging tasks such as measuring the similarity of phonemes in speech and test the robustness by using, for example, affNIST (i.e. MNIST with affine transformations). We would also like to expand the technique to e.g. bigger images that require convolutional networks. This leads to additional complexity caused by the challenge in reversing convolution and pooling layers. However, related work exists for reversing these networks and using them to reconstruct data [22, 10] that could be useful to extend the pipeline presented.

We also plan to research transfer learning scenarios in which this technique could be used as a basis. While there are straightforward applications of the dissimilarity measure, such as source domain selection when multiple sources are available, the initial test using the inter-tasks mapping to do example transfer does not yet seem to lead to a useful approach. The examples transformed through a successful mapping function seem to converge to the center of the target data set, not necessarily leading to additional information about the classes boundaries. To make the approach successful, a way will have to be devised to generate a more diverse set of transferred examples.

Acknowledgments

We would like to thank Chang Wang for providing the source code of the manifold alignment and allowing us to reproduce the experiments in [32]. In addition, we would like to mention the projects Pylearn2 [12] and Scikit-learn [24] which have been used in the experiments. Finally, we would like to thank Haitham Bou Ammar for the interesting discussions.

REFERENCES

- [1] F. Alimoglu and E. Alpaydin, 'Combining multiple representations and classifiers for pen-based handwritten digit recognition', in *Document Analysis and Recognition, 1997., Proceedings of the Fourth International Conference on*, volume 2, pp. 637–640 vol.2, (Aug 1997).
- [2] Haitham Bou Ammar, Eric Eaton, Paul Ruvolo, and Matthew E. Taylor, 'Unsupervised Cross-Domain Transfer in Policy Gradient Reinforcement Learning via Manifold Alignment', in *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*, (January 2015). 27
- [3] Yoshua Bengio, Aaron Courville, and Pascal Vincent, 'Representation learning: A review and new perspectives', *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **35**(8), 1798–1828, (2013).
- [4] Haitham Bou-Ammar, *Automated Transfer in Reinforcement Learning*, Ph.D. dissertation, Maastricht University, 2013.
- [5] Haitham Bou-Ammar, Eric Eaton, Paul Ruvolo, and Matthew E. Taylor, 'Unsupervised cross-domain transfer in policy gradient reinforcement learning via manifold alignment', in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, January 25-30, 2015, Austin, Texas, USA., pp. 2504–2510, (2015).
- [6] Haitham Bou-Ammar, Eric Eaton, Matthew E. Taylor, Decebal Mocuana, Kurt Driessens, Gerhard Weiss, and Karl Tuyls, 'An automated measure of MDP similarity for transfer in reinforcement learning', in *Proceedings of the AAAI'14 Workshop on Machine Learning for Interactive Systems*, (July 2014).
- [7] Haitham Bou-Ammar, Decebal Constantin Mocuana, Matthew E. Taylor, Kurt Driessens, Karl Tuyls, and Gerhard Weiss, 'Automatically mapped transfer between reinforcement learning tasks via three-way restricted boltzmann machines', in *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part II*, pp. 449–464, (2013).
- [8] Haitham Bou-Ammar, Karl Tuyls, Matthew E. Taylor, Kurt Driessens, and Gerhard Weiss, 'Reinforcement learning transfer via sparse coding', in *International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2012, Valencia, Spain, June 4-8, 2012 (3 Volumes)*, pp. 383–390, (2012).
- [9] Barbara Caputo and Novi Patricia, 'Overview of the imageclef 2014 domain adaptation task', in *Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014.*, pp. 341–347, (2014).
- [10] A. Dosovitskiy, T.J. Springenberg, and T. Brox, 'Learning to generate chairs with convolutional neural networks.', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1538–1546, (2015).
- [11] Boqing Gong, Kristen Grauman, and Fei Sha, 'Learning kernels for unsupervised domain adaptation with applications to visual object recognition', *International Journal of Computer Vision*, **109**(1-2), 3–27, (2014).
- [12] Ian J. Goodfellow, David Warde-Farley, Pascal Lamblin, Vincent Dumoulin, Mehdi Mirza, Razvan Pascanu, James Bergstra, Frédéric Bastien, and Yoshua Bengio, 'Pylearn2: a machine learning research library', arXiv preprint arXiv:1308.4214, (2013).
- [13] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten, 'The weka data mining software: An update', *SIGKDD Explor. Newsl.*, **11**(1), 10–18, (November 2009).
- [14] Mohamad H. Hassoun, *Fundamentals of Artificial Neural Networks*, MIT Press, Cambridge, MA, USA, 1st edn., 1995.
- [15] Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury, 'Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups', *Signal Processing Magazine, IEEE*, **29**(6), 82–97, (November 2012).
- [16] Geoffrey Hinton, Simon Osindero, and Yee-Whye Teh, 'A fast learning algorithm for deep belief nets', *Neural computation*, **18**(7), 1527–1554, (2006).
- [17] Geoffrey E. Hinton, 'Training products of experts by minimizing contrastive divergence', *Neural Comput.*, **14**(8), 1771–1800, (August 2002).
- [18] Judy Hoffman, Erik Rodner, Jeff Donahue, Brian Kulis, and Kate Saenko, 'Asymmetric and category invariant feature transformations for domain adaptation', *International Journal of Computer Vision*, **109**(1-2), 28–41, (2014).
- [19] Yann Lecun, Sumit Chopra, Raia Hadsell, Ranzato marc aurelio, and fu-Jie Huang, 'A Tutorial on Energy-Based Learning', in *Predicting Structured Data*, eds., G. Bakir, T. Hofman, B. schölkopf, A. Smola, and B. Taskar. MIT Press, (2006).
- [20] Yann Lecun and Corinna Cortes, 'The MNIST database of handwritten digits'.
- [21] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y. Ng, 'Efficient sparse coding algorithms', in *In NIPS*, pp. 801–808. NIPS, (2007).
- [22] A. Mahendran and A. Vedaldi, 'Understanding deep image representations by inverting them.', in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 5188–5196, (June 2015).
- [23] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller, 'Playing atari with deep reinforcement learning', *CoRR, abs/1312.5602*, (2013).
- [24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, 'Scikit-learn: Machine learning in Python', *Journal of Machine Learning Research*, **12**, 2825–2830, (2011).
- [25] Jürgen Schmidhuber, 'Multi-column deep neural networks for image classification', in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), CVPR '12*, pp. 3642–3649, Washington, DC, USA, (2012). IEEE Computer Society.
- [26] Abhinav Shrivastava, Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros, 'Data-driven visual similarity for cross-domain image matching', in *Proceedings of the 2011 SIGGRAPH Asia Conference, SA '11*, pp. 154:1–154:10, New York, NY, USA, (2011). ACM.
- [27] Matthew E. Taylor, Gregory Kuhlmann, and Peter Stone, 'Autonomous transfer for reinforcement learning', in *Proceedings of the Seventh*

- International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 283–290, (May 2008).
- [28] Matthew E. Taylor and Peter Stone, ‘Transfer learning for reinforcement learning domains: A survey’, J. Mach. Learn. Res., **10**, 1633–1685, (December 2009).
- [29] Matthew E. Taylor, Shimon Whiteson, and Peter Stone, ‘Transfer via inter-task mappings in policy search reinforcement learning’, in Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 156–163, (May 2007).
- [30] L. Torrey and J. Shavlik, ‘Transfer learning’, Handbook of Research on Machine Learning Applications, IGI Global, **3**, 17–35, (2009).
- [31] Laurens Van der Maaten and Geoffrey Hinton, ‘Visualizing data using t-sne’, Journal of Machine Learning Research, **9**(2579-2605), 85, (2008).
- [32] Chang Wang and Sridhar Mahadevan, ‘Manifold alignment without correspondence’, in IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009, pp. 1273–1278, (2009).

On Stochastic Primal-Dual Hybrid Gradient Approach for Compositely Regularized Minimization

Linbo Qiao^{1,2} and Tianyi Lin³ and Yu-Gang Jiang⁴ and Fan Yang⁵ and Wei Liu⁶ and Xicheng Lu^{1,2}

Abstract. We consider a wide spectrum of regularized stochastic minimization problems, where the regularization term is composite with a linear function. Examples of this formulation include graph-guided regularized minimization, generalized Lasso and a class of ℓ_1 regularized problems. The computational challenge is that the closed-form solution of the proximal mapping associated with the regularization term is not available due to the imposed linear composition. Fortunately, the structure of the regularization term allows us to reformulate it as a new convex-concave saddle point problem which can be solved using the Primal-Dual Hybrid Gradient (PDHG) approach. However, this approach may be inefficient in realistic applications as computing the full gradient of the expected objective function could be very expensive when the number of input data samples is considerably large. To address this issue, we propose a Stochastic PDHG (SPDHG) algorithm with either uniformly or non-uniformly averaged iterates. Through uniformly averaged iterates, the SPDHG algorithm converges in expectation with $O(1/\sqrt{t})$ rate for general convex objectives and $O(\log(t)/t)$ rate for strongly convex objectives, respectively. While with non-uniformly averaged iterates, the SPDHG algorithm is expected to converge with $O(1/t)$ rate for strongly convex objectives. Numerical experiments on different genres of datasets demonstrate that our proposed algorithm outperforms other competing algorithms.

1 Introduction

In this paper, we are interested in solving a class of compositely regularized convex optimization problems:

$$\min_{x \in \mathcal{X}} \mathbb{E} [l(x, \xi)] + r(Fx), \quad (1)$$

where $x \in \mathbf{R}^d$, \mathcal{X} is a convex compact set with diameter D_x , $r : \mathbf{R}^l \rightarrow \mathbf{R}$ is a convex regularization function, and $F \in \mathbf{R}^{l \times d}$ is a penalty matrix (not necessarily diagonal) specifying the desired structured sparsity pattern in x . Furthermore, we denote $l(\cdot, \cdot) :$

$\mathbf{R}^d \times \Omega \rightarrow \mathbf{R}$ as a smooth convex function when applying a prediction rule x on a sample dataset $\{\xi_i = (a_i, b_i)\}$, and the corresponding expectation is denoted by $l(x) = \mathbb{E} [l(x, \xi)]$.

When $F = I$, the above formulation accommodates quite a few classic classification and regression models including Lasso obtained by setting $l(x, \xi_i) = \frac{1}{2} \|a_i^\top x - b_i\|^2$ and $r(x) = \lambda \|x\|_1$, and linear SVM obtained by letting $l(x, \xi_i) = \max(0, 1 - b_i \cdot a_i^\top x)$ and $r(x) = (\lambda/2) \|x\|_2^2$, where $\lambda > 0$ is a parameter. Moreover, the general structure of F enables problem (1) to cover more complicated problems arising from machine learning, such as graph-guided regularized minimization [6] and the generalized Lasso model [17].

However, this modeling power also comes with a challenge in computation. In particular, when F is not diagonal, it is very likely that the proximal mapping associated with $r(Fx)$ does not admit a closed-form expression. To cope with this difficulty, we could reformulate problem (1) as a convex-concave saddle point problem by exploiting some special structure of the regularization term, and then resort to the Primal-Dual Hybrid Gradient (PDHG) approach [23]. This approach has exhibited attractive numerical performance in image processing and image restoration applications [5, 3, 23, 19]. We refer readers to [4, 8, 9] to visit convergence properties of PDHG and its variants.

In practice, $\mathbb{E} [l(x, \xi)]$ is often replaced by its empirical average on a set of training samples. In this case, the computational complexity of calling the function value or the full gradient of $l(x)$ is proportional to the number of training samples, which is extremely huge for modern data-intensive applications. This could make PDHG and linearized PDHG suffer severely from the very poor scalability. Therefore, it is promising to propose a Stochastic variant of PDHG (SPDHG). Like many well-studied incremental or stochastic gradient methods [11, 14, 10, 1, 20], we draw a sample ξ^{k+1} in random and compute a noisy gradient $\nabla l(x^k, \xi^{k+1})$ at the k -th iteration with the current iterate x^k . As a result, the proposed SPDHG method enjoys the capability of dealing with very large-scale datasets.

Another way to handle the non-diagonal F and the expected objective function $\mathbb{E} [l(x, \xi)]$ is stochastic ADMM-like methods [12, 21, 7, 15, 18, 2, 22, 16] which aim for solving the following problem after introducing an additional variable z :

$$\min_{x \in \mathcal{X}, z = Fx} l(x) + r(z), \quad (2)$$

whose augmented Lagrangian function is given by $l(x) + r(z) + \lambda^\top (z - Fx) + \frac{\gamma}{2} \|z - Fx\|_2^2$. Comparing this function with the convex-concave problem (1) in Section 3, we can see that ADMM-like methods need to update one more vector variable than PDHG-type methods in every iteration. Thus, it can be expected that the per-iteration computational cost of ADMM-like methods is higher

¹ College of Computer, National University of Defense Technology, Changsha, China. email: {qiao.linbo, xclu}@nudt.edu.cn

² National Laboratory for Parallel and Distributed Processing, National University of Defense Technology, Changsha, China.

³ Research Center for Management Science and Information Analytics, Shanghai University of Finance and Economics, Shanghai, China, email: lin.tianyi@mail.shufe.edu.cn

⁴ School of Computer Science, Fudan University, Shanghai, China, email: ygj@fudan.edu.cn

⁵ Research Center for Management Science and Information Analytics, Shanghai University of Finance and Economics, Shanghai, China, email: fyang11@fudan.edu.cn

⁶ Tencent AI Lab, China, email: wliu@ee.columbia.edu

than our proposed algorithm SPDHG, as confirmed by the numerical experiments in Section 5.

Our contribution. To the best of our knowledge, we propose in this paper a new convex-concave formulation of problem (1), as well as the first stochastic variant of the PDHG algorithm for both uniformly and non-uniformly averaged iterates with achievable iteration complexities. In particular, for uniformly averaged iterates, the proposed algorithm converges in expectation with the rate of $O(1/\sqrt{t})$ and $O(\log(t)/t)$ for convex objectives and strongly convex objectives, respectively. It is worth mentioning that the $O(1/\sqrt{t})$ convergence rate is known to be best possible for first-order stochastic algorithms under general convex objective functions [1], which has also been established for the well-known stochastic ADMM (SADMM) [12]. Moreover, when optimizing strongly convex objectives, non-uniformly averaged iterates generated by SPDHG converge with $O(1/t)$ expected rate, which is the same as that of Optimal SADMM proposed in [2]. However, as mentioned before, the significant advantage gained by SPDHG beyond SADMM is the low per-iteration complexity. The effectiveness and efficiency of the proposed SPDHG algorithm are demonstrated by encouraging empirical evaluation in graph-guided regularized minimization tasks on several real-world datasets.

2 Related Work

Given the importance of problem (1), various stochastic optimization algorithms have been proposed to solve problem (1) or the more general form of problem (1), which can be written into

$$\begin{aligned} \min_{x \in \mathcal{X}, y \in \mathbf{R}^d} \quad & \mathbb{E} [l(x, \xi)] + r(y), \\ \text{s.t.} \quad & Ax + By = b. \end{aligned} \quad (3)$$

It is easy to verify that problem (1) is a special case of problem (3) when $A = F$, $B = -I$ and $b = 0$.

In solving problem (3), Wang and Banerjee [18] proposed an on-line ADMM that requires an easy proximal map of l . However, this is difficult for many loss functions such as logistic loss function. Ouyang et al. [12], Suzuki [15], Azadi and Sra [2], Gao et al. [7], and recently Zhao et al. [21] developed several stochastic variants of ADMM, which linearize l by using its noisy subgradient or gradient and add a varying proximal term. Furthermore, Zhong and Kwok [22] and Suzuki [16] respectively proposed a stochastic averaged gradient-based ADM and a stochastic dual coordinate ascent ADM, which can both obtain improved iteration complexities. However, these methods did not explore the structure of r and need to update one more vector variable than PDHG-type methods in every iteration. We will show in the experiments that our proposed SPDHG algorithm is far more efficient than these algorithms.

It is worth mentioning that another stochastic version of the primal-dual gradient approach was also analyzed in recent work [10]. However, their convex-concave formulation is different from ours, and their algorithm cannot be applied to solve problem (1). Regarding the iteration complexity, the proposed SPDHG algorithm has accomplished the best possible one for first-order stochastic algorithms under general convex objective functions [1]. A better convergence rate of $O(1/t^2 + 1/\sqrt{t})$ can be obtained by using Nesterov's acceleration technique in [11].

The most related algorithm to our proposed SPDHG algorithm is the SPDC algorithm [20] plus its adaptive variant [24]. Similar to our SPDHG algorithm, the SPDC algorithm is also a stochastic variant of the batch primal-dual algorithm developed by Chambolle and Pock

[4], which alternates between maximizing over a randomly chosen dual variable and minimizing over the primal variable. However, the SPDC algorithm does not explore the special structure of the regularization term (Assumption 3), and their convex-concave formulation is different from ours. This leads to the inability of the SPDC algorithm to solve problem (1). Specifically, [20] suggests to reformulate problem (1) as

$$\min_{x \in \mathcal{X}} \max_{y \in \mathbf{R}^d} \{ \mathbb{E} [\langle y, x \rangle - l^*(y, \xi)] + r(Fx) \}, \quad (4)$$

where $l^*(y, \xi) = \sup_{\alpha \in \mathbf{R}^d} \{ \langle \alpha, y \rangle - l(\alpha, \xi) \}$ is the convex conjugate of $l(x, \xi)$. Then the SPDC algorithm in solving problem (4) requires that the proximal map of l^* and $r(Fx)$ be easily computed, which is somewhat strong for a variety of application problems. In addition, the SPDC algorithm requires r to be strongly convex.

In contrast, our SPDHG algorithm only needs the smoothness of l and the convexity of r , and hence efficiently solves a wide range of graph-guided regularized optimization problems, which cannot be solved by the SPDC algorithm and its adaptive variant.

3 Preliminaries

3.1 Assumptions

We make the following assumptions (Assumption 1-4) regarding problem (1) throughout the paper:

Assumption 1 *The optimal set of problem (1) is nonempty.*

Assumption 2 *$l(\cdot)$ is continuously differentiable with Lipschitz continuous gradient. That is, there exists a constant $L > 0$ such that*

$$\|\nabla l(x_1) - \nabla l(x_2)\| \leq L \|x_1 - x_2\|, \forall x_1, x_2 \in \mathcal{X}.$$

Many formulations in machine learning satisfy Assumption 2. The following least square and logistic functions are two commonly used ones:

$$l(x, \xi_i) = \frac{1}{2} \|a_i^\top x - b_i\|^2 \text{ or } l(x, \xi_i) = \log \left(1 + \exp \left(-b_i \cdot a_i^\top x \right) \right),$$

where $\xi_i = (a_i, b_i)$ is a single data sample.

Assumption 3 *$r(x)$ is a continuous function which is possibly non-smooth, and it can be described as follows:*

$$r(x) = \max_{y \in \mathcal{Y}} \langle y, x \rangle,$$

where $\mathcal{Y} \in \mathbf{R}^d$ is a convex compact set with diameter D_y .

Note that Assumption 3 is reasonable for the learning problems with a norm regularization such as ℓ_1 -norm or nuclear norm:

$$\begin{aligned} \|x\|_1 &= \max \{ \langle y, x \rangle \mid \|y\|_\infty \leq 1 \}, \\ \|X\|_* &= \max \{ \langle Y, X \rangle \mid \|Y\|_2 \leq 1 \}. \end{aligned}$$

Assumption 4 *The function $l(x)$ is easy for gradient estimation. That is to say, any stochastic gradient estimation $\nabla l(\cdot, \xi)$ for $\nabla l(\cdot)$ at x satisfies*

$$\mathbb{E} [\nabla l(x, \xi)] = \nabla l(x),$$

and

$$\mathbb{E} [\|\nabla l(x, \xi) - \nabla l(x)\|^2] \leq \sigma^2.$$

Algorithm 1 SPDHG

Initialize: x^0 and y^0 .
for $k = 0, 1, 2, \dots$ **do**
 Choose one data sample ξ^{k+1} randomly.
 Update y^{k+1} according to Eq. (6).
 Update x^{k+1} according to Eq. (8).
end for
Output: $\bar{x}^t = \sum_{k=0}^t \alpha^{k+1} x^{k+1}$ and $\bar{y}^t = \sum_{k=0}^t \alpha^{k+1} y^{k+1}$.

where σ is some small number, and it is used in the proof of Lemma (7).

Assumption 5 $l(\cdot)$ is μ -strongly convex at x . In other words, there exists a constant $\mu > 0$ such that

$$l(y) - l(x) - (y - x)^\top \nabla l(x) \geq \frac{\mu}{2} \|y - x\|^2, \forall y \in \mathcal{X}.$$

We remark that Assumption 5 is optional, and it is only necessary for the theoretical analysis that can lead to a lower iteration complexity.

3.2 Convex-Concave Saddle Point Problem

According to Assumption 3, we are able to rewrite problem (1) as the following convex-concave saddle point problem:

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \{P(y, x) = l(x) + \langle y, Fx \rangle\}. \quad (5)$$

Remark 6 We remark here that the formulation (5) is greatly different from those used in [10, 20, 24], where they formulate problem (1) as another convex-concave saddle point problem (4) by using the convex conjugate of l . Therefore, their algorithms are limited to solving problem (1) due to the fact that the proximal mapping of $r(Fx)$ is difficult to compute.

This problem can be solved by Linearized PDHG (LPDHG) with the following iteration scheme:

$$y^{k+1} := \operatorname{argmax}_{y \in \mathcal{Y}} \left\{ P(y, x^k) - \frac{1}{2s} \|y - y^k\|^2 \right\}, \quad (6)$$

$$x^{k+1} := \left[x^k - \beta \left(\nabla l(x^k) + F^\top y^{k+1} \right) \right]_{\mathcal{X}}. \quad (7)$$

However, the above algorithm is inefficient since computing $\nabla l(x^k)$ in each iteration is very costly when the total number of samples n is large. This inspires us to propose a stochastic variant of PDHG, where only the noisy gradient $\nabla l(x^k, \xi^{k+1})$ is computed at each step.

4 Stochastic PDHG

In this section, we first propose our Stochastic Primal-Dual Hybrid Gradient (SPDHG) algorithms with either uniformly or non-uniformly averaged iterates for solving problem (5); and then provide the detailed convergence analysis of the proposed algorithms.

4.1 Algorithm

The SPDHG is presented in Algorithm 1, where we have addressed the following three important issues: how to apply the noisy gradient, how to choose the step-size, and how to determine the weights for the non-uniformly averaged iterates.

Stochastic Gradient: Our SPDHG algorithm shares some common features with the LPDHG algorithm. In fact, the y -subproblems for both algorithms are essentially the same, while for the x -subproblem we adopt the noisy gradient $\nabla l(x^k, \xi^{k+1})$ in SPDHG rather than the full gradient $\nabla l(x^k)$ in LPDHG, i.e.,

$$x^{k+1} := \left[x^k - \beta^{k+1} \left(\nabla l(x^k, \xi^{k+1}) + F^\top y^{k+1} \right) \right]_{\mathcal{X}}. \quad (8)$$

That is, in SPDHG we first maximize over the dual variable and then perform one-step stochastic gradient descent along the direction $-\nabla l(x^k, \xi^{k+1}) - F^\top y^{k+1}$ with step-size β^{k+1} .

The Step-Size β^{k+1} : The choice of the step-size β^{k+1} varies with respect to the different conditions satisfied by the objective function l . Different step-size rules also lead to different convergence rates. Note that a sequence of vanishing step-sizes is necessary since we do not adopt any technique of variance reduction in the SPDHG algorithm.

Non-uniformly Averaged Iterates: It was shown in [2] that the non-uniformly averaged iterates generated by stochastic algorithms converge with fewer iterations. Inspired by their work, through non-uniformly averaging the iterates of the SPDHG algorithm and adopting a slightly modified step-size, we manage to establish an accelerated convergence rate of $O(1/t)$ in expectation.

For the convenience of readers, we summarize the convergence properties with respect to different settings in Table 1.

Table 1: Convergence properties.

l	General Convex	Strongly Convex	
β^{k+1}	$\frac{1}{\sqrt{k+1}+L}$	$\frac{1}{\mu(k+1)+L}$	$\frac{2}{\mu(k+2)+2L}$
α^{k+1}	$\frac{1}{t+1}$	$\frac{2(k+1)}{(t+1)(t+2)}$	
Rate	$O\left(\frac{1}{\sqrt{t}}\right)$	$O\left(\frac{\log(t)}{t}\right)$	$O\left(\frac{1}{t}\right)$

4.2 Convergence of uniformly averaging under convex objective functions

In this subsection, we analyze the convergence property of the SPDHG algorithm with uniformly averaged iterates for general convex objective functions.

Lemma 7 Let (y^{k+1}, x^{k+1}) be generated by Algorithm 1, and β^{k+1} and α^{k+1} be shown in Table 1. For any optimal solution (y^*, x^*) of problem (5), it holds that

$$\begin{aligned} 0 &\geq \mathbb{E} \left[P(y^{k+1}, x^*) - P(y^*, x^{k+1}) \right] \quad (9) \\ &\geq \frac{\sqrt{k+1}+L}{2} \left(\mathbb{E} \|x^* - x^{k+1}\|^2 - \mathbb{E} \|x^* - x^k\|^2 \right) \\ &\quad + \frac{1}{2s} \left(\mathbb{E} \|y^* - y^{k+1}\|^2 - \mathbb{E} \|y^* - y^k\|^2 \right) \\ &\quad - \frac{\lambda_{\max}(F^\top F) D_y^2 + \sigma^2}{\sqrt{k+1}}. \end{aligned}$$

Proof. For any optimal solution (y^*, x^*) of problem (5), the first-order optimality conditions for Eq. (6) and Eq. (8) are

$$\begin{aligned} 0 &\leq (y^* - y^{k+1})^\top \left(-Fx^k + \frac{1}{s} (y^{k+1} - y^k) \right) \\ 0 &\leq (x^* - x^{k+1})^\top \left[x^{k+1} - x^k + \beta^{k+1} \left(\nabla l(x^k, \xi^{k+1}) + F^\top y^{k+1} \right) \right], \end{aligned}$$

which implies that

$$\begin{aligned}
& \left(x^* - x^{k+1}\right)^\top \nabla l(x^k, \xi^{k+1}) - \left(y^* - y^{k+1}\right)^\top F x^{k+1} \\
& + \left(x^* - x^{k+1}\right)^\top F^\top y^{k+1} \\
\geq & \frac{1}{2\beta^{k+1}} \left(\left\|x^* - x^{k+1}\right\|^2 - \left\|x^* - x^k\right\|^2 + \left\|x^{k+1} - x^k\right\|^2 \right) \\
& + \frac{1}{2s} \left(\left\|y^* - y^{k+1}\right\|^2 - \left\|y^* - y^k\right\|^2 \right) \\
& + \left(y^* - y^{k+1}\right)^\top \left(F x^k - F x^{k+1}\right). \tag{10}
\end{aligned}$$

Furthermore, for any $\gamma > 0$ we have

$$\begin{aligned}
& \left(y^* - y^{k+1}\right)^\top \left(F x^k - F x^{k+1}\right) \\
\geq & -\frac{\lambda_{\max}(F^\top F) D_y^2}{\gamma} - \frac{\gamma}{4} \left\|x^k - x^{k+1}\right\|^2, \tag{11}
\end{aligned}$$

and

$$\begin{aligned}
& \left(x^* - x^{k+1}\right)^\top \nabla l(x^k, \xi^{k+1}) \\
= & \left(x^* - x^{k+1}\right)^\top \nabla l(x^k) + \left(x^* - x^{k+1}\right)^\top \delta^{k+1} \\
\leq & l(x^*) - l(x^{k+1}) + \frac{L}{2} \left\|x^k - x^{k+1}\right\|^2 + \left(x^* - x^{k+1}\right)^\top \delta^{k+1} \\
\leq & l(x^*) - l(x^{k+1}) + \left(x^* - x^k\right)^\top \delta^{k+1} \\
& + \frac{L + \sqrt{k+1}/2}{2} \left\|x^k - x^{k+1}\right\|^2 + \frac{1}{\sqrt{k+1}} \left\|\delta^{k+1}\right\|^2,
\end{aligned}$$

where the first inequality holds due to Lemma 6.2 [7], and $\delta^{k+1} = \nabla l(x^k, \xi^{k+1}) - \nabla l(x^k)$. Then by letting $\gamma = \sqrt{k+1}$ in Eq. (11), we obtain

$$\begin{aligned}
& l(x^*) - l(x^{k+1}) + \left(\begin{array}{c} y^* - y^{k+1} \\ x^* - x^{k+1} \end{array} \right)^\top \left(\begin{array}{c} -F x^{k+1} \\ F^\top y^{k+1} \end{array} \right) \\
\geq & \frac{1}{2\beta^{k+1}} \left(\left\|x^* - x^{k+1}\right\|^2 - \left\|x^* - x^k\right\|^2 \right) \\
& + \frac{1}{2s} \left(\left\|y^* - y^{k+1}\right\|^2 - \left\|y^* - y^k\right\|^2 \right) - \frac{\lambda_{\max}(F^\top F) D_y^2}{\sqrt{k+1}} \\
& - \left(x^* - x^k\right)^\top \delta^{k+1} - \frac{\left\|\delta^{k+1}\right\|^2}{\sqrt{k+1}}.
\end{aligned}$$

Since x^k is independent of ξ^{k+1} , we take the expectation on both sides of the above inequality conditioning on x^k, y^k , and conclude that

$$\begin{aligned}
& \mathbb{E} \left[P(y^{k+1}, x^*) - P(y^*, x^{k+1}) \right] \\
\geq & \frac{1}{2\beta^{k+1}} \left(\mathbb{E} \left\|x^* - x^{k+1}\right\|^2 - \left\|x^* - x^k\right\|^2 \right) - \frac{\mathbb{E} \left\|\delta^{k+1}\right\|^2}{\sqrt{k+1}} \\
& + \frac{1}{2s} \left(\mathbb{E} \left\|y^* - y^{k+1}\right\|^2 - \left\|y^* - y^k\right\|^2 \right) - \frac{\lambda_{\max}(F^\top F) D_y^2}{\sqrt{k+1}}.
\end{aligned}$$

Finally, Eq. (9) follows from the above inequality and Assumption 4. \square

We present the main result for uniformly averaged iterates under general convex objective functions in the following theorem.

Theorem 8 Denote β^{k+1} , α^{k+1} and (\bar{y}^t, \bar{x}^t) as shown in Table 1. For any optimal solution (y^*, x^*) of problem (5), (\bar{y}^t, \bar{x}^t) converges to (y^*, x^*) with $O(1/\sqrt{t})$ rate in expectation.

Proof. Because $(y^k, x^k) \in \mathcal{Y} \times \mathcal{X}$, it holds true that $(\bar{y}^t, \bar{x}^t) \in \mathcal{Y} \times \mathcal{X}$ for all $t \geq 0$. By invoking the convexity of function $l(\cdot)$ and using Eq. (10), we have

$$\begin{aligned}
& \mathbb{E} \left[P(\bar{y}^t, x^*) - P(y^*, \bar{x}^t) \right] \\
\geq & \frac{1}{t+1} \sum_{k=0}^t \left[\frac{1}{2s} \left(\mathbb{E} \left\|y^* - y^{k+1}\right\|^2 - \mathbb{E} \left\|y^* - y^k\right\|^2 \right) \right. \\
& + \frac{\sqrt{k+1} + L}{2} \left(\mathbb{E} \left\|x^* - x^{k+1}\right\|^2 - \mathbb{E} \left\|x^* - x^k\right\|^2 \right) \\
& \left. - \frac{\lambda_{\max}(F^\top F) D_y^2}{\sqrt{k+1}} - \frac{\sigma^2}{\sqrt{k+1}} \right] \\
\geq & -\frac{D_y^2}{2s(t+1)} - \frac{L D_x^2}{2(t+1)} - \frac{D_x^2 + 2\lambda_{\max}(F^\top F) D_y^2 + 2\sigma^2}{\sqrt{t+1}}.
\end{aligned}$$

This together with the fact that $\mathbb{E} \left[P(\bar{y}^t, x^*) - P(y^*, \bar{x}^t) \right] \leq 0$ implies the conclusion in Theorem 8. \square

4.3 Convergence of uniformly averaging under strongly convex objective functions

In this subsection, we analyze the convergence property of the SPDHG algorithm with uniformly averaged iterates for strongly convex objective functions.

Lemma 9 Let (y^{k+1}, x^{k+1}) be generated by Algorithm 1, and β^{k+1} and α^{k+1} be shown in Table 1. For any optimal solution (y^*, x^*) of problem (5), it holds that

$$\begin{aligned}
0 & \geq \mathbb{E} \left[P(y^{k+1}, x^*) - P(y^*, x^{k+1}) \right] \tag{12} \\
& \geq \frac{\mu(k+1) + L}{2} \mathbb{E} \left\|x^* - x^{k+1}\right\|^2 + \frac{1}{2s} \mathbb{E} \left\|y^* - y^{k+1}\right\|^2 \\
& \quad - \frac{\mu k + L}{2} \mathbb{E} \left\|x^* - x^k\right\|^2 - \frac{1}{2s} \mathbb{E} \left\|y^* - y^k\right\|^2 \\
& \quad - \frac{\lambda_{\max}(F^\top F) D_y^2 + \sigma^2}{\mu(k+1)}.
\end{aligned}$$

Proof. By using the same argument as Lemma 7 and the strongly convexity of l , we have

$$\begin{aligned}
& \left(x^* - x^{k+1}\right)^\top \nabla l(x^k, \xi^{k+1}) \tag{13} \\
\leq & l(x^*) - l(x^k) - \frac{\mu}{2} \left\|x^* - x^k\right\|^2 + l(x^k) - l(x^{k+1}) \\
& + \frac{L}{2} \left\|x^k - x^{k+1}\right\|^2 + \left(x^* - x^{k+1}\right)^\top \delta^{k+1} \\
\leq & l(x^*) - l(x^{k+1}) + \left(x^* - x^k\right)^\top \delta^{k+1} - \frac{\mu}{2} \left\|x^* - x^k\right\|^2 \\
& + \frac{L}{2} \left\|x^k - x^{k+1}\right\|^2 + \frac{\kappa}{4} \left\|x^k - x^{k+1}\right\|^2 + \frac{1}{\kappa} \left\|\delta^{k+1}\right\|^2.
\end{aligned}$$

Substituting Eq. (11) with $\gamma = \mu(k+1)$ and Eq. (13) with $\kappa =$

$\mu(k+1)$ into Eq. (10) yields that

$$\begin{aligned} & l(x^*) - l(x^{k+1}) + \begin{pmatrix} y^* - y^{k+1} \\ x^* - x^{k+1} \end{pmatrix}^\top \begin{pmatrix} -Fx^{k+1} \\ F^\top y^{k+1} \end{pmatrix} \\ & \geq \frac{1}{2s} \|y^* - y^{k+1}\|^2 - \frac{1}{2s} \|y^* - y^k\|^2 - \frac{\|\delta^{k+1}\|^2}{\mu(k+1)} \\ & \quad + \frac{\mu(k+1) + L}{2} \|x^* - x^{k+1}\|^2 - \frac{\mu k + L}{2} \|x^* - x^k\|^2 \\ & \quad + \left(\frac{1}{2\beta^{k+1}} - \frac{L + \mu(k+1)}{2} \right) \|x^k - x^{k+1}\|^2 \\ & \quad - \frac{\lambda_{\max}(F^\top F)D_y^2}{\mu(k+1)} - (x^* - x^k)^\top \delta^{k+1}. \end{aligned}$$

Then we obtain Eq. (12) as the same as that in Lemma 7. \square

We present the main result in the following theorem when the objective function is further assumed to be strongly convex.

Theorem 10 Denote β^{k+1} , α^{k+1} and (\bar{y}^t, \bar{x}^t) as shown in Table 1. For any optimal solution (y^*, x^*) of problem (5), (\bar{y}^t, \bar{x}^t) converges to (y^*, x^*) with $O(\log(t)/t)$ rate in expectation.

Proof. Because $(y^k, x^k) \in \mathcal{Y} \times \mathcal{X}$, it holds that $(\bar{y}^t, \bar{x}^t) \in \mathcal{Y} \times \mathcal{X}$ for all $t \geq 0$. By invoking the convexity of function $l(\cdot)$ and using Eq. (12), we have

$$\begin{aligned} & \mathbb{E} [P(\bar{y}^t, x^*) - P(y^*, \bar{x}^t)] \\ & \geq \frac{1}{t+1} \sum_{k=0}^t \left[\frac{1}{2s} \left(\mathbb{E} \|y^* - y^{k+1}\|^2 - \mathbb{E} \|y^* - y^k\|^2 \right) \right. \\ & \quad + \frac{\mu(k+1) + L}{2} \|x^* - x^{k+1}\|^2 - \frac{\mu k + L}{2} \|x^* - x^k\|^2 \\ & \quad \left. - \frac{\lambda_{\max}(F^\top F)D_y^2 + \sigma^2}{\mu(k+1)} \right] \\ & \geq -\frac{D_y^2}{2s(t+1)} - \frac{LD_x^2}{2(t+1)} - \frac{(\lambda_{\max}(F^\top F)D_y^2 + \sigma^2) \log(t+1)}{\mu(t+1)}. \end{aligned}$$

This together with the fact that $\mathbb{E} [P(\bar{y}^t, x^*) - P(y^*, \bar{x}^t)] \leq 0$ implies the conclusion in Theorem 10. \square

4.4 Convergence of non-uniformly averaging under strongly convex objective functions

In this subsection, we analyze the convergence property of the SPDHG algorithm with non-uniformly averaged iterates for strongly convex objective functions.

Lemma 11 Let (y^{k+1}, x^{k+1}) be generated by Algorithm 1, and β^{k+1} and α^{k+1} be shown in Table 1. For any optimal solution (y^*, x^*) of problem (5), it holds that

$$\begin{aligned} 0 & \geq \mathbb{E} [P(y^{k+1}, x^*) - P(y^*, x^{k+1})] \quad (14) \\ & \geq \frac{\mu(k+2) + 2L}{4} \mathbb{E} \|x^* - x^{k+1}\|^2 + \frac{1}{2s} \mathbb{E} \|y^* - y^{k+1}\|^2 \\ & \quad - \frac{\mu k + 2L}{4} \mathbb{E} \|x^* - x^k\|^2 - \frac{1}{2s} \mathbb{E} \|y^* - y^k\|^2 \\ & \quad - \frac{2\lambda_{\max}(F^\top F)D_y^2 + 2\sigma^2}{\mu(k+1)}. \end{aligned}$$

Proof. By substituting Eq. (11) with $\gamma = \frac{\mu(k+1)}{2}$ and Eq. (13) with $\kappa = \frac{\mu(k+1)}{2}$ into Eq. (10), we have

$$\begin{aligned} & \begin{pmatrix} y^* - y^{k+1} \\ x^* - x^{k+1} \end{pmatrix}^\top \begin{pmatrix} Fx^k - Fx^{k+1} \\ F^\top y^{k+1} \end{pmatrix} \\ & \geq -\frac{2\lambda_{\max}(F^\top F)D_y^2}{\mu(k+1)} - \frac{\mu(k+1)}{8} \|x^k - x^{k+1}\|^2, \end{aligned}$$

and

$$\begin{aligned} & \begin{pmatrix} x^* - x^{k+1} \end{pmatrix}^\top \nabla l(x^k, \xi^{k+1}) \\ & \leq l(x^*) - l(x^{k+1}) + \begin{pmatrix} x^* - x^k \end{pmatrix}^\top \delta^{k+1} - \frac{\mu}{2} \|x^* - x^k\|^2 \\ & \quad + \frac{L}{2} \|x^k - x^{k+1}\|^2 + \frac{\mu(k+1)}{8} \|x^k - x^{k+1}\|^2 \\ & \quad + \frac{2}{\mu(k+1)} \|\delta^{k+1}\|^2. \end{aligned}$$

Then we plug the above two inequalities into Eq. (10), and then follow the same argument as Lemma 9 to obtain the desired inequality in Eq. (14). \square

We present the main result for non-uniformly averaged iterates under strongly convex functions in the following theorem.

Theorem 12 Denote β^{k+1} , α^{k+1} and (\bar{y}^t, \bar{x}^t) as shown in Table 1. For any optimal solution (y^*, x^*) of problem (5), (\bar{y}^t, \bar{x}^t) converges to (y^*, x^*) with $O(1/t)$ rate in expectation.

Proof. We have $(\bar{y}^t, \bar{x}^t) \in \mathcal{Y} \times \mathcal{X}$ for all $t \geq 0$. By invoking the convexity of function $l(\cdot)$ and using Eq. (14), we have

$$\begin{aligned} & \mathbb{E} [P(\bar{y}^t, x^*) - P(y^*, \bar{x}^t)] \\ & \geq \frac{2}{(t+1)(t+2)} \sum_{k=0}^t (k+1) \left[-\frac{2\lambda_{\max}(F^\top F)D_y^2 + 2\sigma^2}{\mu(k+1)} \right. \\ & \quad + \frac{\mu(k+2) + 2L}{4} \|x^* - x^{k+1}\|^2 - \frac{\mu k + 2L}{4} \|x^* - x^k\|^2 \\ & \quad \left. + \frac{1}{2s} \left(\mathbb{E} \|y^* - y^{k+1}\|^2 - \mathbb{E} \|y^* - y^k\|^2 \right) \right] \\ & \geq -\frac{D_y^2}{s(t+2)} - \frac{LD_x^2}{t+2} - \frac{4\lambda_{\max}(F^\top F)D_y^2 + 4\sigma^2}{\mu(t+2)} \\ & \quad + \frac{\mu}{2(t+1)(t+2)} \sum_{k=0}^t \left[(k+2)(k+1) \|x^* - x^{k+1}\|^2 \right. \\ & \quad \left. - (k+1)k \|x^* - x^k\|^2 \right]. \end{aligned}$$

Therefore, we conclude that

$$\begin{aligned} 0 & \geq \mathbb{E} [P(\bar{y}^t, x^*) - P(y^*, \bar{x}^t)] \\ & \geq -\frac{D_y^2}{s(t+2)} - \frac{LD_x^2}{t+2} - \frac{4\lambda_{\max}(F^\top F)D_y^2 + 4\sigma^2}{\mu(t+2)}, \end{aligned}$$

which implies the conclusion in Theorem 12. \square

5 Experiments

We conduct experiments by evaluating two models: graph-guided logistic regression (GGLR) (15) and graph-guided regularized logistic regression (GGRLR) (16) [22],

$$\min_{x \in \mathcal{X}} l(x) + \lambda \|Fx\|_1 \quad (15)$$

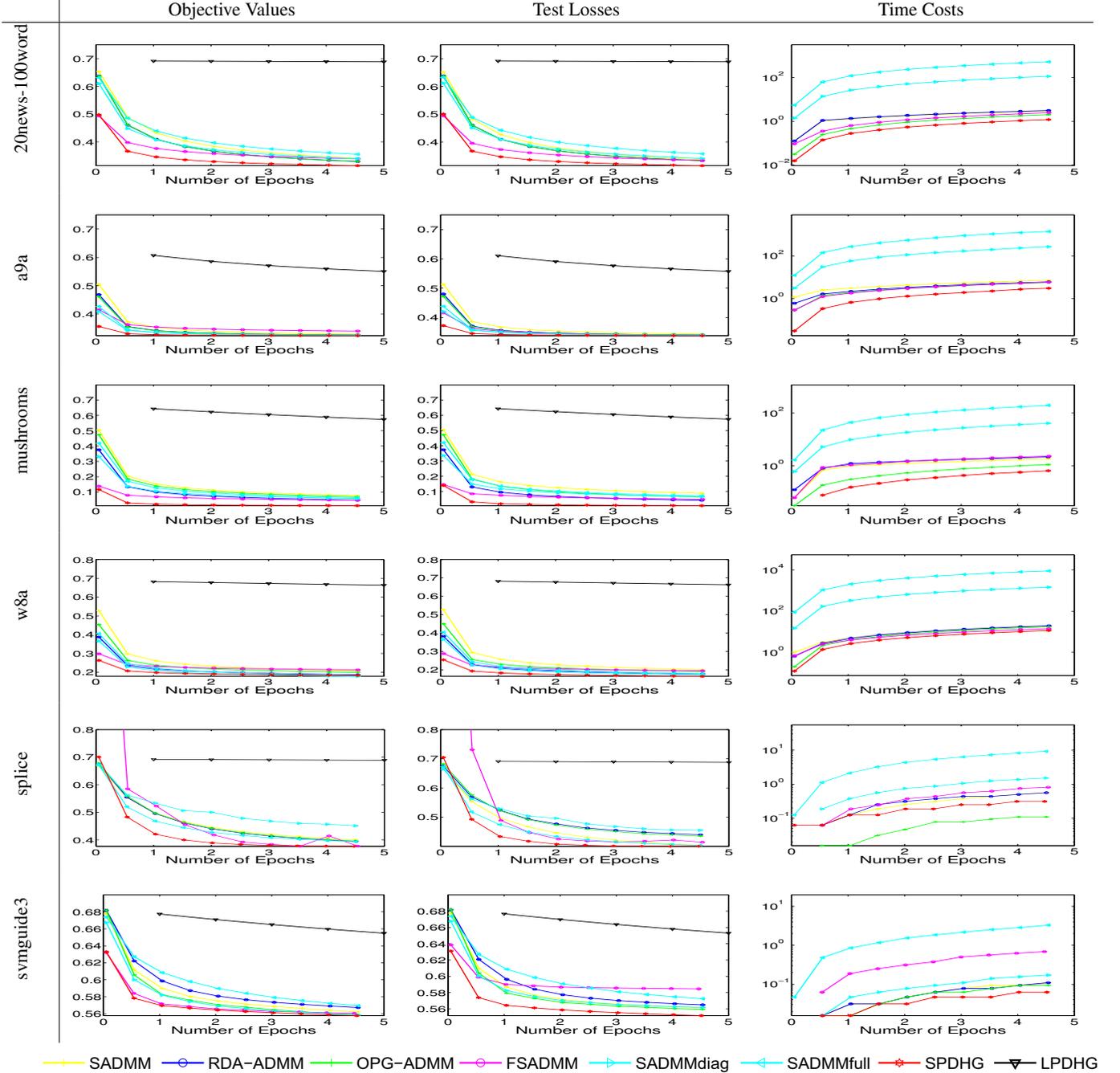


Figure 1: Comparison of SPDHG with STOC-ADMM (SADMM), RDA-ADMM, OPG-ADMM, Fast-SADMM (FSADMM), Ada-SADMMdiag, Ada-SADMMfull and LPDHG on **Graph-Guided Logistic Regression** Task. Epoch for the horizontal axis is the number of iterations divided by the dataset size. **Left Panels:** Averaged objective values. **Middle Panels:** Averaged test losses. **Right Panels:** Averaged time costs (in seconds).

and

$$\min_{x \in \mathcal{X}} l(x) + \frac{\gamma}{2} \|x\|_2^2 + \lambda \|Fx\|_1. \quad (16)$$

Here $l(x) = \frac{1}{N} \left[\sum_{i=1}^N l(x, \xi_i) \right]$ is empirical average of $l(x, \xi_i)$ on a set of samples, and $l(x, \xi_i)$ is logistic function $\log(1 + \exp(-b_i \cdot a_i^\top x))$, where $\xi_i = (a_i, b_i)$. λ is the regularization parameter. F is a penalty matrix promoting the desired sparse structure of x , which is generated by sparse inverse covariance

selection [13]. To proceed, we reformulate problems (15) and (16) into the convex-concave saddle point problem (5) and apply our proposed SPDHG algorithm. On the other hand, we can reformulate problems (15) and (16) into problem (2) by introducing an additional variable $z = Fx$ and then apply stochastic ADMM algorithms.

In the experiments, we compare our SPDHG algorithm with the LPDHG algorithm, and six existing stochastic ADMM algo-

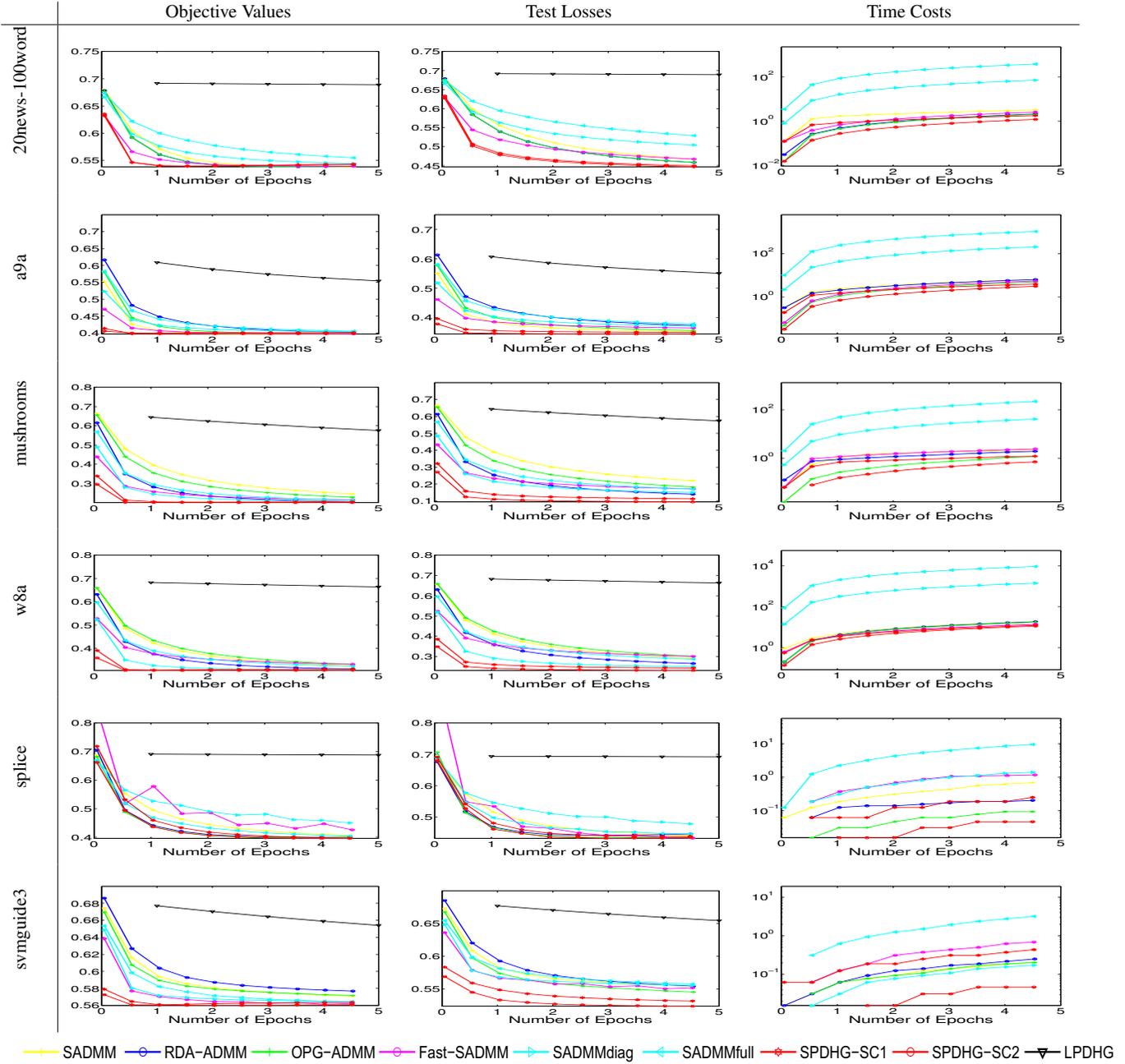


Figure 2: Comparison of SPDHG-SC1 (Uniformly Averaged) and SPDHG-SC2 (Non-Uniformly Averaged) with STOC-ADMM (SADMM), RDA-ADMM, OPG-ADMM, Fast-SADMM (FSADMM), Ada-SADMMdiag, Ada-SADMMfull and LPDHG on **Graph-Guided Regularized Logistic Regression** Task. Epoch for the horizontal axis is the number of iterations divided by the dataset size. **Left Panels:** Averaged objective values. **Middle Panels:** Averaged test losses. **Right Panels:** Averaged time costs (in seconds).

gorithms⁷: SADMM [12], OPG-ADMM [15], RDA-ADMM [15], FSADMM[22], and two variants of adaptive SADMM (*i.e.*, SADMMdiag and SADMMfull) [21]. We do not include online ADMM [18] and SDCA-ADMM [16] since [15] has shown that RDA-ADMM performs better than online ADMM while [20] has shown that the

performance of FSADMM is comparable to that of SDCA-ADMM. Finally, SPDC and Adaptive SPDC are excluded from the experiments since they cannot solve problem (15) and problem (16), as clarified in Section 2.

The experiments are conducted on six binary classification datasets: *20news*⁸, *a9a*, *mushrooms*, *w8a*, *splice* and *svmguide3*⁹. On

⁷ We use the code of SADMM, OPG-ADMM, RDA-ADMM and FSADMM provided by the authors while implementing two variants of adaptive SADMM according to [21].

⁸ www.cs.nyu.edu/roweis/data.html.

⁹ <https://www.csie.ntu.edu.tw/~cjlin/libsvm/>.

Table 2: Statistics of datasets.

dataset	number of samples	dimensionality
<i>svmguid3</i>	1243	21
<i>splice</i>	1000	60
<i>a9a</i>	32,561	123
<i>w8a</i>	64,700	300
<i>20news</i>	16,242	100
<i>mushrooms</i>	8,124	112

each dataset, we use 80% samples for training and 20% for testing, and calculate the lipschitz constant L as its classical upper bound $\hat{L} = 0.25 \max_{1 \leq i \leq n} \|a_i\|^2$. The regularization parameters are set to $\lambda = 10^{-5}$ and $\gamma = 10^{-2}$. To reduce statistical variability, experimental results are averaged over 10 repetitions. We set the parameters of SPDHG exactly following our theory while using cross validation to select the parameters of the other algorithms. Additionally, we use the metrics in [22] to compare our algorithm with the other algorithms, including objective values, test losses and time costs to compare our algorithm with the other. The “test loss” means the value of the empirically averaged loss evaluated on a test dataset, while the “objective value” means the sum of the empirically averaged loss and regularized terms evaluated on a training dataset, and the “time cost” means the computational time consumption of each algorithm. Specifically, we use test losses (*i.e.*, $l(x)$) on test datasets, objective values (*i.e.*, $l(x) + \lambda \|Fx\|_1$ on the GGLR task and $l(x) + \frac{\gamma}{2} \|x\|_2^2 + \lambda \|Fx\|_1$ on the GGRLR task) on training datasets, and computational time costs on training datasets.

Figure 1 shows the objective values, test losses and time costs as the function of the number of epochs on the GGLR task, where the objective function is convex but not necessarily strongly convex. We observe that our algorithm SPDHG mostly achieves the best performance, surpassing six stochastic ADMM algorithms, all of which outperform LPDHG by a significant margin. FSADMM sometimes achieves better solutions but consumes much more computational time than SPDHG. In fact, our algorithm requires the least iterations and computational time among all the evaluated algorithms. Therefore, the performance of our algorithm SPDHG on four datasets is most stable and effective among all algorithms.

We further compare our algorithm against the other algorithms on the GGRLR task, where the objective function is strongly convex. The experimental results are displayed in Figure 2. Our algorithm still outperforms the other algorithms consistently, which supports our analysis in the previous sections. We also find that the difference between uniformly averaging and non-uniformly averaging shown in Figure 2 is not significant. One reason is that our algorithm converges within only one or two effective epochs. In this case, non-uniformly averaging will not exhibit its advantage.

6 Conclusions

In this paper, we proposed a novel convex-concave saddle point formulation to resolve problem (1) as well as the first stochastic variant of the PDHG algorithm, namely SPDHG. The new algorithm can tackle a variety of real-world problems which cannot be solved by the existing stochastic primal-dual algorithms proposed in [10, 20, 24]. We further proved that the proposed SPDHG algorithm converges in expectation with the rate of $O(1/\sqrt{t})$ and $O(\log(t)/t)$ for general and strongly convex objectives, respectively. By averaging iterates non-uniformly, the SPDHG algorithm converges in expectation with the rate of $O(1/t)$ for strongly convex objectives.

The SPDHG algorithm is well-suited for addressing compositely regularized minimization problems when the penalty matrix F is non-diagonal. The experiments in performing graph-guided logistic regression and graph-guided regularized logistic regression tasks

demonstrated that our SPDHG algorithm outperforms the other competing stochastic algorithms.

Acknowledgments

The work was partially supported by the National Natural Science Foundation of China under Grant No. 61303264.

REFERENCES

- [1] A. Agarwal, P. L. Bartlett, P. Ravikumar, and M. J. Wainwright, ‘Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization’, *IEEE Transactions on Information Theory*, **58**(5), 32–35, (2012).
- [2] S. Azadi and S. Sra, ‘Towards an optimal stochastic alternating direction method of multipliers’, in *ICML*, pp. 620–628, (2014).
- [3] S. Bonettini and V. Ruggiero, ‘On the convergence of primal–dual hybrid gradient algorithms for total variation image restoration’, *Journal of Mathematical Imaging and Vision*, **44**(3), 236–253, (2012).
- [4] A. Chambolle and T. Pock, ‘A first-order primal-dual algorithm for convex problems with applications to imaging’, *Journal of Mathematical Imaging and Vision*, **40**(1), 120–145, (2011).
- [5] E. Esser, X. Zhang, and T. F. Chan, ‘A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science’, *SIAM Journal on Imaging Sciences*, **3**(4), 1015–1046, (2010).
- [6] J. Friedman, T. Hastie, and R. Tibshirani, ‘The elements of statistical learning: Data mining, inference, and prediction’, *Springer Series in Statistics*, (2009).
- [7] X. Gao, B. Jiang, and S. Zhang, ‘On the information-adaptive variants of the admm: an iteration complexity perspective’, *Optimization Online*, (2014).
- [8] T. Goldstein, M. Li, X. Yuan, E. Esser, and R. Baraniuk, ‘Adaptive primal-dual hybrid gradient methods for saddle-point problems’, *ArXiv Preprint 1305.0546*, (2013).
- [9] B. He and X. Yuan, ‘Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective’, *SIAM Journal on Imaging Sciences*, **5**(1), 119–149, (2012).
- [10] G. Lan, ‘An optimal randomized incremental gradient method’, *ArXiv Preprint 1507.02000*, (2015).
- [11] X. Lin, ‘Dual averaging methods for regularized stochastic learning and online optimization’, *Journal of Machine Learning Research*, **11**, 2543–2596, (2010).
- [12] H. Ouyang, N. He, L. Tran, and A. Gray, ‘Stochastic alternating direction method of multipliers’, in *ICML*, pp. 80–88, (2013).
- [13] K. Scheinberg, S. Ma, and D. Goldfarb, ‘Sparse inverse covariance selection via alternating linearization methods’, in *NIPS*, pp. 2101–2109, (2010).
- [14] M. Schmidt, N. L. Roux, and F. Bach, ‘Minimizing finite sums with the stochastic average gradient’, *ArXiv Preprint 1309.2388*, (2013).
- [15] T. Suzuki, ‘Dual averaging and proximal gradient descent for online alternating direction multiplier method’, in *ICML*, pp. 392–400, (2013).
- [16] T. Suzuki, ‘Stochastic dual coordinate ascent with alternating direction method of multipliers’, in *ICML*, pp. 736–744, (2014).
- [17] R. J. Tibshirani and J. Taylor, ‘The solution path of the generalized lasso’, *Annals of Statistics*, **39**(3), 1335–1371, (2011).
- [18] H. Wang and A. Banerjee, ‘Online alternating direction method’, in *ICML*, pp. 1119–1126, (2012).
- [19] X. Zhang, M. Burger, and S. Osher, ‘A unified primal-dual algorithm framework based on bregman iteration’, *Journal of Scientific Computing*, **46**(1), 20–46, (2011).
- [20] Y. Zhang and L. Xiao, ‘Stochastic primal-dual coordinate method for regularized empirical risk minimization’, in *ICML*, pp. 353–361, (2015).
- [21] P. Zhao, J. Yang, T. Zhang, and P. Li, ‘Adaptive stochastic alternating direction method of multipliers’, in *ICML*, pp. 69–77, (2015).
- [22] W. Zhong and J. Kwok, ‘Fast stochastic alternating direction method of multipliers’, in *ICML*, pp. 46–54, (2014).
- [23] M. Zhu and T. F. Chan, ‘An efficient primal-dual hybrid gradient algorithm for total variation image restoration’, *UCLA CAM Report*, 8–34, (2008).
- [24] Z. Zhu and A. J. Storkey, ‘Adaptive stochastic primal-dual coordinate descent for separable saddle point problems’, in *Machine Learning and Knowledge Discovery in Databases*, 645–658, Springer, (2015).

Decentralized Large-Scale Electricity Consumption Shifting by Prosumer Cooperatives

Charilaos Akasiadis and Georgios Chalkiadakis¹

Abstract. In this work we address the problem of coordinated consumption shifting for electricity prosumers. We show that individual optimization with respect to electricity prices does not always lead to minimized costs, thus necessitating a cooperative approach. A *prosumer cooperative* employs an internal cryptocurrency mechanism for coordinating members decisions and distributing the collectively generated profits. The mechanism generates cryptocurrencies in a distributed fashion, and awards them to participants according to various criteria, such as contribution impact and accuracy between stated and final shifting actions. In particular, when a scoring rules-based distribution method is employed, participants are incentivized to be accurate. When tested on a large dataset with real-world production and consumption data, our approach is shown to provide incentives for accurate statements and increased economic profits for the cooperative.

1 Introduction

Demand-side management (DSM) in Smart Grid environments generally aims to induce changes to the consumers' demand curves, so as for the total demand to match the production [10, 26, 22]. In order to provide incentives for consumption rescheduling to the actors, variable pricing techniques are often used. This means that instead of applying a flat pricing scheme, time-of-use (TOU), or real-time pricing (RTP) are employed [7, 20]. By setting higher electricity price values for buying energy during intervals of high demand, and lower values during intervals of low demand, it is possible for an electricity consumer to reduce her expenses by rescheduling her energy usage to the most profitable intervals [2]. This is a task that becomes even more important (and challenging) when it comes to *electricity prosumers*. As prosumers both produce and consume energy [3, 24], they can take advantage of fluctuations in prices, and generate even more profit [14].

However, increased participation to DSM schemes often leads to herding effects. As such, the estimated consumption curve could significantly change, both endangering the Grid's stability, and leading to substantially different economic outcomes [28]. For this reason, the formation of consumer cooperatives or virtual power plants has been proposed [2, 5, 13, 28], an approach which, however, requires a centralized entity to serve as the cooperative manager.² To overcome both herding effects and the need for cooperative manager determination, in this work we champion the use of a purpose-designed

cryptocurrency protocol for distributed prosumer cooperative coordination. Cryptocurrencies and blockchain-oriented algorithms run distributedly, and are transparent. Additionally, they use encryption methods, which guarantee that the transactions are secure, and that no third-parties need to take part in the exchanges [8]. A first generation cryptocurrency protocol has already been used in a setting with electricity prosumers, and is called NRGcoin [15]. Although it incentivizes demand and production balancing, that protocol does not promote large-scale cooperative consumption shifting. In our work, we envisage a next-generation, special-purpose cryptocurrency software, which is executed by each cooperative member in a decentralized fashion, and is used for coordinating electricity consumption shifting actions and the sharing of the rewards.

Thus, here we combine for the first time cryptocurrency with mechanism design for cooperatives formation, to achieve large-scale coordinated shifting of electricity prosumers consumption. The cooperative shifting activities result to increased prosumer profits from electricity trading. Using a cryptocurrency protocol, prosumers autonomously create a *virtual* wholesale mediator between the end-users and the Grid. The protocol takes into account prosumer shifting capacity statements, and distributes personalized rewards given the final collective profits achieved, and the cooperative's profits sharing policy of choice. The coins awarded represent shares on the total cooperative profits.

Summarizing, our work has several contributions. First, we model *prosumers* in a market setting with variable prices, and present a *distributed consumption shifting approach for prosumer cooperatives*, which guarantees monetary gains to the participants. We apply a *novel cryptocurrency model* for the coordination and management of the cooperative shifting actions. In the proposed model, the rewards from prosumer participation are determined in a personalized manner, in the form of newly mined coins. We examine different coin mining methods, and champion one that evaluates prosumers via a scoring rule [9] assessing the difference between promised and final actions. *This is the first time that cryptocurrency mining and scoring rules are combined into one method*. By penalizing inaccuracy, this method incentivizes prosumers to provide truthful promises. We propose specific formation techniques, which select members for participation in cooperative actions.

Our approach can be applied in conjunction with any existing regulations or pricing schemes. We evaluate our scheme experimentally on a large dataset that extends over a one-year period, and which is based on real consumption and renewable production data. Simulation results confirm that adopting our mechanism leads to increased profits for the cooperative participants, stabler variable electricity prices, and achieves lower Peak-to-Average Ratio (PAR) values for the difference between electricity supply and demand. Especially

¹ Technical University of Crete, Greece, email: {akasiadi,gehalk}@intelligence.tuc.gr

² The term cooperative refers to conglomerations of prosumers that organize and operate largely on a democratic manner [1]; which is not necessarily the case for Virtual Power Plants.

when using a scoring rules-based reward redistribution method, accuracy is explicitly incentivized with increased gains for the accurate participants.

This paper is further structured as follows: In Section 2 we present the system setting and the individual prosumer financial decisions model. Section 3 presents the cooperative model, the cryptocurrency protocol and three different approaches for personalized reward sharing, as well as methods for contributor selection for cooperative actions. Section 4 presents the experimental results, and, finally, in Section 5 we conclude.

2 A Prosumer Consumption Shifting Model

We assume a setting encompassing *prosumers*, which both import and export energy from and to the Grid; and a *prosumer cooperative*, which is a large coalition of prosumers trading energy as a unique entity. The Grid is regulated by the *distributed system operator (DSO)*, responsible for the transmission of energy and its pricing.

Actors need to take decisions regarding trading at some day-ahead electricity market, or consuming electricity at specific 1 to T intervals during the course of a day (the day-ahead).³ Each actor i is characterized by the amount of electricity (kWh) imported $q_{i,t}^-$, and the amount exported $q_{i,t}^+$, during the time interval t . The aggregate demand and supply levels for each time interval are given by $Q_t^- = \sum_i q_{i,t}^-$, and $Q_t^+ = \sum_i q_{i,t}^+$ kWh. We assume reliable renewable production and demand forecasting techniques that can achieve high precision—lower than 2% mean absolute percentage error [11, 25]. Predictions are noted as $\tilde{q}_{i,t}^-$, and $\tilde{q}_{i,t}^+$ kWh, for imports and exports respectively. The predicted demand and supply for the planning horizon are noted as \tilde{Q}_t^- for the total imports, and \tilde{Q}_t^+ for the total predicted exports.

2.1 Promoting Demand-Side Management

Many methods have been proposed for modelling individual consumption profiles [10, 20, 27]. In our work, we examine the rescheduling of *shiftable loads*, which are those loads that it is possible to shift to in later or earlier time intervals, with minimum impact on the consumer's well being, e.g., battery charging, water-heaters, washing-machines, etc. Now, to promote demand side management operations, prosumers should be offered better prices to counterbalance the associated shifting costs. Following existing dynamic pricing mechanisms, which promote the balancing of demand and renewable energy supply [15], we assume that billing functions are in place (by the DSO) for selling $B_t^{sell}()$, and buying $B_t^{buy}()$ energy to/from the Grid, each with specific properties. First, they are functions of the quantity of energy produced $q_{i,t}^+$ and consumed $q_{i,t}^-$, respectively. Next, and in order to satisfy the supply and demand balancing requirements [23], both also need to be functions of aggregate supply, Q_t^+ , and demand, Q_t^- . Specifically, $B_t^{sell}()$ must take maximum values for fixed $q_{i,t}^-$ s and $q_{i,t}^+$ s, during intervals when $Q_t^+ = Q_t^-$. This incentivizes prosumers to produce exactly the quantity that is required for consumption (since their income is then maximized). Intuitively, it is to the DSO's interest that prosumers decide to sell when $Q_t^+ = Q_t^-$, since this defers the need to import or export energy.

Assumption 1 ([15]) *The pricing for selling energy to the Grid during a time interval t , is a function of the sold quantity, the aggregate quantity produced, and the aggregate quantity consumed during that interval, $B_t^{sell}(q_t^+, Q_t^+, Q_t^-)$; and for fixed q_t^+ , it is maximized as $Q_t^+ - Q_t^- \rightarrow 0$.*

Note that, assuming the electricity production of prosumers originates mainly from wind turbines and photovoltaic panels, the quantity produced q_t^+ cannot be easily controlled [23]. Moreover, the selling prices are also functions of aggregate demand Q_t^- , which we later optimize by shifting consumption tasks in a large-scale cooperative manner.

The $B_t^{buy}()$ on the other hand, should produce lower prices with higher renewable production excess, prompting prosumers to buy energy from the Grid (and perhaps store it future use). Intuitively, it is more efficient to consume the cheap renewable energy produced locally, than import from some external balancing market where prices are in general far worse [23]. This is because exporting or importing electricity involves additional expenditures, e.g. transmission lines, electricity resellers, etc. By contrast, $B_t^{buy}()$ produces higher values as renewable energy supply decreases.

Assumption 2 ([15]) *The pricing for buying energy from the Grid during a time interval t , is a function of the acquired quantity, the aggregate quantity produced, and the aggregate quantity consumed during that interval, $B_t^{buy}(q_t^-, Q_t^+, Q_t^-)$; and for fixed q_t^- , it is minimized as $Q_t^+ - Q_t^- \rightarrow +\infty$.*

2.2 Shifting to Profitable Time Intervals

Given this model, a prosumer can control the quantity consumed during time intervals by shifting consumption tasks. We now characterize each time interval as *peak* or *non-peak*. Peak intervals t_h are those intervals during which reducing consumption can be considered profitable for the prosumer. Specifically, due to Assumptions 1 and 2, this happens when aggregate demand is higher than supply:

Definition 1 (Peak intervals t_h) *Consider a non-negative threshold τ . A time interval t is considered to be a peak interval, t_h , if $\tilde{Q}_t^+ - \tilde{Q}_t^- < \tau$.*

Non-peak intervals t_l are those intervals during which, increasing consumption levels up to the reduced amount of energy that was decreased during t_h , results to lower expenses due to a reduced buying price. Specifically, due to Assumptions 1 and 2, this happens when demand is lower than supply:

Definition 2 (Non-peak intervals t_l) *Consider a non-negative threshold λ . A time interval t is considered to be a non-peak interval, t_l , if $\tilde{Q}_t^+ - \tilde{Q}_t^- > \lambda$.*

Intuitively, variables τ and λ correspond to load difference thresholds that allow profitable shifting actions. Their values can be based on the statistics of \tilde{Q}_t^+ and \tilde{Q}_t^- , according to each actor's business goals.

Now, we assume that each prosumer can alter her baseline demand value q_i^- . More specifically, during peak intervals prosumers can reduce down to $q_{i,t_h}^- - \hat{r}_i^{t_h}$; while for the non-peak intervals consumption can be increased up to $q_{i,t_l}^- + \hat{r}_i^{t_h}$ where $\hat{r}_i^{t_h}$ is the *stated reduction capacity* of each actor i . Also, as in [2], there is a *shifting cost* $c_i^{t_h \rightarrow t_l}$ associated with shifting from a peak to a non-peak interval. The *actual reduction capacity* $r_i^{t_h}$, refers to the load that is reduced during a t_h , and is shifted to some other, non-peak interval

³ A decision theoretic optimization approach for a single prosumer operating in such a setting was proposed by [3, 4]. However, they did not include cooperative electricity trading, nor dealt with consumption shifting.

t_l . These values can be obtained by using appropriate smart metering equipment.

We now discuss the price differences induced when a prosumer shifts $\hat{r}_i^{t_h}$ from a peak interval to a non-peak interval. When prosumers decrease their consumption during peak intervals, they accrue gains from the lower buy prices, and the higher sell prices. Namely, the *estimated profit* by the induced price variations for reducing at t_h is given by:

$$\begin{aligned} \text{profit}_i^{t_h}(\hat{r}_i^{t_h}) &= B_{t_h}^{\text{sell}}(\tilde{q}_{i,t_h}^+, \tilde{Q}_{t_h}^+, (\tilde{Q}_{t_h}^- - \hat{r}_i^{t_h})) \\ &\quad - B_{t_h}^{\text{sell}}(\tilde{q}_{i,t_h}^+, \tilde{Q}_{t_h}^+, \tilde{Q}_{t_h}^-) + B_{t_h}^{\text{buy}}(\tilde{q}_{i,t_h}^-, \tilde{Q}_{t_h}^+, \tilde{Q}_{t_h}^-) \\ &\quad - B_{t_h}^{\text{buy}}((\tilde{q}_{i,t_h}^- - \hat{r}_i^{t_h}), \tilde{Q}_{t_h}^+, (\tilde{Q}_{t_h}^- - \hat{r}_i^{t_h})) \end{aligned} \quad (1)$$

The result from subtracting the second term in Eq. (1) from the first, indicates the profit from the price differences for selling energy; selling during a peak interval t_h , with lowered aggregate demand, $(\tilde{Q}_{t_h}^- - \hat{r}_i^{t_h})$, grants better prices than with the initial demand, $\tilde{Q}_{t_h}^-$ (cf. Assumption 1 & Definition 1 above). Now, the last two terms give the difference in the bill that the prosumer will *pay* for consumption during t_h . Thus, to calculate this quantity, we subtract the billing paid by the prosumer for the reduced consumption $(\tilde{q}_{i,t_h}^- - \hat{r}_i^{t_h})$ from the initial estimated bill $B_{t_h}^{\text{buy}}(\tilde{q}_{i,t_h}^-, \tilde{Q}_{t_h}^+, \tilde{Q}_{t_h}^-)$.

Similarly, the *estimated loss* generated by increasing consumption during non-peak intervals t_l is given by:

$$\begin{aligned} \text{loss}_i^{t_l}(\hat{r}_i^{t_h}) &= B_{t_l}^{\text{sell}}(\tilde{q}_{i,t_l}^+, \tilde{Q}_{t_l}^+, \tilde{Q}_{t_l}^-) \\ &\quad - B_{t_l}^{\text{sell}}(\tilde{q}_{i,t_l}^+, \tilde{Q}_{t_l}^+, (\tilde{Q}_{t_l}^- + \hat{r}_i^{t_h})) \\ &\quad + B_{t_l}^{\text{buy}}((\tilde{q}_{i,t_l}^- + \hat{r}_i^{t_h}), \tilde{Q}_{t_l}^+, (\tilde{Q}_{t_l}^- + \hat{r}_i^{t_h})) \\ &\quad - B_{t_l}^{\text{buy}}(\tilde{q}_{i,t_l}^-, \tilde{Q}_{t_l}^+, \tilde{Q}_{t_l}^-) \end{aligned} \quad (2)$$

To calculate the *estimated gain* for an actor i , for shifting from a t_h to a t_l , we subtract the estimated loss at t_l and the shifting costs $c_i^{t_h \rightarrow t_l}$ per kWh from the estimated profit at t_h :

$$g_i^{t_h \rightarrow t_l}(\hat{r}_i^{t_h}) = \text{profit}_i^{t_h}(\hat{r}_i^{t_h}) - \text{loss}_i^{t_l}(\hat{r}_i^{t_h}) - \hat{r}_i^{t_h} c_i^{t_h \rightarrow t_l} \quad (3)$$

Definition 3 (Eligible interval pairs) *Eligible shifting interval pairs for a prosumer i are those (t_h, t_l) pairs for which the gain associated with the shifting is positive, i.e. $g_i^{t_h \rightarrow t_l}(\hat{r}_i^{t_h}) > 0$, where $\hat{r}_i^{t_h}$ is the actual quantity of the shifted consumption.*

Summarizing, the strategy for individual consumption rescheduling is to find those shifting interval pairs for which the estimated gain is maximized, and shift accordingly.

2.3 Shifting Without Coordination

When optimizing individually, each agent does not take into account other agent rescheduling actions, and considers their consumption to be the baseline. Thus, the optimizer can exhaustively calculate the $g_i^{t_h \rightarrow t_l}(\hat{r}_i^{t_h})$ values for each shifting interval pair, for a stated reduction capacity $\hat{r}_i^{t_h}$. Then, rescheduling takes place (e.g., shifting to the most profitable ones). However, without coordination or constraint enforcements, and since every prosumer optimizes individually, herding effects take place, resulting to substantially different prices during the intervals with the lowest/highest prices, than those anticipated by the prosumers. These ‘‘unexpected’’ price fluctuations are not in favor of the prosumer, as the estimated gains can end up turning to losses:

Lemma 1 *If every participant is rational, and billing follows Assumptions 1 and 2, optimizing the rescheduling of consumption individually does not guarantee positive final gains.*

Intuitively, Lemma 1 states that non-coordinated shifting actions in such settings cannot be expected to always lead to monetary gains for the participants, and cooperation is essential. It is straightforward to show this, considering the fact that estimated gains are calculated based on the values $\tilde{Q}_{t_h}^- - \hat{r}_i^{t_h}$ and $\tilde{Q}_{t_l}^- + \hat{r}_i^{t_h}$, which are used in the first and last term of Eq. (1), and the second and third term of Eq. (2), respectively. However, since participants are rational, every one acts the same manner, resulting to substantially different values finally realized, i.e. final $B^{\text{sell}}()$, $B^{\text{buy}}()$ prices are calculated using $\tilde{Q}_{t_h}^- - (\hat{r}_i^{t_h} + \sum_{j \in C \setminus i} \hat{r}_j^{t_h})$ and $\tilde{Q}_{t_l}^- + (\hat{r}_i^{t_h} + \sum_{j \in C \setminus i} \hat{r}_j^{t_h})$, resulting to lower $B^{\text{buy}}()$ and higher $B^{\text{sell}}()$. Moreover, if the total shifting capacity is not constrained, the conditions in Def. 1 and 2 can stop holding, rendering the intervals ineligible for profitable shifting.

3 Distributed Shifting and Reward Sharing

Now, cooperatives can be key for the effective coordination of consumption shifting actions [2]. Here we describe the workings of *prosumer* cooperatives, allowing members to both sell and buy energy as a single entity. We assume that cooperative members share common estimates regarding the total production and consumption per interval $\tilde{Q}_{t_h}^+$ and $\tilde{Q}_{t_h}^-$ (obtained, e.g., via the summation of communicated individual estimates). Also, participants execute a novel cryptocurrency mechanism, allowing for distributed management, transparency, and personalized rewards. The mechanism awards contributors with new coins, according to specific participation performance measures.

For the scheme to work, each individual i must announce only two values for each shifting interval pair: (a) her reduction capacity, $\hat{r}_i^{t_h}$; and (b) her confidence $\hat{\sigma}_i$ for meeting her reduction promises. The confidence represents the variance of a normal distribution (with mean value 0) over the error between the stated and the final action. This is in line with past approaches [2, 21].

An *optimistic estimate* of the cooperative shifting capacity is then collectively calculated as $\tilde{R}_C^{t_h} = \sum_{i \in C} \hat{r}_i^{t_h}$, and a *pessimistic estimate*, by $\tilde{r}_C^{t_h} = \sum_{i \in C} (1 - \hat{\sigma}_i) \hat{r}_i^{t_h}$. Then, the cooperative determines the shifting interval pairs (t_h, t_l) , as well as the target shifting capacity that will lead to increased profits. In order to guarantee profits, the target shifting capacity is the maximum r^{*,t_h} values to be rescheduled such that Assumptions 1, 2 continue to hold. That is, for each shifting interval pair (t_h, t_l) :

$$\text{maximize } r^{*,t_h} \text{ s.t.}$$

$$\tilde{Q}_{t_h}^+ - (\tilde{Q}_{t_h}^- - r^{*,t_h}) < \tau \quad (4)$$

$$\tilde{Q}_{t_l}^+ - (\tilde{Q}_{t_l}^- + r^{*,t_h}) > \lambda \quad (5)$$

Next, the estimated by the members cooperative gains (minimum and maximum) are calculated, given the total expected consumption and production values of the cooperative for each time interval, $\tilde{q}_{C,t}^- = \sum_{i \in C} \tilde{q}_{i,t}^-$, $\tilde{q}_{C,t}^+ = \sum_{i \in C} \tilde{q}_{i,t}^+$, and the estimates $\tilde{R}_C^{t_h}$, and $\tilde{r}_C^{t_h}$:

$$\tilde{G}_C^{t_h \rightarrow t_l} = \text{profit}_C^{t_h}(\tilde{R}_C^{t_h}) - \text{loss}_C^{t_l}(\tilde{R}_C^{t_h}) \quad (6)$$

$$\tilde{g}_C^{t_h \rightarrow t_l} = \text{profit}_C^{t_h}(\tilde{r}_C^{t_h}) - \text{loss}_C^{t_l}(\tilde{r}_C^{t_h}) \quad (7)$$

To continue, the estimated gains per kWh are calculated:

$$\tilde{G}_{C,kWh}^{t_h \rightarrow t_l} = \frac{\tilde{G}_C^{t_h \rightarrow t_l}}{\tilde{R}_C^{t_h}}, \tilde{g}_{C,kWh}^{t_h \rightarrow t_l} = \frac{\tilde{g}_C^{t_h \rightarrow t_l}}{\tilde{r}_C^{t_h}} \quad (8)$$

For shifting interval pairs $\{t_h, t_l\}$ that

$$\tilde{g}_{C,kWh}^{t_h \rightarrow t_l} > 0 \quad (9)$$

holds, the shifting procedure is expected to be profitable, and the shifting interval pair along with its $\tilde{G}_C^{t_h \rightarrow t_l}$ and $\tilde{g}_C^{t_h \rightarrow t_l}$ values are announced to the members. If $r^{*,t_h} \geq \tilde{r}_C^{t_h}$, every available contributor can participate in the shifting operations. However, in case $r^{*,t_h} < \tilde{R}_C^{t_h}$, the constraint of Eq. (4) does not yet hold and gains are not certain, so some agents must be excluded from action. We will examine different approaches for this in the following section.

Finally, if the cooperative actually reduces $r_C^{t_h} \leq r^{*,t_h}$ given actual Q_t^- & Q_t^+ , the final actual cooperative gain is:

$$g_C^{t_h \rightarrow t_l}(r_C^{t_h}) = profit_C^{t_h}(r_C^{t_h}) - loss_C^{t_l}(r_C^{t_h}) \quad (10)$$

Of course, in order for the final gain levels to be inside the estimated range, two conditions must hold. First, the statements $\hat{r}_i^{t_h}$, $\hat{\sigma}_i$, and the predictions \tilde{Q}_t^- & \tilde{Q}_t^+ must be accurate. As mentioned earlier, the accuracy of \tilde{Q}_t^- & \tilde{Q}_t^+ can be ensured by known methods [11, 25]. We examine how we can achieve the accuracy of $\hat{r}_i^{t_h}$, $\hat{\sigma}_i$ in the next section. Second, the cooperative must be sizeable, meaning that it is the only actor that can induce significant price changes by consumption rescheduling. This helps overcome the problems raised by Lemma 1.

Definition 4 (Sizeable cooperative) A cooperative C is sizeable, if its pessimistic reduction capacity estimate is much greater than the sum of external parties capacity, i.e. when

$$\sum_{i \in C} (1 - \hat{\sigma}_i) \hat{r}_i^{t_h} \gg \sum_{j \notin C} \hat{r}_j^{t_h}, \forall t_h \quad (11)$$

Remark 1 The cooperatives that are formed are sizeable, because, due to Lemma 1, every rational agent avoids optimizing individually, thus seeks to cooperate.

Lemma 2 If statements $\hat{r}_i^{t_h}$, $\hat{\sigma}_i$, are accurate, and the cooperative C is sizeable, then, the cooperative's shifting suggestions include only eligible shifting interval pairs for C , in other words C will have $\tilde{g}_C^{t_h \rightarrow t_l} > 0$.

Proof Since the cooperative has accurate knowledge of the total shifting capacity range $\tilde{R}_C^{t_h}$, and $\tilde{r}_C^{t_h}$, and it is sizeable, the $\tilde{g}_C^{t_h \rightarrow t_l}$ and $\tilde{G}_C^{t_h \rightarrow t_l}$ estimates are more accurate than others calculated based on partial knowledge, thus the following holds: $\tilde{g}_C^{t_h \rightarrow t_l} < g_C^{t_h \rightarrow t_l} < \tilde{G}_C^{t_h \rightarrow t_l}$. Now, due to the enforcement of the constraints from Eq. 4, 5, and 9, for the suggested interval pairs, the minimum gain estimate per kWh is positive, $\tilde{g}_C^{t_h \rightarrow t_l} > 0$. Thus the shifting interval pairs suggested by the cooperative are eligible. \square

In conjunction with Remark 1, this Lemma is important for the following reason. While the calculations above do not take the individual shifting costs into account, cooperative members must weigh the expected gain per kWh (if these are accurate) with their own shifting costs $c_i^{t_h \rightarrow t_l}$ and decide whether they will finally contribute or not. Now, Lemma 2 shows that the cooperative can take advantage of the predicted price differences and create profit by rescheduling consumption. Moreover, since C is sizeable, no other actor can significantly affect prices so that the cooperative does not meet its goals. Therefore, the $\tilde{g}_C^{t_h \rightarrow t_l}$ are accurate (assuming the $\hat{r}_i^{t_h}$, $\hat{\sigma}_i^{t_h}$ are too); and then individuals can safely weigh these against own shifting costs to decide participation. The overall process can be achieved as shown in Alg. 1. The complexity for solving the algorithm's first step,

Algorithm 1 Coordinated shifting for a (t_h, t_l) interval pair

Input: $\tilde{Q}_{t_h}^+, \tilde{Q}_{t_h}^-, \tilde{Q}_{t_l}^+, \tilde{Q}_{t_l}^-, \{\tilde{q}_{i,t}^-\}_C, \{\tilde{q}_{i,t}^+\}_C$

- 1: Determine and announce $\tilde{G}_C^{t_h \rightarrow t_l}, \tilde{g}_C^{t_h \rightarrow t_l}$
- 2: Receive agent bids $\{\hat{r}_i^{t_h}\}_C, \{\hat{\sigma}_i^{t_h}\}_C$,
- 3: Check constraints and select agents
- 4: Wait for shifting actions realization, $\{q_{i,t}^-\}_C, \{q_{i,t}^+\}_C$
- 5: Distribute revenues to contributors

i.e. finding the peak and non-peak intervals, and respective loads and gains for the daily planning horizon, is a function of the number of time intervals. For example, if the cooperative adopted a constrained optimization approach, it would be $\mathcal{O}(t^3)$, where t is the number of time intervals. Next comes the selection of the actual contributors during each peak interval, that of Line 3. The duration of this procedure depends on the selection method that each cooperative adopts. The most expensive step of the selection methods we present in Section 3.3 below, is that of ranking, whose complexity is $\mathcal{O}(n^2t)$ in the worst case, i.e. $\mathcal{O}(n^2)$ for sorting [18], times t time intervals.

3.1 Cooperative Balance Increase

As already discussed (Eq. (10)), prosumers generate gain from the price differences for both buying and selling electricity. However, the gain part from buying is immediately awarded to each prosumer in the form of reduced bills, and cannot be redistributed among the members easily. Better sell prices, on the other hand, result to larger income for the cooperative, and this profit can be concentrated into a collective account. The achieved cooperative balance increase by each collective shifting operation is given by:

$$\begin{aligned} bal_inc(r_C^{t_h}) &= B_{t_h}^{sell}(q_{C,t_h}^+, \tilde{Q}_{t_h}^+, (\tilde{Q}_{t_h}^- - r_C^{t_h})) \\ &\quad - B_{t_h}^{sell}(q_{C,t_h}^+, \tilde{Q}_{t_h}^+, \tilde{Q}_{t_h}^-) - B_{t_l}^{sell}(q_{C,t_l}^+, \tilde{Q}_{t_l}^+, \tilde{Q}_{t_l}^-) \\ &\quad + B_{t_l}^{sell}(q_{C,t_l}^+, \tilde{Q}_{t_l}^+, (\tilde{Q}_{t_l}^- + r_C^{t_h})) \end{aligned} \quad (12)$$

This equation is derived from Eq. (1) and (2) after removing the parts that include $B_{t_h}^{buy}$, and represents the achieved gain from sell prices alone. Assuming that the initial balance of the cooperative is zero, the cooperative balance over the time horizon of shifting operations is simply the sum of the per time step balance increases:

$$b_{COOP} = \sum_{t_h} bal_inc(r_C^{t_h}) \quad (13)$$

However, since each participant contributes to the increase differently, the distribution of rewards must be different as well. A straightforward procedure for this redistribution is to generate and award new coins, which, nevertheless, are returned to the prosumers based on each one's behavior. For this reason we propose a cryptocurrency protocol that is used simultaneously, to both coordinate, and reward prosumers.

3.2 COOPcoin for Prosumer Cooperatives

To achieve effective rescheduling of prosumers consumption, and reward back members according to their behavior, we propose the employment of a specialized cryptocurrency algorithm, designed for the coordination of prosumer cooperative actions. In existing cryptocurrency schemes, the same protocol is used by all members of the community, and each member executes a program that is linked with a distributed database, called the *blockchain* [6, 12].

The program performs certain calculations—e.g., in our case, consumption and production measurements, gain calculations, and so on, which implement Alg. 1. The individual results are then compared with those of other members, and, if validated—i.e. compared and matched, are written to each user’s database that is, added to the blockchain and stored as history. If validation fails for a member, the adopted result is the one calculated by most members. This distributed execution approach removes the need for cooperative managers.

Note that the distributed nature of such an algorithm is guaranteed with the use of existing cryptocurrency protocols. Such protocols offer many desirable features, e.g. distributed consensus, transaction transparency, and anonymized data sharing [19]. Particularly, although data are shared among all participants freely, they are encrypted, and only the issuer and trusted parties can actually recover actual information, and link data with real persons. Increased privacy, transparency, and the ability to operate democratically without a “manager”, are important for cooperatives [1]. Moreover, by ensuring these properties via the use of cryptocurrency, our approach can naturally extend to virtual power plants [5, 21] (where the trust among the constituting entities is even lower).

Our cryptocurrency scheme is specifically designed for prosumer cooperatives, and is called *COOPcoin*. The proposed cryptocurrency protocol “mines” coins according to a small number of measurements and calculations regarding the shifting behaviors, and *does not* require computationally intensive operations like other existing cryptocurrency algorithms, e.g. Bitcoin [12]. Here, “mining”⁴ is performed collectively, i.e. utility is generated by the better electricity rates as a result of collective shifting, and in place of the Bitcoin’s “proof-of-work” concept [12], we use what we term “proof-of-physical-action”: in order to get rewarded with COOPcoins, certain actions (i.e., electricity consumption shifting actions) must take place in the real world. For the sharing of the rewards, the protocol generates COOPcoins based on the collectively achieved profit and uses these to distribute that profit to the participants. The actual number of COOPcoins returned to each prosumer is determined based on their shifting behavior.

More specifically, the number of COOPcoins awarded depends on two terms: the first, $bal_inc(r_C^{t_h})$, is the actual balance increase due to the shifting operations, given by Eq. (12); and the second one, s_k , is a scaling factor used for the personalized rewarding.

$$b_{i,t_h}^+ = bal_inc(r_C^{t_h}) \cdot s_k^{i,t_h} \quad (14)$$

Now, the value of s_k that actually scales each participant’s share over the balance, depends on the reward sharing policies of each cooperative. We examine three approaches:

The *proportionate to estimated reduction (PROPest)* approach distributes back the balance increase to participants in a proportionate manner, according to their capabilities stated prior to the rescheduling actuation.

$$s_{PROPest}^{i,t_h} = \frac{(1 - \hat{\sigma}_i) \hat{r}_i^{t_h}}{\sum_{i \in C} (1 - \hat{\sigma}_i) \hat{r}_i^{t_h}} \quad (15)$$

The *proportionate to actual reduction (PROPact)* approach rewards

according to the achieved individual reduction.

$$s_{PROPact}^{i,t_h} = \frac{r_i^{t_h}}{\sum_{i \in C} r_i^{t_h}} \quad (16)$$

The *accurate (ACCU)* approach uses the normalized *continuously ranked probability score (CRPS)*, a strictly proper scoring rule, to assess the absolute relative error ϵ_i between promised $\hat{r}_i^{t_h}$, and actual $r_i^{t_h}$ performance, with $\hat{\sigma}_i$:

$$s_{ACCU}^{i,t_h} = \frac{1 - CRPS(\mathcal{N}(0, \hat{\sigma}_i), \epsilon_i)}{\sum_{j \in C \setminus \{i\}} (1 - CRPS(\mathcal{N}(0, \hat{\sigma}_j), \epsilon_j)) + 1} \quad (17)$$

CRPS has been used in the past [2, 21] to rank production and consumption reduction forecasts. Here, it reduces the prosumer COOPcoin reward when her actual performance is not inside the stated confidence range. It is used in negative orientation and is normalized, so that perfect forecasts generate a value of zero, while the worst ones produce a value of 1. This incentivizes participants to be accurate.

Theorem 1 *When using ACCU for electricity prosumers cooperative reward sharing, participants are incentivized to be accurate regarding their statements.*

Proof A scoring rule is a function $S(P, Q)$ that assesses the distance between a predictive distribution P and an actual distribution Q . When the rule is *strictly proper*, then $S(Q, Q) \geq S(P, Q)$, with the equality holding iff $P = Q$, i.e. the value is maximized for exact forecasts [9]. Since CRPS is used here in negative orientation, and is normalized, i.e. $CRPS \in [0, 1]$, we have that $CRPS(Q, Q) \leq CRPS(Q, P)$, with the equality holding if and only if $Q = P$. Also, because any affine combination of a strictly proper scoring rule is also strictly proper [16], we exclude agent’s i CRPS from the denominator of Eq. (17), guaranteeing that s_{ACCU}^i is also strictly proper. Now, due to CRPS placement in Eq. (17) for fixed $r_i^{t_h}, r_C^{t_h}$, the share from the positive balance increase (Lemma 2) for the participant i is maximized when $CRPS=0$, leading to $s_{ACCU}^{i,t_h}(Q, Q) \geq s_{ACCU}^{i,t_h}(Q, P)$ with the equality holding iff $Q = P$. Thus, the reward for i is maximized when the forecast $\hat{\sigma}_i$ is accurate. \square

Note that, to maintain strict propriety, i is excluded from the denominator, leading to a small surplus of gain not being directly awarded to the participants in the form of COOPcoins. This *weak budget balancedness* does not affect the other properties of our approach, and the surplus can be returned to the actors in various ways (e.g., via the purchase of new equipment, or as bonus to new members).⁵

3.3 Selection of Contributors

As pointed out earlier, it is probable that shifting capacity is larger than the maximum eligible for profitable actions. In such cases, the cooperative must select only a subset from the available participants in C to include in shifting operations. The actual method used for the selection can vary among cooperatives, according to their business plans and policies. In any case, it is to the actors’ best interest to form reliable cooperatives in order to both achieve gains, and contribute to Grid stability. Here, we examine three selection methods.

⁴ Since cryptocurrency is not issued by a central authority, the process that creates new coins is performed by end-users and it is called *mining*. According to this procedure, users check if the data of the available transactions are valid, i.e. signatures are genuine, amounts in transactions are correct, etc., and are given newly created coins as a reward [17].

⁵ Alternatively, considering that COOPcoins represent shares, if no surplus redistribution actions are performed, the result is an increase to the exchange rate between the COOPcoin and the “external” currency used to pay the cooperative, benefiting this way everyone with COOPcoins in their possession.

The *Random selection* method picks contributors uniformly until the required $\tilde{R}_C^{t_h}$ is covered for each t_h .

The *Reduction Capacity selection* method sorts contributors *wrt.* reduction capacity for each shifting interval pair, in an ascending order. Then, it starts including from the one with the lowest value, until $\tilde{R}_C^{t_h}$ is covered for each t_h . The objective here is to include as many contributors as possible.

The *Engagement selection* method ranks contributors *wrt.* their wealth in COOPcoins. Then, starting from the richer one, it continues with the rest, until $\tilde{R}_C^{t_h}$ is covered for each t_h . The intuition is to include active and valuable members, since the wealth in COOPcoins does not only indicate participation frequency, but overall effectiveness as well.

Following selection, the accepted contributors are called for action, and rewards are dispensed after the actions occur.

4 Experimental Evaluation

In this section we present the dataset and the results from the simulation. First, we report the origin of our dataset, and describe its augmentations to account for missing values. Next, we show the different impacts of individual and cooperative shifting actions, as well as selection methods comparison results; the stability of our proposed scheme is illustrated with a sensitivity test and, finally, we show how different COOPcoin reward sharing techniques incentivize statement accuracy.

4.1 Simulations Setting

Our simulations employ a dataset based on consumption data from Kissamos, a district of Crete, Greece, and renewable production data from Galicia,⁶ Spain, both in 2012. The consumption data represent hourly demand from different contract categories, and include seasonal variabilities.⁷ In particular, the load profiles come from *residential, commercial, industrial, agricultural, public, and municipal* customers. However, due to the nature of agricultural and municipal demand profiles (i.e. mainly pumps, street lighting, etc.), which are tasks that cannot be shifted in time, these two categories do not participate in the prosumer cooperative of our simulation. In total, there are 7,376 prosumers in our setting.

The production data come from real wind generators and solar power plants, and have been scaled and divided, so as to represent the production of each prosumer. Prosumers are equipped either with both wind generators and solar panels, or with a single type of generation only. The average daily prosumer electricity production in our setting is 7.68kWh. The numerical results presented in this section are averages over 10 yearly iterations—that is, averages over 10 simulation runs, with each simulation run encompassing 344 days in 2012.⁸

External Variable Pricing. Now, as the external currency to our mechanism, we assume that the NRGcoin protocol is adopted by the

market and thus $B_t^{sell}()$ and $B_t^{buy}()$ are calculated based on the formulations shown in [15]:

$$B_t^{sell}(q_t^+, Q_t^+, Q_t^-) = (0.1 \cdot q_t^+) + \frac{0.2 \cdot q_t^+}{e^{\left(\frac{Q_t^+ - Q_t^-}{Q_t^-}\right)^2}} \quad (18)$$

and

$$B_t^{buy}(q_t^-, Q_t^+, Q_t^-) = \frac{(0.65 \cdot Q_t^-) \cdot q_t^-}{Q_t^- + Q_t^+} \quad (19)$$

Both billing functions satisfy our assumptions regarding variable pricing, as explained in Section 2.1. The parameter values 0.1,0.2,0.65 are set arbitrarily, so that Eq. (18) and Eq. (19) to produce reasonable results. An illustrative example of the B_t^{buy} and B_t^{sell} values during the eleventh and twelfth weeks of the simulation, are shown in Fig. 1.

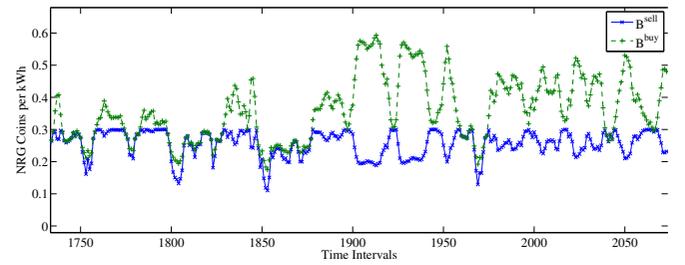


Figure 1. Time-series of B_t^{buy} and B_t^{sell} used in the simulation as the Grid's pricing mechanism.

Shifting Behaviours. Unfortunately, no shifting costs and capacities were available in the dataset, and we are not aware of any datasets including such values. Thus, we assumed shifting capacities that were on average 35% of the demand, varying among categories (e.g. higher for industrial prosumers, and lower for residential ones).

Shifting costs increase inversely proportionally to the prosumer baseline demand, meaning that shifting to an interval which typically has less demand, induces increased comfort loss. In this setting, the shifting costs result to an average value of 3.9 profitable interval pairs/day, for each prosumer.

Moreover, participants are divided into two different accuracy classes that describe the relationship between agent confidence statements $\hat{\sigma}_i$, and final realized shifting actions. The first one contains the *accurate* predictors; this describes the realistic case where agents are mainly confident about their statements, and also have a high probability to deliver what they promised. The second one, corresponds to the *inaccurate* predictors, where prosumers might or might not follow stated forecasts. For *accurate* predictors, the confidence statements and the parameters for calculating the absolute relative error ϵ_i , are sampled from $\mathcal{B}(1, 5)$ and $\mathcal{B}(4, 2)$ respectively; note that the actual $r_i^{t_h}$ is calculated by the product of a sample $\alpha_i^{t_h}$ and the stated shifting capacity $\hat{r}_i^{t_h}$, i.e. $r_i^{t_h} = (1 - \alpha_i^{t_h})\hat{r}_i^{t_h}$. For *inaccurate* agents, confidence statements $\hat{\sigma}_i$ are sampled from a “wider” Gaussian $\mathcal{N}(0.5, 0.15)$, and the $\alpha_i^{t_h}$ parameter is set to $1 - \hat{\sigma}_i$. About 50% of the participants in our setting belong to the accurate class, with the rest being inaccurate agents (since agents are assigned to a specific class with 50% probability).

⁶ <http://www.sotaventogalicia.com>

⁷ The production and consumption values in a Matlab file format can be obtained from <http://intellix2.intelligence.tuc.gr/~akasiadi/ProsumerCoop/>.

⁸ Note that since the simulated time horizon extends to a year, all seasonal variabilities and additional uncertainties (e.g. individual agent availability, individual reduction capacities for each time interval) are sufficiently taken into account.

Parameters τ and λ . To calculate the τ, λ parameters for each day, we first determined the average kWh difference between supply and demand, $Q_t^+ - Q_t^-$, across the 24 values. Then, τ is placed at the 75% of the distance between the average and minimum difference value, while λ at the 25% of the distance between the average and the maximum difference value. Nevertheless, these particular values are application specific, and other algorithms can be used for their calculation as well, according to each cooperative’s capabilities and business goals. We now proceed to describe the numerical results from our experiments.

4.2 Individual vs. Cooperative Action

We first compare two different scenarios, one with the prosumers shifting according to individually optimized plans, and a second where they shift according to the cooperative suggestions. Contributors are selected randomly, and everyone is accurate with respect to their promises and final actions, i.e. no specific “accuracy” classes are used for this set of experiments. Table 1 shows the difference in the total bills of the prosumers, and the average across all year daily peak-to-average ratio (PAR) values, for the total demand and supply difference, buying, and selling prices.

Table 1. Performance of individual and cooperative action.

Method	Total bill difference	Avg. PAR $ Q_t^+ - Q_t^- $	Avg. PAR B^{sell}	Avg. PAR B^{buy}
Initial	-	1.96	1.28	1.21
Individ.	-2.4%	4.75	1.38	1.19
Coop.	-4.4%	1.86	1.24	1.19

First, we observe that the “collective bill” when prosumers cooperate drops by a factor of 2 (-4.4% vs. -2.4%) compared to its reduction when they do not. Additionally, the cooperative approach outperforms individual optimization in terms of peak-to-average ratio (PAR) values (average across 344 days) for the $|Q_t^+ - Q_t^-|$ and B^{sell} columns. Lowering the PAR of $|Q_t^+ - Q_t^-|$ means that demand and supply difference is flattened, thus less electricity is exchanged to the balancing market, and consumption of locally produced electricity is promoted. By contrast, the increase of the PAR for the individual approach shows the scale of the herding effects that take place. Furthermore, reduction in the PAR value of the selling price when cooperating, means smaller fluctuations, a fact that allows for more realistic planning. Lastly, both cooperative and individual optimization leads to buying prices that are quite stable.

4.3 Evaluating Contributor Selection Methods

Henceforth, we assume that each prosumer belongs to the two different accuracy classes introduced earlier. In this setting, we first evaluate the three different contributor selection approaches we put forward, namely *Engagement*, *Reduction Capacity* and *Random*. Table 2 presents the (average) total cooperative gains and balance for 2012, for each of the three proposed contributor selection methods, in NRG coins. The prior COOPcoin wealth distribution required by the *Engagement selection*, is determined by conducting simulations using *Reduction Capacity selection* and ACCU reward sharing. We can observe that *Engagement* achieves the highest cooperative gains, and, consequently, the highest cooperative balance. *Reduction Capacity* also performs well wrt. gains in NRG coins. Lastly, *Random* has the worst outcome, which, nevertheless, is still profitable.

Table 2. Total cooperative gains and balance in 2012 for each selection method (NRG coins).

Measure	<i>Engagement</i>	<i>Reduction Capacity</i>	<i>Random</i>
Total Coop. Gains	133805.51	123813.43	80714.62
Total Coop. Balance	68116.49	58134.89	15029.42

4.4 Shifting Capacity Sensitivity Test

In this set of experiments we gradually change the shifting capacity of every individual, from -50%, to +50% of their initial, and examine the impacts to the average individual gains and average coalition sizes during 2012. Note that during all experiments we enforced two constraints: (a) shifting capacity cannot exceed the hourly demand, and (b) shifting capacity cannot be negative. Results are presented in

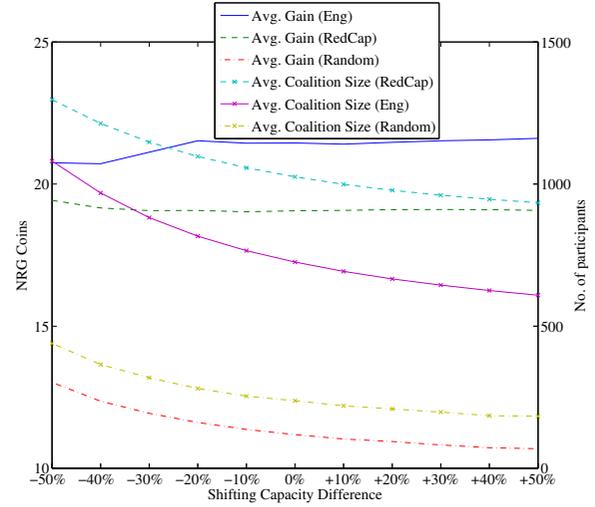


Figure 2. Average individual gains during 2012, and average shifting coalition sizes vs. shifting capacity changes for all three selection methods.

Figure 2. As we observe, the average size of the shifting coalitions drops when the shifting capacity of the prosumers increases, for all selection methods. This is natural, since increase in the shifting capacity helps meeting cooperative shifting requirements with fewer members. Regarding the individual gains of participants for 2012, we can see that *Engagement* and *Reduction Capacity* selection methods are not significantly affected by the difference in the shifting capacity of the individuals. Also, differences in the shifting capacity do not induce changes in the selection methods relative ranking; for all values, *Random* consistently ranks last, while *Engagement* produces higher gains than the *Reduction Capacity*.

The increase in average individual gain for the *Engagement* selection is due to the fact that “better” performing agents are selected continuously, resulting to the gain being shared by fewer agents. Lastly, when contributors are selected using the *Random* method, the average individual gain decreases for higher shifting capacity percentages. This happens because shifting operations are overtaken by fewer members, who, however, are not examined wrt. their truthfulness. Thus, it is highly probable that the final cooperative shifting actions deviate from those promised, leading to less overall gain.

Figure 3 presents the *total cooperative balance* in 2012, for different shifting capacity percentages. As we can see, the balance of the cooperative *decreases* for all methods as prosumers’ shifting capacity

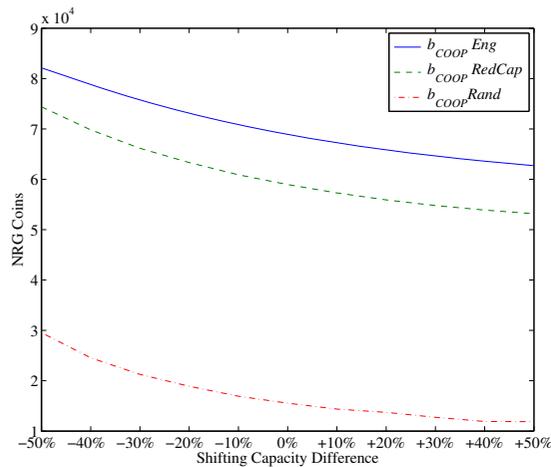


Figure 3. Cooperative balance in 2012 vs. shifting capacity changes for all three selection methods.

increases. This can be interpreted if we consider the average coalition size values from Fig. 2: when the number of acting prosumers decreases, each individual inaccuracy regarding the shifting operation has a larger impact on the cooperative performance, leading to reduced cooperative gains. When actors come in large numbers, their individual errors have less impact, since they cancel out by others to the opposite direction. Regardless of the observed balance reduction, we can see that *Engagement* selection method consistently produces the highest balance, with *Reduction Capacity* following, and *Random* producing the least, irrespective of the way the shifting capacity changes.

4.5 Reward Sharing Methods Evaluation

Finally, we use the three reward sharing methods with each selection method, in order to find the one incentivizing accurate statements the most. Figure 4, presents the *difference* in the total COOPcoin wealth between accurate and inaccurate actors when using different selection and reward sharing approaches. We observe that, for every selection method, this difference is higher when the *ACCU* approach is used for rewarding. Also, as expected, when using *ACCU* redistribution combined with *Engagement selection*, the difference in COOPcoin wealth between accurate and inaccurate participants reaches its highest levels. Interestingly, when using *ACCU*, accurate participants are rewarded more, even when the selection criterion does not distinguish between the two classes (i.e., when using *Reduction Capacity* and *Random* selection). For all these reasons, *ACCU* is clearly the most effective in promoting statements accuracy.

5 Conclusions and Future Work

In this work, we presented a cooperative prosumer consumption shifting scheme that employs a distributed cryptocurrency mechanism for the coordination of members' actions. We proposed contributor selection and reward sharing methods which incentivize truthfulness, guarantee increased profits from the trading of electricity, and help flatten electricity demand curve. Though illustrated in the domain of prosumer cooperatives, our approach immediately applies to the broader domain of prosumer virtual power plants as well.

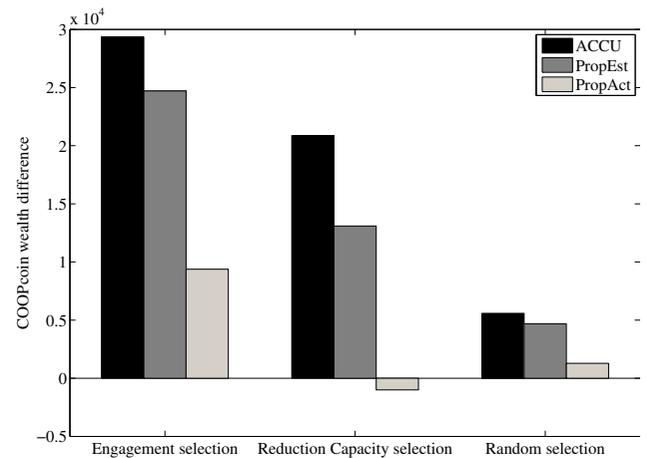


Figure 4. COOPcoin total wealth difference between accurate and inaccurate actors after 344 days.

In future work, we will study settings with multiple cooperatives that can even exchange members. Also, additional selection and rewarding techniques with specific properties will be tested and compared against each other. Furthermore, in the near future we aim to work closely with partners from the European Federation for Renewable Energy Cooperatives [1], in order to see these ideas adopted by real-world Smart Grid businesses.

ACKNOWLEDGEMENTS

Charilaos Akasiadis gratefully acknowledges partial travel support from the Hellenic Artificial Intelligence Society (EETN).

REFERENCES

- [1] European Federation for Renewable Energy Sources Cooperatives. <https://rescoop.eu>. Accessed: 2016-04-15.
- [2] C. Akasiadis and G. Chalkiadakis, 'Agent cooperatives for effective power consumption shifting', in *27th AAAI Conference on Artificial Intelligence*, (2013).
- [3] A. Angelidakis and G. Chalkiadakis, 'Factored MDPs for optimal prosumer decision-making', in *Proc. of the 2015 Int. Conf. on Autonomous Agents and Multiagent Systems, AAMAS '15*, pp. 503–511, (2015).
- [4] A. Angelidakis and G. Chalkiadakis, 'Factored MDPs for optimal prosumer decision-making in continuous state spaces', in *Proc. of EUMAS-2015*, (2015).
- [5] Peter Asmus, 'Microgrids, virtual power plants and our distributed energy future', *The Electricity Journal*, **23**(10), 72–82, (2010).
- [6] A. Back, M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A. Miller, A. Poelstra, J. Timon, and P. Wuille. Enabling Blockchain Innovations with Pegged Sidechains. <https://blockstream.com/sidechains.pdf>, 2014. [Online; accessed 9-April-2016].
- [7] S. Bandyopadhyay, R. Narayanam, R. Kota, H. J. Petra, and Z. Charbiwala, 'Aggregate demand-based real-time pricing mechanism for the smart grid: a game-theoretic analysis', in *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pp. 2554–2560. AAAI Press, (2015).
- [8] P. Franco, *Understanding Bitcoin: Cryptography, Engineering and Economics*. The Wiley Finance Series, Wiley, 2014.
- [9] T. Gneiting and A. E. Raftery, 'Strictly proper scoring rules, prediction, and estimation', *Journal of the American Statistical Association*, **102**(477), 359–378, (2007).
- [10] S. Gottwalt, W. Ketter, C. Block, J. Collins, and C. Weinhardt, 'Demand Side Management — A simulation of household behavior under variable prices', *Energy Policy*, **39**(12), 8163 – 8174, (2011).

- [11] R. K. Jain, K. M. Smith, P. J. Culligan, and J. E. Taylor, 'Forecasting energy consumption of multi-family residential buildings using support vector regression: Investigating the impact of temporal and spatial monitoring granularity on performance accuracy', *Applied Energy*, **123**, 168–178, (2014).
- [12] N. Koblitz and A. J. Menezes, 'Cryptocash, cryptocurrencies, and cryptocontracts', *Designs, Codes and Cryptography*, **78**(1), 87–102, (2015).
- [13] R. Kota, G. Chalkiadakis, V. Robu, A. Rogers, and N. R. Jennings, 'Cooperatives for demand side management', in *The 7th Conf. on Prestigious Applications of Intelligent Systems (PAIS @ ECAI)*, pp. 969–974, (August 2012).
- [14] Y. Luo, S. Itaya, S. Nakamura, and P. Davis, 'Autonomous cooperative energy trading between prosumers for microgrid systems', in *Local Computer Networks Workshops (LCN Workshops), 2014 IEEE 39th Conference on*, pp. 693–696, (Sept 2014).
- [15] M. Mihaylov, S. Jurado, N. Avellana, K. Van Moffaert, I.M. de Abril, and A. Nowe, 'NRGcoin: Virtual currency for trading of renewable energy in smart grids', in *European Energy Market (EEM), 2014 11th Int. Conf. on the*, pp. 1–6, (May 2014).
- [16] N. Miller, P. Resnick, and R. Zeckhauser, 'Eliciting informative feedback: The peer-prediction method', *Management Science*, **51**(9), 1359–1373, (2005).
- [17] K. J. O'Dwyer and D. Malone, 'Bitcoin mining and its energy footprint', in *Irish Signals Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CICT 2014). 25th IET*, pp. 280–285, (June 2014).
- [18] C. H. Papadimitriou, 'Computational complexity', in *Encyclopedia of Computer Science*, 260–265, John Wiley and Sons Ltd., Chichester, UK.
- [19] M. Pilkington, 'Blockchain technology: Principles and applications', *Research Handbook on Digital Transformations*, (2016).
- [20] S. D. Ramchurn, P. Vytelingum, A. Rogers, and N. R. Jennings, 'Agent-based control for decentralised demand side management in the smart grid', in *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pp. 5–12, (2011).
- [21] V. Robu, R. Kota, G. Chalkiadakis, A. Rogers, and N. R. Jennings, 'Cooperative virtual power plant formation using scoring rules', in *Proceedings of the 11th Int. Conference on Autonomous Agents and Multiagent Systems-Vol. 3*, pp. 1165–1166, (2012).
- [22] M. Soliman, H. and A. Leon-Garcia, 'Game-theoretic demand-side management with storage devices for the future smart grid', *Smart Grid, IEEE Transactions on*, **5**(3), 1475–1485, (2014).
- [23] G. Strbac, 'Demand side management: Benefits and challenges', *Energy Policy*, **36**(12), 4419–4426, (2008). Foresight Sustainable Energy Management and the Built Environment Project.
- [24] Q. Sun, A. Beach, M. E. Cotterell, Z. Wu, and S. Grijalva, 'An economic model for distributed energy prosumers', in *System Sciences (HICSS), 2013 46th Hawaii International Conference on*, pp. 2103–2112, (Jan 2013).
- [25] J. W. Taylor, 'Triple seasonal methods for short-term electricity demand forecasting', *European Journal of Operational Research*, **204**(1), 139–152, (2010).
- [26] K. Valogianni, W. Ketter, and J. Collins, 'A multiagent approach to variable-rate electric vehicle charging coordination', in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS '15*, pp. 1131–1139, (2015).
- [27] A. Veit and H.-A. Jacobsen, 'Multi-agent device-level modeling framework for demand scheduling', in *IEEE Int. Conf. on Smart Grid Communications, SmartGridComm '15*, (2015).
- [28] A. Veit, Y. Xu, R. Zheng, N. Chakraborty, and K. Sycara, 'Demand side energy management via multiagent coordination in consumer cooperatives', *Journal of Artif. Intel. Research*, 885–922, (2014).

Analysing Approximability and Heuristics in Planning Using the Exponential-Time Hypothesis¹

Meysam Aghighi, Christer Bäckström, Peter Jonsson, and Simon Ståhlberg²

Abstract. Cost-optimal planning has become a very well-studied topic within planning. Needless to say, cost-optimal planning has proven to be computationally hard both theoretically and in practice. Since cost-optimal planning is an optimisation problem, it is natural to analyse it from an approximation point of view. Even though such studies may be valuable in themselves, additional motivation is provided by the fact that there is a very close link between approximability and the performance of heuristics used in heuristic search. The aim of this paper is to analyse approximability (and indirectly the performance of heuristics) with respect to lower time bounds. That is, we are not content by merely classifying problems into complexity classes — we also study their time complexity. This is achieved by replacing standard complexity-theoretic assumptions (such as $\mathbf{P} \neq \mathbf{NP}$) with the *exponential time hypothesis* (ETH). This enables us to analyse, for instance, the performance of the h^+ heuristic and obtain general trade-off results that correlate approximability bounds with bounds on time complexity.

1 INTRODUCTION

Given a planning instance where each action is assigned (a typically non-negative) weight, the task of *cost-optimal* planning is to find a plan with minimum total weight, or report that no plan exists. This problem has gained a steadily increasing amount of interest in the planning community during recent years. In theory, both satisficing (propositional) planning and cost-optimal planning are \mathbf{PSPACE} -complete problems [8]. However, cost-optimal planning has proven to be more difficult in practice and this has spurred the search for efficient cost-optimal planners.

Since cost-optimal planning is inherently an optimisation problem, it would be natural to try to approximate it instead of finding the best possible solution, especially since the problem has proven to be computationally difficult both theoretically and in practice. The term “approximation” has been assigned different meanings in the literature. We opt for the strict interpretation that is the dominant one in contemporary computer science: we want to produce a solution that has at most α times the cost of an optimal solution (where α may be a constant or a function of some parameter based on the instance under consideration), and preferably we want to produce such a solution faster than it would take to find an optimal solution. This approach for attacking hard optimisation problems is very common

in many different branches of computer science, cf. the textbook by Ausiello et al. [1]. Unfortunately, it appears that approximation has not been used to any significant degree in the context of cost-optimal planning, despite the fact that there are a few very well-known (non-)approximability results reported in the literature. For instance, Selman [29] has shown that blocks world is polynomial-time approximable within 2 but not within every $1 < c < 2$, unless $\mathbf{P} = \mathbf{NP}$, and Betz and Helmert [7] have analysed the approximability of the h^+ heuristics on several different domains.

Besides this obvious use of approximations, there is another usage that may very well be more important. Most state-of-the-art planners (both traditional and cost-optimal) employ heuristic search, and it has been interesting to see that the focus on domain-independent heuristics has fundamentally changed the role of computational complexity in planning. In particular, the identification of tractable fragments has been important, as pointed out by Katz and Domshlak [25].

Computational tractability can be an invaluable tool even for dealing with problems that fall outside all the known tractable fragments of planning. For instance, tractable fragments of planning provide the foundations for most (if not all) rigorous heuristic estimates employed in planning as heuristic search.

The results in this paper will exclusively be of a negative nature: we prove that certain problems cannot be solved within certain time bounds. We claim that negative results are important for the progress of planning research since such results present challenges to the community. For instance, it was hardness results (including the undecidability of first-order planning [9] and plan length arguments for propositional planning) that initiated the research on tractable subclasses in the first place [5]! We are in the fortunate position that the formal nature of many planning heuristics today allows for studying their connections with approximation theory and other concepts in complexity theory, thus making them amenable to a more rigorous formal analysis. Nevertheless, traditional complexity analysis based on polynomial reductions is often a much too coarse tool for deriving interesting results. Hence, alternative analysis methods, established ones (such as parameterized analysis) as well as new ones, are required to advance the theoretical understanding of planning and, in the long run, for obtaining better planners.

The aim of this paper is to analyse approximability (and indirectly the performance of heuristics) with respect to lower time bounds: ultimately, we aim at results such as “approximating this class of planning instances within a given bound takes at least this amount of time”. Non-trivial results of this kind are typically not possible to obtain using standard complexity-theoretic assumptions such as $\mathbf{P} \neq \mathbf{NP}$ or $\mathbf{P} \neq \mathbf{PSPACE}$: superpolynomial lower time bounds tend to resolve this kind of conjectures! Thus, we need stronger assumptions

¹ Meysam Aghighi and Simon Ståhlberg are partially supported by the National Graduate School in Computer Science (CUGS), Sweden. Christer Bäckström is partially supported by the Swedish Research Council (VR) under grant 621-2014-4086.

² Department of Computer and Information Science, Linköping University, SE-58183 Linköping, Sweden. Email: firstname.lastname@liu.se

that give quantitative time bounds. Our approach is based on the *exponential time hypothesis* (ETH) by Impagliazzo and Paturi [20]. It has become an important and widely used assumption when studying the computational complexity of combinatorial problems, cf. the survey by Lokshantov et al. [26]. It is also gaining more and more popularity when studying central problems in AI such as planning and constraint satisfaction [2, 3, 24, 33]. Assessing the plausibility of the ETH is very difficult due to the same reasons as assessing the plausibility of $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$: the available tools for attacking this kind of questions are currently too weak. However, it is clear that the ETH has deep connections with many topics in computer science, such as the existence of subexponential algorithms for \mathbf{NP} -complete problems [19, 23, 28], the complexity and approximability of optimisation problems [11, 27] and parameterised complexity theory [10, 12], and the failure of ETH would have profound consequences.

We begin this paper (in Section 2) by providing some basic definitions and results. In particular, we describe the connections between optimisation, approximability and heuristics in Section 2.4. We continue in Section 3 by analysing the time complexity of cost-optimal planning restricted to actions having no preconditions and only positive effects. Even though this class of planning problems may seem absurdly limited, it allows us to better understand the h^+ heuristics. It is known that computing h^+ is an \mathbf{NP} -hard problem but we do not know anything about the time needed for computing it. We show that for every function $\alpha = (1 - o(1)) \cdot \ln(|V|)$, there exists a constant $c > 0$ such that approximating h^+ within α in time $O(2^{\|P\|^c})$ (where $\|P\|$ is the size of instance P and $|V|$ is the number of variables in P) is impossible unless the ETH is false. One can alternatively view this result as follows: if h^+ can be approximated within some (arbitrarily large) constant in time $2^{\|P\|^c}$ for all $c > 0$, then the ETH is false. In particular, the existence of a polynomial-time algorithm implies that ETH is false. This result holds even if we require that all actions have uniform weight 1.

In the next part (Section 4), we slightly broaden the class of problem instances by allowing both positive and negative preconditions (but still requiring the effects to be positive). We show that (unless the ETH is false) there exists a constant $c > 0$ such that there is no algorithm running in time $O(2^{\|P\|^c})$ that can approximate cost-optimal planning for this class of instances within $2^{p(|V|)}$ for any polynomial function p . This result carries over to the h^+ heuristic extended with negative preconditions. From a computational point of view, it is thus quite a good idea to avoid negative preconditions whenever possible. One should note that, unlike the main result of Section 3, this result does *not* hold if actions are required to have uniform weight.

In the final part (Section 5), we consider unrestricted cost-optimal planning. The main result shows that if the problem can be approximated by some function f within $2^{|V| - 2|V|^{1/c} - 1}$ for some $c > 1$, then f cannot be computed in time $O(2^{|V|^{1/c'}})$ for any $c' > c$, unless the ETH is false. Hence, we obtain a trade-off between approximation bounds and time bounds. This is noteworthy since typical hardness results for approximation simply establish that a problem cannot be approximated within some given function of the problem instance. We conclude the paper in Section 6 by discussing the results and pointing out future research directions.

2 PRELIMINARIES

This section is divided into four parts. In the first three parts, we introduce the planning formalism, the exponential-time hypothesis, and the optimisation framework, respectively. In the fourth part, we es-

tablish and discuss connections between approximability and heuristics.

2.1 Planning

Most results in this paper state lower bounds for the complexity of planning. In such cases, it is beneficial to use a restricted planning language to make the results as strong as possible. Hence, all formal results will be stated using propositional STRIPS throughout. In particular, we will use *propositional STRIPS with negative goals* (PSN) [8], which can alternatively be viewed as SAS^+ restricted to boolean (i.e. two-valued) variable domains [6].

Given a set X of propositional atoms, let $L(X)$ denote the set of all literals over X , i.e. $L(X) = \{x, \bar{x} \mid x \in X\}$. A set of literals $Y \subseteq L(X)$ is *consistent* if there is no $x \in X$ such that $\{x, \bar{x}\} \subseteq Y$. A PSN *frame* is a tuple $F = \langle V, A \rangle$ where V is a set of propositional atoms and A is a set of actions. The *state space* is $S(F) = 2^V$ and its members are called (*total*) *states*. Each action $a \in A$ has a *precondition* $\text{pre}(a) \subseteq L(V)$ and an *effect* $\text{eff}(a) \subseteq L(V)$, which are both consistent. We let $\text{eff}(a)^+ = \text{eff}(a) \cap V$ (the set of *positive effects*) and $\text{eff}(a)^- = \text{eff}(a) \cap \{\bar{v} \mid v \in V\}$ (the set of *negative effects*). The notation $a : X \Rightarrow Y$ defines an action a with $\text{pre}(a) = X$ and $\text{eff}(a) = Y$. For all $s, t \in S(F)$ and $a \in A$, we say that action a is *valid* in s if s satisfies $\text{pre}(a)$ and action a is *from* s to t if a is valid in s and $t = (s \cup \text{eff}(a)^+) \setminus \text{eff}(a)^-$. A sequence $\omega = \langle a_1, \dots, a_\ell \rangle$ of actions in A is a *plan* from $s_0 \in S(F)$ to $s_\ell \in S(F)$ if either (1) ω is the empty sequence and $s_0 = s_\ell$ or (2) there are $s_1, \dots, s_{\ell-1} \in S(F)$ such that a_i is from s_{i-1} to s_i for all i ($1 \leq i \leq \ell$).

A PSN *instance* is a tuple $P = \langle V, A, s_I, s_G \rangle$ such that $F = \langle V, A \rangle$ is a PSN frame, $s_I \in S(F)$ and $s_G \subseteq L(V)$ is consistent. We tacitly assume that instances and frames do not contain superfluous variables, i.e. every variable appears in the precondition or effect of at least one action. A *plan* for P is a plan from s_I to some $s \in S(F)$ that satisfies s_G .

Let PSN denote the set of PSN instances. For any subset $C \subseteq \text{PSN}$ of planning instances, we define the *Plan Existence (PE)* problem.

PE(C)

INSTANCE: A planning instance $P \in C$.

QUESTION: Does P have a plan?

It is also common to consider weighted actions, although we will only consider this topic in Section 4. A *weighted* PSN instance $P = \langle V, A, s_I, s_G, w \rangle$ is a PSN instance with an additional *weight function*, $w : A \rightarrow \mathbb{Z}_+$, that assigns a positive weight to each action. This function is extended to sequences of actions such that $w(\langle a_1, \dots, a_\ell \rangle) = w(a_1) + \dots + w(a_\ell)$. Note that weights are often referred to as *costs* in planning.

Let us now consider plan optimisation. The most common measures to optimise in planning are either the plan length or the total weight of a plan. These two optimisation problems can be defined as follows, where we consider only non-constructive optimisation, i.e. we only ask for the measure of an optimal solution, not for an actual solution. Considering non-constructive optimisation gives stronger lower time bounds since the complexity figures are not affected by the fact that optimal solutions may be exponentially long and thus need an exponential amount of time to be written down. *Length plan optimisation (LPO)* is the minimisation problem of finding the shortest plan and *Cost plan optimisation (CPO)* is the minimisation problem of finding the minimum plan weight.

LPO(C)

INSTANCE: A planning instance $P = \langle V, A, s_I, s_G \rangle$ in C .

ANSWER: $\min\{|\omega| \mid \omega \text{ is a plan for } P\}$ or ∞ if P has no plan.

GPO(C)

INSTANCE: A weighted planning instance $P = \langle V, A, s_I, s_G, w \rangle$, such that $\langle V, A, s_I, s_G \rangle$ is in C and $w : A \rightarrow \mathbb{Z}_+$ is a weight function.

ANSWER: $\min\{w(\omega) \mid \omega \text{ is a plan for } P\}$ or ∞ if P has no plan.

Obviously, LPO can be treated as the special case of GPO where the weight function assigns weight 1 to every action. Since both problems are defined as non-constructive, we can view them as functions that return the measure of an optimal plan, e.g. LPO(P) returns the length of a length-optimal plan for an instance P .

2.2 Satisfiability and the exponential time hypothesis

We will use several different variants of the boolean satisfiability problem during the course of this paper. In its basic form it is defined as follows.

SAT

INSTANCE: A boolean formula F in conjunctive normal form, i.e. F is a conjunction of *clauses* where each clause is a disjunction of literals.

QUESTION: Does F have a satisfying assignment?

The problem k -SAT, $k \geq 1$, is the SAT problem restricted to clauses containing at most k literals. The SAT problem is **NP**-complete and so is the k -SAT problem when $k \geq 3$ [17, problem LO1]. We will also consider the complement of SAT (known as UNSAT): an instance of this problem is considered a 'yes'-instance if and only if there does *not* exist a satisfying assignment. The UNSAT problem is **coNP**-complete and so is the UNSAT problem restricted to clauses of length k (which we denote k -UNSAT) whenever $k \geq 3$.

The complexity of k -SAT is more precisely characterised by the *exponential time hypothesis (ETH)* [20, 19]. This hypothesis is a conjecture stated as follows.

Definition 1 For all constant integers $k > 2$, let s_k be the infimum of all real numbers δ such that k -SAT can be solved in time $O(2^{\delta n})$, where n is the number of variables of an instance. The *exponential time hypothesis (ETH)* is the conjecture that $s_k > 0$ for all $k > 2$.

Informally, ETH says that satisfiability cannot be solved in subexponential time. ETH is not just an arbitrarily chosen concept, but a quite strong assumption that allows for defining a theory similar to the one of **NP**-completeness. There is a concept called SERF (subexponential reduction family) reduction which preserves subexponential time solvability. There is also a concept called SERF-completeness which is similar to **NP**-completeness, but based on SERF-reductions. In other words, there is a subclass of the **NP**-complete problems that are also SERF-complete, meaning that these can all be SERF reduced to each other. Hence, if one of these can be solved in subexponential time, then all of them can. One may thus view the SERF-based theory as a subexponential analogue of the **NP**-completeness theory.

It is possible to equivalently define the ETH for the number of clauses, i.e. replacing $O(2^{\delta n})$ with $O(2^{\delta m})$ in Definition 1, where

m is the number of clauses. It is known that the ETH with respect to the number of clauses holds if and only if the ETH with respect to the number of variables holds [19]. We will use this fact, for instance, in the proof of Lemma 2 below. Also note that since the ETH refers to actual deterministic time bounds it is possible to swap the answer, i.e. if we could solve k -UNSAT in time $O(f(n))$ for some function f , then we could also solve k -SAT in time $O(f(n))$. Hence, the ETH can equivalently be defined in terms of the k -UNSAT problem.

The time complexity of a problem is usually defined as a function of the instance size, while the ETH is defined as a function of the number of variables or clauses. While this allows for sharper results, it is not quite suitable for planning where the instance size is not necessarily polynomial in the number of variables. We will instead use time bounds on the form $O(2^{n^c})$, where n is the instance size. This approach is closely related to the concept of *power indices* [30, 31, 32].

Lemma 2 3 -SAT and 3 -UNSAT cannot be solved in time $O(2^{\|F\|^c})$ for any $c < 1$, unless the ETH is false.

Proof. We begin by analysing the size of 3-CNF formulas. Let F denote an arbitrary 3-CNF formula with n variables and m clauses and assume, without loss of generality, that F contains no repeated clauses, no empty clauses and no unused variables. Then $n \leq 3m$ and each literal can be represented by $\log n + 1$ bits. Hence, $\|F\| \leq 3m(\log n + 1) \leq 3m(\log(3m) + 1) = 3m(\log 3 + \log m + 1) \leq 3m(\log m + 3) \leq 12m \log m$, for $m \geq 2$. Note that an actual representation would also need to either first state the number of variables or use separator symbols or similar. For simplicity, we ignore this minor overhead since it does not matter except for very small instances.

Now suppose 3-SAT can be solved in time $O(2^{\|F\|^c})$ for some $c < 1$. We know that $\|F\| \leq 12m \log m$ when $m \geq 2$. Furthermore, $12m \log m < m^{1+\epsilon}$ for all $\epsilon > 0$ and large m . Choose $\epsilon = 1 - c$, which satisfies that $\epsilon > 0$ since $c < 1$. It then follows from the assumption that 3-SAT can be solved in time

$$O(2^{\|F\|^c}) \subseteq O(2^{m^{(1+\epsilon)c}}) = O(2^{m^d})$$

where $d = (1 + \epsilon)c = (1 + \epsilon)(1 - \epsilon) < 1$. This contradicts the ETH since 2^{m^d} grows slower than $2^{\delta m}$ for all $\delta > 0$. The case for 3-UNSAT is analogous. \square

Lemma 2 states that the ETH must be false if 3-SAT can be solved in time $O(2^{\|F\|^c})$ for some $c < 1$. However, it does not rule out the possibility that the ETH is still false even if 3-SAT cannot be solved in time $O(2^{\|F\|^c})$ for any $c < 1$. This is because $\|F\| \in \Theta(m \log m)$ and there is no c such that $m \in \Theta((m \log m)^c)$. If $c < 1$, then $(m \log m)^c$ grows strictly slower than m , and if $c \geq 1$, then $(m \log m)^c$ grows strictly faster than m . Hence, this lemma uses a somewhat weaker subexponentiality concept than the ETH, but it is still very powerful and useful for our purposes.

2.3 Optimisation and approximation

An *optimisation problem* takes an instance and generates a solution that is optimal according to some specified *measure*, for instance, plan length. It can be either a *minimisation problem* or a *maximisation problem*, depending on if the measure should be minimised or maximised. Alternatively, we may ask only for the actual measure of an optimal solution and not for the solution itself. This is known as *non-constructive optimisation* and it is the approach taken in this paper, as discussed in Section 2.1.

Let opt be a function that denotes the measure of an optimal solution for an instance. For a minimisation problem, we say that a function f approximates opt within a factor α if

$$\text{opt}(I) \leq f(I) \leq \alpha \cdot \text{opt}(I)$$

for all instances I . That is, the function f never underestimates the optimal value but always returns a value which is at most a factor α larger than the optimal value. For a maximisation problem, we similarly say that f approximates opt within α if

$$\frac{\text{opt}(I)}{\alpha} \leq f(I) \leq \text{opt}(I)$$

for all I . The approximation factor α is often also a function of the instance. Furthermore, the purpose of approximation is usually that the approximation function f should be easier to compute than the actual optimum opt . Formally, this can be defined as follows for the case of minimisation.

Definition 3 Let X be some set and let $g : X \rightarrow \mathbb{N}_\infty$ (where $\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}$) be an arbitrary function. We say that g is **NP**-hard if $\mathbf{NP} \subseteq \mathbf{P}^g$ (i.e. if every **NP**-complete problem can be solved by some polynomial-time algorithm having oracle access to g).

Let $g : X \rightarrow \mathbb{N}_\infty$ be a function that we wish to minimise. Let $f : X \rightarrow \mathbb{N}_\infty$ and $\alpha : X \rightarrow \mathbb{N}_\infty$ be functions. We say that f approximates g within α if for every $I \in X$,

$$g(I) \leq f(I) \leq \alpha(I) \cdot g(I).$$

We additionally say that g is *NP-hard to approximate within α* if every function f that approximates X within α is **NP**-hard.

Typically, α is a function of some aspect of the instance, e.g. its size or the number of variables. Also note that $\mathbf{NP} \subseteq \mathbf{P}^g$ if and only if $\mathbf{coNP} \subseteq \mathbf{P}^g$, so there is no need to consider “**coNP**-hardness” in this case.

2.4 Heuristics and approximation

Let $P = \langle V, A \rangle$ be a weighted PSN frame with weight function w and let s_G be a goal, i.e. $s_G \subseteq L(V)$ and s_G is consistent. Let S be the state space for P and let $S_G \subseteq S$ be the set of all states in S that satisfy s_G . Define the *cost function* $c : S \rightarrow \mathbb{N}_\infty$ for P such that for every state $s \in S$,

$$c(s) = \min\{w(\omega) \mid \omega \text{ is a plan from } s \text{ to some } t \in S_G\}$$

or $c(s) = \infty$ if there is no such plan. Computing $c(s)$ is obviously equivalent to solving the CPO problem for the instance $\langle V, A, s, s_G, w \rangle$ and it is thus a minimisation problem.

A *heuristic function* h for a cost function c is itself a cost function that is intended to estimate c while being presumably more efficient to compute. We further say that h is an *admissible* heuristic for c if $h(s) \leq c(s)$ for all $s \in S$, i.e. if h never overestimates c . This is a useful criterion since many heuristic search algorithms are optimal if the heuristic function is admissible, cf. the A^* algorithm [15].

Many heuristic functions are based on the principle of estimating the cost function for an instance by computing the cost function in a relaxed problem instance. Although the details vary, the basic principle is as follows. Let $P = \langle V, A \rangle$ be a weighted PSN frame with cost function c and let s_G be a goal. Another frame $P' = \langle V', A' \rangle$ with cost function c' is first computed. This should be a somehow relaxed version of P that satisfies that $c'(s) \leq c(s)$ for all $s \in S$. Hence, c' can be used as an admissible heuristic for c . Examples of this type

of abstraction are relaxation by state abstraction (projection of an instance to a subset of the variables) and delete relaxation (relaxation by removing all negative effects in the actions), which is also known as the h^+ heuristic [18]. Since computing the cost function is equivalent to computing CPO it is interesting also for this type of heuristics to study the complexity of the CPO problem. Furthermore, even some heuristics are intractable (computing h^+ is **NP**-hard [8]) which makes approximation relevant also for heuristics.

Computing an admissible heuristic function h has both a minimisation and a maximisation aspect. It is a minimisation problem in the sense that it estimates the function c , which should be minimised. However, since admissibility requires that $h(s) \leq c(s)$ we clearly do not want to minimise $h(s)$ any further than we minimise $c(s)$, so we do, in a sense, also want to maximise $h(s)$ under the constraint that $h(s) \leq c(s)$. We can alternatively view this as a multi-objective minimisation where we simultaneously minimise both the value of $h(s)$ and the difference $c(s) - h(s)$. Both these approaches are problematic, though, so it seems more reasonable to minimise $h(s)$ and let the choice of approximation factor α reflect the minimisation of $c(s) - h(s)$.

First suppose we want to use an approximation function f of c within some factor α as a heuristic function. We must then minimise f since c should be minimised. However, this means that f must satisfy the condition

$$c(s) \leq f(s) \leq \alpha \cdot c(s),$$

for all states $s \in S$. Hence, f is not an admissible heuristic. We can achieve admissibility, though, if we divide by α , which yields

$$\frac{c(s)}{\alpha} \leq \frac{f(s)}{\alpha} \leq c(s),$$

and which is admissible. Although this has the form of a maximisation problem, we do actually minimise f and then divide by α .

If we already have an admissible heuristic function h and want to approximate this, presumably because it is still too expensive to compute, then we proceed in the same way. We use a function f that approximates h within some factor α and minimise f , which yields

$$h(s) \leq f(s) \leq \alpha \cdot h(s).$$

We are guaranteed that $h(s) \leq c(s)$, since h is an admissible heuristic, so it is possible that $f(s) \leq c(s)$, but there is no such guarantee in general. To ascertain that f is admissible we can, thus, do as above and divide by α , resulting in

$$\frac{h(s)}{\alpha} \leq \frac{f(s)}{\alpha} \leq h(s) \leq c(s).$$

Just as in the previous case, this looks like a maximisation, but is still a minimisation followed by a scaling with α .

Hence, it is reasonable to treat admissible heuristics in the same way as plan optimisation and approximate them as minimisation problems, making the results that follow relevant in both these cases.

3 POSITIVE PRECONDITIONS AND THE h^+ HEURISTIC

Many successful heuristics for planning are based on the h^+ heuristic [18], also known as the *delete relaxation* or as *ignoring delete lists*. This heuristic is computed by creating a new relaxed planning instance which is identical to the original one, except that all negative effects are removed from the actions. The heuristic value for

the original instance is then computed as the optimal plan length (or plan weight) for the relaxed instance. Unfortunately, computing h^+ is **NP**-complete [8] even if actions are allowed to have at most one precondition and one effect, so in practice it is usually further approximated in one way or another. We refer the reader to Betz and Helmert [7] for an comprehensive overview of approaches to approximate h^+ . The reader should be aware that most of these approaches are not approximability results in the strict sense of Sections 2.3 and 2.4; they are heuristic functions with unclear theoretical properties. We will study the time complexity of computing and approximating h^+ in this section. In fact, we will consider approximation of the even simpler case where the actions do not even have any preconditions. Let C_{0+} denote the set of solvable PSN instances where the actions are restricted to have no preconditions and only positive effects. Observe that restricting ourselves to solvable instances make the forthcoming results even stronger.

We will base the forthcoming proof on the **NP**-complete SET COVER problem [17, SP5].

SET COVER

INSTANCE: A finite set U and a collection S of subsets of U .

OBJECTIVE: Find a smallest subset $S' \subseteq S$ that covers U , i.e. $U = \bigcup S'$

We assume without loss of generality that $\bigcup S = U$ for SET COVER instances. Let $f_{sc}(\langle U, S \rangle) = \min\{|S'| \mid S' \subseteq S \text{ and } S' \text{ covers } U\}$, that is, f_{sc} is a function that gives the size of the smallest cover for a SET COVER instance $\langle U, S \rangle$. The following hardness result for approximation of SET COVER is known from the literature.

Theorem 1 (Dinur and Steurer [16]). *It is **NP**-hard to approximate f_{sc} within any α satisfying $\alpha(\langle U, S \rangle) = (1 - o(1)) \cdot \ln |U|$.*

The factor $1 - o(1)$ tends to 1 so, for example, approximating f_{sc} within $c \cdot \ln |U|$ for any $0 < c < 1$ is **NP**-hard. We can use this result to prove an analogous result for $LPO(C_{0+})$.

Theorem 2 *It is **NP**-hard to approximate $LPO(C_{0+})$ within any factor α such that $\alpha(\langle V, A, s_I, s_G \rangle) = (1 - o(1)) \cdot \ln |V|$.*

Proof. Proof by optimum-preserving polynomial-time reduction from SET COVER. Let $\langle U, S \rangle$ denote an arbitrary instance of SET COVER and define a corresponding PSN instance $P = \langle V, A, s_I, s_G \rangle$ such that

1. $V = \{v_u \mid u \in U\}$;
2. $A = \{a_s \mid s \in S\}$, where $a_s : \emptyset \Rightarrow \{v_u \mid u \in s\}$ for $s \in S$;
3. $s_I = \emptyset$ and
4. $s_G = \{v_u \mid u \in U\}$.

Obviously, $P \in C_{0+}$ since we always assume that $\bigcup S = U$. It is easy to see that the shortest plan for P has exactly as many actions as the size of a minimum cover for U and that $|V| = |U|$. Hence, this is a reduction that preserves optimal solutions, so it follows from Theorem 1 that $LPO(C_{0+})$ is **NP**-hard to approximate within α . \square

We continue by presenting a general result for “lifting” **NP**-hardness results for plan optimisation (such as Theorem 2) into lower bounds on time complexity. The reader should note that the set C can be chosen completely arbitrarily: the restriction to solvable instances only make the result stronger.

Theorem 3 *Let C be a set of solvable PSN instances and let $\alpha : C \rightarrow \mathbb{N}$ be a function. Then, at least one of the following cases holds:*

1. *approximating $LPO(C)$ within α is not **NP**-hard,*
2. *there exists some $c > 0$ such that $LPO(C)$ cannot be approximated within α by any algorithm running in time $O(2^{\|P\|^c})$ or*
3. *the ETH is not true.*

Proof. If it is not **NP**-hard to approximate $LPO(C)$ within α , then we are in case 1. Otherwise, we may assume that every function f that approximates $LPO(C)$ within α is **NP**-hard. Let f be such a function. If there is some $c > 0$ such that f cannot be computed in time $O(2^{\|P\|^c})$, then we are in case 2.

In case 3, the remaining possibility is that there is such a function f that can be computed in time $O(2^{\|P\|^c})$ for all $c > 0$. We know that **NP** \subseteq **P** ^{f} , since f is **NP**-hard, so there is some polynomial-time algorithm \mathcal{A} with oracle access to f that solves 3-SAT. Since \mathcal{A} runs in polynomial time there must be some $k > 1$ such that every input string x to the oracle f satisfies that $\|x\| \leq \|F\|^k$, where F is the input to \mathcal{A} . Choose an arbitrary c such that $0 < c < 1/k$. Let \mathcal{B} be an algorithm that approximates $LPO(C)$ within α and runs in time $O(2^{\|P\|^c})$. Such an algorithm must exist according to the assumption. Let $\mathcal{A}^{\mathcal{B}}$ denote algorithm \mathcal{A} where every oracle call is computed by algorithm \mathcal{B} . There is then some polynomial p such that $\mathcal{A}^{\mathcal{B}}$ runs in time

$$\begin{aligned} O(p(\|F\|) \cdot 2^{(\|F\|^k)^c}) &= O(p(\|F\|) \cdot 2^{\|F\|^{kc}}) \\ &\subseteq O(2^{\|F\|^{kc+\epsilon}}), \end{aligned}$$

for all $\epsilon > 0$. Hence, we can choose ϵ such that $0 < \epsilon < 1 - kc$, since $kc < 1$. However, this contradicts the ETH according to Lemma 2, since $kc + \epsilon < 1$. \square

The value of c can be estimated as follows. First note that $c < 1/k$ and that c , without loss of generality, can be chosen arbitrarily close to $1/k$. Since $\|F\|^k$ is an upper bound on the length of oracle questions, c can be estimated as soon as we have an explicit bound on the length of oracle questions. Obtaining such a bound may of course be more or less difficult depending on the set C of PSN instances.

We arrive at the following result by combining Theorems 2 and 3.

Corollary 4 *Let $\alpha : \text{PSN} \rightarrow \mathbb{N}$ be a function satisfying $\alpha(\langle V, A, s_I, s_G \rangle) = (1 - o(1)) \cdot \ln |V|$. If the ETH holds, then there exists some $c > 0$ such that neither $LPO(C_{0+})$ nor the h^+ heuristic can be approximated within α in time $O(2^{\|P\|^c})$.*

If we merely assume that **P** \neq **NP** then it could be the case that $LPO(C_{0+})$ is approximable within α in time $O(2^{\|P\|^c})$ for every $c > 0$. The function $2^{\|P\|^c}$ grows faster than any polynomial so this fact would not allow us to infer that **P** = **NP**. Thus, the ETH (or some other hypothesis that gives quantitative bounds on time complexity) is instrumental for proving results like Corollary 4.

Stronger bounds for h^+ than those presented in Corollary 4 are probably obtainable by taking instances with preconditions into consideration. We leave this as a plausible research direction. One may note that Betz and Helmert [7] have proven that it is **NP**-hard to approximate $LPO(C_{1+,1+})$ within any constant $c > 0$ where $C_{1+,1+}$ contains all PSN instances where the actions have at most 1 positive precondition and at most 1 positive effect. Theorem 3 implies that for every constant $c > 0$, there exists a $d > 0$ such that $LPO(C_{1+,1+})$ cannot be approximated within c by any algorithm running in time $O(2^{\|F\|^d})$ or faster, unless the ETH does not hold.

Another observation worth making is the following: the result in Corollary 4 is close to optimal for the $LPO(C_{0+})$ problem. It is known that SET COVER can be approximated within $1 + \ln|U|$ [21], which implies that $LPO(C_{0+})$ can be approximated within the same bound. To see this, pick an arbitrary instance $P = \langle V, A, s_I, s_G \rangle$ in C_{0+} . We first preprocess P by repeatedly checking the following cases until none of them is applicable:

1. If there is a variable $v \in V$ such that $v \in s_I$ and $\bar{v} \in s_G$, then P has no solution since no action can make v false. However, this is impossible since $P \in C_{0+}$ and it must thus be solvable.
2. If there is a variable $v \in V$ such that $v \notin s_I$, $v \in s_G$ and there is no action $a \in A$ such that $v \in \text{eff}(a)$, then P has no solution, which is impossible since $P \in C_{0+}$.
3. If there is a variable $v \in V$ such that $v \in s_I$ and $v \in s_G$, then variable v is superfluous and can be removed, since v could never become false.
4. If there is a variable $v \in V$ such that $v \in s_I$ and $\{v, \bar{v}\} \cap s_G = \emptyset$, then variable v is superfluous and can be removed, since all actions have empty preconditions.
5. If there is a variable $v \in V$ such that $v \notin s_I$, $\bar{v} \in s_G$ and there is an action $a \in A$ such that $v \in \text{eff}(a)$, then a cannot be used in any solution and can be removed.

Let the resulting instance be $P' = \langle V', A', s_I', s_G' \rangle$. Note that P' can be computed in polynomial time, P' is solvable, $LPO(P) = LPO(P')$, $s_I' = \emptyset$, and $s_G' = \{v \mid v \in V'\}$. Finally, construct the SET COVER instance $\langle U, S \rangle$ such that $U = V'$ and $S = \{\text{eff}(a') \mid a' \in A'\}$. Then $f_{sc}(\langle U, S \rangle) = LPO(P')$ and the approximability result follows immediately.

Cygan et al. [14] show that improved approximations for SET COVER can be obtained by using algorithms running in low-order exponential time. The transformation above shows that such algorithms are relevant for $LPO(C_{0+})$, too. This indicates that superpolynomial algorithms for obtaining better approximations ought to be further investigated with respect to planning heuristics.

4 POSITIVE AND NEGATIVE PRECONDITIONS

We will now continue with a more expressive class of planning instances without negative effects. Let C_+ denote the set of solvable, weighted PSN instances where the actions are restricted to have positive effects only. Although weighted actions are often considered in the context of h^+ , it is not so common to allow negative preconditions. In that way, C_+ is more expressive than the class of instances usually considered for h^+ .

We will base the forthcoming proof on a reduction from the following NP-complete problem [17, problem GT39].

DIRECTED HAMILTON PATH (DHP)

INSTANCE: A directed graph $G = \langle U, E \rangle$ and vertices $s, t \in U$.

QUESTION: Does G contain a directed Hamilton path from s to t , i.e. a path that starts in s and ends in t and visits every vertex in U exactly once?

We now prove that $CPO(C_+)$ is quite more difficult to approximate than $LPO(C_{0+})$.

Theorem 4 *Let p be an arbitrary polynomial. Then it is NP-hard to approximate $CPO(C_+)$ within α where $\alpha(\langle V, A, I, G \rangle) = 2^{p(|V|)}$.*

Proof. Let p be an arbitrary polynomial. Assume f is a function that approximates $CPO(C_+)$ within $2^{p(|V|)}$. We define a reduction from DHP to $CPO(C_+)$ in two steps as follows. Consider an arbitrary instance of DHP defined by a graph $G = \langle U, E \rangle$ and vertices $s, t \in U$. Without losing generality, assume that $U = \{v_1, \dots, v_n\}$, $s = v_1$ and $t = v_n$.

For the first step, define the weighted directed graph $G^* = \langle U, E^* \rangle$ such that $E^* = \{\langle v_i, v_j \rangle \mid v_i, v_j \in U \text{ and } i \neq j\}$. Then $E \subseteq E^*$, so define the edge weight function $w : E^* \rightarrow \mathbb{N}$ such that $w(e) = 1$ if $e \in E$ and $w(e) = 2^{p(4n)}$ otherwise. Clearly, G^* always has a Hamilton path from v_1 to v_n . Furthermore, if G has a Hamilton path from v_1 to v_n , then G^* has a Hamilton path from v_1 to v_n of weight $n - 1$. On the other hand, if G does not have any Hamilton path from v_1 to v_n , then any such path in G^* must contain at least one edge e with weight $w(e) = 2^{p(4n)}$, so G^* cannot have any Hamilton path from v_1 to v_n of weight less than $2^{p(4n)}$ in this case.

For the second step, construct a corresponding PSN instance $P = \langle V, A, s_I, s_G \rangle$ such that

- $V = \{in_i, out_i, pass_i \mid 1 \leq i \leq n\}$;
- $A = \{a_{i,j} \mid \langle v_i, v_j \rangle \in E^*\} \cup \{b_i \mid v_i \in U\}$, where
 - $a_{i,j} : \{in_i, \overline{out_i}, \overline{in_j}\} \Rightarrow \{out_i, in_j\}$, with weight $w(\langle v_i, v_j \rangle)$, and
 - $b_i : \{in_i, out_i\} \Rightarrow \{pass_i\}$, with weight 1;
- $s_I = \{in_1, out_n\}$ and
- $s_G = \{pass_i \mid 1 \leq i \leq n\}$

Clearly, P is a member of C_+ . The variables and actions have the following purposes. For each vertex v_i , variable in_i is true if we have ever arrived at v_i , variable out_i is true if we have also left v_i and variable $pass_i$ is true if we have ever visited v_i , i.e. if we have both arrived at v_i and left v_i . Action $a_{i,j}$ traverses the edge $\langle v_i, v_j \rangle$, i.e. it moves from vertex v_i to vertex v_j , and action b_i checks that we have visited vertex v_i . We need to prove that P has a plan of weight $w + n$ if and only if G^* has a Hamilton path from v_1 to v_n of weight w .

if: Let σ be a Hamilton path from v_1 to v_n in G^* with weight $w(\sigma)$ and assume without losing generality, that $\sigma = \langle v_1, \dots, v_n \rangle$. Then there is a plan ω for P such that $\omega = \langle a_{1,2}, a_{2,3}, \dots, a_{(n-1),n}, b_1, \dots, b_n \rangle$. Furthermore,

$$\begin{aligned} w(\omega) &= w(\langle a_{1,2}, a_{2,3}, \dots, a_{(n-1),n} \rangle) + w(\langle b_1, \dots, b_n \rangle) \\ &= w(\sigma) + n. \end{aligned}$$

only if: Suppose ω is a plan for P . Clearly, ω must contain each of the actions b_1, \dots, b_n in order to set variables $pass_1, \dots, pass_n$. We can then assume, without losing generality, that ω is of the form $\omega = \langle a_{i_1, j_1}, \dots, a_{i_\ell, j_\ell}, b_1, \dots, b_n \rangle$, since no action has any of the $pass_i$ variables in its precondition and the b_i actions do not depend on each other. For arbitrary i ($1 \leq i \leq n$), suppose action $a_{i,j}$ is in ω for some j . This action requires that out_i is false and changes it to true. Since there is no action that can make out_i false again, there can only be one j such that $a_{i,j}$ is in ω and this action can only appear once. Hence, the sequence $\omega' = \langle a_{i_1, j_1}, \dots, a_{i_\ell, j_\ell} \rangle$ contains at most $n - 1$ actions (it cannot contain any action of type $a_{n,j}$). However, ω' must set all of the variables out_1, \dots, out_{n-1} to true, so for each i ($1 \leq i < n$), there is some k such that it contains action $a_{i,k}$. Hence, ω' contains exactly $n - 1$ actions. This is only possible if $i_1 = 1$, $j_\ell = n$ and $j_k = i_{k+1}$ for all k ($1 \leq k < n$). It follows that ω' corresponds to the Hamilton path $\sigma = v_1, v_{i_2}, \dots, v_{i_\ell}, v_n$ in G^* .

We thus get that

$$\begin{aligned} w(\sigma) &= w(\langle a_{i_1, j_1}, \dots, a_{i_\ell, j_\ell} \rangle) \\ &= w(\omega) - w(\langle b_1, \dots, b_n \rangle) \\ &= w(\omega) - n, \end{aligned}$$

i.e. $w(\omega) = w(\sigma) + n$.

Combining this second step with the relationship between G and G^* and the assumption about f (noting that $|V| = 3n$) yields that:

1. If G has a Hamilton path from v_1 to v_n , then $\text{CPO}(P) = 2n - 1$, so $f(P) \leq 2^{p(3n)}(2n - 1)$.
2. If G has no Hamilton path from v_1 to v_n , then $\text{CPO}(P) \geq 2^{p(4n)}$, so $2^{p(4n)} \leq f(P)$.

Since $2^{p(3n)}(2n - 1) < 2^{p(4n)}$ for sufficiently large values of n , there is a gap such that the value of $f(P)$ can be used to decide if G has a Hamilton path or not; if $f(P) < 2^{p(4n)}$, then G has a Hamilton path from v_1 to v_n , and otherwise G does not have such a path.

The construction above is thus a reduction from DHP to $\text{CPO}(C_+)$ and it is obviously computable in polynomial time. Since also f is assumed to be polynomial-time computable, it follows that we can solve DHP in polynomial time, which is impossible unless $\mathbf{P} = \mathbf{NP}$. Hence, it is \mathbf{NP} -hard to approximate $\text{CPO}(C_+)$ within $2^{p(|V|)}$. \square

Since $C_{0+} \subseteq C_+$ and LPO is a special case of CPO this result involves two additional difficulties compared to the previous theorem. We might, thus, also wonder about the approximability of $\text{LPO}(C_+)$. We only measure the number of actions in this case since every optimal plan can have at most $|V|$ actions when there are no negative effects, since every action in the plan must change at least one variable to true. It immediately follows that $\text{LPO}(C_+)$ can always be approximated within the factor $|V|$; it is trivial to see if the empty plan solves the instance and if not, then the answer $|V|$ will be an approximation within $|V|$ for all plan lengths from 1 to $|V|$. Hence, Theorem 4 does not hold for the LPO case. It is important that we work with non-constructive approximations here: the complexity of computing a concrete solution that approximates within $|V|$ is currently unknown.

We now combine Theorems 3 and 4 to get a quantitative lower bound on the approximability of $\text{CPO}(C_+)$.

Corollary 5 *Arbitrarily choose a polynomial p . If the ETH holds, then there exists some $c > 0$ such that $\text{CPO}(C_+)$ cannot be approximated within $2^{p(|V|)}$ in time $O(2^{||P||^c})$.*

The approximability bound is still valid under severe restrictions on the number of preconditions and effects; all actions a in the reduction above satisfy that $|\text{pre}(a)| \leq 3$ and $|\text{eff}(a)| \leq 2$.

5 GENERAL COST-OPTIMAL PLANNING

We finally turn our attention to the approximability of PSN planning in general. This main result of this section (Theorem 5) goes further than most other approximation results since we obtain a functional relation between approximation bounds and time bounds (and not merely a bound saying that a certain degree of approximability can or cannot be achieved by utilising a certain amount of computational resources). Thus, the result applies to a whole spectrum of approximation bounds and describes a trade-off between approximation bound and running time for achieving this bound. The proof is based on a combination of two results which we present next. The first result is known from the literature.

Lemma 6 (Section 5 in Jonsson [22]) *Arbitrarily choose an integer $c > 1$. There exists a polynomial-time computable function $\rho_c : \text{PSN} \rightarrow \text{PSN}$ with the following properties: for an arbitrary instance $P = \langle V, A, s_I, s_G \rangle$ of $\text{PE}(\text{PSN})$, it holds that:*

1. $\rho_c(P)$ contains $|V|^c$ variables;
2. $\rho_c(P)$ is solvable;
3. if P has a solution, then $\rho_c(P)$ has a solution of length $1 + 2^{|V|}$, or shorter, and
4. if P has no solution, then all solutions for $\rho_c(P)$ are longer than $2^{|V|^c - |V|}$.

The second result is a particular reduction from 3-UNSAT to $\text{PE}(\text{PSN})$ with a very advantageous property: if the given formula contains n variables, then the resulting $\text{PE}(\text{PSN})$ instance contains only $n + 1$ variables. We first recall how to encode an n -bit binary counter in PSN [4]. Let $V = \{x_1, \dots, x_n\}$ and let A contain the n actions

$$c_i : \{x_1, \dots, x_{i-1}, \bar{x}_i\} \Rightarrow \{\bar{x}_1, \dots, \bar{x}_{i-1}, x_i\} \quad (1 \leq i \leq n).$$

Then use V and A to construct a PSN instance $P = \langle V, A, s_I, s_G \rangle$ such that $s_I = \emptyset$ and $s_G = \{x_i \mid 1 \leq i \leq n\}$. This instance is always solvable and it has a shortest plan of length $2^n - 1$ actions that corresponds to a Hamiltonian path from s_I to s_G in the state-transition graph of P .

Construction 7 Define a function $\rho : 3\text{-UNSAT} \rightarrow \text{PSN}$ as follows. Let F be an instance of 3-UNSAT with n variables $\{x_1, \dots, x_n\}$ and m clauses $\{C(F)_1, \dots, C(F)_m\}$. Assume without loss of generality that F contains no clause $C(F)_j$ that is a tautology, i.e. both x_i and \bar{x}_i appear in $C(F)_j$ for some $1 \leq i \leq n$. We construct a corresponding PSN instance $\rho(F) = P_F = \langle V, A, s_I, s_G \rangle$ as follows:

- Let $V = \{x_1, \dots, x_{n+1}\}$.
- Let c_1, \dots, c_{n+1} denote the actions in the $n + 1$ -bit counter over the variables x_1, \dots, x_{n+1} . For each clause $C(F)_j = (\ell_1 \vee \ell_2 \vee \ell_3)$ of F , where $1 \leq j \leq m$, define $T_j = \{\ell_1, \ell_2, \ell_3\}$. Let $A = \{a_{i,j} \mid 1 \leq i \leq n + 1, 1 \leq j \leq m\}$, where the actions are defined as $a_{i,j} : \text{pre}(c_i) \cup T_j \Rightarrow \text{eff}(c_i)$. One may view $a_{i,j}$ as action c_i in a binary counter extended with preconditions saying “clause j is not satisfied by x_1, \dots, x_n ”.
- Let $s_I = \emptyset$.
- Let $s_G = \{\bar{x}_1, \dots, \bar{x}_n, x_{n+1}\}$.

This construction is indeed a polynomial reduction from 3-UNSAT to $\text{PE}(\text{PSN})$.

Lemma 8 *The function ρ in Construction 7 is a polynomial reduction from problem 3-UNSAT to $\text{PE}(\text{PSN})$.*

Proof. Let F be a 3-UNSAT instance and let $P_F = \rho(F)$ be the corresponding PSN instance according to Construction 7. If we treat the variables x_1, \dots, x_{n+1} as encoding a binary number, then every plan must count through all numbers from 0 to 2^n , since these numbers correspond to the initial and goal states and there are only counting actions. Furthermore, for each state $s \in S(P_F)$, only one of the different types c_1, \dots, c_{n+1} of counter actions is valid in s and there are m variants of each such action, one for each clause of F . Hence, for each step in a plan for P_F , there are m different actions to choose from. An action corresponding to a clause $C(F)_j$ of F has a precondition that requires all literals in $C(F)_j$ to be false, i.e. the action is valid only if $C(F)_j$ is false in the current variable

assignment specified by variables x_1, \dots, x_n . This means that for each assignment, i.e. for each state, it is necessary to find an action corresponding to a clause of F that is false for that assignment. In other words, a plan for P_F must verify that for every assignment to x_1, \dots, x_n , at least one clause of F is false, i.e. that F is unsatisfiable. Hence, ρ is a reduction from 3-UNSAT to PE(PSN) and it is obvious that it is polynomial-time computable. \square

We are now ready to prove the main theorem.

Theorem 5 *Arbitrarily choose an integer $c > 1$. If f is a function that can approximate LPO(PSN) within $2^{|V|-2|V|^{1/c}-1}$ then f cannot be computed in time $O(2^{|V|^{1/c'}})$ for any $c' > c$, unless the ETH is false.*

Proof. Suppose f can be computed in time $O(2^{|V|^{1/c'}})$ for some $c' > c$. For simplicity, we show that f cannot approximate LPO(PSN) within $\frac{2^{|V|-|V|^{1/c}}}{2^{|V|^{1/c}+2}}$. We can do this without loss of generality since this bound is slightly more generous than the original bound:

$$\frac{2^{|V|-|V|^{1/c}}}{2^{|V|^{1/c}+2}} \geq \frac{2^{|V|-|V|^{1/c}}}{2^{|V|^{1/c}} \cdot 2} = 2^{|V|-2|V|^{1/c}-1}.$$

We know from Lemma 8 that a 3-UNSAT instance F with n variables can be transformed in polynomial time into a PSN instance $P_F = \langle V, A, s_I, s_G \rangle$ that has a solution if and only if F has no solution, and where $|V| = n + 1$. We consider $P' = \rho_c(P_F) = \langle V', A', s_I', s_G' \rangle$, where the function ρ_c is as defined in Lemma 6. We know that P' can be computed in polynomial time (in the size of F) and that $|V'| = |V|^c$. If P_F has a solution, then

$$\begin{aligned} f(P') &\leq \frac{2^{|V'|-|V'|^{1/c}}}{2^{|V'|^{1/c}+2}} \cdot (1 + 2^{|V|}) = \frac{2^{|V|^c-|V|}}{2^{|V|+2}} \cdot (1 + 2^{|V|}) \\ &< 2^{|V|^c-|V|}, \end{aligned}$$

while if P_F has no solution, then

$$f(P') \geq 2^{|V|^c-|V|}.$$

Thus, the value of $f(P')$ can be used to decide if F is satisfiable or not. By assumption, f can be computed in time $O(2^{|V'|^{1/c'}})$, for some $c' > c$, which implies that the satisfiability of F can be decided in time $O(2^{\binom{(n+1)^c}{1/c'}}) = O(2^{n^{c/c'}})$. However, this contradicts the ETH (via Lemma 2) since $c/c' < 1$. \square

The instances that result from applying ρ are always satisfiable so Theorem 5 still holds if we restrict ourselves to solvable PSN instances. This implies that Theorem 5 is directly comparable to Corollaries 4 and 5.

6 DISCUSSION

We have demonstrated how the approximability of cost-optimal planning can be analysed from the perspective of time complexity. Furthermore, we have argued that such results are important when it comes to understanding the performance of heuristics such as h^+ . Since h^+ has had such a deep impact on the field of planning, it makes sense at this point to discuss some future research directions that are directly related to this heuristic.

Our nonapproximability results for h^+ apply to a very large and unstructured class of planning instances since we do not enforce any restrictions besides positive effects and (in Section 3) positive preconditions. The h^+ heuristic is known to have better performance when applied to more structured sets of instances. For instance, Betz and Helmert [7] report that h^+ for the domain LOGISTICS is NP-hard to compute exactly but it can be polynomial-time approximated within some constant c . Now recall that Theorem 3 gives a time bound whenever PE(C) is NP-hard. Thus, by letting C contain the LOGISTICS instances, Theorem 3 is applicable (with $\alpha = 1$) and we conclude that there exists a $c > 0$ such that computing h^+ for LOGISTICS takes at least $2^{\|F\|^c}$ time. If it is the case that h^+ for LOGISTICS cannot be polynomial-time approximated within every constant (i.e., it does not have a polynomial-time approximation scheme), then we can also conclude that there exist constants $c' > 0$ and $d > 1$ such that it takes at least $2^{\|F\|^{c'}}$ time to compute an approximation within d . Unfortunately, we cannot immediately infer any trade-off results similar to Theorem 5 since we do not have suitable constructions analogous to Construction 7 and the construction in Lemma 6. We view the adaptation of Theorem 5 to various restricted domains as an interesting research direction.

We know from Section 3 that whenever $\alpha = (1 - o(1)) \cdot \ln(|V|)$, then approximating h^+ within α takes at least $O(2^{\|P\|^c})$ time for some $c > 0$ that depends on α . If the instances under consideration are required to have actions without preconditions and we assume that all actions have equal weight, then this bound is essentially optimal since the problem can be approximated within $1 + \ln(|V|)$ in polynomial time. If actions with preconditions are allowed, then we do not know much about polynomial-time approximability. In fact, we basically only have the trivial bound $|V|$ that was pointed out after the proof of Theorem 4 and the hardness result for LPO($C_{1+,1+}$) by Betz and Helmert [7]. If actions with different positive weights are allowed (but actions have no preconditions), then the problem can be approximated within $1 + \ln(|V|)$ by using Chvátal's [13] algorithm for weighted SET COVER. Obtaining sharper bounds is thus an interesting research direction. Approximation algorithms for NP-complete optimisation problems have been studied intensively for a long time and the toolbox has become very powerful. This is evidence for the feasibility of such a project. In the same vein, it would be interesting to obtain lower bounds on polynomial-time approximability of h^+ . Once again, there is a powerful toolbox available.

For many different reasons, it may not always be acceptable to approximate h^+ within the bound that is obtainable in polynomial time. It would thus be interesting to obtain time bounds on approximating h^+ within, for instance, $d \cdot \ln(|V|)$ for some fixed constant $0 < d < 1$ or (the probably more interesting case) of approximating h^+ within some fixed constant. In both cases, we know that there exists some $c > 0$ (that depends on the desired bound) such that computing the approximation takes at least $O(2^{\|P\|^c})$ but we do not know the exact value (nor do we have a close estimate) of c . We have pointed out (after the proof of Theorem 3) that bounds on c can be obtained by analysing connections between 3-SAT and the planning problem at hand. A problem with this approach is that the underlying constructions may be highly involved. This is, for instance, the case for the SET COVER result underlying Corollary 4. Another obstacle that we want to emphasise is that the exact representation of planning instances may be important in this kind of analyses. The oracle function in the proof always take a planning instance as input and the size of this instance (which is directly dependent on the chosen representation) has a strong influence on the time bound.

REFERENCES

- [1] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi, *Complexity and Approximation*, Springer, 1999.
- [2] C. Bäckström and P. Jonsson, ‘A refined view of causal graphs and component sizes: SP-closed graph classes and beyond’, *J. Artif. Intell. Res.*, **47**, 575–611, (2013).
- [3] Christer Bäckström and Peter Jonsson, ‘All PSPACE-complete planning problems are equal but some are more equal than others’, in *Proc. 4th Annual Symposium on Combinatorial Search (SOCS-11)*, Castell de Cardona, Barcelona, Spain, pp. 10–17, (2011).
- [4] Christer Bäckström and Peter Jonsson, ‘Algorithms and limits for compact plan representations’, *J. Artif. Intell. Res.*, **44**, 141–177, (2012).
- [5] Christer Bäckström and Inger Klein, ‘Planning in polynomial time: the SAS-PUBS class’, *Comput. Intell.*, **7**, 181–197, (1991).
- [6] Christer Bäckström and Bernhard Nebel, ‘Complexity results for SAS+ planning’, *Comput. Intell.*, **11**, 625–656, (1995).
- [7] Christoph Betz and Malte Helmert, ‘Planning with h^+ in theory and practice’, in *Proc. Advances in Artificial Intelligence, 32nd Annual German Conference on AI (KI-09)*, Paderborn, Germany, volume 5803 of *Lecture Notes in Computer Science*, pp. 9–16. Springer, (2009).
- [8] Tom Bylander, ‘The computational complexity of propositional STRIPS planning’, *Artif. Intell.*, **69**(1-2), 165–204, (1994).
- [9] David Chapman, ‘Planning for conjunctive goals’, *Artif. Intell.*, **32**(3), 333–377, (1987).
- [10] J. Chen, X. Huang, I. Kanj, and G. Xia, ‘Strong computational lower bounds via parameterized complexity’, *J. Comput. Syst. Sci.*, **72**(8), 1346–1367, (2006).
- [11] Jianer Chen, Benny Chor, Mike Fellows, Xiuzhen Huang, David W. Juedes, Iyad A. Kanj, and Ge Xia, ‘Tight lower bounds for certain parameterized NP-hard problems’, *Inf. Comput.*, **201**(2), 216–231, (2005).
- [12] Y. Chen and M. Grohe, ‘An isomorphism between subexponential and parameterized complexity theory’, *SIAM J. Comput.*, **37**(4), 1228–1258, (2007).
- [13] V. Chvátal, ‘A greedy heuristic for the set covering problem’, *Math. Oper. Res.*, **4**, 233–235, (1979).
- [14] Marek Cygan, Lukasz Kowalik, and Mateusz Wykurz, ‘Exponential-time approximation of weighted set cover’, *Inf. Process. Lett.*, **109**(16), 957–961, (2009).
- [15] Rina Dechter and Judea Pearl, ‘Generalized best-first search strategies and the optimality of A^* ’, *J. ACM*, **32**(3), 505–536, (1985).
- [16] Irit Dinur and David Steurer, ‘Analytical approach to parallel repetition’, in *Proc. 46th ACM Symposium on Theory of Computing (STOC-2014)*, pp. 624–633, (2014).
- [17] Michael R. Garey and David S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, New York, 1979.
- [18] Jörg Hoffmann, ‘Where ‘ignoring delete lists’ works: Local search topology in planning benchmarks’, *J. Artif. Intell. Res.*, **24**, 685–758, (2005).
- [19] R. Impagliazzo, R. Paturi, and F. Zane, ‘Which problems have strongly exponential complexity?’, *J. Comput. Syst. Sci.*, **63**(4), 512–530, (2001).
- [20] Russell Impagliazzo and Ramamohan Paturi, ‘On the complexity of k -SAT’, *J. Comput. Syst. Sci.*, **62**(2), 367–375, (2001).
- [21] David S. Johnson, ‘Approximation algorithms for combinatorial problems’, *J. Comput. Syst. Sci.*, **9**(3), 256–278, (1974).
- [22] Peter Jonsson, ‘Strong bounds on the approximability of two PSPACE-hard problems in propositional planning’, *Annals Math. Artif. Intell.*, **26**, 133–147, (1999).
- [23] Peter Jonsson, Victor Lagerkvist, Gustav Nordh, and Bruno Zanuttini, ‘Complexity of SAT problems, clone theory and the exponential time hypothesis’, in *Proc. 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-13)*, New Orleans, LA, USA, pp. 1264–1277, (2013).
- [24] I. Kanj and S. Szeider, ‘On the subexponential time complexity of CSP’, in *Proc. 27th AAAI Conference on Artificial Intelligence (AAAI-13)*, Bellevue, WA, USA, pp. 459–465, (2013).
- [25] Michael Katz and Carmel Domshlak, ‘New islands of tractability of cost-optimal planning’, *J. Artif. Intell. Res.*, **32**, 203–288, (2008).
- [26] Daniel Lokshтанov, Dániel Marx, and Saket Saurabh, ‘Lower bounds based on the exponential time hypothesis’, *B. EATCS*, **105**, 41–72, (2011).
- [27] D. Marx, ‘On the optimality of planar and geometric approximation schemes’, in *Proc. 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS-07)*, Providence, RI, USA, pp. 338–348, (2007).
- [28] R. Santhanam and S. Srinivasan, ‘On the limits of sparsification’, in *Proceeding of the 39th International Colloquium on Automata, Languages, and Programming (ICALP-12)*, Warwick, UK, pp. 774–785, (2012).
- [29] Bart Selman, ‘Near-optimal plans, tractability, and reactivity’, in *Proc. 4th International Conference on Principles of Knowledge Representation and Reasoning (KR-94)*, pp. 521–529, (1994).
- [30] Richard Edwin Stearns, ‘Turing award lecture: It’s time to reconsider time’, *Commun. ACM*, **37**(11), 95–99, (1994).
- [31] Richard Edwin Stearns, ‘Deterministic versus nondeterministic time and lower bound problems’, *J. ACM*, **50**(1), 91–95, (2003).
- [32] Richard Edwin Stearns and Harry B. Hunt, III, ‘Power indices and easier hard problems’, *Math. Syst. Theory*, **23**(4), 209–225, (1990).
- [33] P. Traxler, ‘The time complexity of constraint satisfaction’, in *Proceeding of the 3rd International Workshop on Parameterized and Exact Computation (IWPEC-08)*, Victoria, BC, Canada, pp. 190–201, (2008).

A Simple Account of Multi-Agent Epistemic Planning

Martin C. Cooper, Andreas Herzig, Faustine Maffre, Frédéric Maris, Pierre Régnier¹

Abstract. A realistic model of multi-agent planning must allow us to formalize notions which are absent in classical planning, such as communication and knowledge. We investigate multi-agent planning based on a simple logic of knowledge that is grounded on the visibility of propositional variables. Using such a formal logic allows us to prove the existence of a plan given the description of the individual actions. We present an encoding of multi-agent planning problems expressed in this logic into the standard planning language PDDL. The solvability of a planning task is reduced to a model checking problem in a dynamic extension of our logic, proving its complexity. Feeding the resulting problem into a PDDL planner provides a provably correct plan for the original multi-agent planning problem. We apply our method on several examples such as the gossip problem.

1 Introduction

Suppose there are n agents each of which knows some secret: a piece of information that is not known to the others. They communicate by phone calls, and whenever one person calls another they tell each other all they know at that time. How many calls are required before each item of gossip is known to everyone? This *gossip problem* can be viewed as perhaps the simplest multi-agent planning problem: it is only the agents' knowledge that evolves, while the facts of the world remain unchanged. We develop a formal framework in which it is possible to express some interesting generalizations of this problem.

Dynamic Epistemic Logic DEL [24] provides a formal framework for the representation of knowledge and update of knowledge, and several recent approaches to multi-agent planning are based on it, starting with [5, 17]. While DEL provides a very expressive framework, it was unfortunately proven to be undecidable even for rather simple fragments of the language [1, 8]. Some decidable fragments were studied, most of which focused on public events [17, 25]. However, the gossip problem requires private communication. There exist other approaches on planning with uncertainty that do not use DEL. The framework presented in [16] allows us to reason about knowledge on literals in a multi-agent setting. A similar approach with beliefs can be found in [19]. While restricted to a single agent, the framework of [20] also deals with 'knowing whether' formulas (i.e., knowing p or knowing $\neg p$).

In this paper we provide a simple multi-agent epistemic logic that we call EL- O^S (Epistemic Logic of Observation), allowing us to model actions and epistemic planning tasks such as the gossip problem. Our logic provides special variables describing what agents can see. These variables determine indistinguishability relations, which

allow us to interpret arbitrary formulas containing epistemic operators in the standard way and to reduce them to boolean formulas. By extending EL- O^S with dynamic operators, we are able to formalize the existence of a plan, giving the complexity result. We also study an encoding of actions into PDDL, the standard Planning Domain Definition Language [18]. This allows us to find a plan efficiently with a PDDL planner, which we do with extensions of the gossip problem and with the 'exam problem' where truth values of facts can also evolve.

The paper is organized as follows. In Section 2 we introduce our epistemic logic EL- O^S . In Section 3 we give a formal definition of actions and planning tasks within our framework. In Section 4 we show how the existence of a plan can be encoded in the extension of EL- O^S with dynamic operators, and give the complexity result. In Section 5 we present the encoding into PDDL. In Section 6 we apply our framework to examples. We conclude in Section 7.

2 A simple epistemic logic

The logic EL- O^S is a fragment of Dynamic Epistemic Logic of Propositional Assignments and Observation DEL-PAO [13], which is a dialect of Dynamic Logic of Propositional Assignments DL-PA [14, 4, 3]. We start by defining its language and semantics.

The basic ingredient of DEL-PAO are atoms of the form $S_i p$, to be read as "agent i sees p " or "agent i knows whether p ." We understand this as follows: when i knows whether p then either p is true and i knows that, or p is false and i knows that. We also allow for higher-order visibility with atoms of the form $S_j S_i p$ (j sees whether i sees p), $S_k S_j S_i p$ (k sees whether j sees whether i sees p), and so on. Along with visibility, DEL-PAO includes joint visibility and deals with a relation of 'introspective consequence' between atoms. Here we simplify things and only consider individual visibility. We call the resulting logic EL- O^S .

2.1 Language

Let $Prop = \{p_1, p_2, \dots\}$ be a countable set of *propositional variables* and $Agt = \{1, \dots, n\}$ a finite set of *agents*. The set of *visibility operators* is $OBS = \{S_i : i \in Agt\}$. The set of all sequences of visibility operators is noted OBS^* ; elements of OBS^* are noted σ, σ' , etc. An *atom* is any sequence of visibility operators S_i followed by a propositional variable. Formally,

$$ATM = \{\sigma p : \sigma \in OBS^*, p \in Prop\}$$

Elements of ATM are noted $\alpha, \alpha', \beta, \beta'$, etc. The depth of an atom is the number of visibility operators composing it.

Then the language of EL- O^S is defined by the following grammar:

$$\varphi ::= \alpha \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi$$

¹ University of Toulouse, IRIT, France.

<http://www.irit.fr/?lang=en>

where α ranges over ATM and i over Agt .

The formula $K_i\varphi$ reads “agent i knows that φ , based on what she observes.” The other boolean operators \perp , \top , \vee , \rightarrow and \leftrightarrow are defined as usual.

A *boolean formula* is a formula without the epistemic operator K_i . The set of all boolean formulas is noted Fml_{bool} . We note $ATM(\varphi)$ the set of atoms appearing in the boolean formula φ . For example, $ATM(q \wedge \mathbf{S}_i p) = \{q, \mathbf{S}_i p\}$. (Note that $p \notin ATM(q \wedge \mathbf{S}_i p)$.)

The visibility operator \mathbf{S}_i can only be applied to atoms: it means that agent i sees the value of this atom (she sees whether it is true or false). On the other hand, the epistemic operator K_i can be applied to any formula and means that i knows that this formula is true. So $K_i\neg p$ is a well-formed formula but $\mathbf{S}_i\neg p$ is not. In Section 2.2 we will show that our logic nevertheless allows us to reason about knowledge of complex formulas.

2.2 States and indistinguishability relations

Our worlds, alias states, are simply subsets of atoms, the ones that are currently true. Therefore the set of all worlds is 2^{ATM} . We denote worlds by s, s', t , etc.

We interpret knowledge operators thanks to the visibility information contained in states. Intuitively, for an atom α , we have:

$$\begin{aligned}\alpha \wedge \mathbf{S}_i \alpha &\leftrightarrow K_i \alpha \\ \neg \alpha \wedge \mathbf{S}_i \alpha &\leftrightarrow K_i \neg \alpha\end{aligned}$$

For complex formulas $K_i\varphi$, we rely on accessible worlds just as in standard epistemic logic [11]: i knows that φ if φ is true in all worlds *indistinguishable* from the current one. Unlike standard epistemic logic however, the indistinguishability relation is not primitive but is constructed from visibility information: two worlds s and s' are indistinguishable for agent i , noted $s \sim_i s'$, if all atoms that i sees in s keep the same value in s' . Formally:

$$s \sim_i s' \text{ iff for every } \alpha \in ATM, \text{ if } \mathbf{S}_i \alpha \in s \text{ then } s(\alpha) = s'(\alpha)$$

where $s(\alpha) = s'(\alpha)$ if and only if either $\alpha \in s$ and $\alpha \in s'$ or $\alpha \notin s$ and $\alpha \notin s'$. It is this construction that allows us to model knowledge in a more compact and natural way: all the information about indistinguishability is contained in the actual state and there is no need to explicitly give possible worlds and how they relate.

In epistemic logic, indistinguishability relations are usually assumed to be equivalence relations. While ours are clearly reflexive, they are neither transitive nor symmetric.² To ensure transitivity and symmetry, we impose *introspection*: agents must always know what they know and what they do not know. In terms of visibility, this means that atoms of the form $\mathbf{S}_i \mathbf{S}_i \alpha$ should always be true. Moreover, every agent should be aware of these facts. Generally: atoms of the form $\sigma \mathbf{S}_i \mathbf{S}_i \alpha$ with σ a sequence of visibility operators, possibly empty, must be true. Let us define the (infinite) set of introspectively valid atoms as:

$$ATM_{INTR} = \{\alpha : \alpha \in ATM \text{ and } \alpha = \sigma \mathbf{S}_i \mathbf{S}_i \alpha'\}.$$

We say that a state containing all these atoms is *introspective* and denote the set of all introspective states by $INTR$. Formally,

$$INTR = \{s \in 2^{ATM} : ATM_{INTR} \subseteq s\}.$$

Clearly, $s \cup ATM_{INTR} \in INTR$ for every state s .

Proposition 1 ([13]). *Relations \sim_i are equivalence relations on $INTR$.*

Proposition 2 ([13]). *Let $s \in INTR$ and $s' \in 2^{ATM}$. If $s \sim_i s'$ then $s' \in INTR$.*

² For example, $\emptyset \sim_i \{\mathbf{S}_i p, p\}$ while $\{\mathbf{S}_i p, p\} \not\sim_i \emptyset$ since $\{\mathbf{S}_i p, p\}(p) \neq \emptyset(p)$.

2.3 Semantics

Formulas are interpreted over states $s \in 2^{ATM}$. The truth conditions are as follows:

$$\begin{aligned}s \models \alpha &\text{ iff } \alpha \in s \\ s \models \neg \varphi &\text{ iff } s \not\models \varphi \\ s \models \varphi \wedge \varphi' &\text{ iff } s \models \varphi \text{ and } s \models \varphi' \\ s \models K_i \varphi &\text{ iff for every } s' \in 2^{ATM} \text{ such that } s \sim_i s', s' \models \varphi\end{aligned}$$

Let us stress that truth conditions are defined on any state $s \in 2^{ATM}$, even if we have seen that the \sim_i are equivalence relations only on introspective states. Moreover, in the truth condition of $K_i\varphi$ we do not require the s' to be introspective: by Proposition 2, s' will belong to $INTR$ if s is introspective. A formula φ is valid if and only if $s \models \varphi$ for every $s \in 2^{ATM}$; it is valid in $INTR$ if and only if $s \models \varphi$ for every $s \in INTR$.

We say that the boolean formula φ is in *normal form* if and only if φ does not contain an introspectively valid atom, i.e., no $\alpha \in ATM(\varphi)$ belongs to ATM_{INTR} .

Proposition 3. *For every $EL\text{-O}^S$ formula φ , there exists a formula φ' in normal form such that $\varphi \leftrightarrow \varphi'$ is valid in $INTR$.*

This formula can be obtained by replacing every introspectively valid atom of φ by \top .

Proposition 4. *For every state $s \in 2^{ATM}$, every boolean formula φ in normal form, and every $\alpha \in ATM_{INTR}$, we have:*

$$s \setminus \{\alpha\} \models \varphi \text{ if and only if } s \cup \{\alpha\} \models \varphi.$$

By Proposition 4, the truth value of a formula in normal form is the same in s and in its introspective, but infinite counterpart $s \cup ATM_{INTR}$.

2.4 From visibility to knowledge

We now show how to reduce $EL\text{-O}^S$ formulas to boolean formulas. This will allow us to reduce multiagent planning problems that are expressible in $EL\text{-O}^S$ to classical planning problems.

Proposition 5 ([13]). *The following equivalences are valid.*

$$\begin{aligned}K_i \alpha &\leftrightarrow \mathbf{S}_i \alpha \wedge \alpha \\ K_i \neg \alpha &\leftrightarrow \mathbf{S}_i \alpha \wedge \neg \alpha \\ K_i(\varphi \wedge \varphi') &\leftrightarrow K_i \varphi \wedge K_i \varphi' \\ K_i \left(\bigvee_{\alpha \in A^+} \alpha \vee \bigvee_{\alpha \in A^-} \neg \alpha \right) &\leftrightarrow \begin{cases} \left(\bigvee_{\alpha \in A^+} K_i \alpha \right) \vee \left(\bigvee_{\alpha \in A^-} K_i \neg \alpha \right) & \text{if } A^+ \cap A^- = \emptyset \\ \top & \text{otherwise} \end{cases}\end{aligned}$$

Moreover, the rule of replacement of equivalents preserves validity.

The last equivalence may appear curious to one familiar with epistemic logic. It is actually inherent to the notion of visibility: if an agent knows that p or q is true by looking at them, she immediately knows which one is true. This is discussed in [13] and [7]; the latter proposes an extension of $DEL\text{-PAO}$ where the equivalence is invalid.

With the help of Proposition 5, starting with the innermost operator K_i (thanks to the rule of replacement of equivalents) we can reduce any $EL\text{-O}^S$ formula to a boolean formula. Let us focus on reducing formulas of the form $K_{i_1} \dots K_{i_m} \alpha$ to a conjunction of atoms. For example, we have by Proposition 5:

$$K_i K_j p \leftrightarrow K_i(\mathbf{S}_j p \wedge p) \leftrightarrow K_i \mathbf{S}_j p \wedge K_i p$$

$$\leftrightarrow \mathbf{S}_i \mathbf{S}_j p \wedge \mathbf{S}_j p \wedge \mathbf{S}_i p \wedge p.$$

We generalize this: define the set of ‘epistemic atoms’ of an epistemic formula φ of the form $K_{i_1} \dots K_{i_m} \alpha$ with $m \geq 0$ such that α is not introspectively valid as follows:

$$EATM(\alpha) = \{\alpha\}$$

$$EATM(K_i \varphi) = EATM(\varphi) \cup \{\mathbf{S}_i \alpha : \alpha \in EATM(\varphi) \text{ and } \alpha \text{ is not of the form } \mathbf{S}_i \alpha'\}$$

The last line ensures that $EATM(\varphi)$ does not contain any introspectively valid atom as we will be interested in formulas in normal form. Denote the conjunction of all these atoms by $\bigwedge_{\alpha \in EATM(\varphi)} \alpha$.

Proposition 6. *The following equivalence is valid in INTR.*

$$K_{i_1} \dots K_{i_m} \alpha \leftrightarrow \bigwedge_{\alpha \in EATM(K_{i_1} \dots K_{i_m} \alpha)} \alpha$$

Lemma 1. *Let $p \geq 0$. Let r_1, \dots, r_p be such that $1 \leq r_1 < \dots < r_p \leq m$. Then $EATM(K_{i_{r_1}} \dots K_{i_{r_p}} \alpha) \subseteq EATM(K_{i_1} \dots K_{i_m} \alpha)$.*

In words, the set of epistemic atoms of $K_{i_1} \dots K_{i_m} \alpha$ includes every epistemic atom of a formula composed of epistemic operators on a subsequence of i_1, \dots, i_m .

We extend $EATM(\cdot)$ to a conjunction of formulas as expected:

$$EATM\left(\bigwedge_{\substack{i_1, \dots, i_m \in \text{Agt} \\ \text{and } \alpha \in A}} K_{i_1} \dots K_{i_m} \alpha\right) = \bigcup_{\substack{i_1, \dots, i_m \in \text{Agt} \\ \text{and } \alpha \in A}} EATM(K_{i_1} \dots K_{i_m} \alpha)$$

where $A \subseteq ATM$ is a set of atoms.

We will use these epistemic atoms in applications.

3 Epistemic planning with conditional effects

In this section, we formally define actions and planning tasks within our framework EL- \mathcal{O}^S . We assume that we perform planning tasks in fully observable, deterministic domains.

3.1 Actions with conditional effects

An *conditional action* is a pair $\mathbf{a} = \langle \text{pre}(\mathbf{a}), \text{eff}(\mathbf{a}) \rangle$ where:

- $\text{pre}(\mathbf{a}) \in Fml_{bool}$ is a boolean formula: the *precondition* of \mathbf{a} ;
- $\text{eff}(\mathbf{a}) \subseteq Fml_{bool} \times 2^{ATM} \times 2^{ATM}$ is a set of triples ce of the form $\langle \text{cnd}(ce), \text{ceff}^+(ce), \text{ceff}^-(ce) \rangle$: the *conditional effects* of \mathbf{a} , where $\text{cnd}(ce)$ is a boolean formula (the condition) and $\text{ceff}^+(ce)$ and $\text{ceff}^-(ce)$ are sets of atoms (added and deleted atoms respectively).

We impose that there is no conflicting effects: for every $ce_1, ce_2 \in \text{eff}(\mathbf{a})$ with $\text{cnd}(ce_1)$ and $\text{cnd}(ce_2)$ consistent, $\text{ceff}^+(ce_1) \cap \text{ceff}^-(ce_2) = \emptyset$.

For example, consider the conditional action toggle_p of flipping the truth value of the propositional variable p . It is described as $\text{toggle}_p = \langle \text{pre}(\text{toggle}_p), \text{eff}(\text{toggle}_p) \rangle$ with $\text{pre}(\text{toggle}_p) = \top$ and $\text{eff}(\text{toggle}_p) = \{\langle p, \emptyset, \{p\} \rangle, \langle \neg p, \{p\}, \emptyset \rangle\}$. The conditions p and $\neg p$ are inconsistent, thus not leading to conflict.

Example 1 (The gossip problem). Let $\text{Agt} = \{1, \dots, n\}$ and $\text{Prop} = \{s_i : i \in \text{Agt}\}$. Each propositional variable s_i represents the secret of agent i . We are not interested in its value, but only in the *knowledge* of its value. (We suppose each s_i is true.)

During the action call_j^i , agents i and j tell each other every secret they know among all n secrets. We have $\text{call}_j^i = \langle \text{pre}(\text{call}_j^i), \text{eff}(\text{call}_j^i) \rangle$ with $\text{pre}(\text{call}_j^i) = \top$ and

$$\text{eff}(\text{call}_j^i) = \{(\mathbf{S}_i s_1 \vee \mathbf{S}_j s_1, \{\mathbf{S}_i s_1, \mathbf{S}_j s_1\}, \emptyset),$$

$\dots,$

$$\langle \mathbf{S}_i s_n \vee \mathbf{S}_j s_n, \{\mathbf{S}_i s_n, \mathbf{S}_j s_n\}, \emptyset \rangle\}.$$

Intuitively, we add visibility of a secret to both agents if at least one knows it. (So we add variables that are already true; in this case there will be no effect.)

There is no possible conflict since call_j^i has no negative effects.

A conditional action \mathbf{a} determines a relation between states that is a partial function:

$$\begin{aligned} s R_{\mathbf{a}} s' \text{ iff } & (1) s \models \text{pre}(\mathbf{a}), \text{ and} \\ & (2) \text{ for every } ce \in \text{eff}(\mathbf{a}) \text{ such that} \\ & (\text{ceff}^+(ce) \cup \text{ceff}^-(ce)) \cap \text{ATM}_{INTR} \neq \emptyset, \\ & s \not\models \text{cnd}(ce), \text{ and} \\ & (3) s' = \left(s \setminus \bigcup_{\substack{ce \in \text{eff}(\mathbf{a}) \\ \text{and } s \models \text{cnd}(ce)}} \text{ceff}^-(ce) \right) \cup \bigcup_{\substack{ce \in \text{eff}(\mathbf{a}) \\ \text{and } s \not\models \text{cnd}(ce)}} \text{ceff}^+(ce). \end{aligned}$$

In words, an action adds and removes atoms as expected if its precondition is satisfied and none of its conditional effects involving an introspective atom can be triggered.

We say that the action \mathbf{a} is in *normal form* if and only if (1) the formulas $\text{pre}(\mathbf{a})$ and $\text{cnd}(ce)$ for every $ce \in \text{eff}(\mathbf{a})$ are in normal form, and (2) for every $ce \in \text{eff}(\mathbf{a})$, if $\alpha \in \text{ceff}^+(ce) \cup \text{ceff}^-(ce)$ then α is not introspectively valid.

Proposition 7. *For every action \mathbf{a} , there exists an action \mathbf{a}' in normal form such that for every $s, t \in INTR$, we have:*

$$s R_{\mathbf{a}} t \text{ if and only if } s R_{\mathbf{a}'} t.$$

Actions in normal form can be obtained by the following modification, (1) for every conditional effect $ce \in \text{eff}(\mathbf{a})$ such that $(\text{ceff}^+(ce) \cup \text{ceff}^-(ce)) \cap \text{ATM}_{INTR} \neq \emptyset$, replace $\text{pre}(\mathbf{a})$ by $\text{pre}(\mathbf{a}) \wedge \neg \text{cnd}(ce)$ and remove ce from $\text{eff}(\mathbf{a})$, and (2) put the resulting $\text{pre}(\mathbf{a})$ and $\text{cnd}(ce)$, for every $ce \in \text{eff}(\mathbf{a})$, in normal form.

Proposition 8. *For every $s, t \in 2^{ATM}$, every action \mathbf{a} in normal form, and every $\alpha \in \text{ATM}_{INTR}$, we have:*

$$s \setminus \{\alpha\} R_{\mathbf{a}} t \setminus \{\alpha\} \text{ if and only if } s \cup \{\alpha\} R_{\mathbf{a}} t \cup \{\alpha\}.$$

As with Proposition 4 for formulas, Proposition 8 implies that if there exists an execution of \mathbf{a} from s that leads to t , then there exists an execution of the same action from $s \cup \text{ATM}_{INTR}$ to $t \cup \text{ATM}_{INTR}$. This is ensured by the fact that actions in normal form neither test nor add nor remove introspectively valid atoms.

3.2 Simple epistemic planning tasks

We say that a state s is *reachable* from a state s_0 via a set of conditional actions Act if there is a sequence of actions $\langle \mathbf{a}_1, \dots, \mathbf{a}_m \rangle$ from Act and a sequence of states $\langle u_0, \dots, u_m \rangle$ with $m \geq 0$ such that $s_0 = u_0$, $s = u_m$ and $u_{k-1} R_{\mathbf{a}_k} u_k$ for every k such that $1 \leq k \leq m$.

A simple epistemic *planning task* is a triple $\langle \text{Act}, s_0, \text{Goal} \rangle$ where Act is a finite set of actions, $s_0 \in ATM$ is a finite state (the initial state) and $\text{Goal} \in Fml_{bool}$ is a boolean formula. It is *solvable* if at least one state s such that $s \models \text{Goal}$ is reachable from s_0 via Act ; otherwise it is unsolvable.

We say that the planning task $\langle \text{Act}, s_0, \text{Goal} \rangle$ is in *normal form* if and only if (1) every action $\mathbf{a} \in \text{Act}$ is in normal form, and (2) the formula Goal is in normal form.

Example 2 (Example 1, ctd.). The planning task corresponding to the gossip problem is $G_1 = \langle \text{Act}^{G_1}, s_0^{G_1}, \text{Goal}^{G_1} \rangle$ with

- $Act^{G_i} = \{\text{call}_j^i : i, j \in \text{Agt} \text{ and } i \neq j\}$;
- $s_0^{G_i} = \{\mathbf{S}_i s_i : i \in \text{Agt}\} \cup \{s_i : i \in \text{Agt}\}$;
- $Goal^{G_i} = \bigwedge_{i,j \in \text{Agt}} \mathbf{S}_i s_j$.

In the initial state, every agent knows her own secret and none of the other secrets. Secrets are also true initially, so that, since no action can change the truth value of s_i , $Goal^{G_i}$ is equivalent to $\bigwedge_{i,j \in \text{Agt}} K_i s_j$.

This planning task is in normal form since every action call_j^i is trivially in normal form.

4 Dynamic extension and complexity results

Consider the planning task $\langle Act, s_0, Goal \rangle$. In this section, we introduce an extension of EL-O^S with dynamic operators, which we call DEL-PAO^S (Dynamic Epistemic Logic of Propositional Assignments and Observation without common knowledge). We show how actions from Act can be encoded into DEL-PAO^S programs. Then we prove that the solvability of $\langle Act, s_0, Goal \rangle$ is in PSPACE by showing that it can be polynomially reduced to a DEL-PAO^S model checking problem.

4.1 A simple dynamic epistemic logic

The language of DEL-PAO^S extends the language of EL-O^S with the dynamic operator $\langle \pi \rangle$, with π a program: the formula $\langle \pi \rangle \varphi$ reads “there exists an execution of π after which φ is true.”

The syntax of programs is defined by the following grammar:

$$\pi ::= +\alpha \mid -\alpha \mid \pi; \pi \mid \pi \sqcup \pi \mid \pi^* \mid \varphi?$$

where α ranges over the set of atomic formulas ATM and φ over the set of formulas.

Atomic programs $+\alpha$ and $-\alpha$ are assignments: they respectively set the value of the atom α to true and to false. Complex programs are composed of sequences of instructions ($\pi; \pi$), non-deterministic choice between instructions ($\pi \sqcup \pi$), repetitions (π^*) and tests ($\varphi?$).

The dual operator $[\pi]\varphi = \neg\langle \pi \rangle\neg\varphi$ (“after every execution of π , φ is true”) is defined as usual. Moreover, **if** φ **then** π abbreviates $(\varphi?; \pi) \sqcup \neg\varphi?$ and **if** φ **then** π **else** π' abbreviates $(\varphi?; \pi) \sqcup (\neg\varphi?; \pi')$.

Semantically, a program is interpreted as a binary relation R_π on states, such that:

$$\begin{aligned} sR_{+\alpha}s' & \text{ iff } s' = s \cup \{\alpha\} \\ sR_{-\alpha}s' & \text{ iff } s' = s \setminus \{\alpha\} \text{ and } \alpha \notin ATM_{INTR} \\ sR_{\pi_1; \pi_2}s' & \text{ iff there exists } u \in 2^{ATM} \text{ such that } sR_{\pi_1}u \text{ and } uR_{\pi_2}s' \\ sR_{\pi_1 \sqcup \pi_2}s' & \text{ iff } sR_{\pi_1}s' \text{ or } sR_{\pi_2}s' \\ sR_{\pi^*}s' & \text{ iff there exist } u_0, \dots, u_m \in 2^{ATM} \text{ with } m \geq 0 \\ & \text{ such that } s = u_0, s' = u_m \\ & \text{ and } u_{k-1}R_\pi u_k \text{ for every } 1 \leq k \leq m \\ sR_{\varphi?}s' & \text{ iff } s = s' \text{ and } s \models \varphi \end{aligned}$$

The truth condition of the new operator is then:

$$s \models \langle \pi \rangle \varphi \text{ iff there exists } s' \in 2^{ATM} \text{ such that } sR_\pi s' \text{ and } s' \models \varphi$$

In words, $\langle \pi \rangle \varphi$ is true if there is a state reachable by executing π where φ is true. An assignment $+\alpha$ or $-\alpha$ updates the state by adding or (unless introspectively valid) removing α ; a sequential composition $\pi_1; \pi_2$ executes first π_1 and then π_2 ; a nondeterministic composition $\pi_1 \sqcup \pi_2$ takes the union of relations for π_1 and for π_2 ; an iteration π^* reaches any state attainable if we repeat π an arbitrary

number of times; a test $\varphi?$ stays in the same state if φ is true there (otherwise the program fails and produces no result world).

Observe that unlike the epistemic operators K_i , the evaluation of dynamic operators may terminate in a non introspective state. However, trying to remove an introspectively valid atom (e.g. by executing $-\mathbf{S}_i s_i$) will lead to a failure of the program because of the definition of $R_{-\alpha}$: a program starting in $INTR$ will never exit $INTR$.

4.2 Storing variables

The conditional effects of the actions that we have defined in Section 3 are produced in parallel. We have to simulate this in DEL-PAO^S by sequential composition. We therefore have to take care that the truth value of no condition is modified by an effect. To achieve this, we store the values of our conditions before executing our action, and evaluate such values. This problem does not arise in PDDL where all conditions are checked before any effects are produced.

We use new atomic variables noted c , called *storage variables*, which we suppose do not appear in the planning task under concern. Then the program storing the value of a formula is defined as:

$$\text{str}(\varphi, c) = \text{if } \varphi \text{ then } +c \text{ else } -c.$$

Proposition 9. *If c does not occur in φ then the equivalence $\varphi \leftrightarrow [\text{str}(\varphi, c)]c$ is valid.*

We will see that after the execution of our program, we will make all the storage variables false so that we do not have to worry about them. The program resetting the value of a given set of storage variables is simply defined as:

$$\text{rst}(\{c_1, \dots, c_m\}) = -c_1; \dots; -c_m.$$

4.3 Encoding of actions

Intuitively, an action is a DEL-PAO^S program, only executed if the precondition is fulfilled, applying each conditional effect whose condition is satisfied. For example, the action toggle_p (flipping the value of the variable p) corresponds to the program $\text{str}(p, c_1); \text{str}(\neg p, c_2); \text{if } c_1 \text{ then } -p; \text{if } c_2 \text{ then } +p$. This highlights the importance of storing values of conditions: the program **if** p **then** $-p$; **if** $\neg p$ **then** $+p$ would actually always make p true.

We first show how to perform one conditional effect ce whose condition's value was stored in c :

$$\text{exeCE}(ce, c) = \text{if } c \text{ then } +\alpha_1; \dots; +\alpha_m; -\beta_1; \dots; -\beta_\ell$$

where $\text{ceff}^+(ce) = \{\alpha_1, \dots, \alpha_m\}$ and $\text{ceff}^-(ce) = \{\beta_1, \dots, \beta_\ell\}$. Note that the ordering of atoms is not important since $\text{ceff}^+(ce) \cap \text{ceff}^-(ce) = \emptyset$. Then we can associate to action a the DEL-PAO^S program $\text{exeAct}(a)$:

$$\begin{aligned} \text{exeAct}(a) &= \text{pre}(a)?; \\ & \text{str}(\text{cnd}(ce_1), c_1); \dots; \text{str}(\text{cnd}(ce_m), c_m); \\ & \text{exeCE}(ce_1, c_1); \dots; \text{exeCE}(ce_m, c_m); \\ & \text{rst}(\{c_1, \dots, c_m\}), \end{aligned}$$

with $\text{eff}(a) = \{ce_1, \dots, ce_m\}$. The ordering of effects is not important since we test values of storage variables.

Proposition 10. *For every $s, t \in 2^{ATM}$ such that s does not contain any storage variable, and every action a in normal form, the program $\text{exeAct}(a)$ behaves like a :*

$$s R_a t \text{ if and only if } s R_{\text{exeAct}(a)} t.$$

Intuitively, our program $\text{exeAct}(a)$ is divided in four parts which are executed in sequence:

1. $pre(a)?$: we test the precondition of the action. Semantically, if $s \models pre(a)$ then we stay in the same state s and continue to execute the program, and if $s \not\models pre(a)$ then the program fails (no world is related to s by $exeAct(a)$). This corresponds to the requirement that $s \models pre(a)$ in R_a .
2. $str(cnd(ce_1), c_1); \dots; str(cnd(ce_m), c_m)$: we store every condition. Recall that $\varphi \leftrightarrow [str(\varphi, c)]c$ is valid, ensuring that testing each c_i after $str(cnd(ce_i), c_i)$ is equivalent to testing $cnd(ce_i)$ before effects are executed. Observe that after the execution of this part, each atom has kept the same value it had in s ; only storage variables have been modified.
3. $exeCE(ce_1, c_1); \dots; exeCE(ce_m, c_m)$: we apply effects based on the truth values of storage variables. For each program $exeCE(ce_i, c_i)$, if $s \models c_i$, then every positive effect from $ceff^+(ce_i)$ is added to s , while every negative effect from $ceff^-(ce_i)$ is removed from s , as specified in R_a . The action being in normal form ensures that $exeCE(ce_i, c_i)$ will not remove an introspectively valid atom, preventing any failure.
4. $rst(\{c_1, \dots, c_m\})$: all storage variables are put to false, like they were in s , so that the execution of $exeAct(a)$ is exactly equivalent to the execution of a (where storage variables do not appear).

Example 3 (Example 1, ctd.). The action $call_j^i$, for any $i, j \in Agt$, is associated to the program:

$$\begin{aligned} exeAct(call_j^i) = & \top?; \\ & str(\mathbf{S}_i s_1 \vee \mathbf{S}_j s_1, c_1); \dots; str(\mathbf{S}_i s_n \vee \mathbf{S}_j s_n, c_n); \\ & \mathbf{if} \ c_1 \ \mathbf{then} \ +\mathbf{S}_i s_1; +\mathbf{S}_j s_1; \\ & \dots; \\ & \mathbf{if} \ c_n \ \mathbf{then} \ +\mathbf{S}_i s_n; +\mathbf{S}_j s_n; \\ & rst(\{c_1, \dots, c_n\}) \end{aligned}$$

Note that in this case, $pre(call_j^i)?$ can clearly be dropped.

4.4 Solvability of a planning task

Now that we have defined the encoding of actions, we can capture the solvability of a planning task in DEL-PAO^S.

Proposition 11. *A planning task $\langle Act, s_0, Goal \rangle$ in normal form such that s_0 does not contain any storage variable is solvable if and only if:*

$$s_0 \models \langle (\bigsqcup_{a \in Act} exeAct(a))^* \rangle Goal.$$

Intuitively, our formula reads “there exists an execution of $(\bigsqcup_{a \in Act} exeAct(a))^*$ after which $Goal$ is true.” The program $(\bigsqcup_{a \in Act} exeAct(a))^*$ non-deterministically chooses an action a from Act and executes the corresponding program $exeAct(a)$, then repeats this a finite number of times. This produces a sequence of actions, i.e., a plan.

We do not impose that s_0 is introspective as it would make it infinite; this is not necessary by Proposition 8 since the planning task is in normal form: if there is an execution of $(\bigsqcup_{a \in Act} exeAct(a))^*$ starting from the introspective state $s_0 \cup ATM_{INTR}$ and leading to a state satisfying $Goal$, then there is one starting from the non-introspective state s_0 and leading to a state satisfying $Goal$.

Proposition 12. *Deciding the solvability of a planning task with DEL-PAO^S is PSPACE-complete.*

The lower bound comes from classical planning [6]; the upper bound is given by Proposition 11, where the problem is reduced to

a model checking problem of DEL-PAO^S, a fragment of DEL-PAO whose model checking problem is in PSPACE.³

This result compares favorably to DEL-based epistemic planning, which is undecidable even for simple fragments [1, 8]. The difference is due to the simplicity of our underlying epistemic logic (cf. Proposition 5) as well as to the limited expressivity of our actions: we can basically model private announcements, while DEL has more general event models.

5 Encoding into PDDL

In this section we present a method for encoding planning problems defined in DEL-PAO^S into PDDL. As already observed, in PDDL we do not need to store conditions as we were obliged to do in DEL-PAO^S. Consider a planning task $\langle Act, s_0, Goal \rangle$. We show how to encode boolean formulas and actions in PDDL.

5.1 Translation of formulas

Some PDDL requirement flags should be set depending on the form of conditions $cnd(ce)$ of conditional effects ce of actions and of the formula $Goal$:

- the default flag `:strips` for conjunctions;
- the flag `:negative-preconditions` for negations;
- the flag `:disjunctive-preconditions` for negations of conjunctions, and disjunctions, if used to simplify writing.

Given a boolean formula $\varphi \in Fml_{bool}$, we define a recursive function $tr_{PDDL}(\varphi)$ which returns the encoding of φ into PDDL:

$$tr_{PDDL}(\mathbf{S}_{i_1} \dots \mathbf{S}_{i_m} p) ::= \begin{cases} (p) & \text{if } m = 0 \\ (\mathbf{S}\text{-}m \ \text{il} \ \dots \ \text{im} \ p) & \text{otherwise} \end{cases}$$

$$tr_{PDDL}(\neg\varphi) ::= (\text{not } tr_{PDDL}(\varphi))$$

$$tr_{PDDL}(\varphi_1 \wedge \varphi_2) ::= (\text{and } tr_{PDDL}(\varphi_1) \ tr_{PDDL}(\varphi_2))$$

with $p \in Prop$, $m \geq 0$, and $i_1, \dots, i_m \in Agt$.

In words, a visibility atom $\alpha = \mathbf{S}_{i_1} \dots \mathbf{S}_{i_m} p$ is encoded by a special fluent with $m+1$ parameters. If $m = 0$, then the propositional variable p is encoded as a fluent without parameters. We note $tr_{PDDL}(\alpha)$ the translation of an atom α in the general case (p or $\mathbf{S}_{i_1} \dots \mathbf{S}_{i_m} p$). Other boolean operators are encoded as expected.

The initial state s_0 is trivially encoded as a set of fluents thanks to $tr_{PDDL}(\alpha)$. $Goal$ and the preconditions of every action can be encoded using $tr_{PDDL}(\varphi)$ since they are all boolean formulas.

5.2 Encoding of actions

The requirement flag `:conditional-effects` must be set.

Consider an action a . For every $ce \in eff(a)$ with $ceff^+(ce) = \{\alpha_1, \dots, \alpha_m\}$ and $ceff^-(ce) = \{\beta_1, \dots, \beta_\ell\}$, we add the conditional effect:

$$\begin{aligned} & (\text{when } tr_{PDDL}(cnd(ce)) \\ & \quad (\text{and } tr_{PDDL}(\alpha_1) \ \dots \ tr_{PDDL}(\alpha_m) \\ & \quad \quad (\text{not } tr_{PDDL}(\beta_1)) \ \dots \ (\text{not } tr_{PDDL}(\beta_\ell)))) \end{aligned}$$

Note that, again, the ordering is not important.

³ The version of DEL-PAO presented in [13] does not include the iteration (represented by the star “*”) in the language of programs. However, a more general result, including the star and with a PSPACE model checking, can be found in [9].

Example 4 (Example 1, ctd.). The action call₂¹ is coded in PDDL as follows:

```
(:action call-1-2
:effect (and
  (when (or (S-1 1 s1) (S-1 2 s1))
    (and (S-1 1 s1) (S-1 2 s1)))
  ...
  (when (or (S-1 1 sn) (S-1 2 sn))
    (and (S-1 1 sn) (S-1 2 sn))))))
```

This is the direct encoding of a call into PDDL. Remark that we could generalize it to any i and j by adding the line ‘:parameters (?i ?j)’ and replacing every ‘(S-1 1 .)’ by ‘(S-1 ?i .)’ and every ‘(S-1 2 .)’ by ‘(S-1 ?j .)’. We will use the latter in experiments because of its succinctness.

Almost all planners from last International Planning Competition (IPC 2014)⁴ handle conditional effects and negative preconditions, and most of them handle disjunctive preconditions. For experiments, we chose to use the planner FDSS-2014 [22] that was satisfying all these preconditions.

6 Applications

In this section, we first study the ‘exam problem’ (a simple illustrative example concerning privacy of information), then generalizations of the gossip problem.⁵ We sometimes write simply ‘a’ for ‘exeAct(a)’ when used within dynamic operators $\langle \cdot \rangle$ and $[\cdot]$.

6.1 The exam problem

Suppose we have two agents: a teacher and a student. The teacher has prepared the exam and keeps it in her office; the goal of the student is to know the exam topic, but without the teacher seeing her doing this. To achieve this goal, the student must enter the teacher’s office, read the exam while the teacher is not inside, and exit the office.

Let the corresponding planning task be $Exam = \langle Act^{Exam}, s_0^{Exam}, Goal^{Exam} \rangle$. Let $Agt = \{t, s\}$ and $Prop = \{exam, open, in_t, in_s\}$. Agent t is the teacher and agent s is the student. The variable $exam$ represents the topic of the exam. Like secrets in the gossip problem, its value is not relevant and we only reason about the knowledge of it (we will assume it is true). The variable $open$ reads “the teacher’s office is open”, and in_i , for i an agent, “agent i is in the teacher’s office”.

Initially, we assume the office is empty and the door is closed:

$$s_0^{Exam} = \{exam\}.$$

As we said, the goal for the student is to know the exam’s topic without being caught by the teacher. The goal is $S_s exam \wedge \neg K_t S_s exam \wedge \neg in_s$. In terms of visibility atoms, this becomes:

$$Goal^{Exam} = S_s exam \wedge \neg S_t S_s exam \wedge \neg in_s.$$

We study two variants of this problem with different actions.

Vigilant teacher. In this first version, we suppose the teacher always closes her office door when leaving. The set of actions is:

$$Act^{Exam} = \{openAndGoln_t, goOutAndClose_t, goln_s, goOut_s, readExam_s\},$$

where

$$openAndGoln_t = \langle \neg in_t, \{\langle \top, \{open, in_t, S_t S_s exam\}, \emptyset \rangle\} \rangle$$

$$goOutAndClose_t = \langle in_t \wedge \neg S_s exam, \{\langle \top, \emptyset, \{S_t S_s exam, in_t, open\} \rangle\} \rangle$$

$$goln_s = \langle open \wedge \neg in_s, \{\langle \top, \{in_s\}, \emptyset \rangle\} \rangle$$

$$goOut_s = \langle open \wedge in_s, \{\langle \top, \emptyset, \{in_s\} \rangle\} \rangle$$

$$readExam_s = \langle in_s, \{\langle \top, \{S_s exam\}, \emptyset \rangle\} \rangle$$

Action $openAndGoln_t$ makes the teacher open and enter the room, and thus watch the exam. Action $goOutAndClose_t$ makes her leave and close the room; she cannot watch the exam anymore. We add the precondition $\neg S_s exam$ to ensure that the teacher cannot leave if she has witnessed the student see the exam, so that she cannot forget this fact. For the student, $goln_s$ and $goOut_s$ makes her enter and leave the office, with the precondition that it is open; $readExam_s$ makes her see the exam topic, acquiring the knowledge on its value.

In this case, no plan exists reaching the goal. Indeed, the student can only enter the room if the door is open, which can only happen when the teacher is inside the room. Therefore the student cannot read the exam’s topic without the teacher knowing it: $S_s exam \rightarrow K_t S_s exam$. This was confirmed by experiments: FDSS-2014 cannot find a plan.

Inattentive teacher. Now we assume that the teacher can leave the room without closing the door. This is done by dividing actions $openAndGoln_t$ and $goOutAndClose_t$ each in two parts:

- we replace $openAndGoln_t$ by:

$$open_t = \langle \neg open, \{\langle \top, \{open\}, \emptyset \rangle\} \rangle$$

$$goln_t = \langle open \wedge \neg in_t, \{\langle \top, \{in_t, S_t S_s exam\}, \emptyset \rangle\} \rangle,$$

- we replace $goOutAndClose_t$ by:

$$goOut_t = \langle open \wedge in_t \wedge \neg S_s exam, \{\langle \top, \emptyset, \{S_t S_s exam, in_t\} \rangle\} \rangle$$

$$close_t = \langle open, \{\langle \top, \emptyset, \{open\} \rangle\} \rangle.$$

Thus the set of actions becomes:

$$Act^{Exam} = \{open_t, close_t, goln_t, goOut_t, goln_s, goOut_s, readExam_s\}.$$

In this setting, the problem becomes solvable: for example, the plan $open_t; goln_t; goln_s; goOut_t; readExam_s; goOut_s$ is a solution plan. More mundanely, the planner finds the shortest plan: $open_t; goln_s; readExam_s; goOut_s$.

In these two examples, we are more interested in the existence of a plan than in the plan itself: the first variant is safe for the teacher, while the second is not.

6.2 The generalized gossip problem

In this section, we present a formalisation of a generalisation of the gossip problem in our framework. A study of this problem and its variants can be found in [10].

The generalized gossip problem. We model the generalized gossip problem, introduced in [15], as a planning task $G_D = \langle Act^{G_D}, s_0^{G_D}, Goal^{G_D} \rangle$. In this generalization, the goal is not only for every agent to know every secret, but also every agent must know this fact, and every agent must know that, and so on until a given depth $D \geq 1$. Let $Agt = \{1, \dots, n\}$ and $Prop = \{s_i : i \in Agt\}$. In terms of knowledge, the goal of the generalized gossip problem is:

$$\varphi_{G_D} = \underbrace{\bigwedge_{i_1 \in Agt} K_{i_1} \dots \bigwedge_{i_D \in Agt} K_{i_D} \bigwedge_{\ell \in Agt} s_\ell}_{D \text{ times}}$$

⁴ <http://helios.hud.ac.uk/scommv/IPC-14/planners.html>

⁵ All resources and PDDL files we used for experiments are available at <http://www.irit.fr/~7EAndreas.Herzig/P/Ecai16.html>.

We have seen in Section 2.4 how to express this with a boolean formula, thanks to our epistemic atoms:

$$Goal^{GD} = \bigwedge EATM(\varphi_{GD}).$$

Recall that introspectively valid atoms are not included in $EATM(\varphi)$, thus the goal is in normal form.

The initial state and the set of actions stay the same:

$$s_0^{GD} = \{\mathbf{S}_i s_i : i \in Agt\} \cup \{s_i : i \in Agt\},$$

$$Act^{GD} = \{\text{call}_j^i : i, j \in Agt, i \neq j\}.$$

The preconditions of calls also remain unchanged: $pre(\text{call}_j^i) = \top$. However, their effects are different. Agents will not only transmit secrets but also *knowledge of secrets*. They will also learn the higher-order knowledge we need in the gossip problem when exchanging secrets. For example, if i knows the secret of ℓ and i calls j , j will learn the secret of ℓ , but also that i knows it; i will learn that j knows that she knows it; and so on until depth D . Moreover, if i knows that i_1 knows the secret of ℓ , then j learns that i_1 knows the secret of ℓ , but also that i knows that, an so on until depth D . As an example, suppose $D = 4$ and we have $K_i K_{i_1} s_\ell$. Then after the call between i and j , we will have, e.g., $K_j K_{i_1} s_\ell$, $K_i K_j K_{i_1} s_\ell$, $K_j K_i K_{i_1} s_\ell$, $K_j K_i K_j K_{i_1} s_\ell$, $K_i K_j K_i K_{i_1} s_\ell$, and so on, that is, any combination of K_i and K_j followed by $K_{i_1} s_\ell$, for a maximum depth of D .

For a given integer m and two agents i and j , we note $\{\mathbf{S}_i, \mathbf{S}_j\}^{\leq m}$ the set all non-empty non-introspective sequences of visibility operators \mathbf{S}_i and \mathbf{S}_j of length at most m . For instance, $\{\mathbf{S}_i, \mathbf{S}_j\}^{\leq 2} = \{\mathbf{S}_i, \mathbf{S}_j, \mathbf{S}_i \mathbf{S}_j, \mathbf{S}_j \mathbf{S}_i\}$.

Thus we have that, during a call between i and j , if i or j knows that $K_{i_1} \dots K_{i_m} s_\ell$, i.e., if $K_i K_{i_1} \dots K_{i_m} s_\ell \vee K_j K_{i_1} \dots K_{i_m} s_\ell$ is true, then $\sigma \mathbf{S}_{i_1} \dots \mathbf{S}_{i_m} s_\ell$ for every $\sigma \in \{\mathbf{S}_i, \mathbf{S}_j\}^{\leq D-m}$ becomes true. Formally:

$$eff(\text{call}_j^i) = \left\{ \left(\bigwedge EATM(K_i K_{i_1} \dots K_{i_m} s_\ell) \vee \bigwedge EATM(K_j K_{i_1} \dots K_{i_m} s_\ell), \right. \right. \\ \left. \left. \left\{ \sigma \mathbf{S}_{i_1} \dots \mathbf{S}_{i_m} s_\ell : \sigma \in \{\mathbf{S}_i, \mathbf{S}_j\}^{\leq D-m} \right\}, \emptyset \right) : \right. \\ \left. 0 \leq m < D \text{ and } i_1, \dots, i_m, \ell \in Agt \text{ such that} \right. \\ \left. \text{for every } 1 \leq k < m, i_k \neq i_{k+1}, \text{ and } i \neq i_1 \text{ and } j \neq i_1 \right\}$$

Consecutive agents in $\mathbf{S}_{i_1} \dots \mathbf{S}_{i_m} s_\ell$ are required to be different so that we do not involve any introspectively valid atom and we obtain an action in normal form. If we take $D = 1$, we retrieve our definition of call_j^i from Example 1 (with tests of secrets that could be omitted).

We require knowledge instead of visibility, i.e., $\bigwedge EATM(K_i K_{i_1} \dots K_{i_m} s_\ell)$ instead of just $\mathbf{S}_i \mathbf{S}_{i_1} \dots \mathbf{S}_{i_m} s_\ell$, so that agents only exchange what they know. For example, we do not want 1 to see whether 2 knows the secret of 3 without 2 knowing the secret of 3: it would imply that 1 watches 2, and that if 2 learns the secret of 3 during a call, 1 will know this even if she did not participate in this call.

Proposition 13. *The equivalence $[\text{call}_j^i] \neg \varphi \leftrightarrow \neg [\text{call}_j^i] \varphi$ is valid.*

This is due to calls being deterministic: executing a call always leads to exactly one state.

Lemma 2. *The following formulas are valid.*

$$\mathbf{S}_{i_1} \dots \mathbf{S}_{i_m} s_\ell \leftrightarrow [\text{call}_j^i] \mathbf{S}_{i_1} \dots \mathbf{S}_{i_m} s_\ell \quad \text{if } i \neq i_1 \text{ and } j \neq i_1 \quad (1)$$

$$\mathbf{S}_{i_1} \dots \mathbf{S}_{i_m} s_\ell \rightarrow [\text{call}_j^i] \mathbf{S}_{i_1} \dots \mathbf{S}_{i_m} s_\ell \quad \text{for any } i, j \quad (2)$$

(1) means that when i_1 is not involved in a call, her knowledge does not evolve. Indeed, along with Proposition 13, it implies:

$$\mathbf{S}_{i_1} \dots \mathbf{S}_{i_m} s_\ell \rightarrow [\text{call}_j^i] \mathbf{S}_{i_1} \dots \mathbf{S}_{i_m} s_\ell$$

$$\neg \mathbf{S}_{i_1} \dots \mathbf{S}_{i_m} s_\ell \rightarrow [\text{call}_j^i] \neg \mathbf{S}_{i_1} \dots \mathbf{S}_{i_m} s_\ell$$

if $i \neq i_1$ and $j \neq i_1$. (2) means that knowledge of agents cannot decrease with a call. Both lines are deduced from the definition of calls.

While the original gossip problem with $n \geq 4$ agents can be solved in $2n - 4$ calls [2, 23, 12], the generalized gossip problem can be solved in at most $(D+1)(n-2)$ calls [15]. For instance, suppose $D = 2$ and $n = 5$, then the sequence $\text{call}_3^1; \text{call}_4^1; \text{call}_5^2; \text{call}_3^1; \text{call}_4^2; \text{call}_5^3; \text{call}_4^1; \text{call}_5^2; \text{call}_3^1$ is a solution with $3 \times 3 = 9$ calls. Our experiments have confirmed that the protocol given in [15] is optimal for $D = 2$ and $n \leq 5$.

Negative goals. We now introduce an extension of the generalized gossip problem where goals can be ‘negative’. We write it $G\text{-neg}_D = \langle Act^{G\text{-neg}_D}, s_0^{G\text{-neg}_D}, Goal^{G\text{-neg}_D} \rangle$. In this variant, we change the goal and impose that some agents do not know some secrets, or some knowledge of secrets, at the end of the sequence of calls. For example, we want 1 not to know the secret of 2, or 1 not to know that 2 knows the secret of 3. The action set, the calls, and the initial state remain the same: $Act^{G\text{-neg}_D} = Act^{GD}$ and $s_0^{G\text{-neg}_D} = s_0^{GD}$.

We note $Goal_A^{G\text{-neg}_D}$ the goal of the generalized gossip problem where only atoms from A , such that $A \cap ATM_{INTR} = \emptyset$, are false. Formally:

$$Goal_A^{G\text{-neg}_D} = \left(\bigwedge_{\alpha \in EATM(\varphi_{GD}) \setminus A} \alpha \right) \wedge \left(\bigwedge_{\alpha \in A} \neg \alpha \right).$$

We present several properties of the gossip problem that will be useful in deciding solvability of $G\text{-neg}_D$.

Lemma 3. *Let $m \geq 2$ be an integer. Let $D \geq m$. Take $m+1$ agents $i_1, i_2, i_3, \dots, i_m, \ell \in Agt$ such that i_1, i_2 and i_3 are distinct. We have:*

$$(\neg \mathbf{S}_{i_1} \mathbf{S}_{i_2} \mathbf{S}_{i_3} \dots \mathbf{S}_{i_m} s_\ell \wedge [\text{call}_j^{i_1}] \mathbf{S}_{i_1} \mathbf{S}_{i_2} \mathbf{S}_{i_3} \dots \mathbf{S}_{i_m} s_\ell) \rightarrow \begin{cases} \bigwedge EATM(K_{i_1} K_{i_3} \dots K_{i_m} s_\ell) \vee \bigwedge EATM(K_{i_2} K_{i_3} \dots K_{i_m} s_\ell) & \text{if } j = i_2 \\ \bigwedge EATM(K_j K_{i_2} K_{i_3} \dots K_{i_m} s_\ell) & \text{otherwise} \end{cases}$$

Proof. In words, we are looking for conditions that make the atom $\mathbf{S}_{i_1} \mathbf{S}_{i_2} \mathbf{S}_{i_3} \dots \mathbf{S}_{i_m} s_\ell$ true after a call. We are only interested in cases where $\mathbf{S}_{i_1} \mathbf{S}_{i_2} \mathbf{S}_{i_3} \dots \mathbf{S}_{i_m} s_\ell$ is not introspective; when it is, it will never be added by an action and thus the implication is trivially true. We examine the two cases.

First case: $j = i_2$. Remember that by the definition of calls, $\text{call}_{i_2}^{i_1}$ only produces atoms beginning with a sequence σ of \mathbf{S}_{i_1} and \mathbf{S}_{i_2} . Since i_3 is distinct from i_1 and i_2 and we avoid introspective atoms, $\text{call}_{i_2}^{i_1}$ can only add our atom by adding $\sigma \mathbf{S}_{i_3} \dots \mathbf{S}_{i_m} s_\ell$ with $\sigma = \mathbf{S}_{i_1} \mathbf{S}_{i_2}$. Then either $\bigwedge EATM(K_{i_1} K_{i_3} \dots K_{i_m} s_\ell)$ or $\bigwedge EATM(K_{i_2} K_{i_3} \dots K_{i_m} s_\ell)$ must be true (before the call), which corresponds to the right side of the implication.

Second case: $j \neq i_2$. Following the same reasoning, $\text{call}_j^{i_1}$ can only add our atom by setting to true $\sigma \mathbf{S}_{i_2} \mathbf{S}_{i_3} \dots \mathbf{S}_{i_m} s_\ell$ with $\sigma = \mathbf{S}_{i_1}$; then either $\bigwedge EATM(K_{i_1} K_{i_2} K_{i_3} \dots K_{i_m} s_\ell)$ must be true or $\bigwedge EATM(K_j K_{i_2} K_{i_3} \dots K_{i_m} s_\ell)$ must be true. We cannot have the former since $\mathbf{S}_{i_1} \mathbf{S}_{i_2} \mathbf{S}_{i_3} \dots \mathbf{S}_{i_m} s_\ell$ is false. The latter corresponds to the right side of the implication. \square

Proposition 14. *Let $m \geq 2$ be an integer. Let $D \geq m$. Take $m+1$ agents $i_1, i_2, i_3, \dots, i_m, \ell \in Agt$ such that i_1, i_2 and i_3 are distinct. Then after a sequence of calls $\mathcal{C} = \text{call}_{j_{r_1}}^{i_{r_1}}; \dots; \text{call}_{j_{r_p}}^{i_{r_p}}$, if $\mathbf{S}_{i_1} \mathbf{S}_{i_2} \mathbf{S}_{i_3} \dots \mathbf{S}_{i_m} s_\ell$ is true then $\mathbf{S}_{i_1} \mathbf{S}_{i_3} \dots \mathbf{S}_{i_m} s_\ell$ is true. Formally:*

$$s_0^{G\text{-neg}_D} \models [\mathcal{C}] (\mathbf{S}_{i_1} \mathbf{S}_{i_2} \mathbf{S}_{i_3} \dots \mathbf{S}_{i_m} s_\ell \rightarrow \mathbf{S}_{i_1} \mathbf{S}_{i_3} \dots \mathbf{S}_{i_m} s_\ell).$$

Proof. We prove it by induction on the sequence of calls. We are only interested in cases where $\mathbf{S}_{i_1}\mathbf{S}_{i_2}\mathbf{S}_{i_3}\dots\mathbf{S}_{i_m}s_\ell$ is not introspective.

Base case: initial situation. We prove:

$$s_0^{G\text{-neg}_D} \models \mathbf{S}_{i_1}\mathbf{S}_{i_2}\mathbf{S}_{i_3}\dots\mathbf{S}_{i_m}s_\ell \rightarrow \mathbf{S}_{i_1}\mathbf{S}_{i_3}\dots\mathbf{S}_{i_m}s_\ell.$$

This is trivially true because only atoms of the form $\mathbf{S}_i s_i$ are true initially.

Inductive case. Suppose:

$$s_0^{G\text{-neg}_D} \models [\mathcal{C}](\mathbf{S}_{i_1}\mathbf{S}_{i_2}\mathbf{S}_{i_3}\dots\mathbf{S}_{i_m}s_\ell \rightarrow \mathbf{S}_{i_1}\mathbf{S}_{i_3}\dots\mathbf{S}_{i_m}s_\ell).$$

We prove that for an arbitrary s :

$$s \models (\mathbf{S}_{i_1}\mathbf{S}_{i_2}\mathbf{S}_{i_3}\dots\mathbf{S}_{i_m}s_\ell \rightarrow \mathbf{S}_{i_1}\mathbf{S}_{i_3}\dots\mathbf{S}_{i_m}s_\ell) \rightarrow [\text{call}_j^i](\mathbf{S}_{i_1}\mathbf{S}_{i_2}\mathbf{S}_{i_3}\dots\mathbf{S}_{i_m}s_\ell \rightarrow \mathbf{S}_{i_1}\mathbf{S}_{i_3}\dots\mathbf{S}_{i_m}s_\ell). \quad (3)$$

First suppose i_1 is not involved in the new call, that is, $i_1 \neq i$ and $i_1 \neq j$. We know by (1) of Lemma 2 that her knowledge (every atom beginning with \mathbf{S}_{i_1}) does not evolve. Thus the implication stays true.

Now suppose i_1 is involved in the new call; without loss of generality, suppose $i = i_1$. By (2) of Lemma 2, we know that a true atom stays true after a call. Then (3) is equivalent to:

$$s \models (\neg\mathbf{S}_{i_1}\mathbf{S}_{i_2}\mathbf{S}_{i_3}\dots\mathbf{S}_{i_m}s_\ell \wedge [\text{call}_j^{i_1}]\mathbf{S}_{i_1}\mathbf{S}_{i_2}\mathbf{S}_{i_3}\dots\mathbf{S}_{i_m}s_\ell) \rightarrow (\mathbf{S}_{i_1}\mathbf{S}_{i_3}\dots\mathbf{S}_{i_m}s_\ell \vee [\text{call}_j^{i_1}]\mathbf{S}_{i_1}\mathbf{S}_{i_3}\dots\mathbf{S}_{i_m}s_\ell). \quad (4)$$

In words, if $\text{call}_j^{i_1}$ makes $\mathbf{S}_{i_1}\mathbf{S}_{i_2}\mathbf{S}_{i_3}\dots\mathbf{S}_{i_m}s_\ell$ true, then either $\mathbf{S}_{i_1}\mathbf{S}_{i_3}\dots\mathbf{S}_{i_m}s_\ell$ was true or it becomes true.

By Lemma 3, we know that the premise of (4) implies either $\bigwedge \text{EATM}(K_{i_1}K_{i_3}\dots K_{i_m}s_\ell)$ or $\bigwedge \text{EATM}(K_{i_2}K_{i_3}\dots K_{i_m}s_\ell)$ if $j = i_2$, or $\bigwedge \text{EATM}(K_jK_{i_2}K_{i_3}\dots K_{i_m}s_\ell)$ otherwise. It is possible to prove that each of these three statements implies either $\mathbf{S}_{i_1}\mathbf{S}_{i_3}\dots\mathbf{S}_{i_m}s_\ell$ or $[\text{call}_j^{i_1}]\mathbf{S}_{i_1}\mathbf{S}_{i_3}\dots\mathbf{S}_{i_m}s_\ell$, using the definition of calls and Lemma 1.

Therefore $\mathbf{S}_{i_1}\mathbf{S}_{i_2}\mathbf{S}_{i_3}\dots\mathbf{S}_{i_m}s_\ell \rightarrow \mathbf{S}_{i_1}\mathbf{S}_{i_3}\dots\mathbf{S}_{i_m}s_\ell$ is preserved by call_j^i , hence the result. \square

With this in mind, we look at some specific examples of goals.

The goal $\text{Goal}_{\{\mathbf{S}_1s_2\}}^{G\text{-neg}_D}$, where only 1 does not know the secret of 2, will always be reachable for $D = 1$ and $n \geq 3$. For example, FDSS-2014 returns the plan $\text{call}_3^1; \text{call}_4^1; \text{call}_5^2; \text{call}_5^3; \text{call}_5^4$ for $n = 5$. More generally, the following protocol gives a solution:

1. call_i^1 for every $i \in \text{Agt} \setminus \{2\}$;
2. solve G_D for $D = 1$ and $\text{Agt} = \{2, \dots, n\}$.

However, it will never be reachable for $D \geq 2$ and $n \geq 3$: by contraposition of Proposition 14, if \mathbf{S}_1s_2 is false then $\mathbf{S}_1\mathbf{S}_3s_2$ is false, thus we cannot reach the goal where only \mathbf{S}_1s_2 is false. FDSS-2014 indeed cannot find any plan for $D = 2$ and $n \leq 4$. (It is obviously unsolvable for any depth when $n = 2$ since the only available action, call_2^1 , establishes \mathbf{S}_1s_2 .)

Now suppose we have $D \geq 2$ and we want 1 not to know whether 2 knows the secret of 3 (but we do want 2 to know the secret of 3): our goal is $\text{Goal}_{\{\mathbf{S}_1\mathbf{S}_2s_3\}}^{G\text{-neg}_D}$. The following protocol produces a solution for $D = 2$ and $n \geq 3$:

1. call_i^2 for every $i \in \text{Agt} \setminus \{3\}$;
2. solve G_D for $D = 2$ and $\text{Agt} = \{1, 3, 4, \dots, n\}$;
3. call_i^2 for every $i \in \text{Agt} \setminus \{1\}$.

One of the plans FDSS-2014 finds is $\text{call}_2^1; \text{call}_4^2; \text{call}_5^2; \text{call}_5^3; \text{call}_5^4; \text{call}_4^1; \text{call}_4^2; \text{call}_5^2; \text{call}_3^2$ for $n = 5$. Again by Proposition 14, we know that if $\mathbf{S}_1\mathbf{S}_2s_3$ is false then $\mathbf{S}_1\mathbf{S}_4\mathbf{S}_2s_3$ is also false. Therefore this goal is always unreachable for $D \geq 3$ and $n \geq 4$. We can generalize this result: we have that $\text{Goal}_{\{\mathbf{S}_{i_1}\dots\mathbf{S}_{i_m}s_\ell\}}^{G\text{-neg}_D}$ is never reachable for $D \geq m+1$ and $n \geq m+2$.

Now consider the goal $\text{Goal}_{\{\mathbf{S}_1s_2, \mathbf{S}_2s_3\}}^{G\text{-neg}_D}$, where 1 must not know the secret of 2 and 2 must not know the secret of 3. For $D = 1$ and $n \geq 4$, the protocol for $\text{Goal}_{\mathbf{S}_1s_2}^{G\text{-neg}_D}$ generalizes as follows:

1. call_i^1 for every $i \in \text{Agt} \setminus \{2, 3\}$ ending with $i = n$;
2. $\text{call}_n^2; \text{call}_3^1$;
3. solve G_D for $D = 1$ and $\text{Agt} = \{3, \dots, n\}$.

For $n = 5$, FDSS-2014 returns $\text{call}_4^1; \text{call}_5^1; \text{call}_3^1; \text{call}_5^2; \text{call}_5^3; \text{call}_5^4$. However, and again by Proposition 14, we know that $\text{Goal}_{\{\mathbf{S}_1s_2, \mathbf{S}_2s_3\}}^{G\text{-neg}_D}$ will never be reachable for $D \geq 2$ and $n \geq 3$ (since, e.g., $\mathbf{S}_1\mathbf{S}_3s_2$ will also be false if \mathbf{S}_1s_2 is false).

7 Conclusion

In this article we have made a first step towards a realistic and provably-correct method for multi-agent epistemic planning. Our use of a logic of action and knowledge together with an state of the art automatic planner (which is assumed to be correct in the case of classical planning with conditional effects) provides a method for producing plans which are guaranteed to be correct.

Our approach contrasts with the undecidability of DEL-based epistemic planning which occurs even for simple fragments. For example, if actions make factual changes to the world, then the problem is undecidable whenever epistemic operators are allowed in preconditions; if actions are purely epistemic, then it is undecidable whenever two agents are involved or the epistemic depth exceeds 2 [1, 8]. Of course, the low complexity of DEL-PAO^S comes at the price of expressivity. We have seen that our epistemic logic EL-O^S has more validities than standard epistemic logic. We have also seen in the exam problem that considering knowledge instead of belief is a restriction leading to counter-intuitive design of actions (the teacher must not exit the room if she has seen the student see the exam). While relaxing knowledge in DEL is simple, this is not easy in DEL-PAO. However, our framework at least allows us to update knowledge along with facts of the world and to specify epistemic preconditions of any form. Since any epistemic formula can be reduced to a boolean formula, the translation to PDDL is immediate.

We intend to continue this line of research by incorporating other important aspects of multi-agent planning, namely control (i.e. which agents are allowed to change the value of which variables) and mutual exclusion (to guarantee that at most one agent has control of a variable at any instant). In the long term, we also aim to generalize this approach to temporal planning where actions are durative and may overlap; flexible planning, where actions may happen between intervals of time; and contingent planning, with uncertainty on the initial state or the effects of actions (and the presence of sensing actions). Another perspective is to encode DEL-PAO^S or even full DEL-PAO into PDDL. This would allow us to perform model checking with optimized PDDL planners.

We can note that, although we have mentioned only PDDL here, alternative approaches exist. For example, it is possible to code a planning problem containing actions with conditional effects directly into SAT and then use an efficient SAT solver to find a plan [21].

Acknowledgements

We would like to thank the anonymous reviewers for their thoughtful reading and comments.

REFERENCES

- [1] Guillaume Aucher and Thomas Bolander, 'Undecidability in epistemic planning', in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, ed., Francesca Rossi, pp. 27–33. AAAI Press, (2013).
- [2] Brenda Baker and Robert Shostak, 'Gossips and telephones', *Discrete Mathematics*, **2**(3), 191–193, (1972).
- [3] Philippe Balbiani, Andreas Herzig, François Schwarzentruber, and Nicolas Troquard, 'DL-PA and DCL-PC: model checking and satisfiability problem are indeed in PSPACE', *CoRR*, **abs/1411.7**, (2014).
- [4] Philippe Balbiani, Andreas Herzig, and Nicolas Troquard, 'Dynamic logic of propositional assignments: a well-behaved variant of PDL', in *Proceedings of the 28th Annual IEEE/ACM Symposium on Logic in Computer Science (LICS)*, ed., Orna Kupferman, pp. 143–152, (2013).
- [5] Thomas Bolander and Mikkel Birkegaard Andersen, 'Epistemic planning for single and multi-agent systems', *Journal of Applied Non-Classical Logics*, **21**(1), 9–34, (2011).
- [6] Tom Bylander, 'The computational complexity of propositional STRIPS planning', *Artificial Intelligence*, **69**, 165–204, (1994).
- [7] Tristan Charrier, Emiliano Lorini, Andreas Herzig, Faustine Maffre, and François Schwarzentruber, 'Building epistemic logic from observations and public announcements', in *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR 2016)*, (2016).
- [8] Tristan Charrier, Bastien Maubert, and François Schwarzentruber, 'On the impact of modal depth in epistemic planning', in *Proc. IJCAI 2016*. AAAI Press, (2016).
- [9] Tristan Charrier, Sophie Pinchinat, and François Schwarzentruber, 'Mental programs and arbitrary public announcement logic: relevance and complexity'. Unpublished manuscript, 2016.
- [10] Martin C. Cooper, Andreas Herzig, Faustine Maffre, Frédéric Maris, and Pierre Régnier, 'Simple epistemic planning: generalised gossiping', *ArXiv e-prints*, **abs/1606.0**, (2016).
- [11] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi, *Reasoning about Knowledge*, MIT Press, 1995.
- [12] Andras Hajnal, Eric C. B. Milner, and Endre Szemerédi, 'A cure for the telephone disease', *Canadian Mathematical Bulletin*, **15**(3), 447–450, (1972).
- [13] Andreas Herzig, Emiliano Lorini, and Faustine Maffre, 'A poor man's epistemic logic based on propositional assignment and higher-order observation', in *Proceedings of the 5th International Conference on Logic, Rationality and Interaction (LORI)*, eds., Wiebe van der Hoek, Wesley H. Holliday, and Wen-fang Wang, pp. 156–168. Springer Verlag, (2015).
- [14] Andreas Herzig, Emiliano Lorini, Nicolas Troquard, and Frédéric Moisan, 'A dynamic logic of normative systems', in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 228–233, (2011).
- [15] Andreas Herzig and Faustine Maffre, 'How to share knowledge by gossiping', in *Proceedings of the 3rd International Conference on Agreement Technologies (AT)*. Springer-Verlag, (2016).
- [16] Filippos Kominis and Hector Geffner, 'Beliefs in multiagent planning: from one agent to many', in *Proceedings of the 25th International Conference on Automated Planning and Scheduling (ICAPS)*, eds., Ronen I. Brafman, Carmel Domshlak, Patrik Haslum, and Shlomo Zilberstein, pp. 147–155. AAAI Press, (2015).
- [17] Benedikt Löwe, Eric Pacuit, and Andreas Witzel, 'DEL planning and some tractable cases', in *Proceedings of the 3rd International International Workshop on Logic, Rationality and Interaction*, pp. 179–192. Springer Berlin Heidelberg, (2011).
- [18] Drew McDermott, Malik Ghallab, Adele Howe, Craig Knoblock, Ashwin Ram, Manuela Veloso, Daniel Weld, and David Wilkins, 'PDDL – The Planning Domain Definition Language', Technical report, Yale Center for Computational Vision and Control, (1998).
- [19] Christian Muise, Vaishak Belle, Paolo Felli, Sheila A. McIlraith, Tim Miller, Adrian R. Pearce, and Liz Sonenberg, 'Planning over multi-agent epistemic states: A classical planning approach', in *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI 2015)*, pp. 3327–3334. AAAI Press, (2015).
- [20] Ronald P. A. Petrick and Fahiem Bacchus, 'Extending the Knowledge-Based Approach to Planning with Incomplete Information and Sensing', in *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling (ICAPS 2004)*, pp. 2–11, (2004).
- [21] Jussi Rintanen, Keijo Heljanko, and Ilkka Niemelä, 'Planning as satisfiability: parallel plans and algorithms for plan search', *Artif. Intell.*, **170**(12-13), 1031–1080, (2006).
- [22] Gabriele Röger, Florian Pommerening, and Jendrik Seipp, 'Fast downward stone soup 2014', in *The 2014 International Planning Competition*, (2014).
- [23] Robert Tijdeman, 'On a telephone problem', *Nieuw Archief voor Wiskunde*, **19**(3), 188–192, (1971).
- [24] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi, *Dynamic Epistemic Logic*, Springer Publishing Company, Incorporated, 1st edn., 2007.
- [25] Quan Yu, Yanjun Li, and Yanjing Wang, 'A dynamic epistemic framework for conformant planning', *Proceedings of TARK*, **15**, 249–259, (2015).

Lexicographic Refinements in Possibilistic Decision Trees

Nahla Ben Amor¹ and Zeineb El Khalfi² and H el ene Fargier³ and R egis Sabbadin⁴

Abstract. Possibilistic decision theory has been proposed twenty years ago and has had several extensions since then. Because of the lack of decision power of possibilistic decision theory, several refinements have then been proposed. Unfortunately, these refinements do not allow to circumvent the difficulty when the decision problem is sequential. In this article, we propose to extend lexicographic refinements to possibilistic decision trees. We show, in particular, that they still benefit from an Expected Utility (EU) grounding. We also provide qualitative dynamic programming algorithms to compute lexicographic optimal strategies. The paper is completed with an experimental study that shows the feasibility and the interest of the approach.

1 Introduction

For many years, there has been an interest in the Artificial Intelligence community towards the foundations and computational methods of decision making under uncertainty (see e.g. [1, 28, 7, 5, 16]). The usual paradigm of decision under uncertainty is based on the *Expected Utility (EU) model* [18, 23]. Its extensions to sequential decision making are *Decision Trees (DT)* [20] and *Markov Decision Processes (MDP)* [6, 19], where the uncertain effects of actions are represented by probability distributions.

When information about uncertainty cannot be quantified in a probabilistic way, possibilistic decision theory is a natural field to consider [14, 27, 12, 15, 10, 11, 15]. Qualitative decision theory is relevant, among other fields, for applications to planning under uncertainty, where a suitable *strategy* (i.e. a set of conditional or unconditional decisions) is to be found, starting from a qualitative description of the initial world, of the available decisions, of their (perhaps uncertain) effects and of the goal to reach (see [1, 3, 9, 8, 21, 22]).

Even though appealing for its ability to handle qualitative problems, possibilistic decision theory suffers from an important drawback. Acts (and strategies in sequential problems) are compared through min and max operators, which leads to a *drowning effect*: plausible enough bad or good consequences may blur the comparison between acts that would otherwise be clearly differentiable.

In order to overcome the drowning effect, refinements of possibilistic decision criteria have been proposed in the non-sequential case [13, 27]. Some refinements have the very interesting property to remain qualitative while satisfying the properties of EU. But these refinements do not extend to sequential decision under uncertainty (in the context of the present work, to decision trees) where the drowning effect is also due to the reduction of compound possibilistic strategies into simple ones [13].

The present paper proposes lexicographic refinements that compare full strategies (and not simply their reductions) and provides a dynamic programming algorithm to compute a lexicographic optimal strategy. It is a technical challenge to establish results of equivalence between lexicographic refinements of utilities of strategies in possibilistic decision trees and EU-based criteria. We prove such results, which opens the way to define dynamic programming solutions or even reinforcement learning algorithms for possibilistic MDPs [26, 25], which would not suffer from the drowning effect.

The paper is structured as follows ; the next Section recalls some results about the comparison of strategies in possibilistic decision trees. In Section 3, we define lexicographic orderings that refine the possibilistic criteria. Section 4 then proposes a dynamic programming algorithm for the computation of lexi-optimal strategies. Section 5 shows that the lexicographic criteria can be represented by *infinitesimal* expected utilities. The last Section reports experiments highlighting the feasibility and interest of the approach⁵.

2 Possibilistic decision trees

Decision trees provide an explicit modeling of sequential decision problems by representing, simply, all possible scenarios. The graphical component of a decision tree is a labelled graph $\mathcal{DT} = (\mathcal{N}, \mathcal{E})$. $\mathcal{N} = \mathcal{N}_D \cup \mathcal{N}_C \cup \mathcal{N}_U$ contains three kinds of nodes (see Figure 1):

- \mathcal{N}_D is the set of decision nodes (represented by squares);
- \mathcal{N}_C is the set of chance nodes (represented by circles);
- \mathcal{N}_U is the set of leaves, also called utility nodes.

For any node N , $Out(N)$ denotes its outgoing edges, $Succ(N)$ the set of its children nodes and $Succ(N, e)$ the child of N that is reached by edge $e \in Out(N)$. This tree represents a sequential decision problem as follows:

- Leaf nodes correspond to states of the world in which a utility is obtained (for the sake of simplicity we assume that utilities are attached to leaves only); the utility of a leaf node $L_i \in \mathcal{N}_U$ is denoted $u(L_i)$.
- Decision nodes correspond to states of the world in which a decision is to be made: $D_i \in \mathcal{N}_D$ represents a decision variable Y_i the domain of which corresponds to the labels a of the edges starting from D_i . These edges lead to chance nodes, i.e. $Succ(D_i) \subseteq \mathcal{N}_C$.
- A state variable X_j is assigned to each chance node $C_j \in \mathcal{N}_C$, the domain of which corresponds to the labels x of the edges starting from that node. Each edge starting from a chance node C_j represents an event $X_j = x$. For any $C_j \in \mathcal{N}_C$, $Succ(C_j) \subseteq \mathcal{N}_U \cup \mathcal{N}_D$ i.e. after the execution of a decision, either a leaf node or a decision node is reached.

¹ LARODEC, Tunisie, email: nahla.benamor@gmx.fr

² LARODEC, Tunisie, IRIT, France, email: zeineb.khalfi@gmail.com

³ IRIT, France, email: fargier@irit.fr

⁴ INRA-MIAT, France, email: rsabbadin@toulouse.inra.fr

⁵ The proofs are omitted for the sake of brevity but are available at <https://www.irit.fr/publis/ADRIA/PapersFargier/ecai2016.pdf>

$Start(\mathcal{DT})$ denotes the first decision nodes of the tree (it is a singleton containing the root of the tree if it is a decision node, or its successors if the root is a chance node). For the sake of simplicity, we suppose that all the paths from the root to a leaf in the tree have the same length: h , the horizon of the decision tree, is the number of decision nodes along these paths. Given a node N of \mathcal{DT} , we shall also consider the subproblem \mathcal{DT}_N defined by the tree rooted in N .

The joint knowledge on the state variables is not given in extenso, but through the labeling of the edges issued from chance nodes. In a possibilistic context the uncertainty pertaining to the possible outcomes of each X_j is represented by a possibility distribution: each edge starting from C_j , representing an event $X_j = x$, is endowed with a number $\pi_j(x)$, the possibility $\pi(X_j = x | past(C_j))$ ⁶. A possibilistic ordered scale, $L = \{\alpha_0 = 0_L < \alpha_1 < \dots < \alpha_l = 1_L\}$, is used to evaluate the utilities and possibilities.

Solving a decision tree amounts to building a *strategy*, i.e. a function $\delta : \mathcal{N}_D \mapsto A$, where A is the set of possible actions, including a special “undefined” action \perp , chosen for action nodes which are left unexplored by a given strategy. Admissible strategies assign a chance node to each reachable decision node, i.e. must be:

- *sound*: $\forall D_i \in \mathcal{N}_D, \delta(D_i) \in Out(D_i) \cup \{\perp\} \subseteq A$, and
- *complete*: (i) $\forall D_i \in Start(\mathcal{DT}), \delta(D_i) \neq \perp$ and (ii) $\forall D_i$ s.t. $\delta(D_i) \neq \perp, \forall N \in Succ(Succ(D_i, \delta(D_i)))$ either $\delta(N) \neq \perp$ or $N \in \mathcal{N}_U$.

We denote by Δ_N (or simply Δ , when there is no ambiguity) the set of admissible strategies built from a tree rooted in N . Each strategy δ defines a connected subtree of DT , the branches of which represent possible scenarios, or *trajectories*. Formally, a trajectory $\tau = (a_{j_0}, x_{i_1}, a_{j_1}, \dots, a_{j_{h-1}}, x_{i_h})$ is a sequence of value assignments to decision and chance variables along a path from a starting decision node (a node in $Start(\mathcal{DT})$) to a leaf: $Y_0 = a_{j_0}$ is the first decision in the trajectory, x_{i_1} the value taken by its first chance variable, X_{j_0} in this scenario, $Y_{i_1} = a_{j_1}$ is the second decision, etc.

We identify a strategy δ , the corresponding subtree and the list of its trajectories represented by a matrix. We also consider subtrees, and thus sub-strategies: let C_j be a chance node, D_{i_1}, \dots, D_{i_k} its successors and, for $l = 1, k$, the strategies $\delta_{i_l} \in \Delta_{D_{i_l}}$ which solve the subproblem rooted in D_{i_l} . $\delta_{i_1} + \dots + \delta_{i_k}$ is the strategy of Δ_{C_j} resulting from the composition of the δ_{i_l} : $(\delta_{i_1} + \dots + \delta_{i_k})(N) = \delta_{i_l}(N)$ iff N belongs to the subtree rooted in D_{i_l} .

Example 1 Let us suppose that a “Rich and Unknown” person runs a startup company. In every state she must choose between Investing (Inv) or Advertising (Adv) and she may be then Rich (R) or Poor (P) and Famous (F) or Unknown (U). Figure 1 shows the possibilistic decision tree (with horizon $h = 2$) that represents this decision problem. This tree has 8 strategies, 16 trajectories:

$\tau_1 = (Adv, R\&U, Inv, P\&U)$, $\tau_2 = (Adv, R\&U, Inv, R\&U)$,
 $\tau_3 = (Adv, R\&U, Adv, R\&U)$, $\tau_4 = (Adv, R\&U, Adv, R\&F)$,
 $\tau_5 = (Adv, R\&F, Adv, R\&U)$, $\tau_6 = (Adv, R\&F, Adv, R\&F)$,
 etc.

The evaluation of a possibilistic strategy, as proposed by [22], relies on the qualitative optimistic and pessimistic decision criteria axiomatized by [11]. The utility of the strategy is computed on the basis of the transition possibilities and the utilities of its trajectories. For each trajectory $\tau = (a_{j_0}, x_{i_1}, a_{j_1}, \dots, x_{i_h})$:

⁶ As in classical probabilistic decision trees, it is assumed that $\pi(X_j = x | past(C_j))$ only depends on the variables in $past(C_j)$ and actually only on the decision made in the preceding node and on the state of the preceding chance node.

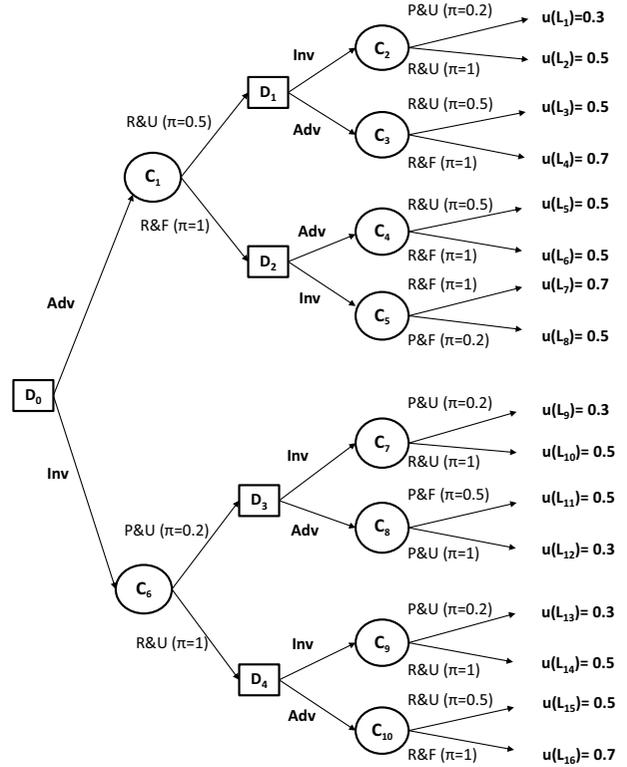


Figure 1. The possibilistic decision tree of Example 1

- Its utility denoted $u(\tau)$, is the utility $u(x_{i_h})$ of its leaf.
- The possibility of τ given that a strategy δ is applied from initial node D_0 is defined by:

$$\pi(\tau | \delta, D_0) = \begin{cases} \min_{k=1..h} \pi_{j_{k-1}}(x_{i_k}) & \text{if } \tau \text{ is a trajectory of } \delta, \\ 0 & \text{otherwise.} \end{cases}$$

where $\pi_{j_{k-1}}$ is the possibility distribution at $C_{j_{k-1}}$.

It is now possible to compute, for any $\delta \in \Delta$ its optimistic and pessimistic utility degrees (the higher, the better):

$$u_{opt}(\delta) = \max_{\tau \in \delta} \min(\pi(\tau | \delta, D_0), u(\tau))$$

$$u_{pes}(\delta) = \min_{\tau \in \delta} \max(1 - \pi(\tau | \delta, D_0), u(\tau))$$

This approach is purely ordinal (only min and max operations are used to aggregate the evaluations of the possibility of events and the ones of the utility of states). We can check that the preference orderings \succeq_O between strategies, derived either from u_{opt} ($O = u_{opt}$) or from u_{pes} ($O = u_{pes}$), satisfy the principle of weak monotonicity:

$\forall C_j \in \mathcal{N}_{C_j}, \forall D_i \in Succ(C_j), \delta, \delta' \in \Delta_{D_i}, \delta'' \in \Delta_{Succ(C_j) \setminus D_i}$:

$$\delta \succeq_O \delta' \implies \delta + \delta'' \succeq_O \delta' + \delta''$$

This property guarantees that *dynamic programming* [2] applies, and provides an optimal strategy in time polynomial with the size of the tree: [21, 22] have proposed qualitative counterparts of stochastic dynamic programming algorithms: in the finite horizon case *backwards induction*, or in the infinite horizon case *value and policy iteration*.

The basic pessimistic and optimistic utilities nevertheless present a severe drawback, known as the "drowning effect", due to the use of idempotent operations. In particular, when two strategies give an identical and extreme (either good, for u_{opt} or bad, for u_{pes}), utility in some plausible trajectory, they may be undistinguished although they may give significantly different consequences in other possible trajectories, as illustrated in Example 2.

Example 2 Let δ and δ' be the two strategies of Example 1 defined by $\delta(D_0) = \delta'(D_0) = Adv$; $\delta(D_1) = Inv$; $\delta'(D_1) = Adv$; $\delta(D_2) = \delta'(D_2) = Adv$. δ gathers 4 trajectories, $\tau_1, \tau_2, \tau_5, \tau_6$ with $\pi(\tau_1|D_0, \delta) = 0.2$ and $u(\tau_1) = 0.3$; $\pi(\tau_2|D_0, \delta) = 0.5$ and $u(\tau_2) = 0.5$; $\pi(\tau_5|D_0, \delta) = 0.5$ and $u(\tau_5) = 0.5$; $\pi(\tau_6|D_0, \delta) = 1$ and $u(\tau_6) = 0.5$. Hence $u_{opt}(\delta) = u_{pes}(\delta) = 0.5$.

- δ' is also composed of 4 trajectories ($\tau_3, \tau_4, \tau_5, \tau_6$). Hence $u_{opt}(\delta') = u_{pes}(\delta') = 0.5$.

Thus $u_{opt}(\delta) = u_{opt}(\delta')$ and $u_{pes}(\delta) = u_{pes}(\delta')$: δ' , which provides at least utility 0.5 in all trajectories, is not preferred to δ that provides a bad utility (0.3) in some non impossible trajectory (τ_1). τ_2 , which is good and totally possible "drowns" the bad consequence of δ in τ_1 in the optimistic comparison; in the pessimistic one, the bad utility of τ_1 is drowned by its low possibility, hence a global degree u_{pes} that is equal to the one of δ' (that, once again, guarantees a 0.5 utility degree at least).

The two possibilistic criteria thus may fail to satisfy the principle of Pareto efficiency, that may be written as follows, for any optimization criterion O (here u_{pes} or u_{opt}):

$\forall \delta, \delta' \in \Delta$, if (i) $\forall D \in \text{Common}(\delta, \delta'), \delta_D \succeq_O \delta'_D$ and (ii) $\exists D \in \text{Common}(\delta, \delta'), \delta_D \succ_O \delta'_D$, then $\delta \succ_O \delta'$ where $\text{Common}(\delta, \delta')$ is the set of nodes for which both δ and δ' provide an action and δ_D (resp. δ'_D) is the restriction of δ (resp. δ') to the subtree rooted in D .

Moreover, neither u_{opt} or u_{pes} do fully satisfy the classical, strict, monotonicity principle, that can be written as follows:

$\forall C_j \in \mathcal{N}_C, D_i \in \text{Succ}(C_j), \delta, \delta' \in \Delta_{D_i}, \delta'' \in \Delta_{\text{Succ}(C_j) \setminus D_i}$,

$$\delta \succeq_O \delta' \iff \delta + \delta'' \succeq_O \delta' + \delta''$$

It may indeed happen that $u_{pes}(\delta) > u_{pes}(\delta')$ while $u_{pes}(\delta + \delta'') = u_{pes}(\delta' + \delta'')$ (or that $u_{opt}(\delta) > u_{opt}(\delta')$ while $u_{opt}(\delta + \delta'') = u_{opt}(\delta' + \delta'')$).

The purpose of the present work is to build efficient preference relations that agree with the qualitative utilities when the latter can make a decision, and break ties when not - to build refinements⁷ that satisfy the principle of Pareto efficiency.

3 Escaping the drowning effect by leximin and leximax comparisons

The possibilistic drowning effect is due to the use of min and max operations. In ordinal aggregations, this drawback is well known and it has been overcome by means of leximin and leximax comparisons [17]. More formally, for any two vectors t and t' :

• $t \succeq_{lmin} t'$ iff $\forall i, t_{\sigma(i)} = t'_{\sigma(i)}$ or $\exists i^*, \forall i < i^*, t_{\sigma(i)} = t'_{\sigma(i)}$ and $t_{\sigma(i^*)} > t'_{\sigma(i^*)}$

• $t \succeq_{lmax} t'$ iff $\forall i, t_{\mu(i)} = t'_{\mu(i)}$ or $\exists i^*, \forall i < i^*, t_{\mu(i)} = t'_{\mu(i)}$ and $t_{\mu(i^*)} > t'_{\mu(i^*)}$

⁷ Formally, a preference relation \succeq' refines a preference relation \succeq if and only if whatever δ, δ' , if $\delta \succ \delta'$ then $\delta \succ' \delta'$.

where, for any vector v (here, $v = t$ or $v = t'$), $v_{\mu(i)}$ (resp. $v_{\sigma(i)}$) is the i^{th} best (resp. worst) element of v .

The refinements of u_{opt} and u_{pes} by lexicographic principles have been considered by [13] for non sequential problems; in this context, a decision is a possibility distribution π over the utility degrees, i.e. a vector of pairs $(\pi(u), u)$. Then it is possible to write:

• $\pi \succeq_{lmax(lmin)} \pi'$ iff $\forall i, (\pi(u), u)_{\mu(i)} \sim_{lmin} (\pi'(u), u)_{\mu(i)}$ or $\exists i^*, \forall i < i^*, (\pi(u), u)_{\mu(i)} \sim_{lmin} (\pi'(u), u)_{\mu(i)}$ and $(\pi(u), u)_{\mu(i^*)} \succ_{lmin} (\pi'(u), u)_{\mu(i^*)}$.

• $\pi \succeq_{lmin(lmax)} \pi'$ iff $\forall i, (1 - \pi(u), u)_{\sigma(i)} \sim_{lmax} (1 - \pi'(u), u)_{\sigma(i)}$ or $\exists i^*, \forall i < i^*, (1 - \pi(u), u)_{\sigma(i)} \sim_{lmax} (1 - \pi'(u), u)_{\sigma(i)}$ and $(1 - \pi(u), u)_{\sigma(i^*)} \succ_{lmax} (1 - \pi'(u), u)_{\sigma(i^*)}$.

where $(\pi(u), u)_{\mu(i)}$ is the i^{th} best pair of $(\pi(u), u)$ according to $lmin$ and $(1 - \pi(u), u)_{\sigma(i)}$ is the i^{th} worst pair of $(1 - \pi(u), u)$ according to $lmax$.

A straightforward way of applying this to sequential decision is to reduce the compound possibility distribution corresponding to the strategy, as usually done in possibilistic (and probabilistic) decision trees. The reduction of δ yields the distribution π_δ on the utility degrees, defined by: $\pi_\delta(u) = \max_{\tau, u(\tau)=u} \pi(\tau|\delta, D_0)$. Then we can write:

$$\begin{aligned} \delta \succeq_{lmax(lmin)} \delta' &\quad \text{iff} \quad \pi_\delta \succeq_{lmax(lmin)} \pi_{\delta'}, \\ \delta \succeq_{lmin(lmax)} \delta' &\quad \text{iff} \quad \pi_\delta \succeq_{lmin(lmax)} \pi_{\delta'}. \end{aligned}$$

$\succeq_{lmax(lmin)}$ (resp. $\succeq_{lmin(lmax)}$) refines $\succeq_{u_{opt}}$ (resp. $\succeq_{u_{pes}}$), but neither $\succeq_{lmax(lmin)}$ nor $\succeq_{lmin(lmax)}$ do satisfy Pareto efficiency, as shown by the following counterexample.

Example 3 Consider a modified version of the problem of Example 1 (Figure 2). δ and δ' are the two strategies defined by: $\delta(D_0) = \delta'(D_0) = Adv$, $\delta(D_1) = Inv$, $\delta'(D_1) = Adv$, $\delta(D_2) = \delta'(D_2) = Adv$. $\text{Common}(\delta, \delta') = \{D_0, D_1, D_2\}$, $\delta_{D_0} = \delta'_{D_0}$, $\delta_{D_2} = \delta'_{D_2}$ and δ_{D_1} dominates δ'_{D_1} w.r.t. $lmax(lmin)$, since $((1, 0.1), (1, 0.9)) \succ_{lmax(lmin)} ((1, 0.1)(0.5, 0.9))$. δ should then be strictly preferred to δ' . By reduction, we get $\pi_\delta(0.9) = \pi_{\delta'}(0.1) = \min(0.4, 1) = 0.4$ and $\pi_\delta(0.8) = \min(1, 1) = 1$ and for δ' we have $\pi_{\delta'}(0.9) = \min(0.4, 0.5) = 0.4$, $\pi_{\delta'}(0.1) = \min(0.4, 1) = 0.4$ and $\pi_{\delta'}(0.8) = \min(1, 1) = 1$: δ and δ' are indifferent for $\succeq_{lmax(lmin)}$. This contradicts Pareto efficiency.

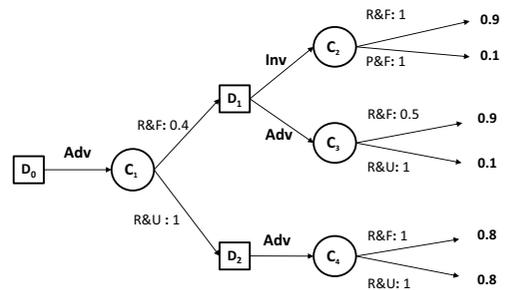


Figure 2. A counter example at the efficiency of $\succeq_{lmax(lmin)}$

The drowning effect at work here is due to the reduction of strategies, namely to the fact that the possibility of a trajectory

is drowned by the one of the least possible of its edges. That is why we propose to give up the principle of reduction and to build lexicographic comparisons on strategies considered *in extenso*.

Recall that: $u_{opt}(\delta) = \max_{\tau \in \delta} \min \left\{ \min_{k=1..h} \pi_{j_{k-1}}(x_{i_k}); u(x_{i_h}) \right\}$.

Then, for any $\tau = (a_{j_0}, x_{i_1}, \dots, a_{j_{h-1}}, x_{i_h})$ and $\tau' = (a_{j'_0}, x_{i'_1}, \dots, a_{j'_{h-1}}, x_{i'_h})$, we define \succeq_{lmin} and \succeq_{lmax} by:

- $\tau \succeq_{lmin} \tau'$ iff $(\pi_{j_0}(x_{i_1}), \dots, \pi_{j_{h-1}}(x_{i_h}), u(x_{i_h})) \succeq_{lmin} (\pi_{j'_0}(x_{i'_1}), \dots, \pi_{j'_{h-1}}(x_{i'_h}), u(x_{i'_h}))$
- $\tau \succeq_{lmax} \tau'$ iff $(1 - \pi_{j_0}(x_{i_1}), \dots, 1 - \pi_{j_{h-1}}(x_{i_h}), u(x_{i_h})) \succeq_{lmax} (1 - \pi_{j'_0}(x_{i'_1}), \dots, 1 - \pi_{j'_{h-1}}(x_{i'_h}), u(x_{i'_h}))$

Hence the proposition of the following preference relations⁸:

- $\delta \succeq_{lmax(lmin)} \delta'$ iff $\forall i, \tau_{\mu(i)} \sim_{lmin} \tau'_{\mu(i)}$ or $\exists i^*, \forall i \leq i^*, \tau_{\mu(i)} \sim_{lmin} \tau'_{\mu(i)}$ and $\tau_{\mu(i^*)} \succ_{lmin} \tau'_{\mu(i^*)}$,
- $\delta \succeq_{lmin(lmax)} \delta'$ iff $\forall i, \tau_{\sigma(i)} \sim_{lmax} \tau'_{\sigma(i)}$ or $\forall i, \tau_{\sigma(i)} \sim_{lmax} \tau'_{\sigma(i)}$ or $\exists i^*, \forall i \leq i^*, \tau_{\sigma(i)} \sim_{lmax} \tau'_{\sigma(i)}$ and $\tau_{\sigma(i^*)} \succ_{lmax} \tau'_{\sigma(i^*)}$,

where $\tau_{\mu(i)}$ (resp. $\tau'_{\mu(i)}$) is the i^{th} best trajectory of δ (resp δ') according to \succeq_{lmin} and $\tau_{\sigma(i)}$ (resp. $\tau'_{\sigma(i)}$) is the i^{th} worst trajectory of δ (resp δ') according to \succeq_{lmax} .

These relations are relevant refinements and escape the drowning effect - they are those we are looking for:

Proposition 1 $\succeq_{lmax(lmin)}$ is complete, transitive and refines $\succeq_{u_{opt}}$; $\succeq_{lmin(lmax)}$ is complete, transitive and refines $\succeq_{u_{pes}}$.

Proposition 2 $\succeq_{lmax(lmin)}$ and $\succeq_{lmin(lmax)}$ both satisfy the principle of Pareto efficiency as well as strict monotonicity.

Propositions 1 and 2 have important consequences; from a prescriptive point of view, they outline the rationality of $lmax(lmin)$ and $lmin(lmax)$ and suggest a probabilistic interpretation, which we develop in Section 5. From a practical point of view, they allow us to define a dynamic programming algorithm to get lexi optimal solutions - this is the topic of the next Section.

4 Dynamic Programming for lexi qualitative criteria

The algorithm we propose (Algorithm 1 for the $lmax(lmin)$ variant; the $lmin(lmax)$ variant is similar) proceeds in the classical way, by backwards induction: when a chance node is reached, an optimal substrategy is recursively built for each of its children; these substrategies are combined but the resulting strategy is NOT reduced, contrarily to what is classically done; when a decision node is reached, the program is called for each child and the best of them is selected.

The comparison of strategies is done on the basis of the matrices of their trajectories (denoted ρ ; each line gathers the possibility and utility degrees of a trajectory $\tau = (a_{j_0}, x_{i_1}, a_{j_1}, \dots, a_{j_h}, x_{i_h})$):

$$\rho_{it} = \begin{cases} \pi_{j_{t-1}}(x_{i_t}) & \text{if } t \leq h, O = lmax(lmin) \\ 1 - \pi_{j_{t-1}}(x_{i_t}) & \text{if } t \leq h, O = lmin(lmax) \\ u(x_{i_h}) & \text{if } t = h + 1. \end{cases}$$

So as to allow fast comparisons, the matrices are built incrementally and ordered on the fly by the function *ConcatAndOrder*: when a

⁸ If the strategies have different numbers of trajectories, neutral trajectories (vectors) are added to the shortest strategy, at the bottom of the shortest list of trajectories

Algorithm 1: DynProgLmaxLmin(N:Node)

Data: δ , the strategy built by the algorithm, is a global variable

Result: Computes δ for \mathcal{DT}_N and returns the matrix of its trajectories, ρ

```

begin
  // Leaves
  if  $N \in \mathcal{N}_U$  then  $\rho = [u(N)]$ ;
  // Chance nodes
  if  $N \in \mathcal{C}$  then
     $k = |Succ(N)|$ ;
    for  $D_i \in Succ(N)$  do
       $\rho_i \leftarrow DynProgLmaxLmin(D_i)$ ;
       $\rho \leftarrow ConcatAndOrder(\rho_1, \dots, \rho_k, \pi_N)$ ;
  // Decision nodes
  if  $N \in \mathcal{D}$  then
     $\rho \leftarrow [0]$ 
    foreach  $a_j \in Out(N)$  do
       $\rho_j \leftarrow DynProgLmaxLmin(Succ(N, a_j))$ ;
      if  $\rho_j \succeq_{lmax(lmin)} \rho$  then
         $\rho \leftarrow \rho_j$  and  $\delta(N) \leftarrow a_j$ ;
  return  $\rho$ ;
```

chance node, say C_j is reached, $k = |Succ(C_j)|$ substrategies are built recursively and their matrices ρ_1, \dots, ρ_k are computed. Matrix ρ of the current (compound) strategy, for the subtree rooted in C_j , is obtained by calling *ConcatAndOrder* ($\rho_1, \dots, \rho_k, \pi_{C_j}$). This function adds a column to each ρ_i , filled with $\pi_j(x_i)$; the matrices are vertically concatenated; then the elements in the lines are ordered in decreasing (resp. increasing) order, and the lines are reordered by decreasing (resp. increasing) order w.r.t. to $lmax$ (resp. $lmin$). As a matter of fact, once ρ has been reordered, $\rho_{1,1}$ is always equal to $u_{opt}(\delta)$ (resp. $u_{pes}(\delta)$).

The lexicographic comparison of two strategies δ and δ' is performed by scanning the elements $\rho_{l,t}$ and $\rho'_{l,t}$ of ρ and ρ' in parallel, line by line from the first one. The first pair of different ($\rho_{l,t}, \rho'_{l,t}$) determines the best matrix/strategy. If the matrices have different numbers of lines, neutral lines are added at the bottom of the shortest one (filled with 0 for the optimistic case, with 1 for the pessimistic one).

Even if working with matrices rather than numerical values, the algorithm is polynomial w.r.t. the size of the original tree. This is because (i) the algorithm crosses each edge of the tree only once (as in the classical version), (ii) the matrices are never bigger than the strategies and (iii) the comparison of strategies is done in time linear with their size - thus linear with the size of the original tree.

5 Lexi comparisons and Expected Utility

If the problem is not sequential, it is easy to see that the comparison of possibilistic utility distributions by $\succeq_{lmax(lmin)}$ and $\succeq_{lmin(lmax)}$ do satisfy the axioms of EU. [13] have indeed shown that these decision criteria can be captured by an EU - namely, relying on infinitesimal probabilities and utilities. In this Section, we claim that such a result can be extended to sequential problems - for decision trees.

The proof relies on a transformation of the possibilistic tree into a probabilistic one. The graphical components are identical and so are the sets of admissible strategies. In the optimistic case the probability and utility distributions are chosen in such a way that the $lmax(lmin)$ and EU criteria do provide the same preference on Δ . To this extent, we build a transformation $\phi : L \subseteq [0, 1] \rightarrow [0, 1]$

that maps each possibility distribution to an additive distribution and each utility level into an additive one; this transformation is required to satisfy the following condition:

$$(R) : \forall \alpha, \alpha' \in L \text{ such that } \alpha > \alpha' : \phi(\alpha)^{h+1} > b^h \phi(\alpha'),$$

where b is the branching factor of the tree. Condition (R) guarantees that if $u_{opt}(\delta) = \alpha > u_{opt}(\delta') = \alpha'$, then a comparison based on a sum-product approach on the new tree will also decide in favor of δ .

For any chance node C_j , a local transformation ϕ_j is then derived from ϕ , such that ϕ_j satisfies both condition (R) and the normalization condition of probability theory. EU_{opt} denotes the preference relation provided by the EU-criterion on the probabilistic tree obtained by replacing each π_j by $\phi_j \circ \pi_j$ and the utility function u by $\phi \circ u$. We show that:

Proposition 3 *If (R) holds, then $\succeq_{EU_{opt}}$ refines $\succeq_{u_{opt}}$.*

Proposition 4 $\delta \succeq_{lmax(lmin)} \delta'$ iff $\delta \succeq_{EU_{opt}} \delta', \forall (\delta, \delta') \in \Delta$.

Example 4 $\phi(1) = 1, \phi(0.9) = 0.2, \phi(0.8) = 0.001, \phi(0.5) = 10^{-10}, \phi(0.4) = 10^{-30}, \phi(0.1) = 10^{-91}$.

It holds that $\phi(\alpha)^3 > \phi(\alpha') * 2^2$, for all $\alpha > \alpha'$. We obtain the transformed conditional distributions by normalizing on each node. For instance for node C_1 , $\phi_1(10^{-30}) = \frac{10^{-30}}{1+10^{-30}}$ and $\phi_1(1) = \frac{1}{1+10^{-30}}$, for node C_2 , $\phi_2(1) = \frac{1}{1+1}$ and $\phi_2(1) = 0.5$, for node C_3 , $\phi_3(10^{-10}) = \frac{10^{-10}}{1+10^{-10}}$ and $\phi_3(1) = \frac{1}{1+10^{-10}}$, for node C_4 , $\phi_4(1) = 0.5$ and $\phi_4(1) = 0.5$.

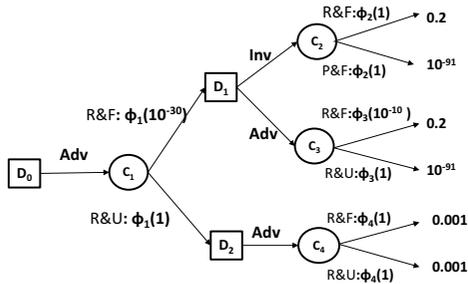


Figure 3. Transformed probabilistic decision tree of possibilistic decision tree of (counter)-example 3

The construction is a little more complex if we consider the $\succeq_{lmin(lmax)}$ comparison, where the utility degrees are not directly compared to possibility degrees π but to degrees $1 - \pi$. Hopefully, it is possible to rely on the results obtained for the optimistic case, since the optimistic and pessimistic utilities are dual of each other.

Proposition 5 *Let \mathcal{DT}^{inv} the tree obtained from \mathcal{DT} by using utility function $u' = 1 - u$ on leaves. It holds that: $u_{pes, \mathcal{DT}}(\delta) \geq u_{pes, \mathcal{DT}}(\delta')$ iff $u_{opt, \mathcal{DT}^{inv}}(\delta') \geq u_{opt, \mathcal{DT}^{inv}}(\delta)$*

As a consequence, we build an EU-based equivalent of $\succeq_{lmin(lmax)}$, denoted $\succeq_{EU_{pes}}$, by replacing each possibility distribution π_i in \mathcal{DT} by the probability distribution $\phi_i \circ \pi_i$, as for the optimistic case and each utility degree u by $\phi(1) - \phi(u)$. It is then possible to show that:

Proposition 6 $\delta \succeq_{lmin(lmax)} \delta'$ iff $\delta \succeq_{EU_{pes}} \delta', \forall (\delta, \delta') \in \Delta$.

Propositions 4 and 6 show that lexi-comparisons have a probabilistic interpretation - actually, using infinitesimal probabilities and utilities. This result comforts the idea, first proposed by [4] and then by [13], of a bridge between qualitative approaches and probabilities, through the notion of big stepped probabilities [4, 24]. We make here a step further, by the identification of transformations that support sequential decision making.

Beyond this theoretical argument, this result suggests an alternative algorithm for the optimization of $lmax(lmin)$ (resp. $lmin(lmax)$): simply transform the possibilistic decision tree into a probabilistic one and use a classical, EU-based algorithm of dynamic programming. In a perfect world, both approaches solve the problem in the same way and provide the same optimal strategies - the difference being that the first one is based on the comparison of matrices, the second one on expected utilities in \mathbb{R}^+ . The point is that the latter handles infinitesimals; then either the program is based on an explicit handling of infinitesimals, and proceeds just like the matrix-based comparison, or it lets the programming language handle these numbers in its own way - and, given the precision of the computation, provides approximations.

6 Experiments

We thus get three criteria for each of the pessimistic and optimistic approaches: the basic possibilistic ones, the lexicographic refinements described in Section 3, and the EU approximations of the latter. We compare the 3 variants within each series with two measures: the CPU time and a pairwise success rate: $Success_{\frac{A}{B}}$ is the percentage of solutions provided by an algorithm optimizing criterion A that are optimal with respect to criterion B ; for instance, the lower $Success_{\frac{u_{opt}}{lmax(lmin)}}$, the more important the drowning effect.

The backward induction algorithms corresponding to the six criteria have been implemented in Java. As to the EU-based approaches, the transformation function depends on the horizon h and the branching factor b (here $b = 2$). We used $\phi(1_L) = 1, \phi(\alpha_i) = \frac{\phi(\alpha_{i+1})^{h+1}}{b^h * 1.1}$, each ϕ_j being obtained by normalization of ϕ on C_j . The experiments have been performed on an Intel Core i5 processor computer (1.70 GHz) with 8GB DDR3L of RAM..

The tests were performed on complete binary decision trees, for $h = 2$ to $h = 7$, that are randomly generated. The first node is a decision node: at each decision level from the root ($i = 1$) to the last level ($i = 7$) the tree contains 2^{i-1} decision nodes. This means that with $h = 2$ (resp. 3, 4, 5, 6, 7), the number of decision nodes is equal to 5 (resp. 21, 85, 341, 1365, 5461). The utility values are uniformly randomly fired in the set $L = \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$. Conditional possibilities relative to chance nodes are normalized, one edge having possibility one and the possibility degree of the other is uniformly fired in L . For each value of h , 100 decision trees are generated.

Figure 4 presents the average execution CPU time for the six criteria. We observe that, whatever the optimized criterion, the CPU time increases linearly w.r.t. the number of decision nodes, which is in line with what we could expect. Furthermore, it remains affordable with big trees: the maximal CPU time is lower than 1s for a decision tree with 5461 decision nodes. It appears that u_{opt} is always faster than EU_{opt} , which is 1.5 or 2 times faster than $lmax(lmin)$. The same conclusion is drawn when comparing $lmin(lmax)$ to u_{pes} and EU_{pes} . These results are easy to explain: (i) the manipulation of matrices is obviously more expensive than the one of numbers and

(ii) the handling of numbers by min and max operations is faster than sum-product manipulations of infinitesimal.

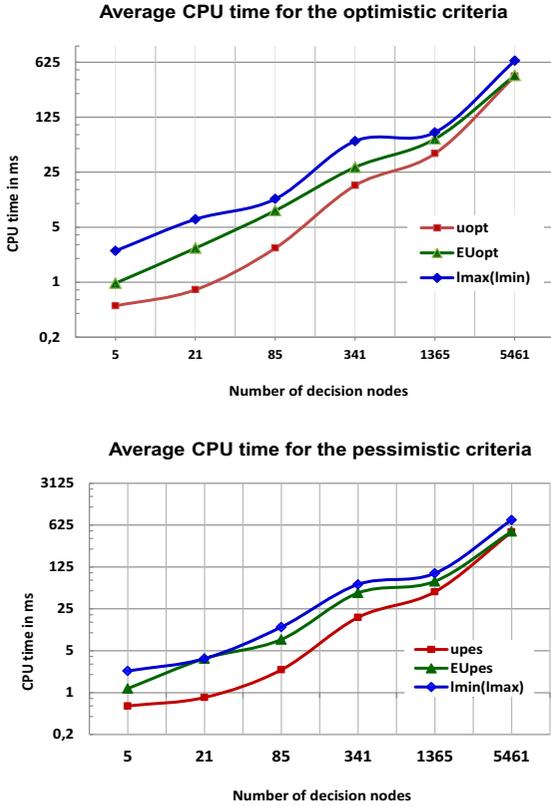


Figure 4. Average CPU time (in ms) for h=2 to 7

As to the success rate, the results are described in Figure 5. The percentage of solutions optimal for u_{opt} (resp. for u_{pes}) that are also optimal for $lmax(lmin)$ (resp. $lmin(lmax)$) is never more than 82%, and decreases when the horizon increases: the drowning effect is not negligible and increases with the length of the trajectories. Moreover EU_{opt} (resp. EU_{pes}) does not perform well as an approximation of $lmax(lmin)$ (resp. $lmin(lmax)$): the percentage of solutions optimal for the former which are also optimal for the latter is lower than 80% in all cases, and decreases when h increases. This is easily explained by the fact that the probabilities are infinitesimals and converge to 0 when the length of the branches (and thus the number of factors in the products) increase, as suggested in Section 5.

These experiments conclude in favor of the lexi refinements in their full definition - their approximation by expected utilities are comparable in terms of CPU efficiency but not precise enough. The EU criteria nevertheless offer a better approximation than u_{opt} and u_{pes} when space is limited (or when h increases).

7 Concluding remarks

This work has both theoretical and practical implications. It extends and generalizes to sequential problems the theoretical links established in [13] between possibilistic utilities and expected utilities. It performs better than the refinement of binary possibilistic utilities

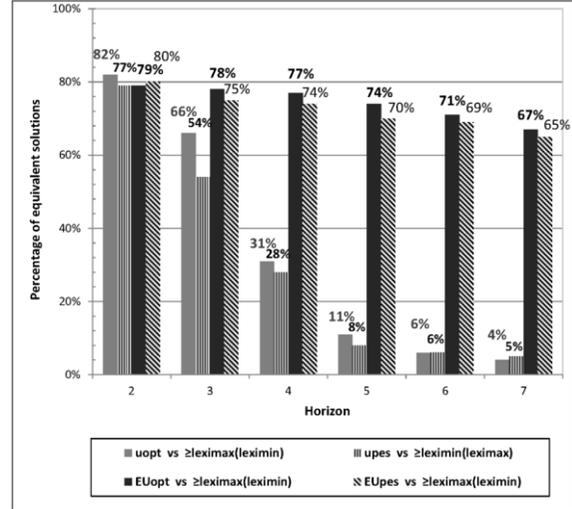


Figure 5. Success rate

(BPU) proposed in [27] for Binary Possibilistic Utilities and as a particular case, to classical, optimistic and pessimistic, possibilistic utilities. In [27]’s treatment indeed, two similar trajectories of the same strategy are merged. The resulting criterion thus suffers from a drowning effect and does not satisfy strict monotonicity: as such, it cannot be represented by an EU-based criterion which “counts” trajectories (weighted by their probabilities). We actually do refine [27]’s criterion. Incorporating our lexicographic refinements in BPU would lead to a more powerful refinement and suggests a probabilistic interpretation of efficient BPU. It also leads to new planning algorithms that are more “decisive” than their original counterparts.

The perspectives of our work are twofold. First, our approach could be naturally extended to solve possibilistic Markov Decision Processes. This extension seems theoretically straightforward, since a finite-horizon MDP can be translated into a set of decision trees (one for each state). Thus, our theoretical results hold for finite-horizon MDPs as well. However, the direct application of the lexicographic approach to possibilistic MDPs may lead to algorithms which are exponential in time and space (w.r.t. the MDP description), since the decision trees associated to a MDP may be of exponential size, while (possibilistic) MDPs can be solved in polynomial time [22, 21]. Determining whether computing lexicographic optimal solutions to possibilistic MDPs is tractable is a perspective of this work.

The second perspective of this work, not unrelated, is to develop simulation-based algorithms for finding lexicographic solutions to MDPs. Reinforcement Learning algorithms [26] allow to solve large size MDPs by making use of simulated trajectories of states to optimize a strategy. It is not immediate to develop RL algorithms for possibilistic MDPs, since no unique stochastic transition function corresponds to a possibility distribution. However, uniform simulation of trajectories (with random choice of actions) may be used to generate an approximation of the possibilistic decision tree (provided that both transition possibilities and utility of the leaf are given with the simulated trajectory). So, interleaving simulations and lexicographic dynamic programming may lead to RL-type algorithms for approximating lexicographic-optimal policies for (large) possibilistic MDPs.

REFERENCES

- [1] Kim Bauters, Weiru Liu, and Lluís Godo, 'Anytime algorithms for solving possibilistic MDPs and hybrid MDPs', in *9th International Symposium on Foundations of Information and Knowledge Systems (FoIKS'16)*, eds., Marc Gyssens and Guillermo Simari, Lecture Notes in Artificial Intelligence, pp. 1–18. Springer International Publishing Switzerland, (2016).
- [2] Richard Bellman, *Dynamic Programming*, Princeton University Press, 1957.
- [3] Nahla Ben Amor, Hélène Fargier, and Wided Guezguez, 'Possibilistic sequential decision making', *International Journal of Approximate Reasoning*, **55**, 1269–1300, (2014).
- [4] Salem Benferhat, Didier Dubois, and Henri Prade, 'Possibilistic and standard probabilistic semantics of conditional knowledge bases', *Journal of Logic and Computation*, **9**, 873–895, (1999).
- [5] Blai Bonet and Hector Geffner, 'Arguing for decisions: A qualitative model of decision making', in *12th Conference on Uncertainty in Artificial Intelligence (UAI-96)*, August 1-4, Portland, Oregon, USA, pp. 98–105, (1996).
- [6] Anthony R. Cassandra, Leslie Pack Kaelbling, and Michael L. Littman, 'Acting optimally in partially observable stochastic domains', in *12th National Conference on Artificial Intelligence (AAAI'13)*, July 31 - August 4 Seattle, WA, USA, pp. 1023–1028, (1994).
- [7] Francis C. Chu and Joseph Y. Halpern, 'Great expectations. part I: on the customizability of generalized expected utility', in *18th International Joint Conference on Artificial Intelligence (IJCAI-03)*, August 9-15, 2013, Acapulco, Mexico, pp. 291–296, (2003).
- [8] Nicolas Drougard, Florent Teichteil-Konigsbuch, Jean-Loup Farges, and Didier Dubois, 'Qualitative possibilistic mixed-observable MDPs', in *29th Conference on Uncertainty in Artificial Intelligence (UAI'13)*, August 11-15, 2013, Bellevue, WA, USA, pp. 192–201, (2013).
- [9] Nicolas Drougard, Florent Teichteil-Konigsbuch, Jean-Loup Farges, and Didier Dubois, 'Structured possibilistic planning using decision diagrams', in *28th Conference on Artificial Intelligence (AAAI'14)*, July 27 -31, 2014, Québec City, Québec, Canada., pp. 2257–2263, (2014).
- [10] Didier Dubois, Lluís Godo, Henri Prade, and Adriana Zapico, 'Making decision in a qualitative setting: from decision under uncertainty to case-based decision', in *6th International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, June 2-5, Trento, Italy, pp. 594–605, (1998).
- [11] Didier Dubois and Henri Prade, 'Possibility theory as a basis for qualitative decision theory', in *14th international joint conference on Artificial intelligence (IJCAI'95)*, August 20-25, Montreal, Quebec Canada, pp. 1925–1930, (1995).
- [12] Didier Dubois, Henri Prade, and Régis Sabbadin, 'Decision-theoretic foundations of qualitative possibility theory', *European Journal of Operational Research*, **128**, 459–478, (2001).
- [13] Hélène Fargier and Régis Sabbadin, 'Qualitative decision under uncertainty: back to expected utility', *Artificial Intelligence*, **164**, 245–280, (2005).
- [14] Phan Giang and Prakash P Shenoy, 'Two axiomatic approaches to decision making using possibility theory', *European Journal of Operational Research*, **162**, 450–467, (2005).
- [15] Lluís Godo and Adriana Zapico, 'On the possibilistic-based decision model: Characterization of preference relations under partial inconsistency', *Applied Intelligence*, **14**, 319–333, (2001).
- [16] Daniel J. Lehmann, 'Generalized qualitative probability: Savage revisited.', in *21st Conference in Uncertainty in Artificial Intelligence (UAI '05)*, July 26-29, Edinburgh, Scotland, pp. 381–388, (1996).
- [17] Hervi Moulin, *Axioms of Cooperative Decision Making*, Cambridge University Press, 1988.
- [18] John Von Neumann and Oskar Morgenstern, *Theory of games and economic behavior*, 1948.
- [19] Martin L. Puterman, *Markov Decision Processes*, John Wiley and Sons, 1994.
- [20] Howard Raiffa, *Decision Analysis: Introductory Lectures on Choices under Uncertainty*, Addison Wesley, 1968.
- [21] Régis Sabbadin, 'Possibilistic Markov decision processes', *Engineering Applications of Artificial Intelligence*, **14**, 287–300, (2001).
- [22] Régis Sabbadin, Hélène Fargier, and Jérôme Lang, 'Towards qualitative approaches to multi-stage decision making', *International Journal of Approximate Reasoning*, **19**, 441–471, (1998).
- [23] Leonard J. Savage, *The Foundations of Statistics*, Wiley, 1954.
- [24] Paul Snow, 'Diverse confidence levels in a probabilistic semantics for conditional logics', *Artificial Intelligence*, **113**, 269–279, (1999).
- [25] Richard S. Sutton, 'Learning to predict by the methods of temporal differences', in *Machine Learning*, pp. 9–44, (1988).
- [26] Richard S. Sutton and Andrew G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 1998.
- [27] Paul Weng, 'Qualitative decision making under possibilistic uncertainty: Toward more discriminating criteria', in *21st Conference in Uncertainty in Artificial Intelligence (UAI'05)*, July 26-29, Edinburgh, Scotland, pp. 615–622, (2005).
- [28] Paul Weng, 'Axiomatic foundations for a class of generalized expected utility: Algebraic expected utility', in *22nd Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-06)*, July 13-16, Arlington, Virginia, pp. 520–527, (2006).

Even Angels Need the Rules: AI, Roboethics, and the Law

Ugo Pagallo¹

Abstract.¹ Over the past years, scholars have increasingly debated over the reasons why we should, or should not, deploy specimens of AI technology, such as robots, on the battlefields, in the market, or at our homes. Amongst the moral theories that discuss what is right, or what is wrong, about a robot's behaviour, virtue ethics, rather than utilitarianism and deontology, offers a fruitful approach to the debate. The context sensitivity and bottom-up methodology of virtue ethics fits like hand to glove with the unpredictability of robotic behaviour, for it involves a trial-and-error learning of what makes the behaviour of that robot good, or bad. However, even advocates of virtue ethics admit the limits of their approach: All in all, the more societies become complex, the less shared virtues are effective, the more we need rules on rights and duties. By reversing the Kantian idea that a nation of devils can establish a state of good citizens, if they "have understanding," we can say that even a nation of angels would need the law in order to further their coordination and collaboration. Accordingly, the aim of this paper is not only to show that a set of perfect moral agents, namely a bunch of angelic robots, need rules. Also, no single moral theory can instruct us as to how to legally bind our artificial agents through AI research and robotic programming.

1 INTRODUCTION

Over the past years the debate on "roboethics" [1, 2], and the legal aspects of robotics [3, 4], has been particularly popular among scholars. As to the technology under scrutiny, some argue that robots are machines basically built upon today's "sense-think-act" paradigm in AI research [5]. Others, as Sebastian Thrun, reckon that robots have to do with the ability of a machine to "perceive something complex and make appropriate decisions" out there [in 6, at 77]. While some others stress that robots should be able to learn and adapt to the changes of the environment, it is important to stress that robots are not a mere "out of the box" machine. As a sort of prolonged epigenetic developmental process, robots progressively gain knowledge or skills from their own interaction with the living beings inhabiting the surrounding environment, so that more complex cognitive structures emerge in the state-transition system of the artificial agent. In addition, robots can respond to stimuli by changing the values of their properties or inner states and, furthermore, they can improve the rules through which those properties change without external stimuli. As a result, we are progressively dealing with agents, rather than simple tools of human interaction. Specimens of the same model will behave in quite different ways, according to the complexity of the context and how humans train, treat, or manage their robots. Both the behaviour and decisions of these artificial agents can thus be unpredictable and risky, hence giving rise to several normative issues.

As to the ethical and legal sides of robotics, there is an ever lasting discussion about their connection. At times, moral theories and the law simply cover different domains, or types of problem. Legal cases of faultless liability in extra-contractual obligations illustrate this point vis-à-vis the claim of virtue-ethicists that define notions of obligation, prohibition, or permission, in light of what makes life good, or bad. Most of what is morally crucial for virtue ethics is not relevant from a legal point of view: we return to this relation below in Section 3. However, contrary to current advocates of "exclusive legal positivism," we may admit that, now and then, moral theories guide the law. Consider cases of general disagreement that regard either the meaning of the terms framing the legal question, or the ways such terms are related to each other in legal reasoning, or the role of the principles that are at stake in the case. As suggested by Ronald Dworkin and his followers, an option for tackling such hard cases is given by the "uniquely right answer"-thesis. According to this stance, a morally coherent narrative should grasp the law in such a way that, given the nature of the legal question and the story and background of the issue, scholars can attain the answer that best justifies or achieves the integrity of the law [7]. By identifying the principles of the system that fit with the established law, jurists could apply such principles in a way that presents the case in the best possible light.

Alternatively, some other scholars represent the hard cases of the law as a class of cases that confront us with something new and moreover, that require a reasonable compromise between many conflicting interests. Although this is of course the stance Herbert Hart made popular with his work [8], it does not follow that we have to buy any of his theoretical assumptions on, for example, the rule of recognition and the minimum content of natural law, to concede that a reasonable compromise has at times to be found in the legal domain. As previous international agreements have regulated technological advancements over the past decades in such fields as chemical, biological and nuclear weapons, or the field of computer crimes since the early 2000s, many claim that a new agreement on some of today's fields of robotics, such as robot soldiers, is necessary [9]. Regardless of the solution to the meta-disagreement on the hard cases of the law, we thus have cases in which the law needs the contribution of moral theories and a set of moral values, in order to define obligations, prohibitions, and permissions, via national statutes and international agreements, such as the Budapest Convention on computer crimes.

The stance of this paper on robots, ethics, and the law aims to explore a further kind of interaction between law and ethics. The attention is drawn here to cases in which moral theories need the contribution of the law. We may assume the ideal scenario of scholars that agree on what is right and what is wrong, what is good and what is bad, about a robot's behaviour, from an ethical point of view, and still two sets of legal issues are fated to remain

¹ Law School, University of Turin (Italy), email: ugo.pagallo@unito.it

open. By reversing the Kantian idea that a nation of devils can establish a state of good citizens, if they “have understanding” [10, at 366], we can say that even a nation of angels need some rules to further their coordination and collaboration. These rules may be interpreted either as moral norms [11], or in the sense of Hart’s “secondary” legal rules, i.e. rules that allow the creation, modification, and suppression of the “primary rules” that govern people’s conduct [8]. The thesis of this paper is not only that a set of perfect moral agents, namely a bunch of angelic robots, would need secondary legal rules to be good “citizens.” In addition, no single moral theory can instruct us as to how we should legally bind our robots. In order to argue these theses, the paper is divided into four sections.

Next, in Section 2, attention is drawn to the debate on roboethics so as to appreciate the complexity of today’s state-of-the-art. More particularly, in Section 3, the focus is on virtue ethics and how the context-sensitivity of this approach, together with its bottom-up methodology, fit like hand to glove with a pragmatic, legal approach to robotics. Section 4 illustrates two reasons why moral theories and current debate on roboethics need the support of the law. Section 4.1 scrutinizes a new generation of robotic crimes that will affect a basic tenet of the rule of law and of its continental European counterpart, the principle of legality, i.e. “no crime, nor punishment without a criminal law.” Section 4.2 dwells on the creation of special, i.e. legally deregulated zones, that should allow us to test unpredictable and risky robots in open environments. The conclusions of the paper insist on how the law may help us better understand risks and threats brought on by possible losses of control of AI systems, and keep them in check. If we are fated to face some of the criminal actions sketched below in the following sections, such as e.g. the “perpetration-by-another” liability model reversed, let us address these scenarios, first, in a living lab.

2 ROBOETHICS TODAY

Scholars have increasingly discussed over the reasons why we should, or should not, deploy robots on the battlefields, in the market, or at our homes. Consider current debate on whether lethal force can be fully automated, or whether the intent to create robots that people bond with is ethically justifiable. In business law, robotic applications trading in auction markets have brought on new moral and legal dilemmas. The random-bidding strategy of these apps clarifies, or even has provoked, real life bubbles and crisis, e.g. the financial troubles of late 2009 that may have been triggered by the involvement of such artificial agents. In this context, suffice it to sum up the debate on “roboethics,” or “moral machines” [12], in accordance with a twofold stance.

On the one hand, as to the strict ethical side of current discussions in the field, we should distinguish meta-ethics, applied ethics, and moral theories, such as deontology, utilitarianism, or virtue ethics. In the field of meta-ethics, the intent is to clarify the basic concepts of the subject-matter, such as notions of right and wrong. In the field of applied ethics, scholars deal with a set of moral dilemmas arising from a specific domain, e.g. robotics. In the field of moral theories, what is at stake concerns the different ways in which we can grasp and define notions of obligation, prohibition, permission, and the like. Correspondingly, in the case of moral theories, a utilitarian would judge the action or behaviour of robots in light of their outcomes; a deontologist in connection

with the intent behind such an action; a virtue-ethicist in light of what makes life good, or bad.

On the other hand, as to the technical side of the debate, there are multiple ways in which we can program our robots. This differentiation, of course, depends on the kind of moral theory we follow. However, once we agree on the content of an ethical code under a given moral theory, we can set up our robots either using deontic logic, or endorsing “principlism” and a theory of *prima facie* duties, or the “divine-command logic,” and so forth [13]. In the case of deontic logic, the aim is to directly formalize and implement an ethical code in terms of what is obligatory, permissible, or forbidden, through an “AI-friendly”-semantics [14], and a corresponding axiomatization [15]. From the point of view of principlism, the attention is drawn to such notions as autonomy, beneficence, and the aim at doing no harm, in order to infer sets of consistent ethical rules through computational inductive logic [16]. In the case of divine-command logic, the goal is the ethical control of robotic behaviour, drawing on both the “logic of requirement” [17, 18], and modal logic [19].

In light of this panoply of approaches, both ethical and technical, we should not miss a crucial point. Regardless of today’s discussions in legal theory, e.g. exclusive vs. inclusive legal positivism, it seems fair to affirm that moral theories often fall short in coping with the complexity of the legal phenomenon. Consider consequentialism, or a utilitarian stance, according to which actions, or behaviours, are judged in light of their outcomes. There are many cases in which, vice versa, “intentions” play a crucial role in the law: think of the intentional misuse of power and the reasons why a certain person committed a criminal offense, so as to evaluate the *actus reus*; the right intention of the proper authority entering into war; the intentions of the parties to a contract, or the wrongful intention that severs the link between claims of extra-contractual liability, i.e. the case of intentional torts as opposed to negligence-based responsibility and strict liability. Although we may aim to design a perfect consequentialist robot, this utilitarian approach would not prevent cases of liability for the behaviour of others in both criminal and civil law, that depend on the “intentions” of the robot.

Against this legal backdrop, some reckon that certain robots can grasp the legal terms of their behaviour and, moreover, humans could blame such machines when they do not keep their own word or when they commit some kind of offense [20, 21, 22]. Others affirm that we should be allowed to expect that a robot really means what it declares when making a contractual offer [23]. In any event, by examining, pace advocates of consequentialism, the intentions of robots, this level of abstraction deepens our understanding of, say, the good faith of humans, rather than the robots’ ability to really understand what they are doing. Leaving aside the field of criminal law, to which we return below in Section 4.1, contemplate today’s “contract problem” in robotics [2, 21]. Here, individuals should be held responsible for the erratic behaviour of robots, by referring the intentions of such machines to existing conventions of business and civil law, e.g. the “objective intention” of a contract. In this latter case, humans should not be able to avoid the usual consequence of robots making a decisive mistake, i.e. the annulment of a contract, when the counterparty had to have been aware of a mistake that due to the erratic behaviour of the robot, clearly concerned key elements of the agreement, such as the market price of the item or the substance of the subject-matter of that contract. Kant would agree on that.

But, reflect now on how deontology in moral theory should address cases in which the law imposes liability regardless of the person's intentions. In addition to individuals' responsibility for the behaviour of their animals and, in most legal systems, their children, this type of faultless liability applies to most producers and users of robots. By following, e.g., Kant's theory of ethics, the aim of design should be the program of a perfect deontologist robot, so that its intentions, i.e. such cognitive states as beliefs, desires, or hopes of the artificial agent, can always be deemed as appropriate. Still, this sort of Kantian robot would not prevent the liability of its "human master," i.e. the latter's strict responsibility in cases where scholars more frequently liken robots to animals [24, 25, 26], rather than products and things. The economic rationale for this legal regime is that strict liability rules represent the best method of accident control by scaling back dangerous activities [27]. From this latter point of view, a Kantian robot, designed in accordance with the tenets of deontology, would not be a good legal agent at all. Some times, the law does not pay any attention to intentions.

Yet, after the "consequentialist robot" and the "deontologist robot," there is a further way to conceive and design our artificial agents, i.e. according to the tenets of virtue ethics. The context sensitivity and bottom-up approach of the latter seems particularly appropriate to tackle the unpredictability and risks of robotic behaviour. As Keith Abney affirms in *Robotics, Ethical Theory, and Metaethics*, virtue ethics, rather than consequentialism, or deontology, appears as "a more helpful approach for robots" [28]. We will explore how far this idea goes in the next section.

3 VIRTUE ROBOTS

An increasing amount of research has been devoted over the past years to the analysis of strong AI systems, trust, and security. Consider current work on the verifiability of systems that change or improve themselves, or on utility functions or decisions processes that aim to avoid that an AI system could try not to be shut down or repurposed. Likewise, reflect on further theoretical frameworks to better appreciate the space of potential systems that avoid undesirable behaviours. At the University of Stanford, an area of study has to do with "loss of control of AI systems." In the words of Eric Horvitz, "we could one day lose control of AI systems via the rise of superintelligences that do not act in accordance with human wishes [so] that such powerful systems would threaten humanity" [29]. Similar risks have been stressed by Bill Gates, Elon Musk, and Stephen Hawking. How should we address these challenges?

As mentioned above in the previous section, some reckon that virtue ethics, rather than utilitarianism, or deontology, may help us tackling the unpredictability and risky behaviour of robots. As Abney argues, there are two reasons why this can be the case. First, this approach does not hinge on any rule-based morality but rather, draws the attention to the context sensitivity of the issues we are dealing with, namely, the disposition to act in a certain way under certain circumstances. In fact, a "proper functioning approach to evaluation appears natural: is the surgical robot operating properly in carving one's chest, or is my new robotic bandsaw dysfunctionally attempting to do the same thing?" [28]. Second, contrary to the top-down approaches of both deontology and utilitarianism, the approach of virtue ethics is bottom-up and involves a trial-and-error learning of what makes the behaviour of

a robot good, or bad. The "hybrid approach" of virtue ethics seems then particularly fruitful to tackle some of the problems with robotic behaviour, such as matters of foreseeability and due care that may trigger new cases of human negligence. The pragmatic and context sensitivity approach of virtue ethics help us indeed to determine how we should address the moral dilemmas of robotics, how we should program these machines, and test them.

However, even Abney admits the limits of this search for the virtues that properly functioning robots, given their appropriate roles, would evince. Simply put, the more societies become complex, the less shared virtues are effective, the more we need rules on rights and duties. In his words, "as the group of those dealing with robots becomes larger and more variegated, social sanctions and shared values gradually become less effective at minimizing them" [28]. Going back to the Kantian idea that even a nation of devils can establish a state of good citizens [10], we should thus admit, on the one hand, that even "virtue robots" demand rules. Yet, on the other hand, this requirement entails a twofold set of further issues. The first problem concerns the different moral rules and multiple ways in which we can embed such rules into robots. Going back to the state-of-the-art illustrated above in the previous section, should we program our robots, following a theory of prima facie duties, or the divine-command logic? Using deontic logic, or endorsing "principlism"? Should we privilege the outcomes of robotic behaviour, or judge them vis-à-vis the intent behind such actions? A mix of them?

The second problem revolves around the nature of the rules that should govern our robots. Here, we can even assume the ideal scenario of scholars that agree on the level of abstraction on what is right and what is wrong, what is good and what is bad, about a robot's behaviour. Yet, even in the case of a common ethical code under a given moral theory, a number of legal issues are fated to remain open. Whereas a set of perfect moral agents, namely a bunch of angelic robots, would still need rules to further their cooperation and collaboration [11], some of these rules are legal, rather than moral. Such rules can be grasped both in the sense of Hart's "secondary rules" that allow the creation, modification, and suppression of the primary legal rules on people's conduct [8], and as procedural rules, or of organization. The set of rules on how to produce enforceable norms at both national and international levels, along with administrative regulation at regional levels, are examples of this class of secondary rules of the law.

However, "virtue robots" also need "primary rules" that govern human and robotic behaviour in legal, rather than moral, terms. Consider cases of individual responsibility that are under a strain, such as immunity for humans bearing responsibility for the care of robots and their behaviour in the field of criminal law, or unjust damages concerning robots as a source of responsibility for other agents in the system [30]. These scenarios appear "hard," for they may spark general disagreement that does not only regard different values and principles of the normative context under examination, on which social acceptability and cohesion ultimately depend. Moreover, these cases require legal expertise to determine whether or not a loophole exists in the field, e.g. in criminal law, and hence, whether or not new primary rules should be added to the legal system.

In addition, the unpredictability of the actions or behaviour of robots, triggers an indefinite kind of cases in which we do not know where we may eventually end up. After all, the UK recorded 77 robot-provoked accidents in 2005 alone in which "people have been crushed, hit on the head, welded and even had molten

aluminium poured over them by robots” [31]. Likewise, current state of the art in technology suggests that the use of, say, unmanned aerial systems (UAS) should still be conceived as an “ultra-hazardous activity,” as much as traditional aviation was considered in the 1930s [30]. Leaving aside further robotic applications, research and the breath-taking progress in AI and robotics then recommend that new levels of risk and unpredictability, e.g. cases of loss of control of AI agents, have to be taken seriously. How should we legally react before such risks and threats?

4 LAW’S EMPIRE

The Dworkinian title of this section intends to stress two different kinds of legal problem in robotics. They have to do with Hart’s “primary legal rules” and their connection with the moral ones through the “secondary rules” of the law. Both problems require a particular expertise for they regard either the identification of a “loophole” in the legal system, or its inner “deadlock.” At times, the behaviour of robots may of course trigger legal hard cases that bring us back to the current meta-disagreement on the hard cases of the law. We mentioned this aspect of the debate above in the introduction, e.g. the “uniquely right answer” [32] vs the “reasonable compromise”-thesis [8], so as to determine for example whether and to what extent lethal force should ever be permitted to be fully automated [9].

Here, the problem is different. It revolves around the ideal scenario of scholars that agree on what is right, or wrong, about robotic behaviour and yet, even in this case, a further set of issues is fated to remain open. These issues concern both the primary and secondary legal rules of the system that should govern the behaviour of robots, and regard either a basic tenet of the rule of law, i.e. the principle of legality, or the unpredictability of robotic behaviour. We will analyse the loopholes of the law in Section 4.1, and its deadlocks in Section 4.2. Then, the time will be ripe for the conclusions of this paper: you do not have to follow the ideas of current “exclusive legal positivism,” i.e. the self-referential completeness of the law and its sources, to admit that the law has some problems of its own, also in the field of robotics.

4.1 Loopholes

The first legal problem of robotics is related to a basic tenet of the rule of law, that is summarized, in continental Europe, with the formula of the principle of legality: “no crime, nor punishment without a criminal law.” Whereas certain behaviours might be deemed as morally bad, or wrong, individuals can be held criminally liable for that behaviour only on the basis of an explicit criminal norm. Contrary to the field of civil (as opposed to criminal) law, in which analogy often plays a crucial role so as to determine individual liability, it is likely that robots will produce a novel generation of loopholes in the criminal law field, forcing lawmakers to intervene at both national and international levels. Robot soldiers are a good example of this first kind of problem, e.g. the aforementioned question on whether lethal force should ever be permitted to be fully automated. But, consider new forms of corporate criminal liability and distributed responsibility that hinge on multiple accumulated actions of humans and computers [22, 33]. It can be extremely difficult to ascertain what is, or should be, the information content of the corporate entity as foundational

to determining the responsibility of individuals. The intricacy of the interaction between humans and computers may lead to cases of impunity that have recommended some legal systems to adopt forms of criminal accountability of corporations. Think of the collective knowledge doctrine, the culpable corporate culture, or the reactive corporate fault, as ways to determine the blameworthiness of corporations and their autonomous criminal liability. Although several critical differences persist between the common law and the civil law traditions, and among the legal systems of continental Europe, we can leave aside this kind of debate, and focus on whether these forms of corporate criminal liability could be applied to the case of the artificial legal agents and the AI smart machines that are under scrutiny in this paper. Noteworthy, over the past years, several scholars have proposed new types of accountability for the behaviour of robots [23, 30, 34, 35, 36, 37], suggesting a fruitful parallelism with those legal systems that admit the autonomous criminal responsibility of corporations.

A true story helps us illustrate this new scenario: in May 2014, Vital, a robot developed by Aging Analytics UK, was appointed as a board member by the Japanese venture capital firm Deep Knowledge, in order to predict successful investments. As a press released was keen to inform us, Vital was chosen for its ability to pick up on market trends “not immediately obvious to humans,” regarding decisions on therapies for age-related diseases. Drawing on the predictions of the AI machines, such trends of humans delegating crucial cognitive tasks to autonomous artificial agents will reasonably multiply in the foreseeable future. But, how about the wrong evaluation of a robot that leads to a lack of capital increase and hence, to the fraudulent bankruptcy of the corporation?

In this latter case, the alternative seems between “crimes of negligence” and the hypothesis of AI corporate liability. As to the crimes of negligence, liability depends on lack of due care, so that a reasonable person fails to guard others against foreseeable harms. The latter hinges on the traditional “natural-probable-consequence” liability model in criminal law that comprises two different types of responsibility. On the one hand, imagine either programmers, or manufacturers, or users who intend to commit a crime through their robot, but the latter deviates from the plan and commits some other offence. On the other hand, think about humans having no intent to commit a wrong but who were negligent while designing, constructing or using a robot. Although this second type of liability is trickier, most legal systems hold humans responsible even when they did not aim to commit any offense. In the view of traditional legal theory, the alleged novelty of all these cases resembles the responsibility of an owner or keeper of an animal “that is either known or presumed to be dangerous to mankind” [26].

Yet, as to the traditional crime of negligence, there is a problem: in the case of the wrong evaluation of the robot that eventually leads to the fraudulent bankruptcy of the corporation, humans could be held responsible only for the crime of bankruptcy triggered by the robot’s evaluation, since the mental element requirement of fraud would be missing in the case of the human members of the board. Therefore, the criminal liability of the corporation and eventually, that of the robot would be the only way to charge someone with the crime of fraudulent bankruptcy. This scenario however means that most legal systems should amend themselves, in order to prosecute either the robot as the criminal agent of the corporation, or the corporation as such.

Further instances of new robotic offenses can be given. After all, we can apply to this context that which James Moor called the “logical malleability” of computers and so, of robots. Since the latter “can be shaped and molded to do any activity that can be characterized in terms of inputs, outputs, and connecting logical operations” [38], the only limits to the new scenarios of robotic crimes are given by human imagination. It is not so hard to envisage a world in which individuals become the innocent agent or instrument of an AI’s bad decision. Certainly, by reversing the usual perspective, the scenario is not entirely new: we have full experience of hackers, viruses or trojan horses, compromising computers connected to the internet, so as to use them to perform malicious tasks under remote direction, e.g. denial-of-service attacks. Yet, what is new in the case of robots concerns their particular role of interface between the online and the offline worlds. In the internet of everything, we may envisage either powerful brain computer interfaces for robots that perceive the physiological and mental states of humans through novel Electroencephalography (EEG) filters, or robots replicating themselves, in order to specialize in infringing practices, so that no human could be held responsible for their autonomous harmful conduct. Legal systems could react either amending once again themselves, e.g. a new kind of autonomous corporate criminal liability for robots, or claiming that the principle of legality does not apply to smart machines after all. In any event, it is likely that a new general type of defence for humans, such as robotic loss of self-control, should be taken into account.

By stressing threats and risks of robotic behaviour, however, we should avert a misunderstanding. We are talking about several applications that, in the words of the UN World Robotics report from 2005, may provide “services useful to the well-being of humans” [39]. Therefore, it seems fair to affirm that the aim of the law to govern the process of technological innovation, should neither hinder it, nor require over-frequent revision to tackle such a progress. The analysis of the loopholes of today’s legal systems in the field of robotics, introduces the examination of its deadlocks. Since robots are here to stay, the aim of the law should be to govern our relationships wisely.

4.2 Deadlocks

The second legal problem of robotics has to do with the unpredictability of the actions or behaviour of robots. From a legal viewpoint, the difficulty of the cases does not only regard how we should represent the web of concepts, ways of interpretation, and principles of the system that are at stake in such cases, through notions of agency, accountability, liability, burdens of proofs, responsibility, clauses of immunity, or unjust damages. Furthermore, legislators can make individuals think twice before using or producing robots, through methods of accident control that either cut back on the scale of the activity via, e.g., strict liability rules, or aim to prevent such activities through the precautionary principle [30]. The recent wave of extremely detailed regulations on the use of drones by the Italian Civil Aviation Authority, i.e. “ENAC,” illustrates this deadlock [40]. How, then, to prevent legislations that may hinder the research in robotics? How to deal with their peculiar unpredictability and risky behaviour? How should we legally regulate the future?

Admittedly, the legal challenges of robotics vary in accordance with the field under examination: international law, criminal law,

civil law, both in contracts and tort law, administrative law, and so forth. Some have proposed that we should register robots just like corporations in business law [34, 35, 36]; while others have recommended that we should bestow robots with capital [37], or that making the financial position of such machines transparent is a priority [23]. In the military sector, scholars and UN special rapporteurs alike have increasingly stressed over the past years, that an international agreement is needed to define the conditions of legitimacy for the employment of robot soldiers. The overall idea is that a detailed set of parameters, clauses and rules of engagement, established by an effective treaty monitoring and verification mechanisms, should allow for a determination of the locus of political and military decisions that, e.g., the increasing complexity of network-centric operations, and the miniaturization of lethal machines, can make very difficult to detect [9].

Still, in many circumstances and with most of the new generation of AI robotic applications, we have a further problem. Current default norms of legal responsibility entail a vicious circle, since the more the strict liability rules are effective, the less we can test our robots. As a result, such primary rules, e.g. the last ENAC regulation from December 2015, can indeed hinder research and development in the field. Correspondingly, we often lack enough data on the probability of events, their consequences and costs, to determine the levels of risk and, thus, the amount of insurance premiums and further mechanisms, on which new forms of accountability for the behaviour of such machines may hinge [30]. This lack of data is crucial, because the unpredictable and risky behaviour of robots affects traditional tenets of the law, such as notions of reasonable foreseeability and due care, on which people’s responsibility may depend. A good example is given by how a new generation of domestic, or service, robots already impact tenets of current legal frameworks in informational privacy and data protection [3, 4, 41, 42]. Therefore, how should legal systems react?

Noteworthy, over the past 13 years, the Japanese government has worked out a way to address these issues through the creation of special zones for robotics empirical testing and development, namely, a form of living lab, or *Tokku*. After the Cabinet Office approved the world’s first special zone in November 2003, covering the prefecture of Fukuoka and the city of Kitakyushu, further special zones have been established in Osaka and Gifu, Kanagawa and Tsukuba. The aim is to set up a sort of interface for robots and society, in which scientists and common people can test whether robots fulfil their task specifications in ways acceptable and comfortable to humans, vis-à-vis the uncertainty of machine safety and legal liabilities that concern, e.g., the protection for the processing of personal data through sensors, GPS, facial recognition apps, Wi-Fi, RFID, NFC, or QC code-based environment interaction [42]. Significantly, this approach to the risks and threats of the human-robot interaction is not only at odds with the typical formalistic and at times, pedantic interpretation of the law in Japan [43]. It is remarkable that such special zones are highly deregulated from a legal point of view. Pace the Italian ENAC, “without deregulation, the current overruled Japanese legal system will be a major obstacle to the realization of its RT [Robot Tokku] business competitiveness as well as the new safety for human-robot co-existence” [43].

So far, the legal issues addressed in the RT special zones regard road traffic laws (Fukuoka 2003), radio law (Kansai 2005), privacy protection (Kyoto 2008), safety governance and tax regulation (Tsukuba 2011), up to road traffic law in highways

(Sagami 2013). These experiments should obviously be extended, so as to further our understanding of how the future of the human-robot interaction could turn out. Some examples were illustrated above in the previous sections, such as matters of foreseeability and due care concerning human negligence, or the unpredictability of robotic behaviour that may trigger novel forms of *actus reus* in criminal law. By testing these scenarios in open, unstructured environments, the Japanese approach does not only show a pragmatic way to tackle the legal challenges of robotics. This sort of interface between strong AI robots and human societies, between present and future, also allows us to better comprehend risks and threats brought on by possible losses of control of AI systems, so as to keep the latter in check.

5 CONCLUSIONS

There are three different ways in which we can grasp the theses and title of this paper, “even angels need the rules.” The first way directly concerns today’s debate in roboethics. From a moral point of view, we can say that even a set of perfect agents, namely a bunch of angelic robots, need rules to further their cooperation and collaboration. Even advocates of virtue ethics concede this point [28]. As illustrated above in Sections 2 and 3, these rules can be interpreted either as moral ones [11], or as secondary legal rules [8].

A second way to interpret the title of the paper involves Hart’s primary rules and more particularly, that which Section 4.1 presented as the loopholes of the law. The legal impact of robotics affects all the sectors in the legal field and still, is especially relevant in criminal law, where analogy should not help tackling the impact of robotics. Although moral theories play a role in this context, so as to either find out the uniquely right answer (Dworkin), or a reasonable compromise (Hart), moral theories do not instruct us as to whether and to what extent we are confronted with a legal loophole and hence, whether or not new legal rules should be added to the system. This is a question that appears crucial for today’s debate on roboethics and still, goes beyond the expertise of robo-ethicists. Is there any loophole in the legal system?

The third way to inflect the title regards the unpredictability and risky behaviour of robots that have been stressed time and again in this paper. Whilst no single moral theory can tackle the complexity of the law and instruct us as to how we should legally bind our robots, the law itself is confronted with that which Section 4.2 summed up as the practical and theoretical “deadlocks” of today’s legal systems. Lawmakers often make individuals think twice before using or producing robots, through methods of accident control that either cut back on the scale of the activity, or aim to stop these activities at all. Here, what the law adds to the current debate in roboethics has to do with the definition of specific secondary legal rules that should allow us to understand what kind of primary legal rules we may need. The creation of legally de-regulated, or special, zones for robotics appears a smart way to overcome such deadlocks and to further theoretical frameworks with which we should better appreciate the space of potential systems that avoid undesirable behaviours. By testing the human-robot interaction outside laboratories, i.e. in open or unstructured areas, we can improve our understanding of how these artificial agents may react in various contexts and satisfy human needs. Also, we can rationally manage the legal aspects of this

experimentation, covering many potential issues raised by the next-generation robots and tackling those requirements that often represent a formidable obstacle for this kind of research, such as public authorisations for security reasons, formal consent for the processing and use of personal data, mechanisms of distributing risk through insurance models and authentication systems, and the like. This is the set of secondary legal rules with which to strengthen our comprehension of the type of primary legal rules we need in order to govern our robots. At the end of the day, this sort of legal de-regulation also offers a fruitful way to deepen our understanding of some moral dilemmas in the field.

REFERENCES

- [1] G. Veruggio, Euron Roboethics Roadmap, *Proceedings Euron Roboethics Atelier*, February 27th-March 3rd, Genoa, Italy (2006)
- [2] J. J. Bryson, The Meaning of the EPSRC Principles of Robotics, *The AISB Workshop on Principles of Robotics*, 4 April 2016.
- [3] U. Pagallo, Killers, Fridges, and Slaves: A Legal Journey in Robotics, *AI & Society*, **26(4)**, 347-354, 2011.
- [4] RoboLaw, Guidelines on Regulating Robotics. EU Project on Regulating Emerging Robotic Technologies in Europe: Robotics facing Law and Ethics, 22 September 2014.
- [5] G. A. Bekey, *Autonomous Robots: From Biological Inspiration to Implementation and Control*, The MIT Press, Cambridge, Mass. & London, 2005.
- [6] P. Singer, *Wired for War: The Robotics Revolution and Conflict in the 21st Century*, Penguin, London, 2009.
- [7] R. Dworkin, *A Matter of Principle*, Oxford University Press, Oxford, 1985.
- [8] H. L. A. Hart, *The Concept of Law*, Oxford University Press, Oxford, 1961.
- [9] U. Pagallo, ‘Robots of Just War: A Legal Perspective’, *Philosophy and Technology*, **24(3)**, 307-323, (2011).
- [10] I. Kant, Perpetual Peace, *The Cambridge Edition of the Works of Immanuel Kant: Practical Philosophy*, trans. by M. Gregor, vol. 8, Cambridge University Press, Cambridge, 1999.
- [11] L. Floridi, ‘Distributed Morality in an Information Society’, *Science and Engineering Ethics*, **19(3)**, 727-743, (2013).
- [12] W. Wallach and C. Allen, *Moral Machines: Teaching Robots Right from Wrong*, Oxford University Press, New York, 2009.
- [13] S. Bringsjord and J. Taylor, The Divine-Command Approach to Robot Ethics, in *Robot Ethics: The Ethical and Social Implications of Robotics*, pp. 85-108, edited by P. Lin, K. Abney and G. A. Bekey, The MIT Press, Cambridge, Mass., 2014.
- [14] J. Horta, *Agency and Deontic Logic*, Oxford University Press, New York, 2001.
- [15] Y. Murakami, Utilitarian Deontic Logic, *Proceedings of the Fifth International Conference on Advances in Modal Logic*, pp. 288-302, edited by R. Schmidt et al. AiML, Manchester UK, 2004.
- [16] M. Anderson and S. Leigh Anderson, Ethical Healthcare Agents, *Advanced Computational Intelligence Paradigms in Healthcare*, pp. 233-257, edited by M. Sordo et al. Springer, Berlin, 2008.
- [17] R. Chisholm, Practical Reason and the Logic of Requirement, *Practical Reason*, 1-17, edited by S. Koerner, Basil Blackwell, Oxford, 1974.
- [18] Ph. Quinn, *Divine Commands and Moral Requirements*, Oxford University Press, Oxford, 1978.
- [19] C. I. Lewis and C. H. Langford, *Symbolic Logic*, Dover, New York, 1959.
- [20] S. J. Hall, *Beyond AI: Creating the Conscience of the Machine*, Prometheus, New York, 2007.
- [21] S. Chopra and L. F. White, *A Legal Theory for Autonomous Artificial Agents*, The University of Michigan Press, Ann Arbor, 2011.

- [22] G. Hallevy, *Liability for Crimes Involving Artificial Intelligence Systems*, Springer, Dordrecht, 2015.
- [23] G. Sartor, 'Cognitive Automata and the Law: Electronic Contracting and the Intentionality of Software Agents', *Artificial Intelligence and Law*, **17(4)**, 253-290, (2009).
- [24] B. Latour, *Reassembling the Social: an Introduction to Actor-Network-Theory*, Oxford University Press, Oxford, 2005.
- [25] D. McFarland, *Guilty Robots, Happy Dogs: The Question of Alien Minds*, Oxford University Press, New York, 2008.
- [26] J. Davis, 'The (common) Laws of Man over (civilian) Vehicles Unmanned', *Journal of Law, Information and Science*, **21(2)**, 10.5778/JLIS.2011.21.Davis.1., (2011).
- [27] R. Posner, *Economic Analysis of Law*. Little Brown, Boston, 1973.
- [28] K. Abney, Robotics, Ethical Theory, and Metaethics: A Guide for the Perplexed, *Robot Ethics: The Ethical and Social Implications of Robotics*, 35-52, edited by P. Lin, K. Abney and G. A. Bekey, The MIT Press, Cambridge, Mass., 2014.
- [29] E. Horvitz, One-Hundred Year Study of Artificial Intelligence: Reactions and Framing. White Paper. Stanford University, at <https://stanford.app.box.com/s/266hrhww2l3gjoy9eur>, (2014).
- [30] U. Pagallo, *The Laws of Robots: Crimes, Contracts, and Torts*, Springer, Dordrecht, 2013.
- [31] R. Noack, A Robot Killed a Factory Worker in Germany. So Who Should Go on Trial?, *The Washington Post*, 2 July 2015.
- [32] R. Dworkin, *Law's Empire*, Harvard University Press, Cambridge, Mass., 1986.
- [33] P. M. Freitas, F. Andrade and P. Novais, Criminal Liability of Autonomous Agents: From the Unthinkable to the Plausible, *AI Approaches to the Complexity of Legal Systems*, 145-156, edited by Pompeu Casanovas et al., Springer, Dordrecht, 2014.
- [34] C. E. A. Karnow, 'Liability for Distributed Artificial Intelligence', *Berkeley Technology and Law Journal*, **11**, 147-183, (1996).
- [35] J.-F. Lerouge, 'The Use of Electronic Agents Questioned under Contractual Law: Suggested Solutions on a European and American Level', *The John Marshall Journal of Computer and Information Law*, **18**, 403, (2000).
- [36] E. M. Weitzenboeck, 'Electronic Agents and the Formation of Contracts', *International Journal of Law and Information Technology*, **9(3)**, 204-234, (2001).
- [37] A. J. Bellia, 'Contracting with Electronic Agents', *Emory Law Journal*, **50**, 1047-1092, (2001).
- [38] J. Moor, 'What Is Computer Ethics?', *Metaphilosophy*, **16(4)**, 266-275, (1985).
- [39] UN World Robotics, Statistics, Market Analysis, Forecasts, Case Studies and Profitability of Robot Investment, edited by the UN Economic Commission for Europe and co-authored by the International Federation of Robotics, UN Publication, Geneva (Switzerland), 2005.
- [40] ENAC, Remoted Piloted Aerial Vehicles Regulation, issue No. 2 dated 16 July 2015. Revision 1 dated 21 December 2015.
- [41] U. Pagallo, 'Robots in the Cloud with Privacy: A New Threat to Data Protection?', *Computer Law & Security Review*, **29(5)**, 501-508, (2013).
- [42] U. Pagallo, The Impact of Domestic Robots on Privacy and Data Protection, and the Troubles with Legal Regulation by Design, *Data Protection on the Move*, 387-410, edited by S. Gutwirth, R. Leenes, and P. de Hert, Springer, Dordrecht, 2016.
- [43] Y.-S. Weng, Y. Sugahara, K. Hashimoto and A. Takanishi, 'Intersection of "Tokku" Special Zone, Robots, and the Law: A Case Study on Legal Impacts to Humanoid Robots', *International Journal of Social Robotics*, published online on February 13, (2015).

One-Class to Multi-Class Model Update Using the Class-Incremental Optimum-Path Forest Classifier

Mateus Riva¹ and Moacir Ponti¹ and Teofilo de Campos²

Abstract. Incremental learning capabilities of classifiers is a relevant topic, specially when dealing with scenarios such as data stream mining, concept drift and active learning. We investigate the capabilities of an incremental version of the Optimum-Path Forest classifier (OPF-CI) in the context of learning new classes and compare its behavior against Support Vector Machines and k Nearest Neighbours classifiers. The OPF-CI classifier is a parameter-free, graph-based approach to incremental training that runs in linear time with respect to the number of instances. Our results show OPF-CI keeps the running time low while producing an accuracy behavior similar to the other classifiers for increments of instances. Also, it is robust to variations on the order of the learned classes, demonstrating the applicability of the method.

1 INTRODUCTION

In incremental learning we consider a process in which new instances emerge and the model adjusts previous knowledge according to the new data. Thus, it does not assume the availability of a sufficient training set in the beginning of the learning process, but rather that training instances appear over time [8]. The ability to learn not only from a fixed training set, but also considering both new data and an already learned model, is an important feature in pattern recognition systems [24]. Moreover, it is important to update models in an efficient way in order to comply with realistic scenarios [9].

One of the issues in this context are the class-incremental problems, in which instances from new classes appear over time [26] [12]. Scenarios where learning new classes is required are not uncommon. A dataset of produce images, for instance, is constantly being updated as new types of fruits and other produce are uploaded into the system. In order to make a viable recognition system, a model with the ability to learn new classes is required, and, as the dataset increases in size, this model must be updated as efficiently as possible [2]. It is also important to keep it robust to order effects [6].

Graph-based approaches represent the feature space structure by capturing relationships among instances (vertices in a graph) and the weights of said relationships. In this manner, algorithms can perform supervised [14], unsupervised [21, 22] and semi-supervised learning [11]. In data stream mining or active learning it is possible to update minimum spanning trees [4] and paths [1] in order to maintain the graph structure and thus the learning model [10].

The Optimum-Path Forest (OPF) classifier [14] is a supervised learning approach that uses graph theory in order to build a model based on Optimum-Path Trees that encode the information of classes

in the feature space. It can be used to develop simple, fast, multiclass and parameter independent classifiers, and shown to be reliable in applications such as intrusion detection, image classification [16] and as a base classifier in ensemble learning for different datasets [18]. It has similarities with the k nearest neighbors (k NN) classifier, specially with the 1NN, but it is often capable of capturing the intrinsic characteristics of the feature space by creating more or less dense trees in different subspaces [23]. Because of its quadratic training complexity (i.e. $\mathcal{O}(n^2)$ for n instances in the training set), strategies were developed to speed-up training [13]. Additionally, a linear incremental-learning version of the OPF was proposed in [17], capable of including p new instances with a $\mathcal{O}(n \cdot p)$ cost, where n is the sample size in the current model.

As incremental learning is still relevant to the analysis of evolving datasets for the purposes of classification and regression [25], novelty detection and concept-drift [7], we report a new algorithm and a series of experiments that tries to incrementally build a model containing from one to multi-class data. In particular, our paper explores the incremental optimum path forest classifier (OPFI) [17], extending it to class-incremental scenarios (OPF-CI), i.e. capable of learning new classes without retraining the whole model. The novelty of the paper relies on building up an OPF model from one-class to multi-class. Besides, we compare our algorithm with other classifiers under different incremental learning conditions, in particular class-incremental, using different class orders. Our method starts by building an initial one-class model upon which the incremental algorithm can run [21], and then include new classes as new trees on the optimum-path forest.

In order to analyse the OPF-CI behavior we compare its results with the classifiers k NN and Support-Vector Machine (SVM) [5], which also has an incremental version [3]. While SVM has a stronger theoretical framework, it is expensive to train and needs parameter tuning. k NN does not need training but it is slow — specially in its original version — for large datasets. OPF is a parameter-free algorithm that works well even with small training sets, which suits several applications, making it relevant to be compared with SVM and k NN. Some characteristic of each method and their complexities are compared in Table 1, in which n is the total of instances available to train a model in a given time.

In this paper, we set out to develop a fast and accurate method based on the Incremental OPF classifier to learn new classes with supervision, from a model with just one class (based on an initial clustering) to a multi-class model, which is our main contribution. This method must be capable of updating itself with the new classes in linear time while maintaining the accuracy of the original OPF, allowing this method to be used in data stream mining, active learning and other related scenarios.

¹ Instituto de Ciências Matemáticas e de Computação, University of São Paulo, Brazil, email: mateusriva@usp.br, ponti@usp.br

² CVSSP, University of Surrey, UK, email: t.decampos@st-annes.oxon.org

Table 1. Comparison of the capabilities and running time to train with n instances of classifiers: k -Nearest Neighbours (k NN), Support Vector Machine (SVM), Incremental SVM (SVMI), Optimum-Path Forest (OPF), Incremental OPF (OPFI) and our Class-Incremental OPF (OPF-CI).

	k NN	SVM	SVMI [3]	OPF [13]	OPFI [17]	OPF-CI
One-class model	✗	✓	✗	✗	✗	✓
Parameter free	✗	✗	✗	✓	✓	✓
Class-incremental	✓	✗	✗	✗	✗	✓
Complexity: training new model	$\Theta(kn)$	$\Theta(n^2)$	$\Omega(1), \mathcal{O}(n^2)$	$\Theta(n^2)$	$\Omega(1), \mathcal{O}(n)$	$\Omega(1), \mathcal{O}(n)$

2 CLASS-INCREMENTAL OPTIMUM PATH FOREST CLASSIFIER

The Optimum-Path Forest (OPF) classifier [14] interprets each instance as a graph node. The edges connecting the instances are defined by an adjacency relation and weighted by a distance function. In this model it is expected that training instances from a given class will be connected by a path of nearby instances. Therefore the model that is learned by the algorithm is composed of several trees, each tree containing instances from a given class, although each class can have several trees in the model. The root of each tree is called a prototype. Each individual sample is connected to the closest prototype of its class.

The model is built by obtaining the minimum spanning tree of the fully-connected graph, then computing the prototypes, which will be the roots of each optimum-path tree. In the basic algorithm, every node that shares an edge with a node of a different class is a prototype. After prototypes are computed, each non-prototype node is then conquered by the prototype which presents the shortest route between itself and the node. This supervised training has a complexity of $\mathcal{O}(n^2)$.

Classification of a new instance on a built OPF model is a matter of deciding to which tree said instance would belong (that is, which prototype provides the shortest route between itself and the new instance), which is not exactly the same as the nearest neighbor. Its predicted class would then be the same as the class of the prototype of this tree. Figure 1 depicts a toy example of OPF training and classification.

The incremental OPF (OPFI) [17] adds an instance to a pre-existing OPF model in linear time, that is, $\mathcal{O}(n + 1)$ in which n is the number of instances already in the model, while maintaining the capabilities and properties of the original OPF, which runs in $\mathcal{O}((n + 1)^2)$. It considers each new instance inserted in the model as one of three possible cases:

- a **new node of an existing tree**, which is simply inserted into the tree while maintaining the properties of the minimum spanning tree, as described by Chin and Houck [4];
- a **replacement prototype** for a tree, which prompts an update of the distance to the root of every node belonging to the tree;
- a **new tree**, which will become a lone prototype, while also prompting a schism in the tree to which it is connected, since its closest neighbour will now also be a prototype.

Because of the property described in this last case, the Incremental OPF classifier is implicitly capable of learning new classes without

needing to rebuild the entire model. However, the OPFI needs an initial forest to extend. In this paper we create an one-class model, therefore extending the OPFI to handle new classes.

In order to build the initial one-class model, an unsupervised learning method using the Optimum-Path Forest is employed [21]; this method works by choosing prototypes as points with maximum local density, then conquering neighbouring nodes based on the relevant maxima. Because this method finds clusters that are optimum-path trees in the feature space, we use those as our initial model. This method is relatively slow, but it can easily and consistently build an 1-class model with well-positioned prototypes, which can help assure good accuracy results for subsequent increments.

Our method is described in Algorithm 1, with focus on lines 1–2, 5–6 and 13–15 of this algorithm. Further description of the functions `recheckPrototype(.)` and `insertIntoMST(.)`, related to lines 7–11 can be found at [17]. Line 2 is responsible for creating an initial one-class model by clustering the instances into optimum-path trees all belonging to a single class. If such model already exists, when inserting an instance $z \in Z$ belonging to an unseen class, we classify it using the current model (line 5), and it will be assigned to a tree that does not match the new class, producing a false output on line 6. Therefore, z becomes a new prototype, as also the vertex of the existing tree that has conquered z , i.e. its predecessor. Because there can be only one prototype per tree, if the predecessor of z was not a prototype, it became one and a process of reconquest will produce two trees, which generates a new decision boundary for future data of the new class. This process is depicted in Figure 2.

Algorithm 1 OPF Class-Incremental training

Require: an OPF model T ; input nodes $z \in Z$, each with its feature vector.

```

1: if  $T = \emptyset$  then
2:    $T \leftarrow \text{OPFClustering}(Z)$ 
3: else
4:   for each  $z \in Z$  do
5:      $\text{OPFClassify}(z, T)$ 
6:     if  $z.\text{label} = z.\text{truelabel}$  then
7:       if  $z.\text{predecessor}$  is a prototype then
8:          $\text{recheckPrototype}(z, z.\text{predecessor}, T)$ 
9:       else
10:         $\text{insertIntoMST}(z, z.\text{predecessor}, T)$ 
11:      end if
12:     else
13:       set  $z$  as prototype
14:       if  $z.\text{predecessor}$  is not a prototype then
15:         set  $z.\text{predecessor}$  as prototype
16:          $\text{reconquest}(z.\text{predecessor}, T)$ 
17:       end if
18:     end if
19:   end for
20: end if
21: return  $T$ 

```

3 EXPERIMENTS

3.1 DATASETS AND REPRODUCIBILITY

The code used to generate the data, the results and those datasets that are not publicly available can be found in the paper webpage³. A total of 6 datasets were used: 2 synthetic and 4 real. They are as follows:

³ <http://www.icmc.usp.br/~moacir/paper/16ecai.html>

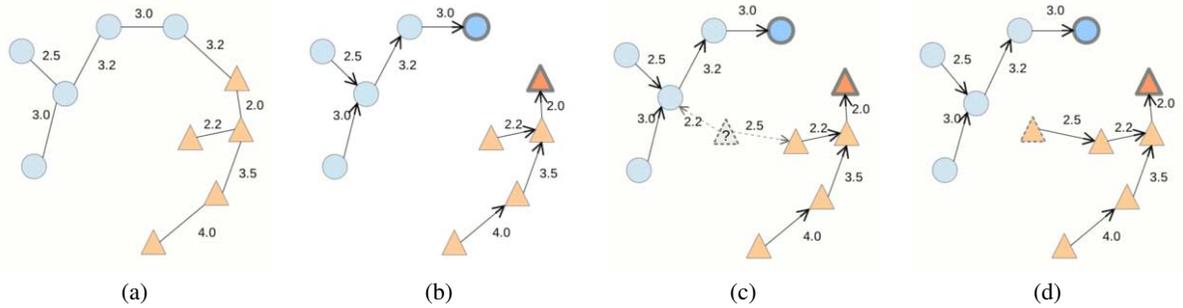


Figure 1. OPF training starts by creating a minimum spanning tree in the feature space (a), and then computing prototypes that will become the root of each optimum-path tree (b). The classification of a new instance is performed by looking at the nearest vertex of each tree (c); it is conquered by the vertex that offers the minimum cost to the root, and assigned to the tree class (d).

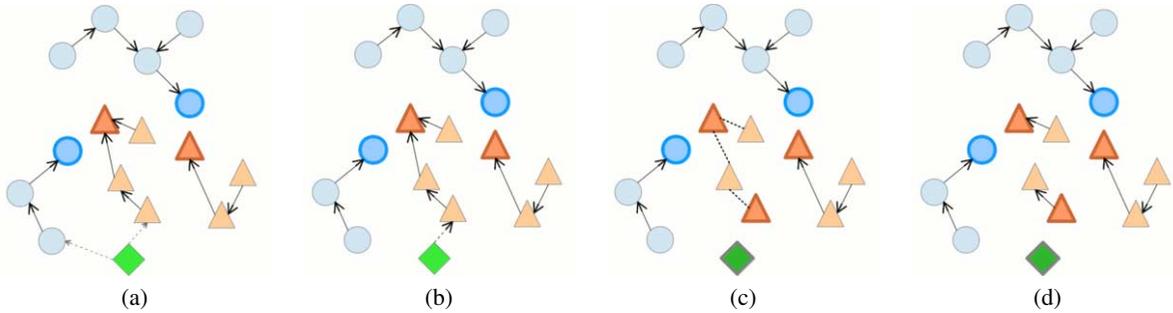


Figure 2. OPF-CI example in class-incremental inclusion: a new pattern, green diamond, can be conquered by one of two classes already in the model, blue circles or orange triangles (a). Here it is conquered by the orange triangle tree (b). Because the instance label does not match the label of the tree, both the new instance and the adjacent vertex become prototypes, which in this case disfigures the previous tree since only one prototype (root) per tree is allowed (c). Then, the reconquest process is performed by the two prototypes, resulting in the new model (d).

1. **Cone-Torus**: a synthetic dataset with three overlapping 2-dimensional classes, one with 92 instances, one with 99 instances and the last one with 209 instances;
2. **L3**: a synthetic dataset with three 2-dimensional overlapping classes, with 500, 250 and 250 instances respectively;
3. **NTL (non-technical losses)**: industrial energy consumption profile dataset, whose attributes describe energy consumption, with 4952 instances, 8 attributes and 2 classes (fraud / non-fraud);
4. **SpamBase**: spam and non-spam emails dataset⁴, with 4601 instances, 56 attributes and 2 classes (spam / non-spam);
5. **MPEG7-BAS**: consists of 70 classes of shapes each having 20 instances⁵. The Beam Angle Statistics (BAS) method was used to extract 180 features.
6. **Produce-BIC-MSB**: consists of 14 classes of produce images [20] from which a Border-Interior Classification (BIC) descriptor was extracted using the MSB (Most Significant Bit) quantization method, with a total of 1400 instances and 64 attributes [15].

3.2 EXPERIMENTAL SETUP

Two main experiments were conducted, both starting with a model trained with instances from a single class. New classes appear in future iterations as follows: **Experiment A**, at each increment a balanced distribution of instances from all classes is added to the model; **Experiment B**, at each increment instances from a single class are added to the model, until all samples of said class are added, until all

classes are added. The appearance order of classes was defined by their label identifiers.

Experiments were performed according to the steps described below. All classes in a given dataset were considered “initial classes” for the purposes of building the 1-class model, one by one. Due to the random nature of the dataset splitting, all experiments were conducted in 10-repeated hold-out sampling.

1. **Splitting of the dataset** into two subsets with 50% of the instances each: S for supervised training and T for testing. The subsets are uniformly distributed;
2. **Splitting of the subset S** into two subsets, S_0 and I , where S_0 contains 50% of the samples of the initial class k present in S , and I contains the remainder;
3. **Initial training on S_0** in order to build a 1-class model and testing against T ;
4. **Splitting of the subset I** into increments S_i :
 - For **Experiment A**, I was split into 20 increments, all uniformly distributed, but maintaining the class balance proportions;
 - For **Experiment B**, I was split into $k \times 10$ increments (k being the total number of classes in the dataset), all uniformly distributed and containing a single class; these increments were then ordered in a class-by-class basis;
 - The sole exception is the MPEG7 dataset, which, due to having a high number of small classes, was split into 5 and $k \times 3$ increments for experiments A and B, respectively;
5. **Inclusion of S_i** in the model and testing against T , for all increments S_i .

⁴ available at <http://archive.ics.uci.edu/ml/datasets/Spambase>

⁵ available at <http://www.dabi.temple.edu/~shape/MPEG7/>

3.3 ACCURACY EVALUATION

A balanced accuracy was used to evaluate the classification:

$$Acc = 1 - \frac{\sum_{i=1}^c E(i)}{2c}$$

where c is the number of classes, and $E(i) = e_{i,1} + e_{i,2}$ is the partial error of c , computed by:

$$e_{i,1} = \frac{FP(i)}{N - N(i)} \text{ and } e_{i,2} = \frac{FN(i)}{N(i)}, i = 1, \dots, c$$

where $FN(i)$ (false negatives) is the number of samples belonging to i incorrectly classified as belonging to other classes, and $FP(i)$ (false positives) represents the samples $j \neq i$ that were assigned to i . Also, $N(i)$ is the number of the instances belonging to the class i and N is the total number of instances.

Therefore, Acc is 1.0 for a 100% accuracy, 0.5 when the classifier assigns all instances to a single class, and 0.0 for an inverse classification (in this case reversing the classification will produce a 100% accuracy). The balanced accuracy is suited for both balanced and class imbalance scenarios [19].

3.4 CLASSIFIERS

In order to gauge the efficiency of the Incremental OPF classifier, two other classifiers were used: the k -Nearest Neighbours classifier (only experiments with $k = 3$ are reported) and the Support Vector Machine classifier (SVC), with a Radial-Basis Function kernel. The Incremental SVM classifier was not used since, being incapable of direct class-incremental learning, its capabilities did not fit the scope of these experiments. The Euclidean distance function was used, and all datasets were normalized using z -score before all experiments (including parameter search).

3.4.1 PARAMETER SEARCH FOR SVM

Grid search space : the grid search was performed on the following logarithmic search space for the SVM with a Radial Basis Function (RBF) kernel: $\gamma \in [2^{-5}, 2^{15}]$, with a step of 2^2 ; and $cost \in [2^3, 2^{-15}]$, with a step of 2^{-2} . The best parameters are available on Table 2.

For the regular experiments, the search was performed using the entire dataset, which included data that would not be available until the last iteration. Thus, the SVM classifier had privileged information, and its accuracy results could be considered an upper bound of what SVM could achieve. Although this provides the SVM with knowledge beyond the others classifiers, it is a way to at least use SVM as a direct comparison method. Also, it allows to see how far the OPF and k NN classifiers would compare to an upper-bound (in terms of parameter tuning) SVM.

Table 2. SVM parameters (cost C and RBF kernel parameter γ) utilized and accuracy achieved using these parameters for each dataset.

Dataset name	Cost (C)	γ	Accuracy
L3	0.125	2	78%
Cone-Torus	1024	1	87%
SpamBase	32	0.0078125	92.5%
MPEG7-B	32	0.0078125	85.4%
NTL	32	8	97.6%
Produce	32	0.0078125	96%

Parameter search in case studies : a case study was carried out in a synthetic dataset, in order to understand the algorithm's behaviour in a more controlled experiment. In this case, we followed the same procedure as experiment B, but the SVM classifier was not provided parameters based on the entire training set. Also, the subset I was split into $k \times 5$ increments instead of $k \times 10$, and only the first class was used as an initial class. As such, in the case study, the 1-class SVM classifier was used while only instances of a single class were available; as soon as the first batch of instances of a new class was presented, a grid search was executed again to find the new parameters for SVM.

This scenario required human intervention since, for each batch of instances with new classes, one has to manually analyze the grid search results and pick the best parameters, possibly requiring expansion of the search space. The time taken to compute the grid search was reported when assessing running time spent during the process of acquiring new classes. However, the time taken to manually analyze the grid search results was not considered.

The objective of this case study was to evaluate the impact of the need for parameters of the SVM classifier in order to contrast it against the parameter-free OPF classifier.

4 RESULTS

4.1 SYNTHETIC CASE-STUDY

The case-study was an in-depth analysis of the effects of the SVM parameter search on total runtime and accuracy in a more realistic incremental scenario, where classes are learned over time, performed on the Cone-Torus dataset.

The full dataset has 400 instances: 92 of class 1, 99 of class 2 and 209 of class 3. Half of these instances were separated to be used as the testing set T . The initial model was built upon 23 instances of the initial class 1. In iterations 1 through 5, a total of 23 instances of class 1 were inserted, spread equally over the 5 iterations. Similarly, 50 instances of class 2 were inserted in iterations 6 to 10, and 104 instances of class 3 were inserted in iterations 10 to 15.

Figure 3 shows the balanced accuracy achieved and cumulative time (in logarithmic scale) spent on each iteration of the case study for the Cone-torus dataset.

These results reveal some interesting points. First and foremost, the parameter searches performed for the SVM classifier (which occurs in iterations 6 and 11) take almost three orders of magnitude of time more to finish than any other step taken by any other classifier. This falls in line with what was expected.

The SVM classifier actually performs better than the others when only provided with two classes (iterations 5 through 10), but the parameters it finds for the set with the initial third class instances are shown to be imperfect, as it loses accuracy as more instances are added to it (while the other classifiers keep improving, or at least maintaining, their current accuracy). This means that, in order to improve the accuracy of the SVM to its optimum, yet another parameter search would be necessary.

4.2 EXPERIMENTS A (BALANCED INCREMENTS)

In these experiments the initial model is one-class and in each iteration new instances from all classes are included in the model. We report the balanced accuracy and the running time for each iteration (note that the time here is not cumulative as it was reported in the

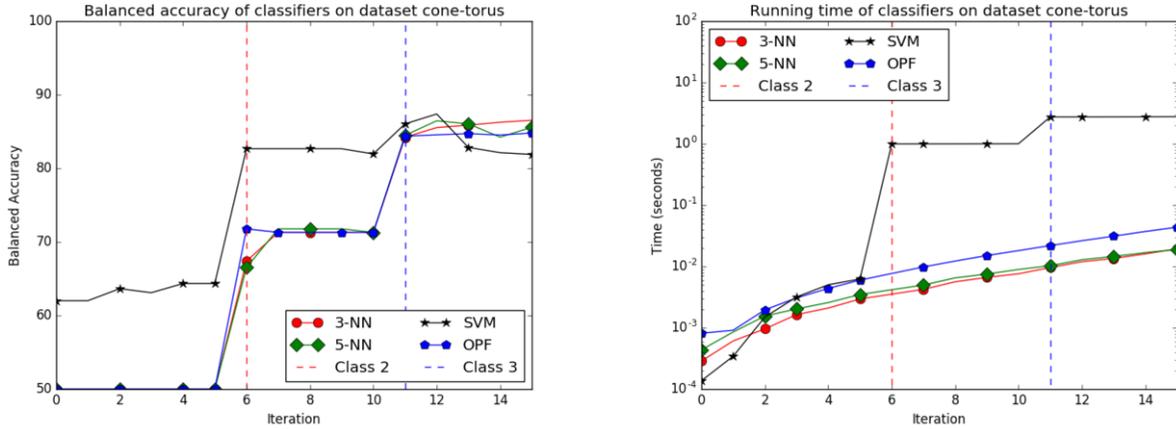


Figure 3. Accuracy and cumulative time results for the case study performed on the Cone-Torus dataset.

case-study). Also, the grid search performed before the first iteration was not included in the SVM running time as it would cause the SVM to dominate the plot and hamper visualization. In Figure 4 we show the results for the datasets L3, NTL and Produce-BIC-MSB, which shows the accuracies on each increment compared with the results obtained by training the original classifiers with the complete training set, referred to as “target” accuracy, which are shown as dotted lines.

In order to see the class-incremental behavior of algorithms when starting with different classes, which could be different for specific distributions or class-imbalanced datasets, we show balanced accuracy results for the datasets Cone-torus, NTL and Produce, respectively in Tables 3, 4 and 5, which includes percentages of the incremental iterations, compared with the target accuracy.

Observe that, with the exception of the NTL dataset, the SVM outperforms the other algorithms in terms of accuracy. This is to be expected, as the SVM classifier is provided with parameters that correspond to the entirety of the dataset, allowing it to better define decision boundaries before the others can. However, with the exception of the L3 dataset, the OPF algorithm performs almost as well as the SVM and better than the 3NN classifier.

In terms of time spent in training and classification, our implementation of OPF spends slightly more time running than our implementation of k NN. Methods to speed up OPF training and/or classification [13] could improve these results. Note however, that using the original OPF classifier would produce a running time of $O(n^2)$ on each iteration, whereas the incremental version used here has $O(n)$ complexity. The time spent running the SVM classifier was highly dependent on the dataset. It is important to remember that the time spent finding the parameters of the SVM was not taken into consideration (see the case-study for details).

One apparent drawback of our method is the complexity of building the initial optimum-path forest using only one class. The chosen algorithm can slow down the first step in some cases, as observed in the iteration zero of some datasets. However, this falls outside of the scope of OPF-CI, which aims to increment pre-existing models in linear time.

The results of these experiments reflect those found in [17]. However, it allow us to see how the incremental OPF algorithm behaves when compared with other classifiers, as well as its capability of acquiring new classes based on a model with only one class.

Table 3. Accuracy results with Cone-torus (synthetic three-class dataset) for different initial classes in the one-class model in experiment A (incrementing all classes)

	initial class	Iterations					Target
		1st	25%	50%	75%	100%	
OPFCI	1	78.2	85.1	84.9	84.5	84.	84.5
	2	84.6	85.4	85.7	85.3	85.0	
	3	69.4	86.0	86.9	86.3	84.8	
SVM	1	78.8	84.5	84.7	87.7	87.9	87.3
	2	74.9	86.2	87.6	87.9	87.4	
	3	61.8	82.8	85.4	87.7	87.3	
3NN	1	70.0	75.7	83.9	85.0	86.5	86.5
	2	81.1	84.1	88.7	87.2	86.5	
	3	71.0	75.5	83.4	84.5	86.5	

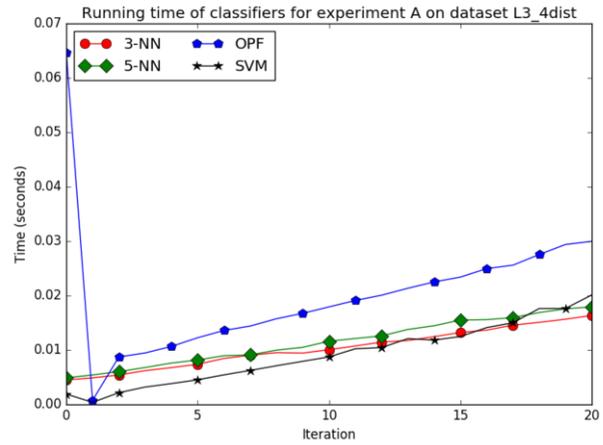
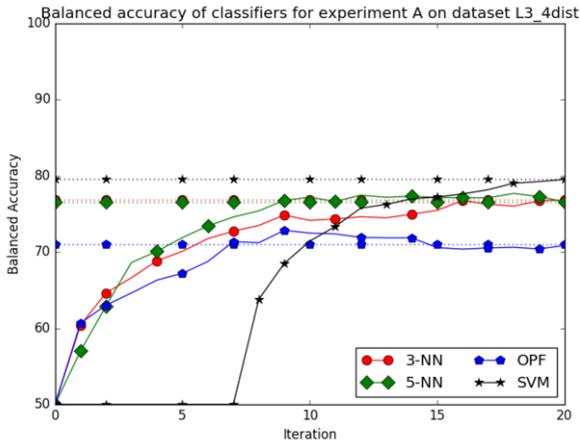
Table 4. Accuracy results with NTL (two-class dataset) for different initial classes in the one-class model in experiment A (incrementing all classes)

	initial class	Iterations					Target
		1st	25%	50%	75%	100%	
OPFCI	1	52.2	68.6	74.7	79.7	80.9	81.5
	2	80.7	81.5	81.4	82.3	80.9	
SVM	1	51.5	63.4	73.4	80.4	80.0	80.0
	2	79.8	77.4	77.4	80.1	80.0	
3NN	1	50.6	51.5	64.4	70.7	73.9	73.9
	2	71.6	72.6	71.1	73.6	73.9	

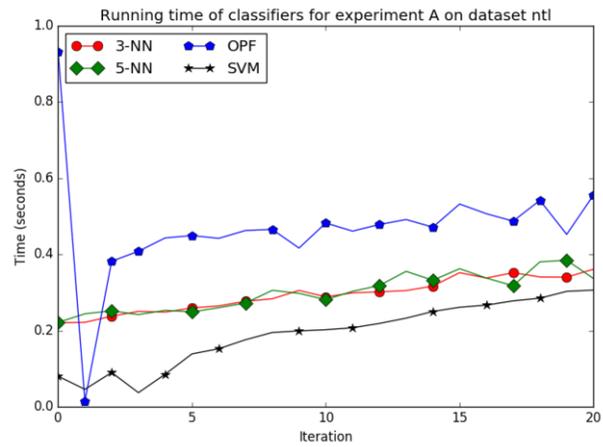
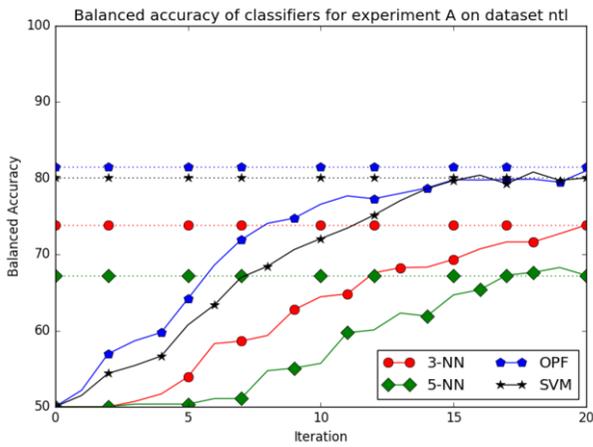
Table 5. Accuracy results with Produce (14-class dataset) for a sample of initial classes in the one-class model in experiment A (incrementing all classes in each iteration)

	initial class	Iterations					Target
		1st	25%	50%	75%	100%	
OPFCI	1	70.9	88.2	91.1	94.1	94.7	94.4
	2	73.5	89.4	92.6	94.0	94.7	
	13	70.9	89.4	92.2	94.2	94.7	
	14	71.2	89.3	92.5	94.4	94.7	
SVM	1	71.5	91.4	93.8	95.5	96.8	96.6
	2	72.6	90.7	93.5	95.4	96.8	
	13	73.6	90.8	93.6	95.5	96.7	
	14	72.6	90.8	93.8	95.5	96.7	
3NN	1	58.7	83.0	87.5	91.2	93.0	93.0
	2	60.8	84.9	89.7	92.0	93.0	
	13	68.9	84.8	90.1	92.0	93.0	
	14	62.7	83.9	90.8	91.2	93.0	

(a) L3



(b) NTL



(c) Produce-BIC-MSB

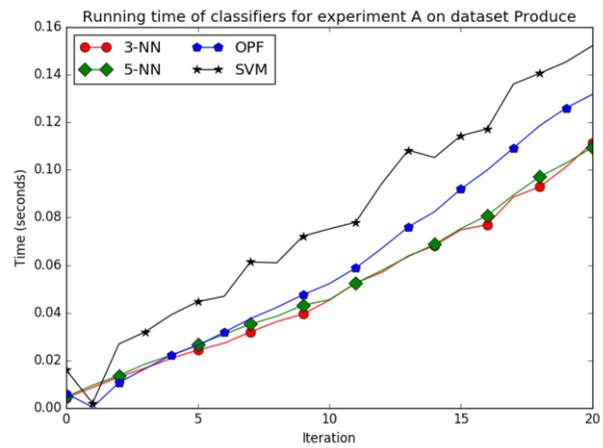
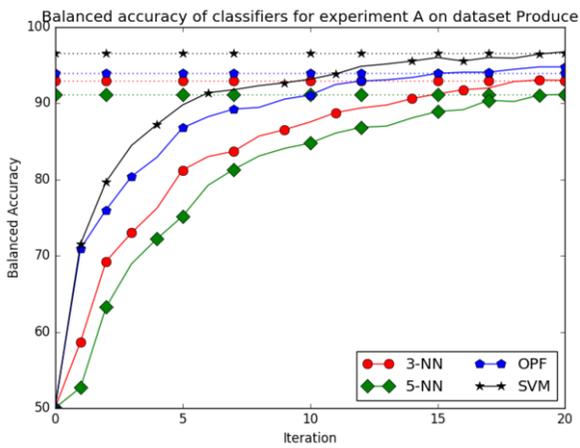


Figure 4. Experiments A, starting with one class models and adding instances of all classes at each iteration: balanced accuracy (left column) and running time (right column) at each iteration for the datasets L3 (a), NTL (b), and Produce-BIC-MSB (c). Solid lines represent the results obtained in each iteration, while dotted lines are the target accuracies for each classifier.

4.3 EXPERIMENTS B (CLASS ORDERED INCREMENTS)

In these experiments the initial model is one-class and in each iteration new instances from a single, unseen class, are included in the model until all classes are added. We report the balanced accuracy and the running time for each iteration. All remaining settings were the same as in experiment A. In Figure 5 we show the results for the datasets Cone-torus, SpamBase and MPEG7, which shows the accuracies on each increment compared with the “target” accuracy, shown as dotted lines.

In order to see the class-incremental behavior of algorithms when starting with different classes, which could be different for specific distributions or class-imbalanced datasets, we show balanced accuracy results for the datasets MPEG7, NTL and Produce, respectively in Tables 6, 7 and 8, which includes percentages of the incremental iterations, compared with the target accuracy.

Table 6. Accuracy results with MPEG7 (70-class dataset) for a sample of initial classes in the one-class model in experiment B (incrementing one class at a time)

initial class	Iterations					Target	
	1st	25%	50%	75%	100%		
OPFCI	1	50.0	62.8	77.3	86.7	90.0	89.9
	2	50.7	61.2	72.6	84.1	89.9	
	3	50.3	61.3	72.6	84.1	89.9	
	4	50.6	61.7	73.4	85.0	89.9	
SVM	1	50.0	60.5	70.0	80.5	91.4	91.4
	2	50.7	60.5	70.0	80.5	91.4	
	3	50.7	60.5	70.0	80.5	91.4	
	4	50.7	60.5	70.0	80.5	91.4	
3NN	1	50.0	60.1	69.7	79.5	84.2	84.2
	2	50.6	60.1	69.7	79.5	84.2	
	3	50.3	60.1	69.7	79.5	84.2	
	4	50.2	60.1	69.7	79.5	84.2	

Table 7. Accuracy results with NTL (two-class dataset) for different initial classes in the one-class model in experiment B (incrementing one class at a time)

initial class	Iterations					Target	
	1st	25%	50%	75%	100%		
OPFCI	1	50.0	50.0	58.1	73.7	81.0	81.5
	2	82.7	78.5	77.3	79.9	80.6	
SVM	1	50.0	50.0	52.9	73.9	80.0	80.0
	2	75.7	73.4	73.3	77.9	80.0	
3NN	1	50.0	50.0	51.1	61.4	73.9	73.8
	2	72.7	65.4	63.9	69.9	73.9	

All of the points of note of experiment A still hold true for experiment B. The results of these experiments prove our point that the OPF incremental algorithm can be used for efficient learning of new classes. As a new class begins to be inserted in the training dataset, the OPF model starts converging to its best possible accuracy given the current number of classes in the training set — the speed of this convergence depends on the dataset.

Also, as pointed out by [6], incremental learning should be robust to the order that the instances appears. The OPF-CI is robust in that sense as showed in Tables 6, 7 and 8, where the algorithm always reaches results similar to the target accuracy.

Due to space limitations, this paper does not include all available results. The complete set of results is available at the repository with

Table 8. Accuracy results with Produce (14-class dataset) for a sample of initial classes in the one-class model in experiment A (incrementing all classes in each iteration)

initial class	Iterations					Target	
	1st	25%	50%	75%	100%		
OPFCI	1	50.0	61.4	72.3	84.9	94.7	94.0
	2	53.8	61.4	72.3	84.9	94.7	
	13	53.8	65.1	75.9	87.9	94.7	
	14	53.1	65.0	75.4	87.7	94.7	
SVM	1	50.0	61.4	74.2	86.6	96.6	96.6
	2	53.6	61.4	74.2	86.6	96.6	
	13	53.5	64.9	77.8	90.0	96.6	
	14	53.2	64.8	77.6	90.0	96.5	
3NN	1	50.0	61.1	71.9	82.6	93.0	93.0
	2	50.0	61.1	71.9	82.6	93.0	
	13	51.8	64.5	75.5	85.9	93.0	
	14	52.0	64.7	74.7	85.9	93.0	

the datasets. The results with the 1NN classifier were omitted because they usually produced similar results when compared with the OPF. Also, the 5NN just produced worst results when compared with 3NN. As pointed out by [23] OPF and k NN have strong similarity and can share equivalent decision boundaries. While it was not our intent to compare such methods, we observed similar results as those of [23].

5 CONCLUSION

In this paper we demonstrated the Incremental OPF capabilities of learning new classes by starting with an optimum path forest composed only by instances from a single class, obtained by an OPF clustering process. The results reinforce previous findings, while comparing those findings with well-known classifiers.

Because the OPF-CI is linear with respect of the number of training instances, it could keep the running time acceptable, while producing an accuracy behavior similar to the other classifiers. Also, OPF is robust even when varying the order of the classes. At the end of each experiment, the OPF-CI algorithm has reached results similar to its target accuracy, thus displaying its capability of automatically learning new classes while keeping robustness to order effects.

The investigation of incremental learning classifiers is still a relevant topic in order to advance the state-of-the-art on algorithms capable of rapidly adapting themselves for future data. Future work can study in more depth the relationship between OPF and k NN and derive prototype selection methods in order to reduce the training set. Further efforts may also explore speeding-up both OPF and OPF clustering using data structures such as skip lists, as well as combining previous OPF algorithms for large scale with OPF-CI.

ACKNOWLEDGEMENTS

The authors would like to thank FAPESP (grants #11/22749-8, #14/04889-5, #15/24217-4 and #15/13504-2). We are also thankful to Mr. Leonardo Ribeiro for his valuable comments on the paper.

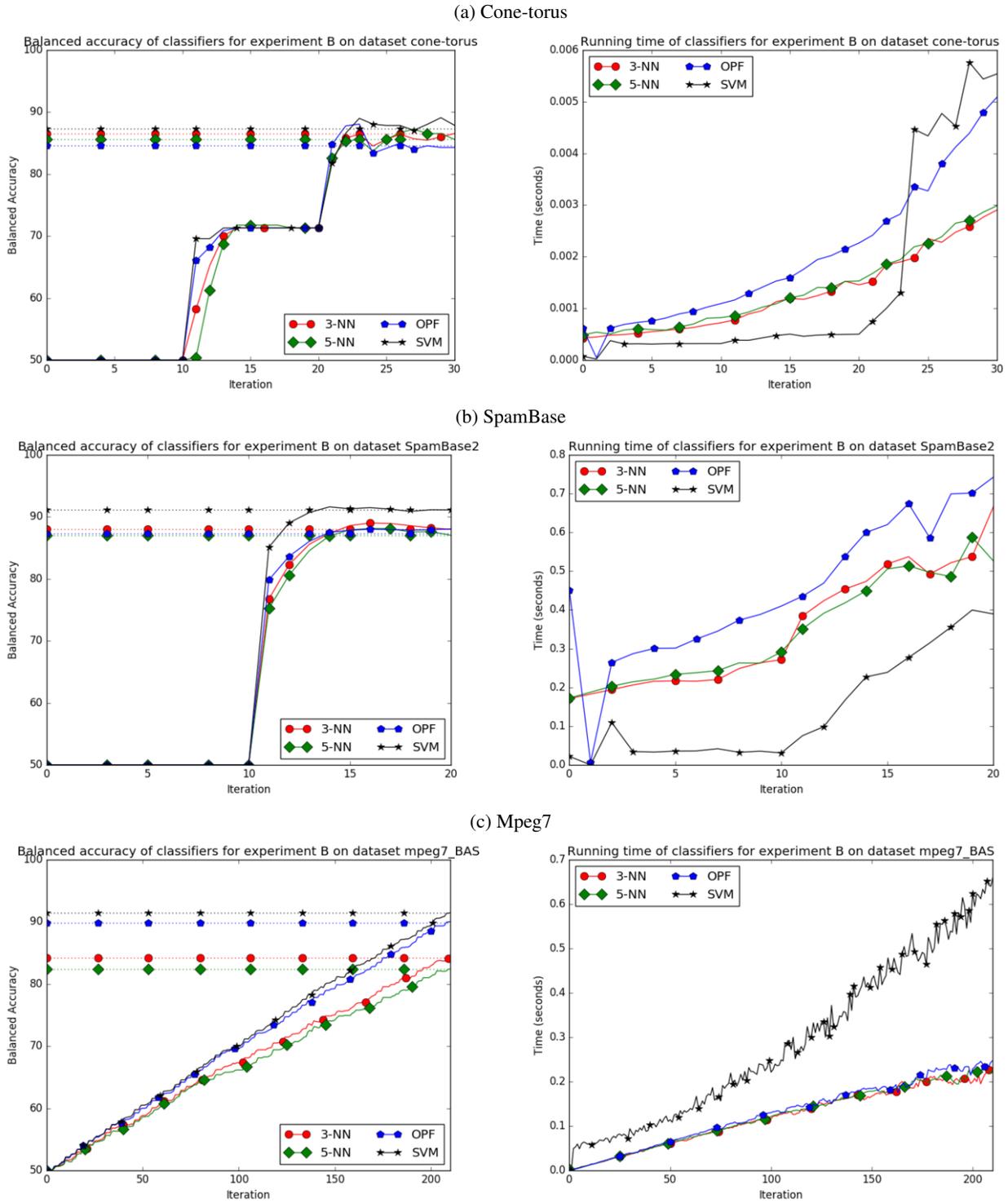


Figure 5. Experiments B (adding one class at a time, starting with class 1): balanced accuracy (left column) and running time (right column) at each iteration for the datasets Cone-Torus (a), SpamBase (b), and MPEG7 (c). Solid lines represent the results obtained in each iteration, while dotted lines are the target accuracies for each classifier.

REFERENCES

- [1] Giorgio Ausiello, Giuseppe F Italiano, Alberto Marchetti Spaccamela, and Umberto Nanni, 'Incremental algorithms for minimal length paths', *Journal of Algorithms*, **12**(4), 615–638, (1991).
- [2] José M Carmona-Cejudo, Manuel Baena-García, José del Campo-Ávila, Rafael Morales Bueno, and Albert Bifet, 'Gnusmail: Open framework for on-line email classification.', in *ECAI*, pp. 1141–1142, (2010).
- [3] Gert Cauwenberghs and Tomaso Poggio, 'Incremental and decremental support vector machine learning', *Advances in Neural Information Processing Systems*, **13**, 409–415, (2000).
- [4] Francis Chin and David Houck, 'Algorithms for updating minimal spanning trees', *Journal of Computer and System Sciences*, **16**(3), 333–344, (1978).
- [5] Corinna Cortes and Vladimir Vapnik, 'Support-vector networks', *Machine learning*, **20**(3), 273–297, (1995).
- [6] Nicola Di Mauro, Floriana Esposito, Stefano Ferilli, and Teresa Maria Altomare Basile, 'A backtracking strategy for order-independent incremental learning', in *ECAI*, volume 16, p. 460, (2004).
- [7] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia, 'A survey on concept drift adaptation', *ACM Computing Surveys (CSUR)*, **46**(4), 44, (2014).
- [8] Xin Geng and Kate Smith-Miles, 'Incremental learning', *Encyclopedia of Biometrics*, 912–917, (2015).
- [9] Christophe Giraud-Carrier, 'A note on the utility of incremental learning', *AI Communications*, **13**(4), 215–223, (2000).
- [10] Marwan Hassani, Pascal Spaus, Alfredo Cuzzocrea, and Thomas Seidl, 'Adaptive stream clustering using incremental graph maintenance', in *Proceedings of the 4th International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*, pp. 49–64, (2015).
- [11] Brian Kulis, Sugato Basu, Inderjit Dhillon, and Raymond Mooney, 'Semi-supervised graph clustering: a kernel approach', *Machine learning*, **74**(1), 1–22, (2009).
- [12] Ilija Kuzborskij, Francesco Orabona, and Barbara Caputo, 'From n to n+1: Multiclass transfer incremental learning', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3358–3365, (2013).
- [13] João P Papa, Alexandre X Falcão, Victor Hugo C De Albuquerque, and Joao Manuel RS Tavares, 'Efficient supervised optimum-path forest classification for large datasets', *Pattern Recognition*, **45**(1), 512–520, (2012).
- [14] João P Papa, Alexandre X Falcão, and Celso TN Suzuki, 'Supervised pattern classification based on optimum-path forest', *International Journal of Imaging Systems and Technology*, **19**(2), 120–131, (2009).
- [15] Moacir Ponti, Tiago S Nazaré, and Gabriela S Thumé, 'Image quantization as a dimensionality reduction procedure in color and texture feature extraction', *Neurocomputing*, **173**, 385–396, (2016).
- [16] Moacir Ponti and Camila T. Picon, 'Color description of low resolution images using fast bitwise quantization and border-interior classification', in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1399–1403, (2015).
- [17] Moacir Ponti and Mateus Riva, 'An incremental linear-time learning algorithm for the optimum-path forest classifier', *arxiv Preprint*, (2016), arxiv:1604.03346 [cs.LG].
- [18] Moacir Ponti Jr and Isadora Rossi, 'Ensembles of optimum-path forest classifiers using input data manipulation and undersampling', in *Multiple Classifier Systems*, pp. 236–246. Springer, (2013).
- [19] Ronaldo C Prati, Gustavo EAPA Batista, and Diego F Silva, 'Class imbalance revisited: a new experimental setup to assess the performance of treatment methods', *Knowledge and Information Systems*, 1–24, (2014).
- [20] Anderson Rocha, Daniel C Hauagge, Jacques Wainer, and Siome Goldenstein, 'Automatic fruit and vegetable classification from images', *Computers and Electronics in Agriculture*, **70**(1), 96–104, (2010).
- [21] L M Rocha, Fabio A M Cappabianco, and Alexandre X Falcão, 'Data clustering as an optimum-path forest problem with applications in image analysis', *International Journal of Imaging Systems and Technology*, **19**(2), 50–68, (2009).
- [22] Satu Elisa Schaeffer, 'Graph clustering', *Computer Science Review*, **1**(1), 27–64, (2007).
- [23] Roberto Souza, Leticia Rittner, and Roberto Lotufo, 'A comparison between k-optimum path forest and k-nearest neighbors supervised classifiers', *Pattern recognition letters*, **39**, 2–10, (2014).
- [24] Sebastian Thrun, 'Is learning the n-th thing any easier than learning the first?', in *Advances in Neural Information Processing Systems*, pp. 640–646, (1995).
- [25] Salvador Tortajada, Montserrat Robles, and Juan Miguel García-Gómez, 'Incremental logistic regression for customizing automatic diagnostic models', *Data Mining in Clinical Medicine*, 57–78, (2015).
- [26] Bo-Feng Zhang, Jin-Shu Su, and Xin Xu, 'A class-incremental learning method for multi-class support vector machines in text classification', in *IEEE International Conference on Machine Learning and Cybernetics*, pp. 2581–2585, (2006).

More than a Name? On Implications of Preconditions and Effects of Compound HTN Planning Tasks

Pascal Bercher and Daniel Höller and Gregor Behnke and Susanne Biundo *

Abstract. There are several formalizations for hierarchical planning. Many of them allow to specify preconditions and effects for compound tasks. They can be used, e.g., to assist during the modeling process by ensuring that the decomposition methods’ plans “implement” the compound tasks’ intended meaning. This is done based on so-called *legality criteria* that relate these preconditions and effects to the method’s plans and pose further restrictions. Despite the variety of expressive hierarchical planning formalisms, most theoretical investigations are only known for standard HTN planning, where compound tasks are just names, i.e., no preconditions or effects can be specified. Thus, up to now, a direct comparison to other hierarchical planning formalisms is hardly possible and fundamental theoretical properties are yet unknown. To enable a better comparison between such formalisms (in particular with respect to their computational expressivity), we first provide a survey on the different legality criteria known from the literature. Then, we investigate the theoretical impact of these criteria for two fundamental problems to planning: *plan verification* and *plan existence*. We prove that the plan verification problem is at most **NP-complete**, while the plan existence problem is in the general case both **semi-decidable** and **undecidable**, independent of the demanded criteria. Finally, we discuss our theoretical findings and practical implications.

1 Introduction

Hierarchical planning approaches are often chosen when it comes to practical real-world planning applications [33]. Examples include composition of web services [29], real-time strategy games [23, 35], robotics [15, 28], or user assistance [11, 10]. While there are several different formalizations for hierarchical planning, it is apparent that most of the *theoretical* investigations are done for a standard formalization (called hierarchical task network (HTN) planning), where compound (or abstract) tasks are just names or symbols [16, 19] – they thus neither show preconditions nor effects. Those investigations include the complexity of the plan existence problem (“Is the problem solvable?”) [16, 19, 5, 2, 3], plan verification (“Is the given plan a solution to the problem?”) [9], changes to plans (“Is the plan still a solution if I change X?”) [8], and expressivity analysis (“What plan structures can be expressed using different language features?”) [21, 22]. The answers to such questions, besides being of theoretical interest, are highly relevant to come up with tractable problem relaxations for heuristics [5] or for problem compilations [4, 1].

For mainly practically motivated reasons, such as providing modeling assistance or generating abstract solutions, several researchers developed hierarchical planning formalizations in which compound

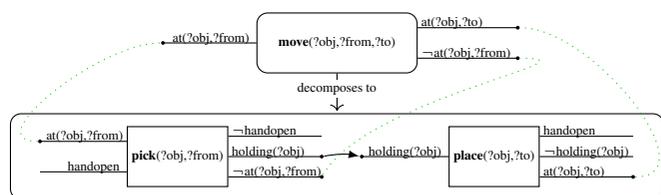


Figure 1. Example for a compound task (*move*) with preconditions and effects and one of its methods. The method’s plan consists of two primitive tasks (*pick* and *place*) and a causal link protecting the condition *holding*. The dotted green lines indicate how the preconditions and effects of the tasks in the plan’s method are related to their more abstract representation.

tasks are allowed to have preconditions and effects [40, 25, 37, 41, 18, 26, 12, 14, 30, 11, 7, 15] (cf. example given in Fig. 1). However, the only theoretical investigations for such a formalization that we are aware of are about the upward and downward refinement properties [40, 6, 30]. So, up to now, for many of such formalisms it is not even clear how hard the respective problems are (given the typical HTN solution criteria), since the plan existence problem was not studied for such a formalization. To close this gap, we investigate the plan existence and the plan verification problems for our formalization that allows to specify preconditions and effects for compound tasks. We survey several legality criteria that define which decomposition methods may be specified for which compound task, depending on its preconditions and effects and take these criteria into account in our complexity analysis. We conclude the paper by discussing our findings and its implications.

2 Problem Formalization

We introduce a hierarchical planning formalism that allows to specify preconditions and effects for compound tasks. In our complexity analysis, we take into account several restrictions on the relationship between compound tasks and the plans that are associated with them via their decomposition methods. The investigated restrictions, called legality criteria, are taken from the literature. To formally study their impact on the complexity results, the formalism needs to be rich enough to be capable of expressing these criteria. Most of the formalisms that define them [40, 41, 37, 12] fuse standard HTN planning [16, 20, 19] with Partial-Order Causal-Link (POCL) planning [31, 36, 20]. We assume this is because the concept of causal links makes it easy to express the desired criteria. Thus, we also use such an hybridization for our investigations. A variety of formalisms fuse HTN with POCL planning [40, 25, 37, 41, 18, 14, 7, 15, 11].

*Institute of Artificial Intelligence, Ulm University, D-89069 Ulm, Germany, {forename.surname}@uni-ulm.de

However, none of these formalizations is both rich enough to allow expressing all legality criteria while being simple enough to easily serve as a basis for proofs. Since all HTN complexity results that are relevant for the sake of this paper have been shown or reproduced in the simplistic (propositional) HTN formalism by Geier and Bercher [19], extending it allows an easy comparison. We therefore extend it by the necessary POCL concepts. In accordance to the literature [26, 11], we refer to the resulting formalism as *hybrid planning*.

Let V be a finite set of *state variables* (or *proposition symbols*). In POCL planning, actions are typically 2-tuples consisting of a precondition and effect, both being conjunctions of literals. Here, we use an equivalent set-based formalization: Actions (or *primitive tasks*) are 4-tuples $(prec^+, prec^-, eff^+, eff^-)$, where $prec^+$ and $prec^-$ denote the positive and negative *preconditions* and eff^+ and eff^- denote the positive and negative *effects*. They describe single state transitions as usual. The *compound* (or *abstract*) tasks have a different underlying meaning, as they need to be decomposed into pre-defined plans by relying on so-called *decomposition methods*. Despite that fact, we allow compound tasks to use preconditions and effects as well (cf. Fig. 1 for an example). Every task has a *task name*. The set of names for the primitive tasks is given by N_p and those of the compound ones by N_c . We define $N := N_p \cup N_c$. The mapping between task names and their actual tasks (i.e., 4-tuples) is established using the function $\delta : N \rightarrow (2^V)^4$. For convenience, we also write $prec^+(n)$, $prec^-(n)$, $eff^+(n)$, and $eff^-(n)$ to refer to n 's positive and negative preconditions and effects, respectively[†]. We call a sequence of tasks $\delta(n_1), \dots, \delta(n_k)$ *executable in a state* $s_0 \in 2^V$ if and only if there is a corresponding sequence of states s_0, \dots, s_k , such that for all $1 \leq i \leq k$ holds $prec^+(n_i) \subseteq s_{i-1}$ and $prec^-(n_i) \cap s_{i-1} = \emptyset$ as well as $s_i = (s_{i-1} \setminus eff^-(n_i)) \cup eff^+(n_i)$. The state s_k is called *the state generated by* $\delta(n_1), \dots, \delta(n_k)$.

In non-linear planning approaches, *plans* are only partially ordered. For a set of ordering constraints \prec , we denote its transitive closure by \prec^* . To differentiate multiple occurrences of the same task within a plan, the partial order is defined over a set of so-called *plan steps* PS , which then map to the actual task name, $\alpha : PS \rightarrow N$. When we mention a *linearization* of the plan steps of a plan, we refer to a total order of PS that does not violate the partial order \prec . Plans may also contain so-called *causal links*. A causal link $(ps, v, ps') \in PS \times V \times PS$ indicates that the precondition v of the *consumer* plan step ps' is *supported* by the *producer* plan step ps . The condition v is also said to be *protected* by the causal link. This means, if v is a positive (resp. negative) precondition of ps' , then no task with v as negative (resp. positive) effect is allowed to be ordered between ps and ps' .

Definition 1 (Plan). A plan P over a set of task names N is a 4-tuple (PS, CL, \prec, α) , where:

- PS is a finite (possibly empty) set of plan steps,
- $CL \subseteq PS \times V \times PS$ is a set of causal links. If $(ps, v, ps') \in CL$, then $v \in prec^+(ps')$ and $v \in eff^+(ps)$ or $v \in prec^-(ps')$ and $v \in eff^-(ps)$. We also require that every precondition variable of all plan steps is protected by at most one causal link,
- $\prec \subseteq PS \times PS$ is a strict partial order. If $(ps, v, ps') \in CL$ with $\alpha(ps)$ and $\alpha(ps')$ being primitive, then $(ps, ps') \in \prec^{\ddagger}$,

[†]To simplify upcoming definitions, we require that for all primitive tasks n_p , $prec^+(n_p) \cap prec^-(n_p) = eff^+(n_p) \cap eff^-(n_p) = \emptyset$.

[‡]As usual in POCL planning, any causal link between primitive tasks implies an ordering. If one of these tasks was compound, this would not be reasonable: Consider the example depicted in Fig. 1 and assume there is a causal link from *move*'s effect $\neg at$ to another task's precondition. If that link

- $\alpha : PS \rightarrow N$ labels every plan step with its task name.

\mathcal{P}_N denotes the set of all plans over the task names N . Two plans are called *isomorphic* if they are identical except for plan step renaming.

Based on the concept of plans, we define *decomposition methods*. The set of all decomposition methods $M \subseteq N_c \times \mathcal{P}_N$ is given in the planning domain and defines how compound tasks can be decomposed. That is, a method $(n_c, P) \in M$ indicates that the compound task (name) n_c can be decomposed into the plan P .

Definition 2 (Hybrid Planning Problem). A hybrid planning problem is a 6-tuple $\pi = (V, N_c, N_p, \delta, M, P^i)$, where:

- V is a finite set of state variables,
- we require $N_c \cap N_p = \emptyset$ and define $N := N_c \cup N_p$, where:
 - N_c is a finite set of compound task names,
 - N_p is a finite set of primitive task names,
 - $\{init, goal\} \subseteq N_p$ denote two special primitive task names,
- $\delta : N \rightarrow (2^V)^4$ is a function mapping the task names to their preconditions and effects[§],
- $M \subseteq N_c \times \mathcal{P}_{\{N\} \setminus \{init, goal\}}$ is a finite set of (decomposition) methods, and
- $P^i = (PS^i, CL^i, \prec^i, \alpha^i) \in \mathcal{P}_N$, is the initial plan. We require that there are plan steps $ps, ps' \in PS^i$ such that:
 - $\alpha^i(ps) = init$ and $\alpha^i(ps') = goal$, and
 - $ps \prec ps'$ and for all $ps'' \in PS^i$ with $ps'' \cap \{ps, ps'\} = \emptyset$ holds $ps \prec ps'' \prec ps'$.

The actual problem that one would like to have solved is given in terms of the initial plan P^i . As done in POCL planning, this plan contains two artificial actions that encode the initial state and the goal description, respectively[§]. Since hybrid planning is a hierarchical setting, P^i usually contains a set of compound tasks for which one needs to find an executable refinement. We added the specification of a goal description for practical reasons: It is especially interesting if one allows the arbitrary insertion of tasks into the plan apart from decomposing compound tasks [19] (not considered in this paper), since hybrid planning then directly captures both classical and POCL planning. Independent of whether task insertion is allowed or not, adding a goal description is not required from a purely theoretical point of view, since one can easily simulate it [19, Sec. 2].

In HTN planning, only those plans are regarded solutions that can be obtained from the initial plan by successively applying decomposition methods to compound tasks. We thus need to define how applying a method transforms one plan into another. Since the decomposed task might serve as a producer or consumer of causal links, we have to decide how such links will be passed down to sub tasks and whether this is mandatory or not (i. e., we could even allow that such links may be deleted upon decomposition). Adding a causal link to a compound task means to commit that the state variable of that link is protected for the complete sequence of states over which this link spans. Allowing to remove that link upon decomposition would remove this constraint and violate the refinement principle [24]. If

would imply an ordering, then after *move* was decomposed, the consumer task had to be ordered behind *place*, which is overly restrictive.

[§]The initial state and goal description are specified in terms of the task names *init* and *goal* and their tuple representation using δ . As usual in POCL planning, the action for *init* does not show a precondition and uses the initial state as effect and, analogously, the action for *goal* has no effects and uses the goal description as precondition.

causal links do have to be passed down to sub tasks, then the respective formalism satisfies the so-called *monotonic property* [27]. If this property does not hold, hierarchical planning systems cannot exploit such causal links to prune plans from the search space, as the constraint imposed by these links could disappear upon decomposition [14]. We hence require that causal links are not allowed to disappear upon decomposition. *How* causal links are passed down has yet to be decided – and different conventions exist [25, p. 204]. We follow the canonical approach by Yang [40] and pass down every causal link to each “compatible” sub task. In other words, if there is a link to a compound task’s precondition/effect v , then for each precondition/effect v in its sub tasks, one successor plan is generated in which the link is passed down to the respective task. In case there is more than one matching sub task, it is not required that the causal link is duplicated to support all these tasks (or a sub set thereof), since the respective plan can be obtained via link insertions from the other plans.

The following definitions formally capture the decomposition of compound tasks and the inheritance of causal links. We first define two functions $in_P, out_P : PS \rightarrow 2^{CL}$ that return the set of incoming, respectively outgoing, causal links of a plan step $ps \in PS$ in a plan $P = (PS, CL, \prec, \alpha)$ as $in_P : ps \mapsto \{(ps', v, ps) \in CL \mid ps' \in PS\}$ and $out_P : ps \mapsto \{(ps, v, ps') \in CL \mid ps' \in PS\}$.

Definition 3 (Decomposition). A method $m = (n_c, P) \in M$ decomposes a plan $P' = (PS', CL', \prec', \alpha')$ into another plan P'' by replacing plan step $ps \in PS'$ with $\alpha'(ps) = n_c$ if and only if:

- there is a plan $\tilde{P} = (\tilde{PS}, \tilde{CL}, \tilde{\prec}, \tilde{\alpha})$ that is isomorphic to P , such that $\tilde{PS} \cap PS' = \emptyset$,
- for each causal link $(ps', v, ps) \in in_{P'}(ps)$ there is a plan step $\tilde{ps}_{(ps', v, ps)} \in \tilde{PS}$, such that:
 - $v \in prec^+(\tilde{\alpha}(\tilde{ps}_{(ps', v, ps)}))$ in case $v \in prec^+(\alpha'(ps))$, or
 - $v \in prec^-(\tilde{\alpha}(\tilde{ps}_{(ps', v, ps)}))$ in case $v \in prec^-(\alpha'(ps))$,
- for each causal link $(ps, v, ps') \in out_{P'}(ps)$ there is a plan step $\tilde{ps}_{(ps, v, ps')} \in \tilde{PS}$, such that:
 - $v \in eff^+(\tilde{\alpha}(\tilde{ps}_{(ps, v, ps')}))$ in case $v \in eff^+(\alpha'(ps))$, or
 - $v \in eff^-(\tilde{\alpha}(\tilde{ps}_{(ps, v, ps')}))$ in case $v \in eff^-(\alpha'(ps))$,
- $P'' = (PS'', CL'', \prec'', \alpha'')$ is given as follows:

$$\begin{aligned}
 PS'' &:= (PS' \setminus \{ps\}) \cup \tilde{PS} \\
 CL'' &:= (CL' \setminus (in_{P'}(ps) \cup out_{P'}(ps))) \\
 &\quad \cup \{(ps', v, \tilde{ps}_{(ps', v, ps)}) \mid (ps', v, ps) \in in_{P'}(ps)\} \\
 &\quad \cup \{(\tilde{ps}_{(ps, v, ps')}, v, ps') \mid (ps, v, ps') \in out_{P'}(ps)\} \\
 \prec'' &:= (\prec_1 \cup \tilde{\prec} \cup \prec_2 \cup \prec_3)^*, \text{ with} \\
 \prec_1 &:= (\prec' \setminus \{(ps', ps'') \in \prec' \mid ps \cap \{ps', ps''\} \neq \emptyset\}) \\
 \prec_2 &:= \{(ps', ps'') \in PS' \times \tilde{PS} \mid (ps', ps) \in \prec'\} \cup \\
 &\quad \{(ps', ps'') \in \tilde{PS} \times PS' \mid (ps, ps'') \in \prec'\} \\
 \prec_3 &:= \{(ps', \tilde{ps}_{(ps', v, ps)}) \mid (ps', v, \tilde{ps}_{(ps', v, ps)}) \in CL'' \\
 &\quad \text{and } \{\alpha(ps'), \alpha(\tilde{ps}_{(ps', v, ps)})\} \subseteq N_p\} \cup \\
 &\quad \{(\tilde{ps}_{(ps, v, ps')}, ps') \mid (\tilde{ps}_{(ps, v, ps')}, v, ps') \in CL'' \\
 &\quad \text{and } \{\alpha(\tilde{ps}_{(ps, v, ps')}, \alpha(ps'))\} \subseteq N_p\} \\
 \alpha'' &:= (\alpha' \setminus \{(ps, n_c)\}) \cup \tilde{\alpha}
 \end{aligned}$$

As noted, we require that causal links involving the decomposed plan step ps (i.e., $in_{P'}(ps)$ and $out_{P'}(ps)$) are passed down upon

decomposition (cf. CL''). For this, any compatible precondition that is not yet protected by a causal link inside the method’s plan \tilde{P} can be used. The definition of the ordering constraints of the new plan P'' , \prec'' comprises all ordering constraints of the original plan P' except the ones involving the decomposed plan step ps , \prec_1 . It further contains all ordering constraints of the method’s plan \tilde{P} , $\tilde{\prec}$, as well as those that are inherited from the orderings involving ps , \prec_2 . All new causal links only involving primitive tasks are responsible for adding further orderings, \prec_3 . Apart from the necessary extensions to handle causal links, our definition of decomposition is identical to the one from HTN planning [19, Def. 3].

In HTN planning, any solution to a planning problem (1) needs to be obtainable from the initial task network via the application of a sequence of decompositions and (2) needs to contain an executable sequence of its actions [19, Def. 5, 6]. We consider the second criterion as impractical: One is usually interested in executable action sequences, but finding one from a “solution” task network is still **NP-hard** [34, Thm. 15], [16, Thm. 8]. Further, such a sequence itself is in general not regarded a solution (but just the task network in which this sequence occurs), which we regard contra-intuitive. Instead, we require that *all* linearizations are executable and that each executable linearization is considered a solution as well. To support this stronger notion of solutions, we also allow the insertion of causal links and ordering constraints.

Definition 4 (Causal Link Insertion). Let $P = (PS, CL, \prec, \alpha)$ and $P' = (PS, CL', \prec', \alpha)$ be plans. P' can be obtained from P by insertion of a causal link $(ps, v, ps') \notin CL$ with $ps, ps' \in PS$ if and only if:

- $v \in eff^+(\alpha(ps))$ and $v \in prec^+(\alpha(ps'))$ or $v \in eff^-(\alpha(ps))$ and $v \in prec^-(\alpha(ps'))$,
- $CL' = CL \cup \{(ps, v, ps')\}$, and
- $\prec' = (\prec \cup \{(ps, ps') \mid \{\alpha(ps), \alpha(ps')\} \subseteq N_p\})^*$

Definition 5 (Ordering Insertion). Let $P = (PS, CL, \prec, \alpha)$ and $P' = (PS, CL, \prec', \alpha)$ be plans. P' can be obtained from P by insertion of an ordering constraint $(ps, ps') \notin \prec$, $ps, ps' \in PS$ if and only if $\prec' = (\prec \cup \{(ps, ps')\})^*$.

Definition 6 (Solution). A plan $P = (PS, CL, \prec, \alpha)$ is a solution to a planning problem π , if and only if:

1. P is a refinement of P^i . That is, there is a sequence of decompositions (cf. Def. 3), causal link insertions (cf. Def. 4), and ordering constraint insertions (cf. Def. 5) transforming P^i into P ,
2. P is primitive, i.e., $\{\alpha(ps) \mid ps \in PS\} \subseteq N_p$, and
3. P is executable in the standard POCL sense:
 - There are no unprotected preconditions. A precondition $v \in prec^+(\alpha(ps))$ (resp. $v \in prec^-(\alpha(ps))$) of a plan step $ps \in PS$ is called *unprotected*, if and only if there is no plan step $ps' \in PS$ with a causal link (ps', v, ps) for $v \in eff^+(\alpha(ps))$ (resp. $v \in eff^-(\alpha(ps))$).
 - There are no causal threats. A plan contains a causal threat if and only if there is a causal link $(ps, v, ps') \in CL$ with $v \in prec^+(\alpha(ps'))$ (resp. $v \in prec^-(\alpha(ps'))$) and a plan step $ps'' \in PS$ with $v \in eff^-(\alpha(ps''))$ (resp. $v \in eff^+(\alpha(ps''))$), such that neither $(ps'', ps) \in \prec$ nor $(ps', ps'') \in \prec$ holds.

The first solution criterion corresponds to the standard HTN criterion that requires every solution to be in the refinement space of the initial plan. The second solution criterion demands the respective

plan to be primitive, as only primitive plans are typically regarded executable. The third criterion requires executability as it is done in POCL planning; these criteria ensure that every linearization of the plan steps corresponds to a sequence of tasks that is executable in the initial state and generates a state satisfying the goal description.

3 Legality Criteria – A Survey and Discussion

Provided the planning domain allows to specify preconditions and effects for compound tasks, there should be a clearly-defined criterion stating which decomposition methods are allowed to be specified for such compound tasks [17, 14]. When considering a compound task as an abstraction of a certain plan, it does not seem to make much sense to specify a precondition or effect of that task if it does not occur anywhere in the plan. So, if the domain modeler decides to specify preconditions and effects for a compound task, he or she has a certain idea on how the plans of the respective decomposition methods should look like. Thus, several researchers have formalized possible relations between a compound task’s preconditions and effects and its methods’ plans. We call these criteria *legality criteria* and plans that respect them *implementations* of their compound task. For each criterion, we give a small example illustrating it. Further, we want to note that the example depicted in Fig. 1 satisfies all of the legality criteria discussed in this section⁴.

The first and weakest criterion that we investigate is closely related to the criteria that ensure that a compound task can be decomposed (cf. Def. 3). We restrict to models where the plan of a compound task’s method makes use of the task’s preconditions and effects.

Definition 7 (Downward Compatible). *A method $(n_c, P) \in M$ with $P = (PS, CL, \prec, \alpha)$ is called downward compatible if and only if:*

- for each $v \in prec^+(n_c)$ (resp. $v \in prec^-(n_c)$) there is a plan step $ps \in PS$ with an unprotected precondition $v \in prec^+(\alpha(ps))$ (resp. $v \in prec^-(\alpha(ps))$).
- for each $v \in eff^+(n_c)$ (resp. $v \in eff^-(n_c)$) there is a plan step $ps \in PS$ with the same effect $v \in eff^+(\alpha(ps))$ (resp. $v \in eff^-(\alpha(ps))$).

Downward compatible methods (n_c, P) are always applicable to a plan as long as it contains n_c – a property shared with standard HTN planning (without method preconditions). If the method was not downward compatible, causal links involving n_c influence its applicability, which also causes unintended “strange” behavior during search: Let us assume a plan P' contains a plan step ps with the name n_c that has an unprotected effect $v \in V$. Now, assume that (n_c, P) does violate the legality criterion. Whether this method can be used to generate a successor plan now depends on the planner’s choice whether it first decomposes ps (this would work since all causal links can be correctly passed down to sub tasks in P') or first adds a causal link from $ps \in PS$ protecting v (then, ps can not be decomposed because the newly inserted link cannot be passed down, which violates the decomposition criteria given in Def. 3).

While this criterion clearly ensures that the “most obvious” modeling errors are prevented, it is not yet clear whether the user’s intent about the relationship between the compound task and its methods’ plans is actually satisfied. This is due to the fact that there are various possibilities what the preconditions and effects of the compound task are meant to entail. For example, if a primitive action n_p is within a plan, we know that – assuming that plan is executable – there are also

states s and s' , such that s satisfies n_p ’s precondition and s' satisfies n_p ’s effects. For compound tasks, this is not necessarily the case. In particular for the downward compatibility, it is clear that the compound task’s effects do not need to be true in one single state, but its state variables might only hold in different states.

Several more restrictive criteria have been proposed in the literature. They can be categorized into two classes: one *enforces* that compound tasks have non-empty preconditions/effects under certain circumstances, and one where the specification thereof is *optional*.

We are only aware of one legality criterion that falls into the first class: It was proposed by Russell and Norvig for their fusion of HTN with POCL planning [37]. For each method (n_c, P) and every effect of n_c , they require that “it [is] asserted by at least one step of P and is not denied by some other, later step”. Further, they require that “every precondition of the steps in P must be achieved by a step in P or be one of the preconditions of n_c ”. This implies that every open precondition in P needs to be a precondition of the compound task. This, in turn, has further consequences: Consider the case where the task n_c has two decomposition methods (n_c, P) and (n_c, P') . According to this criterion, n_c must use all open preconditions of both P and P' . As a consequence, both the downward compatibility and hence the monotonic property [27] may be violated. As argued in Sect. 2 (motivation for Def. 3), in this paper, we do not allow that commitments on an abstract level may be removed upon decomposition. We are thus not investigating this criterion in more detail.

The next criterion that we discuss was proposed by Biundo and Schattenberg [12]. As their planning formalism shows substantial differences to the one in this paper, we do not capture every detail, but only the main ideas. Their formalism is based upon a many-sorted first-order logic that features abstract state variables and so-called decomposition axioms defining them. They enable abstract tasks to make use of abstract state variables thereby fusing action abstraction with state abstraction. Further, their definition of methods only featured totally ordered plans: there, a method contains a set of task sequences, each of which is an implementation of the compound task. So, each method containing n sequences is a compact representation of n methods with totally ordered plans. Allowing for methods with partially ordered plans strictly increases expressivity (with respect to the plan existence problem [16, 2] and the generated solutions [21]), so legality should also be defined for such methods. We hence adapt their definition to partially ordered plans.

They require that if a method’s task sequence is executable in a state satisfying the compound task’s precondition, then it generates a state that satisfies the compound task’s effects. They further require the compound task’s precondition to be an abstraction of the task sequence’s first task’s precondition. Our generalization to partial orders states that this property must hold for every linearization that is induced by the given plan. However, we further demand that there also needs to be a state in which all linearizations are executable. As discussed earlier, causal links involving compound tasks do not (directly) induce ordering constraints (cf. Def. 1). However, since we consider a plan legal if all its linearizations are legal (which in turn need to respect the causal links), we here interpret causal links as additional ordering constraints. We thus define $\prec^+ = (\prec \cup \{(ps_i, ps_j) \mid (ps_i, v, ps_j) \in CL\})^*$ and only consider linearizations with respect to \prec^+ . Then, a plan step linearization is said to *respect* a set of causal links CL if none of the protected conditions is violated by the induced state sequence.

Definition 8 (along Biundo and Schattenberg [12]). *A method $(n_c, P) \in M$, $P = (PS, CL, \prec, \alpha)$, is called legal if and only if:*

⁴For the sake of readability, the example is given with variables, whereas our formalization assumes a ground (i.e., propositional) representation.

- (n_c, P) is downward compatible,
- let $N_{\text{first}} := \{\alpha(ps) \mid ps \in PS \text{ and there is no } ps' \in PS \text{ with } (ps', ps) \in \prec^+\}$. Then, $\text{prec}^+(n_c) \subseteq \bigcap_{n \in N_{\text{first}}} \text{prec}^+(n)$ and $\text{prec}^-(n_c) \subseteq \bigcap_{n \in N_{\text{first}}} \text{prec}^-(n)$,
- for all states $s \in 2^V$ holds: if (a) $s \supseteq \text{prec}^+(n_c)$ and $s \cap \text{prec}^-(n_c) = \emptyset$, (b) every task sequence \bar{t} corresponding to a plan step linearization $ps_1, \dots, ps_{|PS|}$ with respect to \prec^+ is executable in s , and (c) respects CL , then (d) \bar{t} generates a state s' , such that $\text{eff}^+(n_c) \subseteq s'$ and $\text{eff}^-(n_c) \cap s' = \emptyset$. Further, there needs to be a state $s \in 2^V$, such that (a) to (d) hold.

Concerning the intention behind the compound task's preconditions and effects, this criterion is already much stronger than the simple downward compatibility criterion. As opposed to that criterion, we here get the property that in any solution plan that is obtained from decomposing a compound task n_c , there is a *single* state in which n_c 's preconditions hold. This trivially follows from the fact that the preconditions of n_c are abstractions of any first task and any compound task needs to be decomposed into a primitive plan. However, the criteria do not imply that there is a single state in which the compound task's effects hold – despite the restrictions imposed on the relationship between n_c 's effects and its methods' plans. The reason for this is that the criterion does not take into account additional effects of the compound tasks that are in the method's plan, other than those explicitly specified in their preconditions and effects. This issue is addressed by Marthi et al.'s *possible* effects [30]. They are, however, restricting to totally ordered methods.

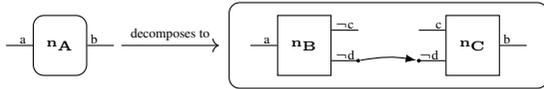


Figure 2. Illustration of a method that is supposed to implement the compound task n_A . The tasks n_B and n_C are primitive.

Concerning executability, Def. 8 requires that there is at least one state in which the respective plan's linearizations are executable, which might be considered too restrictive. The example given in Fig. 2 is not legal with respect to Def. 8, since – due to the variable c – there cannot be a state in which n_A 's method's plan is executable. Since other tasks could be ordered in between n_B and n_C to support n_C 's precondition c , one could also relax this criterion. This is done by the next legality criterion (for which the example is legal), as it allows that open preconditions of a method's plan may be supported by tasks at an arbitrary “position” within a plan (i.e., it does not require that a single state enables the execution of the plan's tasks).

The criterion was proposed by Yang [40, p. 14] and consists of three sub criteria. The first two imply downward compatibility, but require more. Criterion 1 demands, similar to Def. 8, that none of the tasks in the plan invalidates the compound task's effects. Criterion 2 requires that every precondition variable of the compound task has to occur in the plan in such a way that none of its sub tasks can be used to establish it. Criterion 3 is closely related to the concept of causal threats. Each precondition within the plan might not be violated by a converse effect of another task. Note that this is neither equivalent to stating that the respective plan must be free of causal threats (as the criterion must also hold in the absence of any causal links) nor that the plan is executable in any way (as none of the criteria ensures preconditions to be supported).

Definition 9 (Yang [40]). A method $(n_c, P) \in M$ with $P = (PS, CL, \prec, \alpha)$ is called legal if and only if:

1. for all $v \in \text{eff}^+(n_c)$ (resp. $v \in \text{eff}^-(n_c)$) there exists a $ps \in PS$ with $v \in \text{eff}^+(\alpha(ps))$ (resp. $v \in \text{eff}^-(\alpha(ps))$), such that for all $ps' \in PS$, $ps' \neq ps$ holds: if $v \in \text{eff}^-(\alpha(ps'))$ (resp. $v \in \text{eff}^+(\alpha(ps'))$), then $(ps', ps) \in \prec$.
2. for all $v \in \text{prec}^+(n_c)$ (resp. $v \in \text{prec}^-(n_c)$) there exists a $ps \in PS$ with $v \in \text{prec}^+(\alpha(ps))$ (resp. $v \in \text{prec}^-(\alpha(ps))$), such that for all $ps' \in PS$, $ps' \neq ps$ holds: if $v \in \text{eff}^+(\alpha(ps'))$ (resp. $v \in \text{eff}^-(\alpha(ps'))$), then $(ps, ps') \in \prec$.
3. for all $ps \in PS$, for all $v \in \text{prec}^+(\alpha(ps))$ (resp. $v \in \text{prec}^-(\alpha(ps))$), and for all $ps' \in PS$ with $ps' \neq ps$ and $(ps, ps') \notin \prec$ it holds: if $v \in \text{eff}^-(\alpha(ps'))$ (resp. $v \in \text{eff}^+(\alpha(ps'))$), then there exists a $ps'' \in PS$, such that $\{(ps', ps''), (ps'', ps)\} \subseteq \prec$ and $v \in \text{eff}^+(\alpha(ps'))$ (resp. $v \in \text{eff}^-(\alpha(ps'))$).

The next legality criterion is by Young et al. [41]. They argue that Yang's criterion of threat-free plans was too strong. Instead, their only requirement is that any of the compound task's preconditions “contributes” to at least one of its effects (and vice versa), which they ensure by requiring the existence of a chain of causal links within the plans connecting them with each other [41, p. 191]. Thus, our example illustrated in Fig. 2 also satisfies this criterion, but it would not do so if both the variable d and the respective causal link were not present (while assuming that n_C is still ordered behind n_B).

They model their criterion by including artificial start and end actions, which use the compound task's precondition and effect as effect and precondition, respectively, and require the causal link chains between them. Upon decomposition, those actions disappear, but the causal links involving them are reused to be linked to the plan steps that share causal links with the compound task. Those actions thus do not imply that there are states, in which the compound task's preconditions and effects hold. In the following definition, we therefore did not include the artificial start and end tasks.

Definition 10 (Young et al. [41]). A method $(n_c, P) \in M$ with $P = (PS, CL, \prec, \alpha)$ is called legal if and only if:

- (n_c, P) is downward compatible and
- for each $v \in \text{eff}^+(n_c)$ (resp. $v \in \text{eff}^-(n_c)$), there is a sequence of plan steps ps_1, \dots, ps_k with $ps_i \in PS$ for $1 \leq i \leq k$, $v \in \text{eff}^+(\alpha(ps_k))$ (resp. $v \in \text{eff}^-(\alpha(ps_k))$), $v' \in \text{prec}^+(\alpha(ps_1)) \cup \text{prec}^-(\alpha(ps_1))$, and a chain of causal links connecting them.
- for each $v \in \text{prec}^+(n_c)$ (resp. $v \in \text{prec}^-(n_c)$), there is a sequence of plan steps $ps_1, \dots, ps_{k'}$ with $ps_i \in PS$ for $1 \leq i \leq k'$, $v \in \text{prec}^+(\alpha(ps_1))$ (resp. $v \in \text{prec}^-(\alpha(ps_1))$), $v' \in \text{eff}^+(\alpha(ps_{k'})) \cup \text{eff}^-(\alpha(ps_{k'}))$, and a chain of causal links connecting them.

Due to space restrictions, we cannot include all the work related to formalisms that allow to specify preconditions and effects for compound tasks. Concerning the surveyed legality criteria, we restricted the presentation to approaches which *explicitly* mention the demanded criteria. We further want to mention the work by Castillo et al. [14], which also fuses HTN planning with POCL planning. Since they infer the methods automatically, their plans also fulfill certain criteria, which seem to be closely related to Def. 8. Further attention deserves the *angelic semantics* by Marthi et al. [30]. Their conditions (“high-level action descriptions”) take all states into account that are generated by any primitive plan that is reachable by decomposing the respective compound task. In contrast, the legality

criteria surveyed before relate a compound task's preconditions and effects directly to its methods' plans (in particular to the preconditions and effects of its tasks).

4 Complexity Results

In this section we investigate the complexity of the plan verification and the plan existence problem for the hybrid planning formalism.

For some of our hardness results, we reduce a certain set of HTN planning problems to hybrid problems. Since all required results are proved or reproduced in the HTN planning framework by Geier and Bercher [19] (or based upon it), we first state a proposition that every HTN planning problem can be expressed as a hybrid planning problem with the same set of solutions and without violating any of the legality criteria for decomposition methods. Concerning notation, note that a *task network* (T, \prec, α) [19, Def. 1] in HTN planning is a special case of plans (cf. Def. 1), as task networks do not contain causal links. What we call plan steps is referred to as *tasks* T in HTN planning. We use both notations, depending on the context.

Theorem 1. *Let \mathcal{P} be an HTN planning problem according to Def. 2 by Geier and Bercher [19]. Then, \mathcal{P} can be transformed into a hybrid planning problem π according to Def. 2 that satisfies the legality criteria in Defs. 7, 8, 9, and 10, such that:*

1. **Refinement Correspondence.** *Let tn be a (not necessarily primitive) task network that can be obtained via decomposition in \mathcal{P} . Then, the plan P that is isomorphic to tn can be obtained via decomposition in π . Conversely, let P be a primitive plan that can be obtained via decomposition in π . Then, the task network tn that is isomorphic to P can be obtained via decomposition in \mathcal{P} .*
2. **Solution Correspondence.** *Let tn be a solution task network for \mathcal{P} . Then, for each executable task sequence t_1, \dots, t_n thereof there is a solution plan P to π containing exactly this task sequence. Conversely, let P be a solution plan for π . Then, for all linearizations \overline{ps} of the plan steps of P there is a task network tn that is a solution for \mathcal{P} , such that \overline{ps} is an executable linearization of the tasks in tn .*

Proof. Let $\mathcal{P} = (L, C, O, M, c_I, s_I)$ with L being a finite set of proposition symbols and C and O finite sets of compound and primitive task name symbols, respectively. Each primitive task name $o \in O$ has a unique operator $(prec(o), add(o), del(o)) \in (2^L)^3$ corresponding to an action without negative preconditions. M is a finite set of methods mapping compound tasks to task networks. $c_I \in C$ is the initial compound task name and $s_I \in 2^L$ the initial state.

Transformation. We transform the HTN planning problem \mathcal{P} into a hybrid planning problem $\pi = (V, N_c, N_p, \delta, M', P^i)$. We define $V := L$. The initial compound task c_I and the initial state s_I of \mathcal{P} are encoded in π by the initial plan $P^i = (PS^i, CL^i, \prec^i, \alpha^i)$. That is, $PS^i = \{ps_{init}, ps, ps_{goal}\}$, $CL^i = \emptyset$, $\prec^i = \{(ps_{init}, ps), (ps, ps_{goal})\}^*$, and $\alpha^i = \{(ps_{init}, init), (ps, c_I), (ps_{goal}, goal)\}$. The actions of *init* and *goal* are given by $\delta(init) = (\emptyset, \emptyset, s_I, V \setminus s_I)$ and $\delta(goal) = (\emptyset, \emptyset, \emptyset, \emptyset)$. We are now defining the tasks and methods of π in such a way that all methods in M' satisfy all legality criteria. We construct a model in which no compound task has preconditions or effects and all methods contain only compound tasks or at most one task. We do this by introducing an additional compound task o_{clone} for every primitive task $o \in O$ and replace all primitive tasks in every decomposition method's plan by the respective compound task. To ensure that this does not change the set of solutions, each of these

compound tasks o_{clone} has exactly one decomposition method with a plan containing exactly o : Let $N_c := C \cup \{o_{clone} \mid o \in O\}$ and for all $n_c \in N_c$, let $\delta(n_c) := (\emptyset, \emptyset, \emptyset, \emptyset)$. For the primitive tasks, let $N_p := O \cup \{init, goal\}$ and for all $o \in O$, let $\delta(o) = (prec(o), \emptyset, add(o), del(o))$. The methods M' are given by

$$M' := \{(c, (T, \emptyset, \prec, \alpha')) \mid (c, (T, \prec, \alpha)) \in M, \text{ with} \\ \alpha' := \{(t, c) \mid (t, c) \in \alpha, c \in C\} \cup \\ \{(t, o_{clone}) \mid (t, o) \in \alpha, o \in O\}\} \\ \cup \{(o_{clone}, (\{ps\}, \emptyset, \emptyset, \{(ps, o)\})) \mid o \in O\}$$

Legality. The downward compatibility (Def. 7) and legality criterion that requires chains of causal links (Def. 10) trivially hold for all methods, since none of the compound tasks have preconditions or effects (so, the methods' plans are not further restricted). The other legality criteria (Defs. 8 and 9) require further restrictions on the plans even if the (parent) compound task does not have preconditions or effects. It is easy to see that all these restrictions hold, since – by construction – plans only contain compound tasks or at most one task. In the first case, there are no preconditions and effects, hence all criteria hold. In the second, the preconditions or effects of the single task do not violate any of the criteria as well.

Refinement Correspondence. Let tn_1, \dots, tn_k be a sequence of task networks, such that $tn_1 = (\{t\}, \emptyset, \{(t, c_I)\})$, $tn_k = tn$, and any task network tn_j can be obtained from tn_{j-1} , $1 < j \leq k$, via decomposition. Let the corresponding sequence of decomposed task names be c_1, \dots, c_{k-1} . Since no compound task $c \in C$ was removed from any of the decomposition methods' task networks, and since none of them contains causal links, there is also a sequence of plans P_1, \dots, P_k , such that $P_1 = P^i$ and the plan P_j results from decomposing c_{j-1} in P_{j-1} , $1 < j \leq k$. The resulting plan P_k is isomorphic to the task network tn_k except that P_k contains a compound task $o_{clone} \in N_c \setminus C$ for any primitive task $o \in O$ in tn_k . Thus, using the methods for those $o_{clone} \in N_c \setminus C$, there is a sequence of decompositions that transform P_k into P'_k with P'_k being isomorphic to tn_k . For the other direction, let P be a primitive plan that can be obtained via decomposition in π . Because the decomposition of a compound task $o_{clone} \in N_c \setminus C$ does not introduce further compound tasks, we can assume that first tasks c_1, \dots, c_{k-1} , $c_i \in C$, $1 \leq i < k$ are decomposed leading to a plan P_k and then only tasks in $o_{clone} \in N_c \setminus C$ leading to a primitive plan P'_k . When decomposing c_1, \dots, c_{k-1} in \mathcal{P} , we obtain a task network tn_k that is isomorphic to P'_k .

Solution Correspondence. Let tn be a solution to \mathcal{P} . It is thus reachable via decomposition in \mathcal{P} . Due to the refinement correspondence, P being isomorphic to tn can be obtained via decomposition in π . Thus, for any executable linearization of the tasks of tn , P can be turned into a totally ordered solution plan P' containing exactly that sequence via ordering and causal link insertions. For the other direction, let P be a solution for π . Without loss of generality we can assume that P can be generated by first decomposing, then inserting causal links, then ordering constraints. Let P' be the last primitive plan before any ordering or causal link insertion. Due to the refinement correspondence, tn being isomorphic to P' is reachable via decomposition in \mathcal{P} . Then, tn contains all linearizations of the plan steps of P' (in particular the executable ones). \square

Plan Verification. We now investigate how hard it is to verify whether a given plan is a solution to a hybrid planning problem. While in classical, non-hierarchical planning, this question can be answered in linear time w.r.t. the size of the input plan [16,

Thm. 8], the corresponding problem is much harder in the HTN setting. Behnke et al. [9] proved that the HTN plan verification problem is **NP-complete** even under several restrictions.

We first investigate the special case where there is no hierarchy. In HTN planning, this means to decide whether a primitive plan has an executable linearization, which is already **NP-complete** [34, Thm. 15], [16, Thm. 8]. In hybrid planning, *all* linearizations need to be executable. The respective problem is hence equivalent to verifying whether a POCL plan is a solution to a POCL planning problem, which is commonly known to be tractable.

Theorem 2. *Let P be a plan and $\pi = (V, N_c, N_p, \delta, M, P^i)$ a hybrid planning problem without hierarchy, i.e., $N_c = M = \emptyset$. Deciding whether P is a solution to π is in **P**.*

Proof. Checking that every precondition is supported by exactly one causal link can be done in linear time w.r.t. the number of all preconditions and causal links. Checking the absence of causal threats can be done in quadratic time. Let $P = (PS, CL, \prec, \alpha)$. For each causal link $(ps, v, ps') \in CL$ iterate over all plan steps $ps'' \in PS$, $ps'' \notin \{ps, ps'\}$. If none of these ps'' has an effect conflicting with v and can be ordered between ps and ps' without violating \prec , continue to the next causal link, otherwise fail. \square

Please note that the reason why this verification problem is easier than in HTN planning cannot be attributed to the fact that compound tasks show preconditions and effects, but to the fact that in hybrid (and POCL) planning, all linearizations need to be executable, whereas HTN planning only requires that there exists one.

Next we consider the general case, in which there are no restrictions on the hierarchy. We start by showing **NP** membership.

Lemma 1. *Let P be a plan and π a hybrid planning problem. Independently of the demanded legality criteria (Def. 7 to 10), it is in **NP** to decide whether P is a solution to π .*

Proof sketch: For HTN planning, we showed that the corresponding verification problem is in **NP** [9, Thm. 1]. We show that the two main proof steps (not emphasizing some special cases due to lack of space) remain applicable despite the extension of the formalism to hybrid planning: First, we show that any plan that can be obtained via decomposition can be obtained by a polynomial number of methods. Second, we give a guess-and-verify algorithm that runs in **NP**.

The first main step consists of five sub steps. First, we construct a so-called ε -extended planning problem π' from π that has the same set of solutions as π , but allows for shorter decomposition sequences. We call methods that contain an empty plan ε -methods. Now, for any compound task name that can be transformed into an empty plan by an arbitrary number of decomposition methods (due to ε -methods already present in M), we introduce an additional ε -method in M' of π' . For HTN planning, this can be done in **P** [9, after Def. 1]. Since causal links are not allowed to disappear upon decomposition, the same result applies to hybrid planning. Second, given a sequence \bar{m}_1 of methods in M' leading from the initial plan to a plan P' , the methods are reordered as follows: all ε -methods immediately follow the method that inserted the plan step they erase into the plan, resulting in the sequence \bar{m}_2 . Note that reordering these decomposition methods is possible, since all legality criteria satisfy Def. 7, downward compatibility (cf. footnote on page 4). Third, if for any non- ε -method in \bar{m}_2 , all plan steps of its plan are thereafter erased, all those methods are replaced by one single ε -method resulting in a shorter sequence \bar{m}_3 . Let $\bar{P} = P_1, \dots, P_n$ with $P_1 = P^i$ and $P_n = P'$ be the corresponding sequence of plans. Its subsequence

\bar{P}' that consists of the plans to which non- ε -methods are applied and P' forms a sequence with non-decreasing plan step size. Due to plateaus (which can be caused, e.g., by so-called unit-methods, which decompose a compound task into a single other task), \bar{P}' can still be arbitrary long. Step four is a preparation to obtain a bound on their lengths: We reorder methods between the plateaus such that in every plateau only methods remain that decompose the plan step that is decomposed last in the respective plateau (as this step ends the plateau) resulting into \bar{m}_4 . As argued before, reordering is also possible in the hybrid planning setting. In step five we can now shorten the new sequence of plans corresponding to \bar{m}_4 . Every plateau in the new plan sequence can now be limited to at most $|N_c|$ plans, as otherwise cycles must occur. Because there are at most k plateaus (k being $|PS|$ of P'), we can state that the final sequence of methods \bar{m}_5 has at most $2k(|N_c| + 1)$ non- ε -methods^{||} and thus $2k(|N_c| + 1)\Delta$ methods in total, Δ being the maximal number of plan steps of the plans in the methods and P^i . We thereby conclude that if a plan P' can be obtained via decomposition in π , it can be obtained by a polynomial number of methods in π' .

For the second main step, we guess a polynomially bounded sequence of methods in π' and calculate the resulting plan P' . We then guess additional ordering constraints (bounded by k^2) and causal links (bounded by $k|V|$), insert them into P' resulting in P'' , guess a bijection between the plan steps in P'' and P and verify isomorphism. We then verify executability of P in **P** (Thm. 2). \square

We now show that the plan verification problem is also **NP-hard**.

Theorem 3. *Let P be a plan and π a hybrid planning problem. Independently of the demanded legality criteria (Def. 7 to 10), it is **NP-complete** to decide whether P is a solution to π .*

Proof. Membership is stated in Lem. 1. For hardness, we use a corollary of HTN plan verification [9, Cor. 5]. According to this, it is **NP-complete** to verify, given a sequence of tasks \bar{t} and a totally unordered precondition- and effect-free HTN problem \mathcal{P} , whether there is a solution task network tn , such that \bar{t} is an (executable) linearization of tn 's tasks. (Note that the complexity of the problem does not stem from finding an *executable* linearization, as there are no preconditions and effects, but from finding the right decompositions leading to the desired plan. The original proof reduces vertex cover to HTN plan verification.)

Let π be a hybrid planning problem that is constructed from \mathcal{P} with the properties stated in Thm. 1. Further, let P be a totally ordered plan containing \bar{t} as plan step sequence. If there is a solution tn of \mathcal{P} , such that \bar{t} is an executable linearization of tn , then we can conclude that P is a solution to π (Thm. 1). Conversely, if P is a solution to π , then there exists a solution tn to \mathcal{P} , such that P 's plan step linearization is an executable linearization of tn (Thm. 1). \square

Plan Existence. In general, HTN planning is **undecidable** [16, 19]. We now show that this also holds for hybrid planning.

Theorem 4. *Hybrid planning is undecidable. That is, it is undecidable to determine whether a hybrid planning problem has a solution – no matter, which of the legality criteria of Def. 7 to 10 hold.*

Proof. Since we can encode any HTN planning problem into a solution-conserving hybrid planning problem that satisfies all legality criteria (Thm. 1), we can reduce the undecidable plan existence problem for HTN planning [19, Thm. 1] to hybrid planning. \square

^{||}We previously stated a bound of only $|k|(|N_c| + 1)$ [9, Lem. 1], as we handled a special case wrong (details omitted due to space restrictions).

From this theorem we can conclude that hybrid planning is as expressive as HTN planning, since it allows to encode undecidable problems. However, HTN planning is also known to be semi-decidable (or recursively enumerable, **RE**) [16, Thm. 1], which implies that for any HTN planning problem, a solution can be eventually found if one exists (while the undecidability prevents one from proving – in general – that there is no solution in case there actually is none). We now show that this is also true for hybrid planning.

Theorem 5. *Hybrid planning is semi-decidable. That is, the set of all hybrid planning problems that possess a solution is in RE.*

Proof. We give a partial recursive function f that, given a hybrid planning problem π , returns *true* if π has a solution and that may not halt, otherwise. We define f as the algorithm that enumerates all plans and verifies whether they solve π . It may run infinitely long, but as soon as it finds a solution, f returns *true*. Although there are infinitely many plans, enumeration is possible by starting with all plans of length two (the only tasks of which are the artificial *init* and *goal* actions) and then successively incrementing plan length. For each plan, verify in **NP** whether it is a solution (Thm. 3). \square

As a further corollary from Thm. 1, it also follows that many sub classes of hybrid planning are as hard as the respective problem classes in HTN planning. Such restrictions include syntactical ones (such as totally ordered task networks [2] or delete-relaxed actions [5]) and structural restrictions on the hierarchy (such as tail-recursive or acyclic problems [2]). To formally prove this, we would have to show that the respective restrictions still hold in the hybrid planning problem after the translation process done in the proof of Thm. 1.

5 Discussion

As a corollary from the last section’s results, we can observe that for the studied legality criteria, allowing preconditions and effects for compound tasks does neither increase nor decrease the expressivity of the formalism with regard to the plan existence problem in the general case. We want to emphasize that this is caused by the fact that none of the studied criteria (Def. 7 to 10) enforces to specify preconditions or effects for compound tasks (such as the one by Russell and Norvig [37]). Legality criteria that enforce to specify such preconditions or effects might influence the respective results and therefore also reduce expressivity, as they might prevent to specify computationally hard problems. Having the *option* to model such preconditions and effects still serves several practically relevant purposes, however. We shortly discuss some of them in this section and give pointers to the literature for further details.

As argued by Fox [17, p. 196], “one of the strongest motivations for using some form of abstraction in planning is the observation that people use it to great effect in their problem-solving”, which is also backed up by psychological studies [13]. Consequently, people should already be supported during the process of constructing (hierarchical) planning domains. Modeling support has attracted increased interest during the last years, which is one of the reasons that lead to the establishment of the *International Competition on Knowledge Engineering for Planning and Scheduling (ICKEPS)***. Nevertheless, there is still only very little research to automatically support the modeling process, which is particularly true for hierarchical models. The tool by McCluskey and Kitchin [32] as well as GIPO [39] for hierarchical models expressed in the modeling language OCL_h

checks certain properties and reports violations. In analogy to the respective properties that these tools verify, the legality criteria allow to automatically verify the relationship between compound tasks and their methods’ plans. As Fox points out, “*abstract plans have intentional meaning*” – and so do compound tasks. Thus, adhering some desired legality criterion is a possible way to automatically verify whether the model complies with the user’s intent.

Apart from providing modeling assistance, another main purpose of being able to specify preconditions and effects for compound tasks is to exploit them during search. Reasoning about these preconditions and effects may result in a smaller search space, as irresolvable flaws, such as open preconditions, can be detected earlier, i.e., before the respective compound task is decomposed. This further allows to generate “solution” plans on different levels of abstraction. Such plans look like ordinary (primitive) solutions with the difference that some tasks are still compound. When the model fulfills further prerequisites, it is guaranteed that such abstract solutions can be refined into a primitive one [40, 30]. Then, the model is said to fulfill the downward refinement property [6].

Preconditions and effects of compound tasks can also improve plan explanations. Plan explanations as developed by Seegebarth et al. [38] give a justification about the purpose of a primitive action questioned by the user. It is based upon a sequence of arguments, each being of the form (a) “action a is required as it supports a precondition variable of another action a' by a causal link” or (b) “action a is required as it was introduced via decomposition of a compound task c ”. Necessity of the other task a' (resp. c) is proved similarly. Preconditions and effects of compound tasks now allow to combine these two argument types [38]. Then, the causal chain argument (a) can be extended from just primitive actions to compound tasks, as they show preconditions and effects as well, which allows for much shorter and more abstract explanations.

6 Conclusion

To finally answer the question whether compound tasks with preconditions and effects are more than just names: We can state *no* in the sense that for the criteria we studied in more detail, we were able to show that in the general case, the formalism is equally expressive (with respect to the plan existence problem) than the HTN formalism, in which compound tasks are just names. For many sub classes, however, we only showed lower bounds – upper bounds still need to be proved. It might also be that other, more restrictive, legality criteria influence the hardness of the problem, in which case we also had to state *yes*. We can already answer the question with *yes* with regard to practical considerations, such as modeling assistance: The preconditions and effects, when combined with a desired legality criterion, can be exploited to provide assistance to ensure that the methods comply with the user’s intent – or at least to rule out some of the modeling flaws.

ACKNOWLEDGEMENTS

We thank Thomas Geier for discussions that helped improving this paper as well as Kathi Krammer for proof reading an early draft. This work was done within the Transregional Collaborative Research Centre SFB/TRR 62 “Companion-Technology for Cognitive Technical Systems” funded by the German Research Foundation (DFG).

** <http://www.icaps-conference.org/index.php/Main/Competitions>

REFERENCES

- [1] Ron Alford, Gregor Behnke, Daniel Höller, Pascal Bercher, Susanne Biundo, and David Aha, 'Bound to plan: Exploiting classical heuristics via automatic translations of tail-recursive HTN problems', in *Proc. of the 26th Int. Conf. on Automated Planning and Scheduling (ICAPS)*, pp. 20–28. AAAI Press, (2016).
- [2] Ron Alford, Pascal Bercher, and David Aha, 'Tight bounds for HTN planning', in *Proc. of the 25th Int. Conf. on Automated Planning and Scheduling (ICAPS)*, pp. 7–15. AAAI Press, (2015).
- [3] Ron Alford, Pascal Bercher, and David Aha, 'Tight bounds for HTN planning with task insertion', in *Proc. of the 25th Int. Joint Conf. on AI (IJCAI)*, pp. 1502–1508. AAAI Press, (2015).
- [4] Ron Alford, Ugur Kuter, and Dana S. Nau, 'Translating HTNs to PDDL: A small amount of domain knowledge can go a long way', in *Proc. of the 21st Int. Joint Conf. on AI (IJCAI)*, pp. 1629–1634. AAAI Press, (2009).
- [5] Ron Alford, Vikas Shivashankar, Ugur Kuter, and Dana Nau, 'On the feasibility of planning graph style heuristics for HTN planning', in *Proc. of the 24th Int. Conf. on Automated Planning and Scheduling (ICAPS)*, pp. 2–10. AAAI Press, (2014).
- [6] Fahiem Bacchus and Qiang Yang, 'Downward refinement and the efficiency of hierarchical problem solving', *Artificial Intelligence*, **71**(1), 43–100, (1994).
- [7] Patrick Bechon, Magali Barbier, Guillaume Infantes, Charles Lesire, and Vincent Vidal, 'HiPOP: Hierarchical partial-order planning', in *Proc. of the 7th Europ. Starting AI Researcher Symposium (STAIRS)*. IOS Press, (2014).
- [8] Gregor Behnke, Daniel Höller, Pascal Bercher, and Susanne Biundo, 'Change the plan – how hard can that be?', in *Proc. of the 26th Int. Conf. on Automated Planning and Scheduling (ICAPS)*, pp. 38–46. AAAI Press, (2016).
- [9] Gregor Behnke, Daniel Höller, and Susanne Biundo, 'On the complexity of HTN plan verification and its implications for plan recognition', in *Proc. of the 25th Int. Conf. on Automated Planning and Scheduling (ICAPS)*, pp. 25–33. AAAI Press, (2015).
- [10] Pascal Bercher, Susanne Biundo, Thomas Geier, Thilo Hörml, Florian Nothdurft, Felix Richter, and Bernd Schattenberg, 'Plan, repair, execute, explain - how planning helps to assemble your home theater', in *Proc. of the 24th Int. Conf. on Automated Planning and Scheduling (ICAPS)*, pp. 386–394. AAAI Press, (2014).
- [11] Susanne Biundo, Pascal Bercher, Thomas Geier, Felix Müller, and Bernd Schattenberg, 'Advanced user assistance based on AI planning', *Cognitive Systems Research*, **12**(3-4), 219–236, (2011). Special Issue on Complex Cognition.
- [12] Susanne Biundo and Bernd Schattenberg, 'From abstract crisis to concrete relief – a preliminary report on combining state abstraction and HTN planning', in *Proc. of the 6th Europ. Conf. on Planning (ECP)*, pp. 157–168. AAAI Press, (2001).
- [13] Richard Byrne, 'Planning meals: Problem solving on a real data-base', *Cognition*, **5**, 287–332, (1977).
- [14] Luis A. Castillo, Juan Fernández-Olivares, and Antonio González, 'On the adequacy of hierarchical planning characteristics for real-world problem solving', in *Proc. of the 6th Europ. Conf. on Planning (ECP)*, pp. 169–180. AAAI Press, (2001).
- [15] Filip Dvořák, Arthur Boit-Monnot, Flix Ingrand, and Malik Ghallab, 'A flexible ANML actor and planner in robotics', in *Proc. of the 2nd Workshop on Planning and Robotics (PlanRob)*, pp. 12–19, (2014).
- [16] Kutluhan Erol, James A. Hendler, and Dana S. Nau, 'Complexity results for HTN planning', *Annals of Mathematics and Artificial Intelligence*, **18**(1), 69–93, (1996).
- [17] Maria Fox, 'Natural hierarchical planning using operator decomposition', in *Proc. of the 4th Europ. Conf. on Planning (ECP)*, pp. 195–207. Springer, (1997).
- [18] Maria Fox and Derek Long, 'Hierarchical planning using abstraction', *IEE Proc. – Control Theory Appl.*, **142**(3), 197–210, (1995).
- [19] Thomas Geier and Pascal Bercher, 'On the decidability of HTN planning with task insertion', in *Proc. of the 22nd Int. Joint Conf. on AI (IJCAI)*, pp. 1955–1961. AAAI Press, (2011).
- [20] Malik Ghallab, Dana S. Nau, and Paolo Traverso, *Automated Planning: Theory and Practice*, Morgan Kaufmann, 2004.
- [21] Daniel Höller, Gregor Behnke, Pascal Bercher, and Susanne Biundo, 'Language classification of hierarchical planning problems', in *Proc. of the 21st Europ. Conf. on AI (ECAI)*, pp. 447–452. IOS Press, (2014).
- [22] Daniel Höller, Gregor Behnke, Pascal Bercher, and Susanne Biundo, 'Assessing the expressivity of planning formalisms through the comparison to formal languages', in *Proc. of the 26th Int. Conf. on Automated Planning and Scheduling (ICAPS)*, pp. 158–165. AAAI Press, (2016).
- [23] Éric Jacopin, 'Game AI planning analytics: The case of three first-person shooters', in *Proc. of the 10th AI and Interactive Digital Entertainment Conf. (AIIDE)*, pp. 119–124. AAAI Press, (2014).
- [24] Subbarao Kambhampati, 'Refinement planning as a unifying framework for plan synthesis', *AI Magazine*, **18**(2), 67–98, (1997).
- [25] Subbarao Kambhampati and James A. Hendler, 'A validation-structure-based theory of plan modification and reuse', *Artificial Intelligence*, **55**, 193–258, (1992).
- [26] Subbarao Kambhampati, Amol Mali, and Biplav Srivastava, 'Hybrid planning for partially hierarchical domains', in *Proc. of the 15th Nat. Conf. on AI (AAAI)*, pp. 882–888. AAAI Press, (1998).
- [27] Craig A. Knoblock, 'Automatically generating abstractions for planning', *Artificial Intelligence*, **68**, 243–302, (1994).
- [28] Raphaël Lallement, Lavindra De Silva, and Rachid Alami, 'HATP: An HTN planner for robotics', in *Proc. of the 2nd Workshop on Planning and Robotics (PlanRob)*, pp. 20–27, (2014).
- [29] Naiwen Lin, Ugur Kuter, and Evren Sirin, 'Web service composition with user preferences', in *Proc. of the 5th Europ. Semantic Web Conf. (ESWC)*, pp. 629–643. Springer, (2008).
- [30] Bhaskara Marthi, Stuart J. Russell, and Jason Wolfe, 'Angelic semantics for high-level actions', in *Proc. of the 17th Int. Conf. on Automated Planning and Scheduling (ICAPS)*, pp. 232–239. AAAI Press, (2007).
- [31] David McAllester and David Rosenblitt, 'Systematic nonlinear planning', in *Proc. of the 9th Nat. Conf. on AI (AAAI)*, pp. 634–639. AAAI Press, (1991).
- [32] Thomas Lee McCluskey and Diane E. Kitchin, 'A tool-supported approach to engineering HTN planning models', in *In Proc. of 10th IEEE Int. Conf. on Tools with AI (ICTAI)*, pp. 272–279. IEEE, (1998).
- [33] Dana S. Nau, Tsz-Chiu Au, Okhtay Ilghami, Ugur Kuter, Dan Wu, Fusun Yaman, Héctor Muñoz-Avila, and J. William Murdock, 'Applications of SHOP and SHOP2', *Intelligent Systems, IEEE*, **20**, 34–41, (2005).
- [34] Bernhard Nebel and Christer Bäckström, 'On the computational complexity of temporal projection, planning, and plan validation', *Artificial Intelligence*, **66**(1), 125–160, (1994).
- [35] Santiago Ontañón and Michael Buro, 'Adversarial hierarchical-task network planning for complex real-time games', in *Proc. of the 24th Int. Conf. on AI (IJCAI)*, pp. 1652–1658. AAAI Press, (2015).
- [36] J. Scott Penberthy and Daniel S. Weld, 'UCPOP: A sound, complete, partial order planner for ADL', in *Proc. of the 3rd Int. Conf. on Principles of Knowledge Representation and Reasoning (KR)*, pp. 103–114. Morgan Kaufmann, (1992).
- [37] Stuart Russell and Peter Norvig, *Artificial Intelligence – A modern Approach*, chapter 12: Practical Planning, 367–376, Prentice-Hall, 1 edn., 1994.
- [38] Bastian Seegebarth, Felix Müller, Bernd Schattenberg, and Susanne Biundo, 'Making hybrid plans more clear to human users – a formal approach for generating sound explanations', in *Proc. of the 22nd Int. Conf. on Automated Planning and Scheduling (ICAPS)*, pp. 225–233. AAAI Press, (2012).
- [39] Ron M. Simpson, Diane E. Kitchin, and Thomas Lee McCluskey, 'Planning domain definition using GIPO', *The Knowledge Engineering Review*, **22**, 117–134, (2007).
- [40] Qiang Yang, 'Formalizing planning knowledge for hierarchical planning', *Computational Intelligence*, **6**(1), 12–24, (1990).
- [41] Robert Michael Young, Martha E. Pollack, and Johanna D. Moore, 'Decomposition and causality in partial-order planning', in *Proc. of the 2nd Int. Conf. on AI Planning Systems (AIPS)*, pp. 188–193. AAAI Press, (1994).

A Probabilistic Logic Programming Approach to Automatic Video Montage

Bram Aerts and Toon Goedemé and Joost Vennekens¹

Abstract. Hiring a professional camera crew to cover an event such as a lecture, sports game or musical performance may be prohibitively expensive. The CAMETRON project aims at drastically reducing this cost by developing an (almost) fully automated system that can produce video recordings of such events with a quality similar to that of a professional crew. This system consists of different components, including intelligent Pan-Tilt-Zoom cameras and UAVs that act as “virtual camera men”. To combine the footage of these different cameras into a single coherent and pleasant-to-watch video, a “virtual editor” is needed. This paper describes the development of such a component. We adopt a declarative approach, in which we build a model of the decision process that a human editor might follow to edit a video. To achieve a montage that obeys various cinematographic rules while at the same time retaining a natural, non-mechanical feel, we construct this model in a Probabilistic Logic Programming language. We demonstrate that the resulting system can be run in real-time and that it delivers montages that are almost indistinguishable from those made by a professional editor.

1 INTRODUCTION

Events such as lectures, sports games, musical performance, etc. are often recorded on video. When produced by a professional video crew, these recordings are typically of high quality, but also quite expensive. Because good-quality cameras are no longer overly expensive, organizers may be tempted to instead produce a “home-made” recording of their event, filmed by amateurs. However, while this option is much cheaper, the end results tend to be noticeably lower quality than those produced by a professional crew.

The research described in this paper fits within the CAMETRON project. The goal of this project is to fill the gap between videos produced by a professional crew and home-made productions, by creating a system to produce recordings whose quality approaches that of professionally produced video, but without the price tag. It will rely on an almost completely automatic system, thereby eliminating the expensive personnel cost. Its focus is on producing full-length reports of events, i.e., our goal is to produce videos that span the entire lengths of the event itself, rather than a summary or a set of highlights.

The CAMETRON system will consist of two separate components, mimicking the roles found in a typical film crew: virtual “camera men” (consisting of fixed cameras, pan-tilt-zoom cameras and cameras mounted on UAVs, together with the software to operate them) will capture the footage, while a virtual “editor” combines footage of the different cameras. The ongoing development of the first component has been described elsewhere [11, 5]. In this paper,

we present the second component: a virtual editor system, that is able to produce a single coherent, qualitative video from a number of different feeds of raw footage from the same event.

Creating a video that is both interesting and easy-to-follow is not a straightforward task. Human editors typically follow a number of different—and sometimes contradictory—cinematographic “rules” to accomplish this task. To develop our virtual editor, we will follow a declarative approach, in which we explicitly represent these rules. This approach has the benefit that it offers a great deal of flexibility in deciding which rules should be taken into account and how they should take priority over each other. An additional advantage is that it also allows us to reuse the same knowledge to perform different tasks: we cannot only use the rules to generate a montage, but also to evaluate the quality of a given montage or to learn certain properties of good montages from given examples.

To represent the rules, we need a suitable knowledge representation language. A particular challenge in this application is that cinematographic rules are not strict: they are guidelines that are typically followed, but not always. Indeed, the rules may sometimes contradict each other, and even if they do not, a human editor may still choose to ignore a rule, simply because the result “feels” better. A virtual editor should therefore not rigidly follow the rules, but it should sometimes deviate from them in order to give the montage a more interesting and natural flavour, thereby mimicking the creativity of a human editor. For this reason, we have chosen to make use of a Probabilistic Logic Programming (PLP) language, which allows us to represent these rules in a non-deterministic way. This has the additional benefit that—just like a human editor—the system is able to produce different montages from the same input streams.

In this paper, we restrict attention to videos of relatively simple settings, like a lecture, a sports event or a concert. More challenging settings, such as fiction, are left for future work. Our work differs from existing approaches in various ways. First, unlike techniques that construct summaries of an event’s highlights [9, 7], our method produces a video of the complete event. Second, unlike approaches that are developed for a specific setting [13, 18, 8], we do not assume a fixed number of cameras, a fixed camera setup or a particular subject matter. Third, our system assumes as input only video footage, in contrast to, e.g., the virtual editor systems that are present in game design and which use a 3D model of the scene [3, 10]. Fourth, our system is able to make the required editing decisions in real-time (for 25fps video), unlike, e.g., [4, 12]. Fifth, we incorporate variety of different cinematographic rules into our system and demonstrate that the resulting montages are almost indistinguishable from those produced by a professional editor. Existing work either does not perform such validation [12, 18, 8] or appears to have worse results than our method [4, 13]. Finally, systems such as [4, 12] are determinis-

¹ KU Leuven, Technology Campus De Nayer, Research Group EAVISE, Belgium, email: {b.aerts, toon.goedeme, joost.vennekens}@kuleuven.be

tic, in the sense that they always produce the same montage for the same video stream. The fact that our approach is more flexible in this respect may prove useful if the user for some reason does not like the montage initially proposed by the system.

The remainder of this paper is structured as follows. We describe Problog, the PLP system that we will use, in Section 2. In Section 3, we discuss a number of cinematographic rules. Section 4 describes our editing system in detail, in Section 5 we describe how this system can be used. Section 6 discusses the computational performance of this system, while Section 7 investigates the quality of its output. In Section 8, related work is discussed in more detail. We conclude in Section 9.

2 PRELIMINARIES: CP-LOGIC AND THE PROBLOG SYSTEM

CP-logic [20] is an expressive PLP language, based on Sato’s distribution semantics [19]. A theory in CP-logic consists of a set of rules of the following form:

$$\alpha_0 :: A_0 \ ; \ \dots \ ; \ \alpha_n :: A_n \quad :- \ \phi. \quad (1)$$

Here, ϕ is a formula (typically, a conjunction of literals), the A_i are atoms and the α_i are probabilities with $\sum_i \alpha_i \leq 1$. Each such rule represents a non-deterministic causal mechanism: the body ϕ triggers a non-deterministic event which causes at most one of the A_i ; for each i , α_i is the probability that A_i is the outcome of this event.

We first consider only the ground case, in which no variables are allowed. In this case, the formal semantics of CP-logic can be characterized in terms of the well-founded semantics for normal logic programs as follows. Suppose that, for a rule of form (1), we probabilistically either replace this rule by one of the rules $A_i :- \phi$, each with probability α_i , or we remove this rule altogether with probability $1 - \sum_i \alpha_i$. If we do this independently for each of the rules r in a CP-logic theory T , then we probabilistically reduce T to a normal logic program P . By π_T , we denote the resulting probability distribution over these normal logic programs P . The formal semantics of the CP-logic theory is then the probability distribution P_T over possible worlds (i.e., Herbrand interpretations) that is defined by $\pi_T^*(I) = \sum_{P=I} \pi_T(P)$; here, the sum is taken over all normal logic programs P that have the interpretation I as their well-founded model. These semantics are only well-defined for theories T that have the property that all logic programs P that can be produced from them (with non-zero probability) have a two-valued well-founded model (which is then also the unique stable model of P). CP-logic disallows theories that do not have this property.

A small example is the following CP-logic theory T_{SB} . It describes two children, Billy and Suzy, who each may decide to throw a rock at a bottle. Both children throw with 75% accuracy.

```
0.5::throws(billy).
0.6::throws(suzy).
0.75::breaks :- throws(billy).
0.75::breaks :- throws(suzy).
```

According to this theory, for instance $\pi_{T_{SB}}^*(breaks) = 0.5 * 0.6 * 0.75 + 0.5 * 0.4 * 0.75 + 0.5 * 0.6 * (1 - (1 - 0.75)^2) = 0.65625$.

This semantics is extended to the non-ground case by viewing a non-ground rule as a template for the set of all of its ground instantiations. This approach is identical to how variables are treated in Answer Set Programming [14, 15]. It requires all rules to be finitely

groundable. This can be ensured by disallowing function symbols and requiring that all variables must appear in a positive body atom.

The above example is therefore equivalent to:

```
0.5::throws(billy).
0.6::throws(suzy).
0.75::breaks :- throws(X).
```

The Problog system [16] is a state-of-the-art inference system for CP-logic. It supports such inference tasks as querying the probability $\pi_T^*(A)$ of an atom A and sampling a particular interpretation I according to this distribution π_T^* . Its input language extends CP-logic with a number of features that make it easier to write Prolog-style programs. In particular, it allows lists, the use of predicates such as `findall` and the use of variables as probability labels in the head. In the remainder of this paper, we will use this input language.

The small example given above can also be tried out in the online Problog inference engine: <http://tinyurl.com/jqdt1rm>

3 CINEMATOGRAPHIC MODEL

Whether they are movies, documentaries, lecture recordings, or any other form of edited video, all compositions tend to follow some generally accepted cinematographic rules. These rules exist to avoid confusing the viewer. While more experimental movies may deliberately violate these rules to shock or confuse the viewer, they are typically obeyed in the kind of objective event reports that form the focus of this paper. Indeed, following these rules leads to informative, pleasant-to-watch reports, that do not confuse or distract the viewer.

Cinematographic rules can be divided in two main groups. The first group are rules that need to be taken into account while filming. These rules concern concepts such as depth of field, rule of thirds, headroom and the 180 degrees rule as described in e.g. [11]. These rules are outside the scope of this paper, because they need to be taken into account by the virtual “camera men”, not the virtual editor. We therefore assume that the raw video feeds satisfy these rules.

The second group of rules concern the ways in which multiple video streams should be put together. These are the focus of our editing system. Before discussing these rules, we define some terminology. A *shot* is a sequence of successive frames from the same camera. A *cut* is a transition from one shot to the next. A *montage* is a sequence of shots. Figure 1 shows a montage composed of shots taken from different video streams. Note also that, at any particular point in time, some of the cameras may not be producing footage.

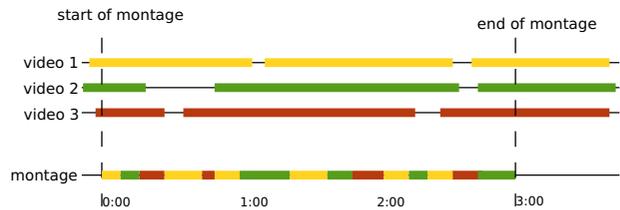


Figure 1. Making a montage out of video streams.

Length of shots. A shot should neither be too long nor too short. If it is too long, the viewer will get bored and his attention will drift. On the other hand, when shots are too short, the video becomes too jumpy and the viewer is presented with too much information in too little time. In general, shot lengths between 7 and 90 seconds are considered acceptable. However, the length of a shot also depends

on the amount of action in the scene: in energetic scenes, shorter shots can accentuate the ongoing action, while longer shots are more suitable for static scenes. For instance, a lecture recording will tend to have longer shots than a recording of a basketball game.

In order to prevent the montage from appearing too mechanical or predictable, there should also be some variation in the shot length. Instead of striving towards some “ideal” shot length, we should therefore make use of different shot lengths, within the bounds of what is neither too long nor too short.

Order of shots. Certain transitions between shots work well, while others should be avoided. When two consecutive shots mismatch, this is referred to as a *jump cut*. Jump cuts can occur when there is either too much or too little difference between the shots. When the footage of two different cameras is almost, but not completely the same, a “jumpy” effect occurs when switching from one to the other. As a rule, there should be a minimum angle of 30 degrees between the view points of two different cameras in order to avoid these kinds of jump cuts.

On the other hand, when the footage of two different cameras is completely different, the viewer will feel lost when switching from one to the other, because he has no clue about the context of the new subject. A quite common order of shots is therefore to start with a *long shot*, that establishes an overview of the scene. The next shot is then typically a *medium shot*, that shows a closer picture of, e.g., a particular person in the scene. Then, a *close-up shot* of the person’s face could follow. The close-up can then be followed up with another overview or medium shot.

Although this order of shots is often pleasing, it is not necessarily always followed. Indeed, as with the shot length, there should also be some variation in the sequence of different kinds of shots in order to avoid montages that appear too mechanical.

Continuity. To ensure coherence between shots, continuity should be respected. Cutting between disparate locations should be avoided, because it violates spatial continuity. When multiple cameras are filming one scene, all actions taking place in this scene should appear fluid and continuous. In other words, when switching between cameras, footage before and after the switch should be contiguous. This principle is called temporal continuity.

In this paper, our goal is to record footage of an event that takes place in a single location. We can therefore assume that spatial continuity is satisfied by our camera setup. In addition, our goal is also to cover the full length of the event (as opposed to summarizing the highlights). Therefore, temporal continuity will be satisfied because subsequent shots in the video will correspond to subsequent events in real life.

Action and reaction. Video coverage of an event should capture all the relevant action. When filming an action, three separate phases are important: premeditation, the action itself and the reaction. First, whenever an action is about to take place, there is a brief moment in advance when the person is thinking about undertaking this action. To prepare the viewer for the action that is about to take place, showing this premeditation is important. Second, showing the action itself is of course the most important thing. Third, when the action is complete, the viewer expects to see the reaction of a person to it.

4 OVERVIEW OF THE EDITING SYSTEM

An overview of our editing system is shown in Figure 2. It takes as input a number of different video streams, together with an analysis of each of these streams. This analysis is performed by computer vision algorithms, which have been described elsewhere [11, 1, 2, 6]. For each frame in each stream, we expect these algorithms to provide the following information:

- Whether people are present in the shot;
- Which kind of shot (long shot, medium shot, ...) it is;
- Which person is the most prominent subject of the shot;
- Which action (walking, talking, ...) the main subject performs.

The goal of our editing system is to decide for each point in time which of the available camera feeds will be used. The output of the system is the single video stream that is thus constructed.

The Editing system we propose consists of 3 components. The first component is the preprocessor, which comes in after the computer vision. This is a program written in python, which synchronizes camera footage, cleans up noisy detection data and parses this data to a format readable by problog. The problog core works out a montage, taking into account the cinematographic rules described in 3. After that, another python program edits the original video streams into one final montage, in the order the problog core calculated.

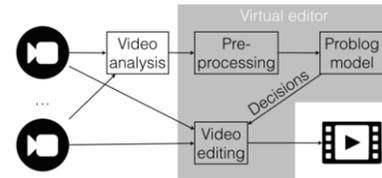


Figure 2. Overview of the editing system.

4.1 Data Representation

We assume that all cameras are synchronized with respect to a global time line. This time line is divided into discrete frames. For each frame, we need to represent which kind of footage each of the available cameras is providing for that particular frame.

We represent camera sources as $camera(ID)$. When a new camera connects, or any active camera disconnects, it is possible to add or remove the corresponding ID. A connected camera does not necessarily produce useful footage at each point in time, e.g., a UAV does not produce footage while at its loading station. Valid video footage is represented as $frame(Frame, CameraID)$. Even when a camera is providing footage, this may not be usable if the camera is being adjusted. We denote this as $adjust(Frame, CameraID, Type)$ where $Type$ is either *zoom* or *focus*.

Valid video footage may contain people or not. The video analysis system assigns each person—or rather, group of pixels that might be a person—a detection score that indicates how likely this is to be an actual person. We assume that the detection with the highest score corresponds to the most prominent person in the frame. We identify this person by means of a $person(Frame, CameraID, PersonID)$ fact. When a person is detected, we make a distinction between the different shot types: long shot, medium shot or close-up. We represent this as $shot.type(Frame, CameraID, Type)$. In addition, the person may perform a specific action: stand still, walk, point, or talk. We

represent this as $action(Frame, CameraID, Action)$. For images in which no person is detected, we make a distinction between *overview* shots and *noperson* shots. An *overview* shot is one in which people are too small to be detected by the video analysis, whereas a *noperson* shot is one that actually does not contain any people. The video analysis may use knowledge of the position of the different cameras in the scene to try to distinguish between the two.

4.2 Cinematographic model

The editor system is based on a Prolog model of the decision process that a human editor might follow to produce a live montage. At each point in time (i.e., for each frame), the editor needs to decide which camera to use, based on the footage that he has previously used and the footage is currently available from the different cameras (and possibly also a small lookahead at future footage to avoid, e.g., switching to a camera that is about to stop producing footage).

For every time point T , the program decides which camera C to use at that time. This decision is recorded in a predicate $use_camera(T, C, Torig)$. The third argument records the starting time point $Torig$ of the shot that is ongoing at time T . This is merely for efficiency reasons, since we could also recompute $Torig$ by looking at the preceding $use_camera(T', C, _)$ atoms with $T' < T$.

We define $use_camera(T, C, Torig)$ by means of the following two rules. The value of this predicate depends on two decisions that the editor must make: first, whether to cut away from the ongoing shot and, if so, which other camera to switch to. The first decision is recorded by the predicate $change(T)$ and we will discuss below how it is made. The effect of not changing is of course simply that the ongoing shot continues.

```
use_camera(T, C, Torig) :-
    previous_camera(T, C, Torig),
    not(change(T)).
previous_camera(T, C, Torig) :-
    time(T), Tp is T-1, use_camera(Tp, C, Torig).
```

If the editor does decide to change at time T , it will pick at random one of the other cameras C that are good candidates to switch to at that particular time (as given by $change_candidate(T, C)$).

```
use_camera(T, C, T) :-
    time(T), change(T),
    findall(Cam, change_candidate(T, Cam), Cams),
    uniform(Cams, C, T).
```

The predicate $uniform(Cams, C, T)$ expresses that C is randomly selected, according to a uniform distribution, from the list $Cams$. The third argument T is present to ensure that, at different time points, a different camera may be selected from the same list. The definition of $uniform$ is as follows. In order to make a uniform selection from a list $[H | T]$ of length N , this predicate either selects the head H with probability $\frac{1}{N}$, or it performs a uniform selection from the list T of length $N-1$ with the remaining probability $1 - \frac{1}{N}$.

```
uniform(List, El, ID) :-
    length(List, L), uniform(List, L, El, ID).
uniform([H|T], N, H, ID) :-
    P is 1/N, s(P, [H|T], ID).
uniform([H|T], N, E, ID) :-
    P is 1/N, not(s(P, [H|T], ID)),
    NN is N-1, uniform(T, NN, E, ID).
P::s(P, _, _).
```

The predicate $change(T)$ contains the essence of our cinematographic model. In general, there are two reasons to change: we can either do so because we want to, or because we have to. The latter case occurs when the current camera no longer provides usable footage, either because it no longer produces any footage at all, or because it starts to zoom or refocus.

```
change(T) :- not(can_stay(T)).
can_stay(T) :-
    previous_camera(T, C, _),
    frame_ok(T, C).
frame_ok(T, C) :- frame(T, C), not(adjust(T, C, _)).
```

In addition to changing because we have to, we might also switch cameras because we want to. In general, this will occur if there is at least one camera whose footage we prefer over that of the current shot and if we have already met the minimal shot length requirement.

```
change(T) :-
    change_candidate(T, _), not(too_short(T)).
```

In order to decide whether the current shot is long enough, we of course need to know the current shot length. This is easily computed by means of the third argument of the use_cam predicate:

```
shot_length_until(T, Len) :-
    previous_camera(T, C, Torig), Len is T-Torig.
```

The minimal shot length is not a fixed number. Some shots are definitely long enough and some shots are definitely too short, but there is also a gray area, in which an editor might make a judgment call to cut away slightly sooner than he would like in order to, e.g., better capture the beginning of an action. For this reason, we define too_short in a probabilistic way, as shown in Figure 3.

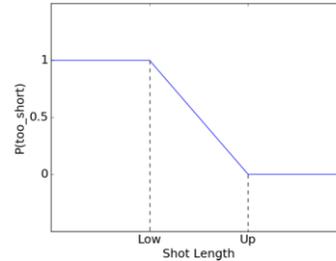


Figure 3. The probability of a shot being considered too short.

Here, Up is an upper bound on the minimal shot length (i.e., longer than Up is never too short), while Low is a lower bound (i.e., shorter than Low is always too short).

```
P::too_short(T) :-
    shot_length_until(T, Len),
    min_length_ub(Up), min_length_lb(Low),
    Len <= Up, P is min(1, (Up-Len) / (Up-Low)).
```

We similarly define a predicate too_long in terms of max_length_lb and max_length_ub . The effect of the current shot becoming too long is that this gives a reason to switch to a different camera:

```
P::too_long(T) :-
    shot_length_until(T, Len),
    max_length_ub(Up), max_length_lb(Low),
    Len >= Low, P is min(1, (Len-Low) / (Up-Low)).
should_switch(T) :- too_long(T).
```

We can now determine whether there exist cameras that it would be appropriate to switch to. Recall that if we find at least one such *change_candidate*, we will terminate the current shot unless it is still *too_short*. To make this decision, we divide the other cameras into three different categories: poor, fair and good change candidates. A good candidate is one that we always want to switch to, i.e., the existence of such a camera is in itself a reason to change. Poor and fair candidates are those that we might switch to if the current shot is getting too long and there is no better alternative available, i.e., we only consider switching to a fair candidate if there are no good candidates and we only consider switching to a poor candidate if there are neither fair nor good candidates.

```
change_candidate(T,C):-
  change_candidate(T,C,good).
change_candidate(T,C):-
  change_candidate(T,C,fair),
  should_switch(T),
  not(change_candidate(T,_,good)).
change_candidate(T,C):-
  change_candidate(T,C,poor),
  should_switch(T),
  not(change_candidate(T,_,fair)),
  not(change_candidate(T,_,good)).
```

A poor candidate is any camera to which it is possible to change to. If, in addition, the candidate camera provides a good transition from the camera that is currently in use, it is considered a fair candidate. Finally, a good candidate is a fair candidate that also has more interesting footage than the current camera.

```
change_candidate(T,C,good):-
  change_candidate(T,C,fair),
  better_frame(T,C).
change_candidate(T,C,fair):-
  change_candidate(T,C,poor),
  good_transition(T,C).
change_candidate(T,C,poor):-can_change(T,C).
```

The predicate *can_change* is similar to the predicate *can_stay* that was defined above, but it is more stringent. In addition to demanding that the camera is currently providing a usable frame (*frame_ok(T,C)*), *can_change* also demands that the camera will keep on providing usable frames in the near future. This is to prevent us from falling below the minimal shot length by cutting to a camera that is about to stop filming.

```
can_change(T,C1):-
  previous_camera(T,C,_),camera(C1),C\C1,
  future_ok(T,C1).
future_ok(T,C):-
  camera(C),min_length_lb(M),T2 is T+M,
  not(between(T,T2,T1),not(frame_ok(T1,C))).
```

As discussed above, the difference between a poor candidate and a fair one lies in the quality of the transition that can be achieved by switching from the current camera to the candidate.

```
good_transition(T,C):-
  previous_camera(T,Cp,_),Tp is T-1,
  shot_type(Tp,Cp,STp),shot_type(T,C,STcur),
  shot_transition(T,STp,STcur).
```

Here, *shot_transition(T,STp,STcur)* represents the fact that, at time *T*, it is a good idea to switch from shot type *STp* to shot

type *STcur*. As mentioned before, we do not want to impose a strict order in which the different shot types are used. For this reason, *shot_transition* will be a probabilistic predicate. It will allow all possible transitions in principle, but assign a higher probability to those transitions that are generally considered better.

```
P::shot_transition(T,From,To):-
  time(T),quality(From,To,P).
quality(ls,ls,0.8).
quality(ls,ms,1).
...
```

The first argument of the *shot_transition* predicate is needed to allow different choices to be made at different time points, i.e., it should be possible that a transition from *ls* to *ms* is considered a good idea at time point *t* but not at $t' \neq t$. The *quality* predicate defines the probability that each kind of transition is considered a good idea, as shown in Table 1.

Table 1. The probabilities of different shot transitions.

From	To				
	ls	ms	cu	os	np
long shot	0.8	1	0.2	0.8	0.1
medium shot	1	1	0.6	0.8	0.1
close up	0.8	0.7	0.2	0.8	0.1
overview shot	0.8	0.7	0.1	0.8	0.1
no person	1	1	0.2	1	0.1

The difference between a fair candidate and a good candidate is that the latter not only offers a good transition but also provides more interesting footage. This will mainly be determined by the actions that are taking place in the footage. Again, in order to avoid giving the montage a mechanical feel, we do not place a strict priority on the different kinds of actions that might occur. Instead, we assign the different kinds of actions a probability that they will be selected in the video. Selecting an action can only happen at the time point when it starts. The video analysis module detects the actions of talking, walking and pointing. We also introduce a special action, called *emerging*, that we assign to a person who newly appears in the footage.

```
0.55::select_action(Taction,C,emerge):-
  start_person(Taction,C).
0.80::select_action(Taction,C,talk):-
  start_action(Taction,C,talk).
0.55::select_action(Taction,C,walk):-
  start_action(Taction,C,walk).
0.65::select_action(Taction,C,point):-
  start_action(Taction,C,point).
start_action(T,C,AT):-
  action(T,C,AT),Tprev is T-1,
  not(action(Tprev,C,AT)).
start_person(T,C):-
  person(T,C,ID),Tprev is T-1,
  not(person(Tprev,C,ID)).
```

As explained before, it is important to not just show the action itself, but also the “premeditation” leading up to it. The length of this lead-in may depend on the action in question. Moreover, as with shot lengths, there is a window of acceptable options, rather than a single fixed value. We define these windows for the different actions as follows:

```
pre_action_window(merge, 0, 1).
pre_action_window(talk, 2, 10).
pre_action_window(walk, 0, 2).
pre_action_window(point, 0, 5).
```

If an action AT starts at time T_{action} in the footage of camera C , then we can cut to camera C at any time point in the interval $[T_{action} - Max, T_{action} - Min]$, where $[Min, Max]$ is the pre-action window of action AT , as defined by the above predicate $pre_action_window(AT, Min, Max)$. For instance, if a *talk* action is initiated at time point 20, we can cut at any time point in the interval [10,18]. However, in doing so, we should take care that whenever we decide to switch at time T_{switch} , the camera actually produces a stream of valid frames between T_{switch} and T_{action} . For instance, if the camera produces no frames between time point 14 and 16, we should not yet switch to it at time point 12.

Starting from time point T , the following predicate gathers into a list all the time points $T' \leq T$ such that camera C has produced only valid frames between T' and T . We do not go back arbitrarily far in the history, but only look Len frames back.

```
valid_since(T,C,Len,[]) :-not(frame_ok(T,C)).
valid_since(T,C,-1, []).
valid_since(T,C,Len,[T|Tail]) :-
    Len >= 0, frame_ok(T,C),
    Tp is T-1, Newlen is Len-1,
    valid_since(Tp,C,Newlen,Tail).
```

We can now use this predicate to gather into a *List* all time points at which we could possibly switch to a camera C that shows action AT starting at T_{action} .

```
valid_action_window(Taction,C,List,AT) :-
    pre_action_window(AT,Min,Max),
    T is Taction-Min, Delta is Max-Min,
    valid_since(T,C,Delta,List).
```

If we now switch to camera C at any time point $T_{switch} \in List$, we will have good footage from T_{switch} up to the time point T_{action} when action AT starts. In addition to this, we also want to avoid showing an action that does not last long enough for the viewer to make sense of it. We therefore also require that the person doing the action remains visible for a minimal amount of time after the start of the action.

```
action_visible(Taction,C,AT) :-
    minimal_length(M), Tmin is Taction+M-1,
    not(between(Taction,Tmin,T)),
    not(frame_ok(T,C), person(T,C,PID)).
```

If the starting action that we have selected (with *select_action*) satisfies both of these conditions (i.e., before and after the starting time of the action there is a sufficient amount of valid footage), then we can select one of its possible switching times (as given by the *valid_action_window* predicate) as a time to cut to the camera in question.

```
better_frame(T,C) :-
    select_action(TAct,C,Act),
    action_visible(TAct,C,Act),
    valid_action_window(TAct,C,List,Act),
    uniform(List,T,TAct).
```

In addition to switching to a different camera because it is starting an interesting new action, we might also switch because the footage of the current camera has become less interesting than the footage of another. We consider footage showing a person to be typically more interesting than footage not showing a person, and footage in which someone is talking more interesting than other footage.

```
0.55::better_frame(T,C) :-
    previous_camera(T,Cp,_),
    not(any_person(T,Cp)), any_person(T,C,_).
any_person(T,C) :- person(T,C,_).
0.60::better_frame(T,C) :-
    previous_camera(T,Cp,_),
    not(action(T,Cp,talk)), action(T,C,talk).
```

This concludes the usual decision process of the editor. A special case that we have not yet discussed is the very first time point of the montage. Here, we prefer an overview shot. Failing that, we might choose a long shot, medium shot, close-up, or, as a final resort, a no-person shot. We record this preference in a list and gather the list of all possible candidates, taking these preferences into account. Then we select randomly one of the candidates.

```
initial_priority([os, ls, ms, cu, np]).
initial_candidate(T,C) :-
    initial_priority(List), T2 is T+1,
    initial_candidate(T2, C, List).
initial_candidate(T, C, [First|Tail]) :-
    shot_type(T,C,First), future_ok(T,C).
initial_candidate(T,C,[First,Second|Tail]) :-
    not(shot_type(T,C,First), future_ok(T,C)),
    initial_candidate(T,C,[Second|Tail]).
use_camera(T,C,T) :-initial_time_point(T),
    findall(Cam,initial_candidate(T,Cam),Cams),
    uniform(Cams,C,T).
```

5 USING THE CINEMATOGRAPHIC MODEL

The Problog model of the previous section describes the decision process that an editor might follow in order to produce a montage in accordance with cinematographic rules. This model defines a probability distribution over all possible ways in which a number of given input streams can be edited into a single video. It can be used to perform different tasks.

By *querying* the probability of a given montage according to this distribution, we can obtain an estimate of the quality of this montage: highly probable montages satisfy many of the rules, whereas unlikely montages contain many “violations”. *Sampling* from this distribution will produce a single montage. This is of course the task that we focus on in this paper. Because the probability distribution defined by the Problog program assigns a higher probability to montages that respect more of the cinematographic rules, sampling is more likely to produce a good montage than a poor one. If we compute different samples for the same input, we will obtain a different montage each time. We view this as a desirable property, because it corresponds to how human editors perform their task: it is unlikely that a human editor would produce precisely the same output each time, if he were asked to edit together the same input streams multiple times. Capturing this variance in our virtual editor helps to ensure that our montages have a natural feel.

An alternative to sampling is to compute the *most likely* montage, given a set of input streams. However, such an approach would have significant disadvantages when compared to the sampling approach:

- The task of computing the most likely outcome is computationally significantly harder than that of sampling from a distribution.
- To compute the most likely montage, we need to consider the entire time line at once. Such an approach would therefore only be usable in an offline editing system. By contrast, as we will discuss in Section 6, sampling can be used to implement a system that edits input streams in (slightly delayed) real-time.
- Under the same circumstances, the most likely montage will always make the same choices (namely, it will choose the most likely alternative). By contrast, sampling will occasionally make a less likely choice, thereby producing a less mechanical and more interesting montage.

However, a disadvantage of the greater variation allowed by the sampling approach is that it may occasionally produce a poor montage. To compensate for this, our editing system will construct a set of samples. From this set, the most likely sample will be returned as the output of our system. By making this set larger, we increase the computational cost of our method, but reduce the risk of returning a poor montage. Our current implementation uses 10 samples.

In addition to *querying* and *sampling*, the Problog system also contains algorithms that perform the task of *parameter learning*. Currently, the probabilistic parameters of our Problog model have been manually filled in, based on our understanding of cinematographic rules. An alternative approach would be to derive these parameters automatically from a number of examples videos. In this case, we might also be able to train our model to emulate particular editing styles. We will investigate this topic in future work.

6 ACHIEVING REAL-TIME PERFORMANCE

In order to create a montage, our editing system calls on Problog to compute a number of different samples according to the probability distribution defined by the cinematographic model. There are two main parameters that affect the computational performance of the sampling algorithm: the number of time points and the number of cameras. Indeed, for each time point T and camera C , the sampling procedure must decide whether to switch to C at time T . We expect that the number of cameras will not vary greatly and will be dictated to a large extent by the setting in which the video must be produced. By contrast, the number of time points is a more flexible parameter. Indeed, the granularity of the time line used by our Problog model need not coincide with that of the actual input video. In other words, it is not necessary that each frame of the incoming video streams corresponds to an individual time point in the Problog model.

We will group together a number of actual frames into a sequence that we call a *pframe*. These pframes will act as time points for the Problog model. The number of real frames that go into one pframe is therefore a key parameter of our system. On the one hand, this number affects the performance of the sampling algorithm: the higher it is, the faster execution will be. On the other hand, this number also affects the quality of the montage that is produced, because our system will only be able to switch cameras at the start of a new pframe. Therefore, the smaller the pframes are, the more fine-grained this decision process will be and the higher the quality of the output.

Figure 4 shows the execution time needed to compute 10 samples in function of the length of a single pframe. The y-axis expresses the execution time as a fraction of the length of the video stream that needs to be edited. If this value is ≤ 1 , we have a system that is able to make the required editing decisions in real-time. The figure shows a graph for both a 3-camera and a 6-camera setup. In both cases,

taking a pframe to be equal to a single real frame ($= 0.04s$, since the video was shot at $25fps$) produces a runtime that is much too slow. However, as the length of a pframe increases, the runtime drastically decreases. For the 3-camera setup, we reach real-time performance as soon as at least 6 frames are combined in a single pframe. With 6 cameras, we need 10 frames per pframe. Because it is unlikely that a viewer would be able to tell the difference between cutting 0.4s earlier or later in a montage, we consider this an acceptable way of reaching real-time performance.

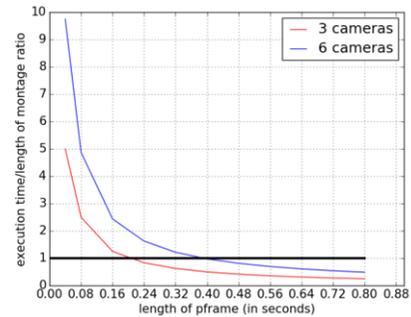


Figure 4. Execution time vs. length of pframes (in seconds)

7 EXPERIMENTAL RESULTS

Our system is able to produce real-time edits of different video streams. The only question remaining is whether the resulting montages are of a quality similar to those produced by professional editors. Since there is no single right way to edit video, we have no “ground truth” to compare the output of our system to. Instead, we have subjected the virtual editor to a “Turing test”: we have asked a number of test subjects to distinguish between the output of our system and a professionally made montage of the same video streams.

The test case used in this experiment is a lecture recording made by three cameras. This footage was edited by a professional editor, who was present during the recording. From the entire video stream, we selected a fragment of three minutes in which there was a lot of “action”, namely one speaker introducing another speaker, who then came to the stage. We presented 58 students with two video clips: this particular fragment as edited by the professional and the same fragment edited from the same input streams by our system. The student were then asked to identify the clip produced by the professional. As an incentive, a small prize (worth around 75€) was given to a random student among those who guessed correctly.

The two video clips can be viewed online:

1. <https://www.youtube.com/watch?v=7vrfhzD4G0c>
2. <https://www.youtube.com/watch?v=Bz110YKGeI4>

In case the reader would like to perform this experiment himself: the professionally edited video is Montage i , where i is the 22nd digit in the decimal expansion of π .

Of the 58 students, 31 ($= 53\%$) correctly identified the professionally edited clip. This difference between this outcome and one that could be produced by random guessing is not statistically significant ($T[57] = 0.5$; $p = 0.6$), i.e., the data does not allow to reject the hypothesis that the subjects were unable to tell the difference between the professional editor and our system. In addition, the subjects were also asked to indicate (on a scale of 1 to 3) how confident they were in

their choice. The results are shown in Figure 5. Those students who guessed *incorrectly* were on average actually slightly *more* confident of their choice than those who guessed correctly, although again the difference is not statistically significant (a $\chi^2[2] = 1.69$ test provides a p-value of 43%).

We conclude that our editing system indeed provides a good approximation of the quality delivered by a professional editor for this particular case study of lecture recording.

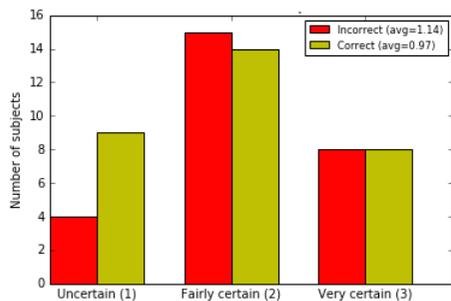


Figure 5. Confidence of subjects in their choice.

8 RELATED WORK

There exist several systems that perform automatic video montage for the purpose of creating a summary of some event(s). Examples in the scientific literature are [7, 9]. Also commercially available software offers this functionality, e.g., Muvee², Aescripts³, Magisto⁴ and Google Photos⁵. While related, this is essentially a different problem from the one that we have considered in this paper, where we want to produce full-length coverage of an event.

This problem of producing full-length coverage is also studied by [12]. Their approach also takes into account cinematographic rules regarding shot transitions and view selection. However, it requires about one second of computation time per frame and does not include an experimental validation of the quality of the produced video. In [17], a general scheduling algorithm is proposed to achieve optimal observability using multiple adjustable sensors. This algorithm is applied to the video editing problem in [4]. It requires cameras with (partially) overlapping field of views, and constructs a 3D-map with areas of higher interest. Quality of view is determined by amount of action, number of objects, visible events and a combined object score. The video montage is then made by maximizing this quality, while simultaneously minimizing inter-camera switching. This is a significantly different approach from ours, in which the cinematographic rules are not as explicitly present. The quality of videos created by this system was compared to that of professionally edited videos in an experiment similar to ours: 83% of their test subjects labeled the computer generated montage as professionally edited, while 93% of the test subjects labeled the professionally edited video as such. While this is not precisely the same experiment as we performed, this 10% difference strikes us as more significant than the 3% deviation from a 50-50 split that we observed. In addition, this method also requires 0.16s of decision making time per frame, which does not suffice to reach real-time performance at 25fps.

² <http://www.muvee.com/home>

³ <http://aescripts.com/automated-video-editing>

⁴ <https://www.magisto.com/how-it-works>

⁵ <https://photos.google.com>

Both of these methods reduce the video editing problem to an optimisation problem. On the one hand, this explains their greater computational complexity, while, on the other hand, it also means that these methods can only edit each particular set of input streams in one particular way. By contrast, our probabilistic sampling method is more flexible and may offer the user a number of different alternative montages to choose from.

In [13], an automatic video editor is developed specifically for lecture recording. Three cameras are used, each with a specific purpose: an overview camera, an audience facing camera and a lecturer tracker. This work includes cinematographic rules that are specific to this setting (e.g., if a person in the audience asks a question, show that person in the montage). The quality of the produced montages was tested by an experiment in which subjects were asked to assign a score of 1 to 5 to both an automatically produced and professionally made montage. The automatically generated video scored an average of 2.8, while man-made video had an average score of 3.7. Again, this difference of 22.5% is more significant than the differences we observed in our experiments. This system was later extended to allow a greater variety of settings [18]. This work also added some additional rules to the system, but their impact on the quality of the montages was not experimentally verified.

Another setting-specific system is that of [8], which performs off-line automatic video editing of a theater play. This system uses a setup with one static camera, from which a range of sub-views are cut to create multiple shots. The quality of the montages produced by this system was not experimentally verified.

9 CONCLUSIONS AND FUTURE WORK

This paper has presented an automated video editing system, which may play a role in reducing the cost of producing a video report of an event such as a lecture, sports game or musical performance. We have developed this system in a declarative way, by building a model of the non-deterministic decision process that a human editor could follow in order to produce a montage. By using the state-of-the-art Probabilistic Logic Programming system Problog, we are to sample from this distribution and thereby produce a montage. We have demonstrated that the computations needed to make all the required decisions can be performed in real-time and that—at least for the particular case study of lecture recording—the quality of the produced montage is almost indistinguishable from that produced by a professional editor.

In future work, we will examine the performance of this system in more complex settings than lecture recording and investigate how the same declarative model may be used for machine learning of the probabilistic parameters.

ACKNOWLEDGEMENTS

This work was funded by the KU Leuven Research Fund as part of the GOA project “CAMETRON”. The authors thank Dries Hulens for providing the video analysis module used in this work.

REFERENCES

- [1] Punarjay Chakravarty, Sayeh Mirzaei, Tinne Tuytelaars, et al., ‘Who’s speaking?: Audio-supervised classification of active speakers in video’, in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pp. 87–90. ACM, (2015).
- [2] Punarjay Chakravarty and Tinne Tuytelaars, ‘Cross-modal supervision for learning active speaker detection in video’, *arXiv preprint arXiv:1603.08907*, (2016).
- [3] David B Christianson, Sean E Anderson, Li-wei He, David H Salesin, Daniel S Weld, and Michael F Cohen, ‘Declarative camera control for automatic cinematography’, in *AAAI/IAAI, Vol. 1*, pp. 148–155, (1996).
- [4] Fahad Daniyal and Andrea Cavallaro, ‘Multi-camera scheduling for video production’, in *Visual Media Production (CVMP), 2011 Conference for*, pp. 11–20. IEEE, (2011).
- [5] F. De Smedt, D. Hulens, and T. Goedem, ‘On-board real-time tracking of pedestrians on a UAV’, in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops. Embedded Vision Workshop*, (2015).
- [6] Ali Diba, Ali Mohammad Pazandeh, Hamed Pirsiavash, and Luc Van Gool, ‘Deepcamp: Deep convolutional action & attribute mid-level patterns’, in *CVPR 2016, International Conference on Computer Vision and Pattern Recognition.*, (2016).
- [7] Yanwei Fu, Yanwen Guo, Yanshu Zhu, Feng Liu, Chuanming Song, and Zhi-Hua Zhou, ‘Multi-view video summarization’, *Multimedia, IEEE Transactions on*, **12**(7), 717–729, (2010).
- [8] Vineet Gandhi, Remi Ronfard, and Michael Gleicher, ‘Multi-clip video editing from a single viewpoint’, in *Proceedings of the 11th European Conference on Visual Media Production*, p. 9. ACM, (2014).
- [9] Andreas Girgensohn, John Boreczky, Patrick Chiu, John Doherty, Jonathan Foote, Gene Golovchinsky, Shingo Uchihashi, and Lynn Wilcox, ‘A semi-automatic approach to home video editing’, in *Proceedings of the 13th annual ACM symposium on User interface software and technology*, pp. 81–89. ACM, (2000).
- [10] Li-wei He, Michael F Cohen, and David H Salesin, ‘The virtual cinematographer: a paradigm for automatic real-time camera control and directing’, in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 217–224. ACM, (1996).
- [11] Dries Hulens, Toon Goedemé, and Tom Rumes, ‘Autonomous lecture recording with a ptz camera while complying with cinematographic rules’, in *Computer and Robot Vision (CRV), 2014 Canadian Conference on*, pp. 371–377. IEEE, (2014).
- [12] Hao Jiang, Sidney Fels, and James J Little, ‘Optimizing multiple object tracking and best view video synthesis’, *Multimedia, IEEE Transactions on*, **10**(6), 997–1012, (2008).
- [13] Qiong Liu, Yong Rui, Anoop Gupta, and Jonathan J Cadiz, ‘Automating camera management for lecture room environments’, in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 442–449. ACM, (2001).
- [14] V.W. Marek and M. Truszczyński, ‘Stable models and an alternative logic programming paradigm’, in *The Logic Programming Paradigm: a 25-Year Perspective*, eds., K.R. Apt, V.W. Marek, M. Truszczyński, and D.S. Warren, 375–398, Springer, Berlin, (1999).
- [15] I. Niemelä, ‘Logic programs with stable model semantics as a constraint programming paradigm’, *Annals of Mathematics and Artificial Intelligence*, **25**, 241–273, (1999).
- [16] L. De Raedt, A. Kimmig, and H. Toivonen, ‘ProbLog: A probabilistic Prolog and its application in link discovery’, in *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 2462–2467, (2007).
- [17] Mohammad Rezaeian, ‘Estimation entropy and optimal observability’, in *PerCom/PerSeNS conference*, (2006).
- [18] Yong Rui, Anoop Gupta, Jonathan Grudin, and Liwei He, ‘Automating lecture capture and broadcast: technology and videography’, *Multimedia Systems*, **10**(1), 3–15, (2004).
- [19] T. Sato and Y. Kameya, ‘PRISM: A language for symbolic-statistical modeling’, in *Proceedings of IJCAI*, (1997).
- [20] J. Vennekens, M. Denecker, and M. Bruynooghe, ‘CP-logic: A language of causal probabilistic events and its relation to logic programming’, *Theory and Practice of Logic Programming*, **9**(3), 245–308, (2009).

Checking the Conformance of Requirements in Agent Designs Using ATL

Nitin Yadav and John Thangarajah¹

Abstract. Intelligent agent systems built using the BDI model of agency have grown in popularity for implementing complex systems such as UAVs, military simulations, trading agents and intelligent games. The robust and flexible behaviours that these systems afford also makes testing the ‘correctness’ of these systems a non-trivial task. Whilst the main focus on existing work has been on checking the correctness of agent-programs, in this work we present an approach to formally verify agent-based designs for a particular BDI agent design methodology. The focus is on verifying whether the detailed design of the agents conform to the requirements specification. We present a sound and complete approach, formally verifiable properties, and an evaluation with respect to time and effectiveness.

1 Introduction

Intelligent agent systems have been used to develop software systems in a variety of application areas [19]. Agent systems of this kind are often designed and implemented in terms of software structures that are based on metaphors of humans and human societies; for example, events, beliefs, goals, plans and intentions. While there are many agent development paradigms, the *Belief-Desire-Intention* (BDI) model [24] is a mature paradigm that has been adopted by several agent development platforms such as JACK [28], JASON [8] and JadeX [7]. As with any software development, testing the ‘correctness’ of these BDI-based agent systems is an important task. See [20] for a recent survey of the state of the art in testing agent systems.

Most existing work on verifying BDI agent systems has focused on formal verification (e.g. [10]), particularly using model checking techniques (e.g. [13]) and theorem proving (e.g. [25]), or on runtime testing (e.g. [30, 31]) of *agent programs*. That is, testing the correctness after the system (or part of it) has already been implemented. However, the notion that identifying and correcting errors early in the software development cycle is well accepted in software engineering [6, Page 1466]. In this work we present a formal model-checking based approach for verifying the correctness of the agent design models at the *design stage* prior to implementation.

There has not been much work on testing the correctness of detailed agent designs with the exception of the recent work by Abushark et al. [1, 2]. They provide an approach for checking the correctness of interaction protocols [1] and requirement models [2]. Their approach in [1] is to extract all possible behaviour traces of the detailed agent design (comprising goals, plans and message exchanges) related to a particular protocol and report the ones that do not conform by checking against an execution structure of the protocol. They adapt a similar approach for checking requirements in [2].

Although their approach is able to identify traces that do not conform to the interaction or requirement specifications, there are some significant limitations. The first is that the approach has no formal semantics and as mentioned by the authors in [2] the approach is neither sound nor complete. Secondly, in the design, plans that post two or more sub-goals where the execution can be interleaved can create a large number of traces caused by the interleaving of all the steps of those sub-goals. As the number of parallel steps increases the possible behaviour traces grows at least exponentially and hence the time and space to extract them. This is particularly problematic for checking the requirements using scenarios, where a scenario specifies a particular sequence of steps. If the sequence specified is one of the possible parallel interleaving, all possible traces must be extracted to find the one that matches. Finally, the trace extraction is carried out independent of the requirements specification. Hence, no matter how long or short the scenario is, the number and size of traces is only dependent on the detailed design, which can be inefficient.

In this work, we present a formal approach to verifying the correctness of the detailed agent designs with respect to the requirement specifications via a model checking approach. Similar to the work of Abushark et al., we use Prometheus agent design models [22] as the basis of our work. The proposed approach however, is formal, sound and complete, uses model checking rather than trace extraction, and presents the designer with a model as well as traces. In addition, the approach is general and can be adapted to other agent design methodologies that follow the BDI model of agency [12] as they all share a set of common design concepts.

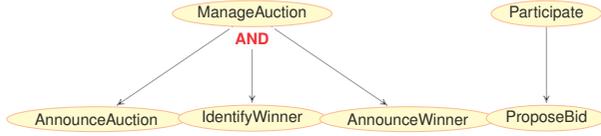
The key contributions of this paper are: (i) we formalise, rather informal and abstract, agent-oriented software design concepts; (ii) we provide precise semantics for the verification problem that we (and Abushark et al. [1]) are trying to address; (iii) we provide a formal design framework that is amenable to automatic testing and verification; (iv) we demonstrate how verification tools developed within the agent community can be used for checking conformance within agent designs; and (v) we provide an initial evaluation on the scalability of the proposed approach.

2 Background and related work

2.1 Prometheus- BDI agent design paradigm

The Prometheus methodology [22], together with the design tool (PDT) [23], supports the complete development of agent systems from specification and design through to implementation. The design methodology presents well-defined notation and processes for developing three key phases. The *System specification* where the interface of the system is specified in terms of inputs (percepts), outputs (actions), the external entities that interact with the system, and

¹ RMIT University, Australia, email: firstname.lastname@rmit.edu.au



(a) Goal overview for the auction example.

No	Type	Name	Role
1	Percept	StartAuction	Auctioneer
2	Goal	AnnounceAuction	Auctioneer
3	Goal	Participate	Bidder
4	Goal	IdentifyWinner	Auctioneer
5	Goal	AnnounceWinner	Auctioneer

(b) A scenario for the auction example.

Figure 1: Requirements spec. for the auction example.

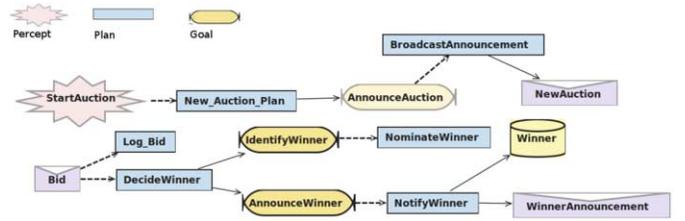
the requirements of the system specified via scenarios and goal diagrams. Scenarios specify a particular run of the system akin to use cases in traditional Object-Oriented design. Table 1b, illustrates an example scenario related to an auction system. The Goal diagrams specify the functionality of the system and how they may be decomposed into smaller sub-goals. Figure 1a, illustrates a goal diagram for an auction system.

The *Architectural design* specifies the internals of the system in terms of agents and any communication between them. The *detailed design* details the internals of each agent in terms of plans, messages, and goals that they handle and produce amongst other things. Figure 2, illustrates the detailed design of an Auctioneer agent, in a simple auction system. The plan `New_Auction` is triggered by the percept `StartAuction`, produces the (sub)goal `AnnounceAuction`, this goal then triggers the execution of the plan `BroadcastAnnouncement`.

2.2 Alternating-time temporal logic

Alternating-time Temporal Logic (ATL) [4] is a logic for reasoning about the ability of agent coalitions in *multi-agent game structures*. ATL formulae are obtained by combining propositional formulas, the usual temporal operators—namely, \bigcirc (“in the next state”), \square (“always”), \diamond (“eventually”), and \mathcal{U} (“strict until”)—and a *coalition path quantifier* $\langle\langle A \rangle\rangle$ taking a set of agents A as parameter. Conceptually, an ATL formula $\langle\langle A \rangle\rangle\phi$, where A is a set of agents, holds in an ATL model if the agents in A can *force* ϕ true, by choosing their actions, no matter how the agents *not* in A happen to move. The semantics of ATL is defined in concurrent game structures where, at each state, all agents simultaneously choose their actions from a finite set, and the next state deterministically depends on such choices. More concretely, a concurrent game structure is a tuple $\mathcal{M} = \langle \mathcal{A}, Q, \mathcal{P}, Act, d, \mathcal{V}, \sigma \rangle$, where $\mathcal{A} = \{1, \dots, k\}$ is a finite set of agents, Q is the set of states, \mathcal{P} is the set of propositions, Act is the set of all domain actions, $d : \mathcal{A} \times Q \mapsto 2^{Act}$ indicates all available actions for an agent in a state, $\mathcal{V} : Q \mapsto 2^{\mathcal{P}}$ is the valuation function stating what is true in each state, and lastly $\sigma : Q \times Act^{|\mathcal{A}|} \mapsto Q$ is the transition function mapping a state q and a joint-move $\vec{a} \in \mathcal{D}(q)$, where $\mathcal{D}(q) = \times_{i=1}^{|\mathcal{A}|} d(i, q)$ is the set of legal joint-moves in q , to the resulting next state q' .

A *path* $\lambda = q_0q_1 \dots$ in a structure \mathcal{M} is a, possibly infinite, sequence of states such that for each $i \geq 0$, there exists a joint-move $\vec{a}_i \in \mathcal{D}(q_i)$ for which $\sigma(q_i, \vec{a}_i) = q_{i+1}$. To provide semantics to formulas $\langle\langle \cdot \rangle\rangle\phi$, ATL relies on the notion of agent strategies. Techni-

**Figure 2:** Auctioneer agent - detailed design.

cally, an ATL *strategy* for an agent agt is a function $f_{agt} : Q^+ \mapsto Act$, where $f_{agt}(\lambda q) \in d(agt, q)$ for all $\lambda q \in Q^+$, stating a particular action choice of agent agt at path λq . A *collective strategy* for group of agents $A \subseteq \mathcal{A}$ is a set of strategies $F_A = \{f_{agt} \mid agt \in A\}$ providing one specific strategy for each agent $agt \in A$. For a collective strategy F_A and an initial state q , the set of all *possible outcomes* of F_A starting at state q , denoted $out(q, F_A)$, are the set of all computation paths that may ensue when the agents in A behave as prescribed by F_A , and the remaining agents follow any arbitrary strategy (see [4]). The semantics for the coalition modality is then defined as follows (here ϕ is a *path formula*; and $\mathcal{M}, \lambda \models \phi$ is defined in the usual way [4]):

$\mathcal{M}, q \models \langle\langle A \rangle\rangle\phi$ iff there is a collective strategy F_A such that for all computations $\lambda \in out(q, F_A)$, we have $\mathcal{M}, \lambda \models \phi$.

Given a concurrent game structure \mathcal{M} and an ATL formula ϕ , the *model checking problem* of ATL asks for the set of states in \mathcal{M} that satisfy formula ϕ . Let $[\phi]_{\mathcal{M}}$ denote the *maximal* set of states of \mathcal{M} that satisfy ϕ . A state q in \mathcal{M} is said to be *winning* for ϕ if $q \in [\phi]_{\mathcal{M}}$.

2.3 Related work

The BDI agent-oriented paradigm is a popular and successful approach for building agent systems. We have a long history (20+ years) of collaboration with multiple industry partners that use BDI agents and the Prometheus methodology (or its variants) to design the agents. The overarching goal in this paper is to provide our user community with tools that would enable them to develop more reliable systems which is a need that has emerged from them.

As is with any software development approach, a complex system is generally conceptualized first using software design methodologies (e.g., Prometheus in our case), and then this design is implemented using a programming language (e.g., JACK, JASON, JADEX, etc). Although there is a large body of work on verifying BDI agent systems using formal verification techniques [17, 13, 26, 3, 11], these approaches either assume the presence of a formal BDI system or testing is done on an agent system that is already implemented. Though recent work [1, 2] has tried to address verification at the level of agent designs, that is before a system is even programmed, the use of formal verification techniques for this purpose is not widespread.

With respect to agent designs, the Tropos methodology supports validating requirements using T-Tool [14] and reasoning about agent goals using the GR-Tool [15]. However, Tropos does not provide support for verification of requirements against agent designs. The approach that comes closest to ours is that of Abushark et.al. [2]. In their work the authors verify requirements against detailed designs via an algorithmic approach. The key idea there is to construct a Petri net from the given requirements and verify the detailed designs by extracting the behavior traces and executing these traces

against the constructed Petri net. The authors in [2] address the issue of requirements verification in an ad-hoc manner without formalising the problem, and leave the soundness and completeness of their approach as future work.

In [27], scenarios in Prometheus are extended such that they may be propagated to the detailed design, thus reducing the chance of error, and use them in generating scenario-based run-time test cases. Their approach complements the formal verification framework presented in this paper.

In this work we are interested in formally verifying requirements against agent details and on the way of achieving this provide a way to formalise agent designs based on Prometheus methodology. Further, as shown in [9], BDI design methodologies share similar structures, and hence we believe that the key formal notions developed here can be adapted to other BDI design methodologies.

3 Framework

We begin with formally defining the core elements of an agent design. Goals are usually captured by defining *goal trees*. A goal tree is a tree (in its usual sense) whose nodes are agent goals and branches are labelled either an AND (all sub-goals required to achieve the goal) or an OR (only one sub-goal required to achieve the goal). Formally, a *goal tree* is a tuple $T = \langle G, g_0, \mathcal{R}, \mu \rangle$ where G is the set of goals, $g_0 \in G$ is the top level goal, relation \mathcal{R} defines the parent-child relationship between goals where $\mathcal{R}(g_1, g_2)$ implies that g_1 is the parent goal of g_2 , and $\mu : G \rightarrow \{\text{AND}, \text{OR}\}$ provides AND-OR mapping for sub-goals. Given a goal g and its sub-goals g_1, \dots, g_n , let $\text{sg}(g) = \{g_1, \dots, g_n\}$. That is, the function sg returns the set of all sub-goals of g .

A scenario consists of a sequence of steps that need to be achieved for it to be completed. Each step is either a *percept*, a *goal* or an *action* and the entity responsible for it is defined by an agent *role*. Formally, each step in a scenario is a pair (o, r) where o is a *step type* and r is an agent role. A scenario is then an ordered tuple $S = \langle (o_1, r_1), \dots, (o_n, r_n) \rangle$ where $n \geq 1$. Given a scenario S consisting of n steps we denote the size of S by $|S|$ (i.e., $|S| = n$), its i^{th} step by $S[i]$, and the step type and role of the i^{th} step by $S[i].\text{type}$ and $S[i].\text{role}$, respectively. A *requirements specification* simply consists of a scenario S and a set of k goal trees $T_i = \langle G_i, g_{i0}, \mathcal{R}_i, \mu_i \rangle$, where $1 \leq i \leq k$.

A detailed design on the other hand consists of entities used in scenarios (i.e., goals, percepts, and actions) and additionally messages and plans. A *message* is of the form $\text{from} \rightarrow \text{to} : \text{msg}$ where from and to are agents and msg is the name of the message. A plan in an agent design is defined in terms of its trigger and outputs such as messages, actions, and sub-goals. Details of a plan body are not defined at the design level (it is an implementation level detail). A *plan* is a tuple $p = \langle \text{name}, \text{trigger}, O \rangle$ where name is a unique identifier for p , trigger is its trigger, and O is the set of its outputs. For technical convenience, we shall refer to components of plan p as $p.\text{name}$, $p.\text{trigger}$, and $p.O$. In the rest of the paper, we refer to the set of all goals by *Goals*. Similarly, *Percepts*, *Actions*, *Messages*, and *Plans*.

In order to track each possible plan instance we need to track *how* it was triggered. We assign each plan instance an *id* to track a sequence of plan activations that preceded it. Formally, the set of *ids* for plan $p = \langle \text{name}, \text{trigger}, O \rangle$ is defined inductively by $\Delta(p) = \{\text{id} \cdot \text{name} \mid \exists p'(\text{trigger} \in p'.O, \text{id} \in \Delta(p'))\}$ where for the base case we have that $\Delta(p) = \{\text{name} \mid \text{trigger} \in \text{Percepts}\}$. Generally, an *id* for a plan instance p will start with a plan name

that handles a percept, followed by a sequence of plan names, subsequently ending with p 's name, such that the trigger for a plan was in the output of plan preceding it in the *id*. Given an $\text{id} = \text{id}' \cdot \text{name}$ for a plan $p = \langle \text{name}, \text{trigger}, O \rangle$ let $\text{history}(\text{id}) = \text{id}'$ and let $\text{active}(\text{id}) = p$.

An agent for design purposes is a collection of plans along with its roles. A *design agent* is a tuple $\text{Ag} = \langle n, R, Pl \rangle$ where n is agent's name, R is a set of agent's roles, and Pl is agent's plan library. A *detailed design* consists of a set of design agents. Formally, a detailed design D is a set $\{\text{Ag}_1, \dots, \text{Ag}_n\}$ where Ag_i are design agents for $1 \leq i \leq n$. Observe that goals, actions, messages are closely linked to plans via its trigger and outputs and can be directly inferred from the agents plan library.

3.1 Conformance of requirements and designs

The problem we are interested in is to *automatically check if a detailed design D conforms to a requirements specification R* . Observe that though all scenario entities (i.e., step types) are also available in the detailed design there may not be an exact one to one mapping between them (else the verification task will be simple). For example, a scenario may specify a goal g whereas the detailed design may only have plans for g 's sub-goals. Additionally, there may be multiple plans to handle a given goal, some of which may not conform to the requirements. In order to precisely define what it means for a detailed design to conform to a requirements specification we introduce a notion of *traces* for a requirement specification and detailed design.

Due to the subjective nature of the design process a requirement may be captured in multiple ways. Hence, in general, just by looking at the sub-goals one cannot infer if the designer has specified an ordering between sub-goals or that she is indifferent towards it. For consistency and uniformity we will assume the following interpretations:

- Listing the parent goal implies that ordering of sub-goals does not matter; and
- Listing of sub-goals without a parent goal implies that ordering of sub-goals matter.

Requirement traces: Informally, we say a goal is *met* if the goal itself is posted, or all its sub-goals are posted in case of AND sub-goal type, or at least one of its sub-goals is posted in case of a OR sub-goal type. Formally, a goal g from a goal tree $\langle G, g_0, \mathcal{R}, \mu \rangle$ is *met* by a sequence of goals g_1, \dots, g_n if either one of the following holds: (i) $n = 1$ and $g_1 = g$; or (ii) if $\mu(g) = \text{AND}$, then $\text{sg}(g) = \{g_1, \dots, g_n\}$; or (iii) if $\mu(g) = \text{OR}$, then $n = 1$ and $g_1 \in \text{sg}(g)$.

Conceptually, a trace for a requirements specification is *one* possible way in which an underlying scenario can be achieved. Formally, a *trace for a requirement specification* with scenario $S = \langle (o_1, r_1), \dots, (o_n, r_n) \rangle$ and goal trees $\{T_1, \dots, T_k\}$ is a sequence of pairs $\tau = (o'_1, r_1) \cdot \dots \cdot (o'_m, r_m)$ with $m \geq n$ such that for each step type o_i there exists indices s_i and e_i where:

1. if $o_i.\text{type} \in \text{Percepts} \cup \text{Actions}$, then $s_i = e_i$ and $o_i = o'_{s_i}$;
2. if $o_i.\text{type} \in \text{Goals}$, then o_i is met through $o'_{s_i} \cdot \dots \cdot o'_{e_i}$;
3. for all indices it holds that $e_j = s_{j+1} - 1$, $s_1 = 1$, and $e_n = m$, where $1 \leq j < n$.

Informally, a trace for a requirements specification is a concatenation of sequences (with start index s_i and end index e_i) such that each sequence achieves a scenario step. If the step is a percept or

an action then the sequence is just one element containing the same percept or action. If a scenario step is a goal, then the sequence is such that the goal in the scenario step is met as per the assumptions discussed before.

Design traces: We define a trace for a detailed design incrementally by introducing traces for a plan, an agent, and finally for a group of agents. The idea is to define a trace for a design agent as an interleaving of its plan traces such that each posted goal is handled by at most one plan (goals are posted internally in an agent). We define a trace for a design as a collection of design agent traces such that each posted message is handled by at most one plan in the receiver agent (messages are posted across agents).

A *trace for a plan* $p = \langle \text{name}, \text{trigger}, O \rangle$ is a sequence of the form $\tau = \text{trigger} \cdot \text{name} \cdot o_1, \dots, o_n$ where $|O| = n$ and o_1, \dots, o_n is a *permutation* of elements in O . (An agent activates a plan by acknowledging its trigger, executing the plan body, and then posting the plan outputs one by one.)

Since an agent may have more than one active plan at a time, a trace for a design agent will consist of interleaved active plan traces with certain constraints. We define an interleaved trace resulting from two traces $\tau_1 = t_1^1 \dots t_n^1$ and $\tau_2 = t_1^2 \dots t_m^2$ to be a trace $\tau_{1+2} = t_1 \dots t_{m+n}$ such that $\tau_{1+2}^{\uparrow \tau_1} = \tau_2$ and $\tau_{1+2}^{\uparrow \tau_2} = \tau_1$ where $\tau_{1+2}^{\uparrow \tau}$ is a trace obtained by projecting out τ from τ_{1+2} . Given a set of plan traces $\Gamma = \{\tau_1, \dots, \tau_n\}$, a *trace for a design agent* is an interleaved trace $\tau = t_1, \dots, t_\ell$ over agent's plan traces Γ such that:

1. if $t_i = \mathbf{g}$ is a trigger in plan trace τ' and $\mathbf{g} \in \text{Goals}$, then there exists $t_j = \mathbf{g}$ where \mathbf{g} is output in a plan trace $\tau'' \neq \tau'$ where $\tau', \tau'' \in \Gamma$, and $j < i$;
2. for any two $t_i = t_j = \mathbf{g}$, where $i \neq j$, $\mathbf{g} \in \text{Goals}$ and t_i, t_j are triggers for plan traces τ^1 and τ^2 , there must exist $t_{i'} = t_{j'} = \mathbf{g}$ with $i' \neq j', i' < i, j' < j$ such that $t_{i'}, t_{j'}$ are plan outputs in traces $\tau^{1'}, \tau^{2'}$, respectively.

Intuitively:(1) a goal should be posted before it can be handled; and (2) only one plan gets activated for a posted goal.

A design trace then is simply a set of traces, one for each design agent, where we allow an agent to be inactive if it does not have any active plans. Formally, a *design trace* for a set of k agents is a sequence $\tau = (t_1^1, \dots, t_n^1) \dots (t_1^k, \dots, t_n^k)$ such that each element t_j^{agt} can either be ϵ (here ϵ denotes an empty token) or from design agent agt 's trace with the following constraints:

1. for all agents i it holds that $t_m^i \dots t_n^i$ is agent i 's trace such that either $m = 1$ or $t_{m-1}^i = \epsilon$, and $n = \ell$ or $t_{n+1}^i = \epsilon$;
2. for each $t_m^i = \text{msg}$ where $\text{msg} \in \text{Messages}$ is a trigger of agent i , there exists a unique $t_n^{i'} = \text{msg}$ where $t_n^{i'}$ is an output in a plan of agent $i' \neq i$, $n < m$, agent i is msg 's receiver and agent i' is msg 's sender;
3. if $t_m^i = \text{msg}$ is a message (that is, $\text{msg} \in \text{Messages}$) and in output of agent i 's plan, then for an agent i' such that i' is msg 's receiver and for all indices $m < j < n$ where $t_n^{i'} = \text{msg}$ is a trigger for a plan in agent i' 's trace it is the case that $t_j^{i'} \neq \epsilon$.

Intuitively:(1) an agent trace cannot have empty tokens; (2) each message is handled by at most one plan; (3) the receiver agent must handle a message posted for it (i.e., it cannot choose to stay idle in presence of a pending message).

3.1.1 Comparing requirements and design traces

Conceptually, a trace for a detailed design is said to conform with a requirements trace if the design trace contains elements from the requirements trace in the right order. Formally, a design trace $\tau_D = (t_1^1, \dots, t_1^k) \dots (t_\ell^1, \dots, t_\ell^k)$ for a design D *conforms* to a trace $\tau_R = (o_1, r_1) \dots (o_m, r_m)$ of requirements specification R if there exists a set of indices j_1, \dots, j_m such that for all $1 \leq i \leq m$ there exists agent agt where $t_{j_i}^{\text{agt}} = o_i$ and r_i is in agt 's roles. We say a detailed design D conforms to a requirements specification R if there exists a trace τ_D of D for which there exists a trace τ_R of R such that τ_D conforms to τ_R .

A design trace will generally be much longer in length than a scenario trace because a detailed design fleshes out *how* a particular requirement is achieved.

4 Model checking agent designs

Conceptually, an ATL model (also known as concurrent game structure) consists of a set of agents that act concurrently in order for the game to progress. The game consists of a set of states that evolve based on agents' moves and one checks temporal formulae against these states to verify properties. For our conformance checking problem, the set of agents will consist of a requirements agent and a number of detailed design agents. The requirements agent will select its actions as per the underlying scenario and goal trees, whereas the design agents will move as per their active plans. In the ATL game structure we will match each design agent's action against the action of the requirements agent to check if a scenario step has been achieved. The objective then is to verify if all scenario steps have been achieved in the right order.

We do this in two steps: (i) we build two kinds of finite state automaton (see [16] for details on FSA.) one that will accept all traces for a requirements specification, and second that will accept all traces for a given plan; (ii) we use these automata to build a concurrent game structure that will serve as our ATL model for model checking. A game state in our ATL model will consist of underlying states of these automata (a requirements automaton and multiple plan automata) and these states will be appropriately updated based on the agents' actions (that is, based on scenario steps achieved, plan triggers, and plan completions). Finally, to verify that a given detailed design conforms to the requirements specification, we will check the formula $\langle\langle \mathcal{A} \rangle\rangle \diamond \text{final}$, where \mathcal{A} is the set of all design agents and *final* captures the condition that the requirements automaton has reached its final state.

4.1 Requirements and agent plan automata

We introduce two technical notations required for constructing the automata. First, given a sequence $\tau = t_1 t_2 \dots t_n t_{n+1}$ let a set of states for accepting τ indexed by a number be $\theta(\tau, i) = \{q_\epsilon^i, q_{t_1}^i, q_{t_1 t_2}^i, \dots, q_{t_1 \dots t_n}^i\}$. Intuitively, the role of states in $\theta(\tau, i)$ is to track the sequence τ and since we may need to track the same sequence more than once we index it by a number. Second, given a set of elements E , let the set of all possible sequences (that is, permutations) of length $|E|$ that can be generated from E be $\text{perm}(E)$.

Next, we build an automaton such that its language is the set of traces for a given requirements specification. Formally, an automaton for a given scenario $S = \langle (o_1, r_1), \dots, (o_n, r_n) \rangle$ and set of k goal trees $T_i = \langle G_i, g_{i0}, \mathcal{R}_i, \mu_i \rangle$, where $1 \leq i \leq k$, is a tuple $F_R = \langle Q, q_0, \Sigma, \delta, \{q_f\} \rangle$ where:



Figure 3: Automata for the auction example.

1. $Q = \{q_{o_0}, q_{o_n}\} \cup \{q_o \mid \exists i S[i].\text{type} = o\} \cup \{\theta(\tau, i) \mid \tau \in \text{perm}(\text{sg}(g)), \exists i(g = S[i].\text{type}), g \in \mathcal{G}_S^\wedge\} \cup \{q_g \mid \exists i, j S[i].\text{type} = g', g \in \text{sg}(g'), g' \in \mathcal{G}_S^\vee\}$ where \mathcal{G}_S^\wedge (\mathcal{G}_S^\vee) is the set of AND (OR) goals in scenario S ;
2. q_{o_0} is the initial state and q_f is the final state where $q_f = q_{o_n}$;
3. $\Sigma = \{(o, r) \mid \exists i S[i] = (o, r)\} \cup \{(o, r) \mid \exists i, g S[i] = (g, r), o \in \text{sg}(g)\}$ is the alphabet consisting of elements in traces of requirements R ; and
4. $\delta: Q \times \Sigma \rightarrow Q$ is the transition function where $\delta(q, \sigma) = q'$ if:
 - (a) $\sigma = (o_i, r_i), q = q_{o_{i-1}}, q' = q_{o_i}$ for $1 \leq i \leq n$;
 - (b) $\sigma = (o'_i, r_i)$ such that there exists $o_i \in \mathcal{G}_S^\vee, o'_i \in \text{sg}(o_i), q = q_{o_{i-1}}, q' = o_i$ where $1 \leq i \leq n$;
 - (c) $\sigma = (o'_i, r_i)$ such that there exists $o_i \in \mathcal{G}_S^\wedge, o'_i \in \text{sg}(o_i)$ and one of the following holds:
 - i. $q = q_{o_{i-1}}$ and $q' = q_{o'_i}$;
 - ii. $q = q_\tau^i$ and $q' = q_{o_i}$ where $\tau \cdot o'_i \in \text{perm}(\text{sg}(o_i))$;
 - iii. $q = q_\tau^i$ and $q' = q_{\tau'}^i$ where $\tau \cdot o'_i = \tau'$ and $\tau' \cdot \tau'' \in \text{perm}(\text{sg}(o_i))$ for some $|\tau''| \geq 1$.

Intuitively, the set of states (1) of an automaton for a requirements specification consists of a state per each scenario step and additional states to cater for sub-goals where a scenario step has AND sub-goals. In addition, we index the states required for AND sub-goals (with the step number) as the same goal may appear more than once in a scenario. The alphabet (3) of the automaton consists of requirements trace elements, that is, a pair consisting of step type and step role. The first condition (4a) of the transition function connects each scenario step sequentially, the second (4b) provides alternatives where an OR goal could be achieved by one of its sub-goals, and the third (4c) caters for each permutation of an AND goal. Figure 3 (a) shows the finite state automaton for the auction requirements specification described in Figure 1b.

Theorem 1. *The language of automaton F_R for requirements specification R having a scenario S and goal trees $\{T_1, \dots, T_k\}$ is the set of all possible traces of R .*

PROOF (SKETCH). Let $w = \sigma_1 \dots \sigma_\ell \in L(F_R)$, automaton $F_R = \langle Q, q_0, \Sigma, \delta, \{q_f\} \rangle$ and scenario $S = \langle (o_1, r_1), \dots, (o_n, r_n) \rangle$. Hence, there exists a sequence of transitions $\lambda = q_0 \xrightarrow{\sigma_1} q_1 \xrightarrow{\sigma_2} \dots \xrightarrow{\sigma_\ell} q_\ell$ where $\delta(q_{i-1}, \sigma_i) = q_i$ for $1 \leq i \leq \ell$ and $q_\ell = q_f$. The sequence λ contains states q^1, \dots, q^n such that $q^i = q_{o_i}$ for $1 \leq i \leq n$ (from condition 4). From the states q^1, \dots, q^n one can extract index pairs (s_j, e_j) where $\lambda[s_j] = q^j$ and $e_j = s_{j+1} - 1$ where $1 \leq j < n$ and $e_n = \ell$. Each sub-sequence $\lambda[s_j] \dots \lambda[e_j]$ achieves step o_j , for $1 \leq j \leq \ell$; hence w is a trace for requirements R . For the other side, assume $\tau = (o'_1, r'_1) \dots (o'_m, r'_m)$ is a trace for requirements R . Then, there exists indices s_i, e_i for each step o_i , where $1 \leq i \leq m$, such that sequence $o'_{s_i} \dots o'_{e_i}$ achieves step o_i . If o_i is a percept or action, for $1 \leq i \leq m$, then there exists transition from q_{o_i} to $q_{o_{i+1}}$ on symbol (o_i, r_i) (condition 4a in definition of F_R). If o_i is a goal, for $1 \leq i \leq m$, then either (i) $s_i = e_i$ and $\tau[s_i] = o_i$ (condition 4a in definition of F_R), or (ii) $s_i = e_i$, $\tau[s_i] \in \text{sg}(o_i)$, and $\mu(g) = \text{OR}$ (condition 4b in definition of F_R), or (iii) o_i is met through sequence $\tau[s_i] \dots \tau[e_i]$ (condition 4c in definition of F_R). Observe that the first condition in transition function

of F_S caters for condition (i), and second and third condition in transition function of F_S cater for condition (ii) above. Combined with the fact that q_{o_n} is the final state, we get that $\tau \in L(F_R)$. \square

Next we construct an automaton for a plan such that the automaton will accept only the possible traces of the plan. An automaton for a plan $p = \langle n, t, O \rangle$ is a tuple $F_p = \langle Q, q_0, \Sigma, \delta, q_f \rangle$ where:

1. $Q = \{q_0, q_1, q_f\} \cup \{\theta(\tau, 1) \mid \tau \in \text{perm}(O)\}$ is set of states;
2. q_0 and q_f are the initial and final states, respectively;
3. $\Sigma = \{n, t\} \cup O$ is the alphabet;
4. $\delta: Q \times \Sigma \rightarrow Q$ is the transition function where $\delta(q, \sigma) = q'$ if:
 - (a) $q = q_0, \sigma = t$, and $q' = q_1$; or
 - (b) $q = q_1, \sigma = n$, and $q' = q_\tau^1$; or
 - (c) $q = q_\tau^1, \sigma \in O$, and $q' = q_{\tau \cdot \sigma}^1$ where $\tau \cdot \sigma \cdot \tau' \in \text{perm}(O)$ for some $|\tau'| \geq 1$; or
 - (d) $q = q_\tau^1, \sigma \in O$, and $q' = q_f$ where $\tau \cdot \sigma \in \text{perm}(O)$.

The alphabet of the automaton is the elements of the plan. The transition function ensures that any trace accepted by the automaton starts with the plan trigger (4a), followed by its name (4b), and then any legal permutation of the plan's outputs (4c, 4d). Figure 3 (b) shows the finite state automaton for the DecideWinner plan of the Auctioneer. For the purpose of implementation the number of states of the automaton of plan $p = \langle n, t, O \rangle$ will be exponential in $|O|$.

Theorem 2. *The language of the automaton F_p for a plan $p = \langle \text{name}, \text{trigger}, O \rangle$ is the set of all traces of p .*

PROOF (SKETCH). Observe that any word in the language of automaton F_p has the form $\text{trigger} \cdot \text{name} \cdot \lambda$ where λ is from the set $\text{perm}(O)$. Hence, any word in the language of F_p is a trace of plan p . Similarly, it can be shown that the first two symbols in a trace τ of plan p result F_p to transition from its initial state to state q_τ^1 ; and subsequent symbols of τ cause F_p to transition from q_τ^1 to q_f . \square

These automata provide a cleaner mapping to build the ATL game structure and also provide a straightforward translation to ISPL encoding for MCMAS [18] (please see the Appendix for the ISPL encoding for the auction example.)

4.2 ATL concurrent game structure

The ATL model that we will build consists of a requirements agent and a number of design agents. Observe that the automaton for a given scenario encapsulates all legal scenario traces (see Theorem 1), and hence models the behavior of the requirements agent. A design agent on the other hand consists of multiple plans, and therefore its behavior will be modelled by multiple automata, one for each of its (potential) plans that might get activated.

A concurrent game structure for an agent design consists of an ATL requirements agent and one ATL agent per detailed design agent. The requirements agent executes actions as per the automaton obtained from the requirements specification and design agents execute actions as per automata of their active plans. A game state captures the completion status of the requirements specification along

with the status of plan instances. A key feature in our reduction is that the state of requirements specification progresses only when a design agent executes an action as expected by the requirements. Intuitively, this implies that a next step in a (partial) requirements trace has been achieved by one of the detailed design agents. We assume that the percept in the first scenario step is posted by default, and hence a plan that can handle it will be the first to get activated. Formally, given an automaton $F_R = \langle Q^R, q_0^R, \Sigma^R, \delta^R, \{q_f^R\} \rangle$ for a requirements specification R (consisting of a scenario and a set of goal trees) and a detailed design D containing k design agents $agt_i = \langle n_i, R_i, Pl_i \rangle$ and automata $F_p = \langle Q^p, q_0^p, \Sigma^p, \delta^p, \{q_f^p\} \rangle$ for each plan $p \in Pl$ (let $Pl = \cup_{1 \leq i \leq k} Pl_i$), a concurrent game structure for R and D is a tuple $\mathcal{M}_{\langle R, D \rangle} = \langle \{Req, n_1, \dots, n_k\}, Q, \mathcal{P}, Act, d, \mathcal{V}, \delta \rangle$ where:

1. There are $k + 1$ agents: **Req** is the requirements agent, and n_1, \dots, n_k are design agents (one per detailed design agent);
2. States Q consist of the following finite range functions:
 - (a) $\text{scn} \in Q^R$ (Q^R is set of states for automaton F_R);
 - (b) $\text{plan}_i^{\text{id}} \in Q^p \cup \{\text{inact}\}$ where $\text{id} \in \Delta(p)$, $p \in Pl_i$ where $1 \leq i \leq k$, and **inact** is used to capture if a plan is inactive;
3. \mathcal{P} is the set of propositions asserting value assignments to the above defined functions and \mathcal{V} is the mapping from a game state q to the values returned by the above defined functions. For convenience, we will write $(\text{scn}(q) = q_r) \in \mathcal{V}(q)$ as $\text{scn}(q) = q_r$.
4. $Act = \{a \mid \exists r(a, r) \in \Sigma^R\} \cup \{\text{id} \cdot a \mid a \in \Sigma^p, \text{id} \in \Delta(p), p \in Pl\} \cup \{\text{fin}, \text{nop}\}$ is the set of domain actions, where Σ^R is the alphabet for the automaton F_R , Σ^p is the alphabet for the automaton for plan p , **fin** and **nop** are special actions to denote that a scenario has finished and an agent has no active plans, respectively. The action $\text{id} \cdot a$ denotes symbol a of alphabet Σ^p prefixed by id of its plan instance p . Given an annotated action $\text{id} \cdot a$, let $\text{action}(\text{id} \cdot a) = a$.
5. $d(j, q)$ defines the moves available for agent j in state q :
 - (a) Requirements agent ($j = \text{Req}$):

$$d(j, q) = \begin{cases} \{a \mid \exists r, q^R(\text{scn}(q), (a, r), q^R) \in \delta^R\}, & \text{if } \text{scn}(q) \neq q_f^R \\ \{\text{fin}\}, & \text{otherwise.} \end{cases}$$

- (b) Design agents ($j \in \{n_1, \dots, n_k\}$):

$$d(j, q) = \begin{cases} \text{next-actions}(j, q), & \text{if } |\text{next-actions}(j, q)| > 0 \\ \{\text{nop}\}, & \text{otherwise.} \end{cases}$$

$$\text{where, next-actions}(j, q) = \{\text{id} \cdot a \mid \exists q_p(\text{plan}_j^{\text{id}}(q), \text{action}(\text{id} \cdot a), q_p) \in \delta^p, \text{id} \in \Delta(p)\}$$

6. $\delta: Q \times Act^k \rightarrow Q$ is the transition function such that $\delta(q, \vec{a}) = q'$, where $\vec{a} = a_r, a_1, \dots, a_k$ is the move vector containing actions for requirements and design agents, and q' is as follows:

- (a) Requirements: updated if a design agent acts as expected:

$$\text{scn}(q') = \begin{cases} s = \delta^R(\text{scn}(q), (\text{action}(a_i), r)), & \text{if } s \text{ is defined} \\ \text{for some } 1 \leq i \leq k \text{ where } r \in R_i; \\ \text{scn}(q), & \text{otherwise} \end{cases}$$

- (b) Plans: updated as follows ($j \in \{n_1, \dots, n_k\}$):

- i. if $\text{plan}_j^{\text{id}}(q) = q_0$ and $a_j = \text{id} \cdot a$, then $\text{plan}_j^{\text{id}}(q') = q_1$ and $\text{plan}_j^{\text{id}}(q') = \text{inact}$ where $\text{history}(\text{id}) = \text{history}(\text{id}')$ and $\text{active}(\text{id}) \cdot \text{trigger} = \text{active}(\text{id}') \cdot \text{trigger}$;
- ii. if $\text{plan}_j^{\text{id}}(q) = q_1$ and $a_j = \text{id} \cdot a$, then $\text{plan}_j^{\text{id}}(q') = q_e^1$;

- iii. if $\text{plan}_j^{\text{id}}(q) = q^p$ and $a_j = \text{id} \cdot a$, then $\text{plan}_j^{\text{id}}(q') = \delta^p(q^p, a)$ where $p = \text{active}(\text{id})$, and:
 - A. if $a \in \text{Goals}$, then $\text{plan}_j^{\text{id}}(q') = q_0$ where $\text{id}' = \text{id} \cdot \text{name}$ such that $p = \langle \text{name}, a, O \rangle \in Pl_j$;
 - B. if $a \in \text{Messages}$, then $\text{plan}_j^{\text{id}}(q') = q_0$ where $\text{id}' = \text{id} \cdot \text{name}$ such that $p = \langle \text{name}, a, O \rangle \in Pl_i$ where $i \neq j$;
- iv. if $a_j = \text{nop}$, then $\text{plan}_j^{\text{id}}(q') = \text{plan}_j^{\text{id}}(q)$;

Given a requirements specification R and detailed design D , the states of the concurrent game structure $\mathcal{M}_{\langle R, D \rangle}$ consist of states from the automaton of requirements R , and states from the automata of possible plan instances of design agents in D . The function **scn** returns the current state of automaton F_R whereas functions $\text{plan}_j^{\text{id}}$ returns the current state of plan **active**(id) of design agent j (2). Moves of the requirements agent (5a) from a state q consist of scenario step types that are part of outgoing transitions of current state of automaton F_R (that is, $\text{scn}(q)$). If there are no outgoing transitions (because F_R has reached its final state) then the requirements agent's moves consist of the special action **fin**, implying that the requirements have been met. Moves of a design agent (5b) consists of a union of all possible symbols that are part of outgoing transitions of automata of its active plans. A design agent j does the special action **nop** in a state q if it does not have any active plans, that is, the set of **next-actions**(j, q) is empty. The transition function of game structure $\mathcal{M}_{\langle R, D \rangle}$ models how the underlying automaton states are updated (6). For practical purposes, encoding a concurrent game structure to ISPL [18] is straightforward. We present the ISPL encoding for the auction example in the Appendix.

4.3 Verifiable design properties

The ATL model $\mathcal{M}_{\langle R, D \rangle}$ for a requirements specification R and detailed design D can be now used to verify design time properties such as *conformity* and *coverage*. Requirements conformity (as formalized in Section 3.1.1) deals with checking if a detailed design can achieve a given requirements specification, whereas coverage signifies in how many different ways can a requirements specification be achieved.

Requirements conformity: Given an ATL model $\mathcal{M}_{\langle R, D \rangle}$ for a requirements specification R and detailed design D we are interested in checking whether design D conforms to requirements R . Observe that the requirements agent in the ATL model has limited freedom in its behavior: it repeatedly selects one of its expected scenario steps until one of the agents with an appropriate role executes an expected action. What it implies is that we are trying to match a trace of detailed design with a trace of requirement. Hence, in order to check the conformity of requirements with detailed design we model check the formula $\varphi = \langle\langle \mathcal{A} \rangle\rangle \diamond \text{final}$ where \mathcal{A} is the set of all design agents in the ATL model and **final** is defined as $\text{sch} = q_f^R$ where q_f^R is the final state of scenario automaton F_R . Finally, we check this formula from a state q_I (in the game structure) such that the requirements automaton and the plans that will handle that percept in the first scenario step are in their initial states, and all other design agent plan instances have the value **inact**. Formally, given a ATL model $\mathcal{M}_{\langle R, D \rangle}$ for requirements R with scenario $S = \langle (o_1, r_1), \dots, (o_n, r_n) \rangle$ and detailed design D it is the case that: (i) $\text{sch}(q_I) = q_0^R$ where q_0^R is the initial state of requirements automaton F_R , (ii) $\text{plan}_j^{\text{id}}(q_I) = q^0$ for all design agents j such that $\text{active}(\text{id}).\text{trigger} = o_1$, and (iii) $\text{plan}_j^{\text{id}}(q_I) = \text{inact}$ for all design agents j such that $\text{active}(\text{id}) \cdot \text{trigger} \neq o_1$.

Theorem 3. A detailed design D conforms to requirements specification R iff $\mathcal{M}_{(R,D)}, q_I \models \langle\langle \mathcal{A} \rangle\rangle \diamond (\text{scn} = q_f^R)$ where \mathcal{A} is the set of all detailed design agents in $\mathcal{M}_{(R,D)}$ and q_f^R is the final state of requirements automaton F_R .

PROOF (SKETCH). Suppose $\mathcal{M}_{(R,D)}, q_I \models \langle\langle \mathcal{A} \rangle\rangle \diamond (\text{scn} = q_f^R)$. Hence, there exists a collective strategy F_A , one for each agent in \mathcal{A} such that $\diamond(\text{scn} = q_f^R)$ is satisfied in all computations $\lambda \in \text{out}(q^I, F_A)$. Let $\lambda = q_0 \cdots q_\ell$ where $q_0 = q^I$, $\text{sch}(q_\ell) = q_f^R$, and $\vec{a}_1 \cdots \vec{a}_\ell$ be the sequence of move vectors such that $\delta(q_{i-1}, \vec{a}_i) = q_i$, for $1 \leq i \leq \ell$, where δ is the transition function of $\mathcal{M}_{(R,D)}$. Let each $\vec{a}_i = \langle a_i^r, a_i^1, \dots, a_i^k \rangle$ where a_i^r is the action of requirements agent and a_i^1, \dots, a_i^k are actions of the design agents. From the definition of δ (condition 6) one can observe that by construction $(a_1^1, \dots, a_1^k) \cdots (a_\ell^1, \dots, a_\ell^k)$ is the design trace that conforms to the scenario trace $\tau = (a_1^r, r_1) \cdots (a_m^r, r_m)$ where τ is extracted from $a_1^r \cdots a_\ell^r$ by removing adjacent duplicate elements. For the other side, given a trace $\tau_D = (a_1^1, \dots, a_1^k) \cdots (a_\ell^1, \dots, a_\ell^k)$ of design that conforms to a trace $\tau_R = (a_1^r, r_1) \cdots (a_m^r, r_m)$ of requirements R (that is, there exists indices w_1, \dots, w_m where each step type was achieved for scenario), one can construct a path $\lambda = q_0 \cdots q_\ell$ following the transition function δ of $\mathcal{M}_{(R,D)}$ such that (i) actions of design agents between states q_{i-1} and q_i are $\tau_D[i]$, (ii) requirements agent executes action (o_i, r_i) at w_i and repeats its action between each w_i 's, for $1 \leq i \leq m$, (iii) $q_0 = q^I$, and (iv) $\text{scn}(q_\ell) = q_f^R$. \square

Design nonconformity: Even though a given design may conform to a scenario, it may still contain traces that do not achieve any scenario trace. These design traces are not necessarily faulty as the ordering of plan outputs is not specified in the detailed design. Nonetheless, such design traces should be highlighted to the designer to spot potential errors in the agent design. Interestingly, we can extract all such traces from the winning states of the concurrent game structure $\mathcal{M}_{(R,D)}$ by checking the formula $\tilde{\varphi} = \langle\langle \mathcal{A} \rangle\rangle \square \neg \text{final}$ where \mathcal{A} is the set of all design agents in the ATL model. In general, such traces will have incorrect ordering of scenario steps or plan activations that do not achieve a scenario step. Intuitively, one extracts such traces by extracting actions of design agents from move vectors responsible for transitions in the winning states. We omit the details of the approach to extract such traces due to space limitations.

Requirements coverage: Given the set of maximal winning states $[\varphi]$ in ATL model $\mathcal{M}_{(R,D)}$ one can extract all the requirements specification traces that can be achieved by at least one design trace. Since the language of automaton $F_R(L(F_R))$ is all the possible requirements traces, one can compute the words in $L(F_R)$ that are not in the traces extracted from the winning game states to denote requirements traces that a designer might want to cater for in the detailed design. For the discussed design properties, the set of maximal winning states serves as a model from which all the relevant design and requirements traces can be extracted.

5 Scalability Evaluation

In this section, we present initial scalability results to show the feasibility of our approach. We generated test cases with a single scenario and multiple goal trees, and a single detailed design agent. The detailed designs were varied by: (i) goals per plan (g/p), (ii) plans per goal (p/g), and (iii) the depth of the design (d). (i) and (ii) were varied from 1 to 3, and depth was sequentially changed from 1 to 8, resulting in 72 test cases. Each scenario, with the number of steps matching the depth of the design, was constructed such that its corresponding design always conforms to it. These designs were then verified for conformance in McMAS [18] and their execution time

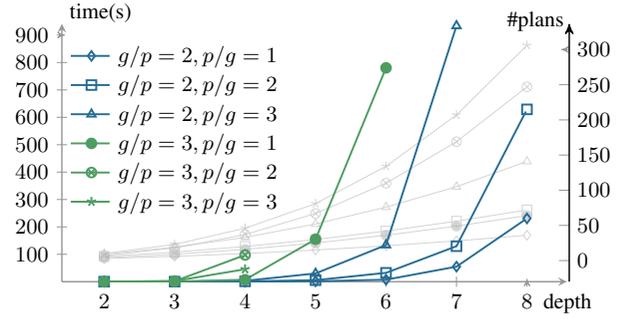


Figure 4: Scalability analysis of approach (see text for details).

was recorded from its output.² We had put an upper time limit of one hour.

In addition, to control the branching factor the percentage of target goals and plans at a depth d was always $1/d$. The rest of the goals/plans were simply one p/g and one g/p . If we do not control the branching factor, the number of possible traces (due to interleavings) grows at a rate factorial to the depth and branching. Given that in practice, not all plans and goals have high branching factors, our approach provides a balance between relevance and reliability.

Figure 4 shows the scalability results of our approach where on the x-axis is the depth, on the left y-axis is the time in seconds, and on the right y-axis is the number of plans in the design. For example, the line with square marks has: $g/p = 2$ and $p/g=2$; each point in the line is a test case; the last point is test case of depth 8 with an average execution time of more than 600 seconds and more than 70 plans. Note that the colored lines denote time and the faded gray lines denote the number of plans.

All cases with depth less than 4 were trivial to solve in terms of time (maximum time of 2.5 sec. for case 3-3-3). With depth 4 onwards the branching contributes significantly and as it appears from Figure 4 (respective cases with $g/p = 3$ have higher execution times than cases with $g/p = 2$) that goals per plan affect the execution time more than plans per goal. However, we point out that our approach tests one scenario at a time (along with its goal trees) against its relevant detailed design. A design that caters for one scenario, generally, will not have a large number of plans. Even so, our approach demonstrates that it can handle large design components within reasonable time bounds for design time verification.

To compare our technique with the one presented in [2], we requested the authors to run our test designs. For smaller test cases their approach was faster than ours (0.11 sec vs 1.2 sec. for case 2-2-4 and 0.13 vs 1.8 sec. for case 3-2-3). However, by increasing the depth by one in both test cases the trace based approach lagged considerably. For example, it took 70.57 sec. for the case 2-2-5 and it was still executing after 4.5 hours for the case 3-2-4. The same in our approach took 5.5 sec. and 97.5 sec., respectively. This is expected, as mentioned, since our approach relies on model checking, it scales significantly better than the trace based approach.

6 Conclusion

This paper presented a framework for formally verifying agent design models with respect to the requirements specifications. We show how informal and semi-structured design artefacts can be transformed into formal structures that can be model checked and we provide details on how to model check via ATL games.

² The system had a quad core i7 3.4GHz CPU with 8GB RAM.

```

1 Agent Requirements
2   Vars:
3     state: {s0, s1, s2, s3, s4, s5};
4   end Vars
5   Actions = {Ann_Auction, Start, Manage, finish,
6     Bid, Ann_Winner, Id_Winner, nop};
7   Protocol:
8     state = s0: {Start_Auction};
9     state = s1: {Ann_Auction, Manage};
10    state = s2: {Bid};
11    state = s3: {Id_Winner, Manage};
12    state = s4: {Ann_Winner, Manage};
13    state = s5: {finish};
14    Other: {nop};
15  end Protocol
16  Evolution:
17    state = s1 if (state = s0) and
18      (( Auctioneer.Action = Start));
19    state = s2 if (state = s1) and
20      (( Auctioneer.Action = Ann_Auction));
21    state = s2 if (state = s1) and
22      (( Auctioneer.Action = Manage));
23    state = s3 if (state = s2) and
24      (( P1.Action = Bid) or (P2.Action = Bid));
25    state = s4 if (state = s3) and
26      (( Auctioneer.Action = Id_Winner));
27    state = s4 if (state = s3) and
28      (( Auctioneer.Action = Manage));
29    state = s5 if (state = s4) and
30      (( Auctioneer.Action = Ann_Winner));
31    state = s5 if (state = s4) and
32      (( Auctioneer.Action = Manage));
33  end Evolution
34 end Agent

```

Figure 5: ISPL encoding for the requirements agent.

While there is existing work on formally verifying the correctness of agent programs via model checking (e.g. [13, 10, 29] and theorem proving (e.g. [25]) to the best of our knowledge this paper is the first to propose a formal verification of detailed agent designs, even prior to any implementation. This allows early detection of errors which is well known to save costs in software development. With respect to agent implementations recent work by Zhang et al.[21] has shown that, across 14 different agent programs, 34% of errors were due to faults in the design.

Although, in general, there is a lack of much work on checking the correctness of detailed agent designs with respect to the requirements specification, recent work by Abushark et al. [2] and Thangarajah et al. [27] are exceptions. We have highlighted some of the limitations of [2], and shown empirically that our approach is significantly more computationally effective as the parallelism within the design increases.

The problem we address also bears resemblance to specifications modelled using modal transition systems [5] in the sense that agent designs have parts that are “required” and parts that are “allowed”. This required part is checked with respect to a scenario. However, due to the presence of goal hierarchies/decomposition we usually do not get a one to one mapping between the transitions.

Our main purpose in this paper was to demonstrate the use of a formal verification approach and we chose ATL for two key reasons. First, ATL model checking supports synthesis of strategies and this is essential for checking properties such as requirements coverage. Second, we are currently extending this framework to verifying AUML protocols and in that setting we require the ability to model an environment for messages that arise external to an agent. In addition, ATL’s multi agent modelling allows for natural extensions that can be built in our framework, such as verifying agent capabilities. This work presents a base from which various other aspects of the agents’ designs could be formally verified.

7 Appendix: ISPL Encoding

In this section we present the ISPL [18] encoding for the auction example used in the paper. Briefly, an agent in ISPL is modelled by four key elements: i) set of local states, ii) set of actions that the agent can

```

1 Agent Auctioneer
2   Vars:
3     P_NewAuction: {..};
4     P_Broadcast: {..};
5     P1_DecdWin: {init, s0, s1, s2, s3, s4, s5};
6     P2_DecdWin: {..};
7     P1_NotifyWin: {..};
8     P2_NotifyWin: {..};
9   ..
10  end Vars
11  Actions = {NewAuction, Auction, finish, Ann_Winner,
12    Id_Winner, nop, NotifyWin, Ann_Auction, Start,
13    LogBid, Bid, WinnerAnnouncement, Broadcast,
14    DecdWin, NewAuction, Bid, NomWin};
15  Protocol:
16    P1_DecdWin = DecdWin0: {Bid};
17    P1_DecdWin = DecdWin1: {DecdWin};
18    P1_DecdWin = DecdWin2: {Id_Winner, Ann_Winner};
19    P1_DecdWin = DecdWin4: {Ann_Winner};
20    P1_DecdWin = DecdWin5: {Id_Winner};
21  ..
22  end Protocol
23  Evolution:
24    P1_DecdWin=s1 if (P1_DecdWin=s0) and (Action=Bid);
25    P1_DecdWin=s2 if (P1_DecdWin=s1) and (Action=DecdWin);
26    P1_DecdWin=s3 if (P1_DecdWin=s4) and (Action=Ann_Winner);
27    P1_DecdWin=s4 if (P1_DecdWin=s2) and (Action=Id_Winner);
28    P1_DecdWin=s5 if (P1_DecdWin=s2) and (Action=Ann_Winner);
29  ..
30  end Evolution
31 end Agent

```

Figure 6: ISPL encoding for the Auctioneer agent.

execute, iii) a protocol that defines which actions are legal to execute based on its state, and iv) an evolution function that defines how states evolve. An ISPL file consists of the agent definitions, the winning condition, the initial states, and the coalition formula to check. In our case, the winning condition in the ISPL encoding is simply the last state of the requirements agent. In our auction example this is specified as `win if Requirements.state = s5;`. The formula we check for verifying requirements is `<g1> F win;` where `g1` is the coalition of agents.

The encoding for our auction example consists of 4 agents, a requirements agent named `Requirements` that models the underlying scenario, an auctioneer agent named `Auctioneer`, and two bidder agents named `P1` and `P2` (not shown here). Figure 5 shows the requirements agent. Observe the straightforward translation from the scenario (please see Figure 1b) in the agent design to its automaton (as shown in Figure 3) and finally to an ISPL encoding. The critical part in the encoding for the requirements agent is its evolution function. The requirements agent changes its state only when an expected action is executed by the correct design agent. For example, the requirements agent will update its state from `s2` to `s3` only when one of the bidder agents sends their bid (lines 23-24 in Figure 5).

Figure 6 shows the (partial) code for the auctioneer design agent. The state variables of design agents consist of variables to track the plans that the agent has. For example, the auctioneer agent has plans to start an auction, broadcast the announcement, handle bids, notify winner, etc. Multiple copies of a plan are required where its trigger can occur multiple times. For example, since multiple agents can send their bids, the auctioneer agent has multiple copies of the `DecideWinner` plan, one per bidder agent (that is, `P1_DecdWin` and `P2_DecdWin` in Figure 6). Also observe that the encoding for a design agent consists of an aggregation of plans it has and the plan automata built earlier provide a clean way to map these to ISPL. The evolution function in the encoding of a design agent keeps a track of its active plans and allows the agent to progress one plan at a time. The resulting behavior of a design agents emerges from the possible ways the active plans can be interleaved.

REFERENCES

- [1] Y. Abushark, J. Thangarajah, T. Miller, and J. Harland. Checking consistency of agent designs against interaction protocols for early-phase defect location. In *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14*, pages 933–940, Paris, France, May 2014.
- [2] Yoosuf Abushark, Michael Winikoff, Tim Miller, James Harland, and John Thangarajah. Checking the correctness of agent designs against model-based requirements. In *ECAI 2014*, volume 263, pages 953–954, Prague, 2014. IOS Press.
- [3] N. Alechina, M. Dastani, B. S. Logan, and John-Jules Meyer. A logic of agent programs. pages 795–800, 2007.
- [4] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, (49):672–713, 2002.
- [5] Adam Antonik, Michael Huth, Kim Guldstrand Larsen, Ulrik Nyman, and Andrzej Wasowski. 20 years of modal and mixed specifications. *Bulletin of the European Association for Theoretical Computer Science*, (95), 2008.
- [6] B. Boehm. Understanding and controlling software costs. *Journal of Parametrics*, 8(1):32–68, 1988.
- [7] R. Bordini, L. Braubach, H. Dastani, A. El-Fallah-Seghrouchni, J. Gomez-Sanz, J. Leite, G. O'Hare, A. Pokahr, and A. Ricci. A survey of programming languages and platforms for multi-agent systems. *Informatica*, 30(1):33–44, 2006.
- [8] Rafael H Bordini, Jomi Fred Hübner, and Michael Wooldridge. *Programming multi-agent systems in AgentSpeak using Jason*, volume 8. John Wiley & Sons, 2007.
- [9] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos. Tropos: An agent-oriented software development methodology. *JAA-MAS*, 8(3):203–236, 2004.
- [10] M. Dastani, K. Hindriks, and J.J. Meyer, editors. *Specification and Verification of Multi-agent systems*. Springer, Berlin/Heidelberg, 2010.
- [11] M. Dastani and W. Jamroga. Reasoning about strategies of multi-agent programs. pages 997–1004, 2010.
- [12] S. DeLoach, L. Padgham, J. Thangarajah, A. Perini, and A. Susi. Using three AOSE toolkits to develop a sample design. *IJAOSE*, 3(4):416–476, 2009.
- [13] L. Dennis, M. Fisher, M. Webster, and R. Bordini. Model checking agent programming languages. *Automated Software Engineering*, 19(1):5–63, 2012.
- [14] Ariel Fuxman, Marco Pistore, John Mylopoulos, and Paolo Traverso. Model checking early requirements specifications in Tropos. In *Requirements Engineering, 2001. Proceedings. Fifth IEEE International Symposium on*, pages 174–181. IEEE, 2001.
- [15] Paolo Giorgini, John Mylopoulos, and Roberto Sebastiani. Goal-oriented requirements analysis and reasoning in the Tropos methodology. *Engineering Applications of Artificial Intelligence*, 18(2):159–171, 2005.
- [16] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Pearson Addison-Wesley, 3 edition, 2007.
- [17] Wojciech Jamroga and Wojciech Penczek. Specification and verification of multi-agent systems. In *Lectures on Logic and Computation*, pages 210–263. Springer, 2012.
- [18] Alessio Lomuscio, Hongyang Qu, and Franco Raimondi. MCMAS: A model checker for the verification of multi-agent systems. pages 682–688, 2009.
- [19] S. Munroe, T. Miller, R. Belecheanu, M. Pechoucek, P. McBurney, and M Luck. Crossing the agent technology chasm: Lessons, experiences and challenges in commercial applications of agents. *Knowledge engineering review*, 21(4):345, 2006.
- [20] CuD. Nguyen, Anna Perini, Carole Bernon, Juan Pavn, and John Thangarajah. Testing in multi-agent systems. In Marie-Pierre Gleizes and Jorge J. Gomez-Sanz, editors, *Agent-Oriented Software Engineering X*, volume 6038 of *Lecture Notes in Computer Science*, pages 180–190. Springer Berlin Heidelberg, 2011.
- [21] L. Padgham, J. Thangarajah, Z. Zhang, and T. Miller. Model-based test oracle generation for automated unit testing of agent systems. *IEEE Transactions on Software Engineering*, 39(9):1230–1244, 2013.
- [22] L. Padgham and M. Winikoff. *Developing intelligent agent systems: A practical guide*. John Wiley & Sons, Chichester, 2004.
- [23] Lin Padgham, John Thangarajah, and Michael Winikoff. Prometheus design tool. In *Proceedings of The AAAI Conference on Artificial Intelligence*, pages 1882–1883, Chicago, USA, 2008.
- [24] Anand S Rao, Michael P Georgeff, et al. Bdi agents: From theory to practice. In *ICMAS*, volume 95, pages 312–319, 1995.
- [25] S. Shapiro, Y. Lespérance, and H. Levesque. The cognitive agents specification language and verification environment for multiagent systems. In *AAMAS'02*, pages 19–26, 2002.
- [26] Steven Shapiro, Y Lespérance, and HJ Levesque. The cognitive agents specification language and verification environment. In *Specification and Verification of Multi-agent Systems*, pages 289–315. Springer, 2010.
- [27] John Thangarajah, Gaya Buddhinath Jayatilleke, and Lin Padgham. Scenarios for system requirements traceability and testing. In *Proceedings of 10th International Conference on Autonomous Agents and Multiagent Systems AAMAS 2011*, pages 285–292, Taipei, Taiwan, 2011.
- [28] Michael Winikoff. JACK Intelligent Agents: An Industrial Strength Platform. In *Multi-Agent Programming*, pages 175–193. Springer, 2005.
- [29] Nitin Yadav and Sebastian Sardina. Reasoning about BDI agent programs using ATL-like logics. volume 7519, pages 437–449, 2012.
- [30] Zhiyong Zhang, John Thangarajah, and Lin Padgham. *Model Based Testing for Agent Systems*, pages 399–413. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [31] Zhiyong Zhang, John Thangarajah, and Lin Padgham. *Automated Testing for Intelligent Agent Systems*, pages 66–79. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

A Uniform Account of Realizability in Abstract Argumentation

Thomas Linsbichler¹ and Jörg Pührer² and Hannes Strass²

Abstract. We introduce a general framework for analyzing realizability in abstract dialectical frameworks (ADFs) and various of its subclasses. In particular, the framework applies to Dung argumentation frameworks, SETAFs by Nielsen and Parsons, and bipolar ADFs. We present a uniform characterization method for the admissible, complete, preferred and model/stable semantics. We employ this method to devise an algorithm that decides realizability for the mentioned formalisms and semantics; moreover the algorithm allows for constructing a desired knowledge base whenever one exists. The algorithm is built in a modular way and thus easily extensible to new formalisms and semantics. We have implemented our approach in answer set programming, and used the implementation to obtain several novel results on the relative expressiveness of the abovementioned formalisms.

1 Introduction

The abstract argumentation frameworks (AFs) introduced by Dung [9] have garnered increasing attention in the recent past. In his seminal paper, Dung showed how an abstract notion of argument (seen as an atomic entity) and the notion of individual attacks between arguments together could reconstruct several established KR formalisms in argumentative terms. Despite the generality of those and many more results in the field that was sparked by that paper, researchers also noticed that the restriction to *individual attacks* is often overly limiting, and devised extensions and generalizations of Dung’s frameworks: directions included generalizing individual attacks to *collective attacks* [23], leading to so-called SETAFs; others started offering a *support* relation between arguments [8], preferences among arguments [1, 22], or attacks on attacks into arbitrary depth [2]. This is only the tip of an iceberg, for a more comprehensive overview we refer to the work of Brewka, Polberg, and Woltran [5].

One of the most recent and most comprehensive generalizations of AFs has been presented by Brewka and Woltran [6] (and later continued by Brewka et al. [4]) in the form of *abstract dialectical frameworks (ADFs)*. These ADFs offer any type of link between arguments: individual attacks (as in AFs), collective attacks (as in SETAFs), and individual and collective support, to name only a few. This generality is achieved through so-called *acceptance conditions* associated to each statement. Roughly, the meaning of relationships between arguments is not fixed in ADFs, but is specified by the user for each argument in the form of Boolean functions (acceptance functions) on the argument’s parents. However, this generality comes with a price: Strass and Wallner [29] found that the complexity of the associated reasoning problems of ADFs is in general higher

than in AFs (one level up in the polynomial hierarchy). Fortunately, the subclass of *bipolar ADFs* (defined by Brewka and Woltran [6]) is as complex as AFs (for all considered semantics) while still offering a wide range of modeling capacities [29]. However, there has only been little concerted effort so far to exactly analyze and compare the expressiveness of the abovementioned languages.

This paper is about exactly analyzing means of expression for argumentation formalisms. Instead of motivating expressiveness in natural language and showing examples that some formalisms seem to be able to express but others do not, we tackle the problem in a formal way. We use a precise mathematical definition of expressiveness: a set of interpretations is *realizable* by a formalism under a semantics if and only if there exists a knowledge base of the formalism whose semantics is exactly the given set of interpretations. Studying realizability in AFs has been started by Dunne et al. [11, 10], who analyzed realizability for extension-based semantics, that is, interpretations represented by sets where arguments are either accepted (in the extension set) or not accepted (not in the extension set). While their initial work disregarded arguments that are never accepted, there have been continuations where the existence of such “invisible” arguments is ruled out [3, 20]. Dyrkolbotn [12] began to analyze realizability for labeling-based semantics of AFs, that is, three-valued semantics where arguments can be accepted (mapped to true), rejected (mapped to false) or neither (mapped to unknown). Strass [28] started to analyze the relative expressiveness of two-valued semantics for ADFs (relative with respect to related formalisms). Most recently, Pührer [26] presented precise characterizations of realizability for ADFs under several three-valued semantics, namely admissible, grounded, complete, and preferred. The term “precise characterizations” means that he gave necessary and sufficient conditions for an interpretation set to be ADF-realizable under a semantics.

The present paper continues this line of work by lifting it to a much more general setting. We combine the works of Dunne et al. [10], Pührer [26], and Strass [28] into a unifying framework, and at the same time extend them to formalisms and semantics not considered in the respective papers: we treat several formalisms, namely AFs, SETAFs, and (B)ADFs, while the previous works all used different approaches and techniques. This is possible because all of these formalisms can be seen as subclasses of ADFs that are obtained by suitably restricting the acceptance conditions.

Another important feature of our framework is that we uniformly use three-valued interpretations as the underlying model theory. In particular, this means that arguments cannot be “invisible” any more since the underlying vocabulary of arguments is always implicit in each interpretation. Technically, we always assume a fixed underlying vocabulary and consider our results parametric in that vocabulary. In contrast, for example, Dyrkolbotn [12] presents a construc-

¹ Institute of Information Systems, TU Wien, Vienna, Austria

² Computer Science Institute, Leipzig University, Leipzig, Germany

tion for realizability that introduces new arguments into the realizing knowledge base; we do not allow that. While sometimes the introduction of new arguments can make sense, for example if new information becomes available about a domain or a debate, it is not sensible in general, as these new arguments would be purely technical with an unclear dialectical meaning. Moreover, it would lead to a different notion of realizability, where most of the realizability problems would be significantly easier, if not trivial.

The paper proceeds as follows. We begin with recalling and introducing the basis and basics of our work – the formalisms we analyze and the methodology with which we analyze them. Next we introduce our general framework for realizability; the major novelty is our consistent use of so-called characterization functions, firstly introduced by Pührer [26], which we adapt to further semantics. The main workhorse of our approach will be a parametric propagate-and-guess algorithm for deciding whether a given interpretation set is realizable in a formalism under a semantics. We then analyze the relative expressiveness of the considered formalisms, presenting several new results that we obtained using an implementation of our framework. We conclude with a discussion.

2 Preliminaries

We make use of standard mathematical concepts like functions and partially ordered sets. For a function $f : X \rightarrow Y$ we denote the *update of f with a pair $(x, y) \in X \times Y$* by $f|_y^x : X \rightarrow Y$ with $z \mapsto y$ if $z = x$, and $z \mapsto f(z)$ otherwise. For a function $f : X \rightarrow Y$ and $y \in Y$, its preimage is $f^{-1}(y) = \{x \in X \mid f(x) = y\}$. A *partially ordered set* is a pair (S, \sqsubseteq) with \sqsubseteq a partial order on S . A partially ordered set (S, \sqsubseteq) is a *complete lattice* if and only if every $S' \subseteq S$ has both a greatest lower bound (glb) $\prod S' \in S$ and a least upper bound (lub) $\bigsqcup S' \in S$. A partially ordered set (S, \sqsubseteq) is a *complete meet-semilattice* iff every non-empty subset $S' \subseteq S$ has a greatest lower bound $\prod S' \in S$ (the *meet*) and every ascending chain $C \subseteq S$ has a least upper bound $\bigsqcup C \in S$.

Three-Valued Interpretations Let A be a fixed finite set of statements. An *interpretation* is a mapping $v : A \rightarrow \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ that assigns one of the truth values true (\mathbf{t}), false (\mathbf{f}) or unknown (\mathbf{u}) to each statement. An interpretation is *two-valued* if $v(A) \subseteq \{\mathbf{t}, \mathbf{f}\}$, that is, the truth value \mathbf{u} is not assigned. Two-valued interpretations v can be extended to assign truth values $v(\varphi) \in \{\mathbf{t}, \mathbf{f}\}$ to propositional formulas φ as usual.

The three truth values are partially ordered according to their information content: we have $\mathbf{u} <_i \mathbf{t}$ and $\mathbf{u} <_i \mathbf{f}$ and no other pair in $<_i$, which intuitively means that the classical truth values contain more information than the truth value unknown. As usual, we denote by \leq_i the partial order associated to the strict partial order $<_i$. The pair $(\{\mathbf{t}, \mathbf{f}, \mathbf{u}\}, \leq_i)$ forms a complete meet-semilattice with the information meet operation \prod_i . This meet can intuitively be interpreted as *consensus* and assigns $\mathbf{t} \prod_i \mathbf{t} = \mathbf{t}$, $\mathbf{f} \prod_i \mathbf{f} = \mathbf{f}$, and returns \mathbf{u} otherwise.

The information ordering \leq_i extends in a straightforward way to interpretations v_1, v_2 over A in that $v_1 \leq_i v_2$ iff $v_1(a) \leq_i v_2(a)$ for all $a \in A$. We say for two interpretations v_1, v_2 that v_2 *extends* v_1 iff $v_1 \leq_i v_2$. The set \mathcal{V} of all interpretations over A forms a complete meet-semilattice with respect to the information ordering \leq_i . The consensus meet operation \prod_i of this semilattice is given by $(v_1 \prod_i v_2)(a) = v_1(a) \prod_i v_2(a)$ for all $a \in A$. The least element of (\mathcal{V}, \leq_i) is the valuation $v_{\mathbf{u}} : A \rightarrow \{\mathbf{u}\}$ mapping all statements to unknown – the least informative interpretation. By \mathcal{V}_2 we denote the set of two-valued interpretations; they are the \leq_i -maximal elements of

the meet-semilattice (\mathcal{V}, \leq_i) . We denote by $[v]_2$ the set of all two-valued interpretations that extend v . The elements of $[v]_2$ form an \leq_i -antichain with greatest lower bound $v = \prod_i [v]_2$.

Abstract Argumentation Formalisms An *abstract dialectical framework (ADF)* is a tuple $D = (A, L, C)$ where A is a set of statements (representing positions one can take or not take in a debate), $L \subseteq A \times A$ is a set of links (representing dependencies between the positions), $C = \{C_a\}_{a \in A}$ is a collection of functions $C_a : 2^{\text{par}(a)} \rightarrow \{\mathbf{t}, \mathbf{f}\}$, one for each statement $a \in A$. The function C_a is the *acceptance condition of a* and expresses whether a can be accepted, given the acceptance status of its parents $\text{par}(a) = \{b \in S \mid (b, a) \in L\}$. We usually represent each C_a by a propositional formula φ_a over $\text{par}(a)$. For the acceptance condition C_a , we take $C_a(M \cap \text{par}(a)) = \mathbf{t}$ to hold iff M is a model of φ_a .

Brewka and Woltran [6] introduced a useful subclass of ADFs: an ADF $D = (A, L, C)$ is *bipolar* iff all links in L are supporting or attacking (or both). A link $(b, a) \in L$ is *supporting in D* iff for all $M \subseteq \text{par}(a)$, we have that $C_a(M) = \mathbf{t}$ implies $C_a(M \cup \{b\}) = \mathbf{t}$. Symmetrically, a link $(b, a) \in L$ is *attacking in D* iff for all $M \subseteq \text{par}(a)$, we have that $C_a(M \cup \{b\}) = \mathbf{t}$ implies $C_a(M) = \mathbf{t}$. Intuitively, a link $(b, a) \in L$ is supporting iff it can never be the case that there is some state of affairs where we accept a and reject b , but after additionally also accepting b do not accept a any more. Symmetrically, a link $(b, a) \in L$ is attacking iff it can never be the case that we reject a and b , but after accepting b also accept a . If a link (b, a) is both supporting and attacking then b has no actual influence on a . (But the link does not violate bipolarity.) We write BADFs as $D = (A, L^+ \cup L^-, C)$ and mean that L^+ contains all supporting links and L^- all attacking links; see also Example 1 below.³

The semantics of ADFs can be defined using an operator Γ_D over three-valued interpretations [6, 4]. For an ADF D and a three-valued interpretation v , the interpretation $\Gamma_D(v)$ is given by

$$a \mapsto \prod_i \{w(\varphi_a) \mid w \in [v]_2\}$$

That is, for each statement a , the operator returns the consensus truth value for its acceptance formula φ_a , where the consensus takes into account all possible two-valued interpretations w that extend the input valuation v . If this v is two-valued, we get $[v]_2 = \{v\}$ and thus $\Gamma_D(v)(a) = v(\varphi_a)$.

The standard semantics of ADFs are now defined as follows. For ADF D , an interpretation $v : A \rightarrow \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ is

- *admissible* iff $v \leq_i \Gamma_D(v)$;
- *complete* iff $\Gamma_D(v) = v$;
- *preferred* iff it is \leq_i -maximal admissible;
- a *two-valued model* iff it is two-valued and $\Gamma_D(v) = v$.

We denote the sets of interpretations that are admissible, complete, preferred, and two-valued models by $\text{adm}(D)$, $\text{com}(D)$, $\text{prf}(D)$ and $\text{mod}(D)$, respectively. These definitions are proper generalizations of Dung's notions for AFs: For an AF (A, R) , where $R \subseteq A \times A$ is the attack relation, the *ADF associated to (A, R)* is $D_{(A, R)} = (A, R, C)$ with $C = \{\varphi_a\}_{a \in A}$ and $\varphi_a = \bigwedge_{b: (b, a) \in R} \neg b$ for all $a \in A$. AFs inherit their semantics from the definitions for ADFs [4, Theorems 2 and 4]. In particular, an interpretation is *stable* for an AF (A, R) if and only if it is a two-valued model of $D_{(A, R)}$.

³ Other than a part of the name, there is no relationship of bipolar ADFs with the bipolar framework of Cayrol and Lagasque-Schiex [8]; Brewka and Woltran gave a more detailed comparison of the two formalisms [6].

Example 1. Consider the bipolar ADF $D = (A, L^+ \cup L^-, C)$ over vocabulary $A = \{a, b, c\}$ with

$$\varphi_a = b \wedge c, \quad \varphi_b = \neg a, \quad \varphi_c = a \vee \neg b$$

whence it follows that $L^+ = \{(b, a), (c, a), (a, c)\}$ and $L^- = \{(a, b), (b, c)\}$. (The types of links can be read off the polarities of the statements in the acceptance formulas [28, Theorem 1]; statements occurring only positively are supporting, those that occur only negatively are attacking.) Intuitively, the acceptance condition φ_a is a group support: a can only be accepted if *both* b and c are accepted. For b , we have an individual attack just like in standard AFs: b is attacked by a , and therefore only be accepted if a is not accepted. The acceptance condition of c consists of a support by a that overpowers an attack by b ; in other words, to be able to accept c , the support from a must be present or the attack from b must be absent, and if both are present then the support is stronger. (We could have specified that the attack is stronger than the support by writing $\varphi_c = a \wedge \neg b$.) Regarding the semantics of D , we find that $\text{mod}(D) = \text{prf}(D) = \{v_1\}$ with $v_1 = \{a \mapsto \mathbf{f}, b \mapsto \mathbf{t}, c \mapsto \mathbf{f}\}$. Furthermore, we have $\text{adm}(D) = \text{com}(D) = \text{prf}(D) \cup \{v_2\}$ where $v_2 = \{a \mapsto \mathbf{u}, b \mapsto \mathbf{u}, c \mapsto \mathbf{u}\}$. Intuitively, setting all statements to \mathbf{u} is always admissible; in this case it is also complete because no statement is unconditionally accepted or rejected. The non-trivial interpretation v_1 is a model of the BADF because intuitively: a is rejected since it misses the support of c ; b is accepted because the attack from a does not materialize; c is rejected because it misses support from a and at the same time is attacked by b . ■

A SETAF is a pair $S = (A, X)$ where $X \subseteq (2^A \setminus \{\emptyset\}) \times A$ is the (set) attack relation. We define three-valued counterparts of the semantics introduced by Nielsen and Parsons [23], following the same conventions as in three-valued semantics of AFs [7] and argumentation formalisms in general. Given a statement $a \in A$ and an interpretation v we say that a is *acceptable* with respect to v if and only if $\forall (B, a) \in X \exists a' \in B : v(a') = \mathbf{f}$ and a is *unacceptable* with respect to v if and only if $\exists (B, a) \in X \forall a' \in B : v(a') = \mathbf{t}$.

For an interpretation $v : A \rightarrow \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$ it holds that

- $v \in \text{adm}(S)$ iff for all $a \in A$, a is acceptable wrt. v if $v(a) = \mathbf{t}$ and a is unacceptable wrt. v if $v(a) = \mathbf{f}$;
- $v \in \text{com}(S)$ iff for all $a \in A$, a is acceptable wrt. v iff $v(a) = \mathbf{t}$ and a is unacceptable wrt. v iff $v(a) = \mathbf{f}$;
- $v \in \text{prf}(S)$ iff v is \leq_i -maximal admissible; and
- $v \in \text{mod}(S)$ iff $v \in \text{adm}(S)$ and $\nexists a \in A : v(a) = \mathbf{u}$.

For a SETAF $S = (A, X)$ the corresponding ADF D_S has acceptance formula $\varphi_a = \bigwedge_{(B, a) \in X} \bigvee_{a' \in B} \neg a'$ for each statement $a \in A$.

Proposition 1. For any SETAF $S = (A, X)$ it holds that $\sigma(S) = \sigma(D_S)$, where $\sigma \in \{\text{adm}, \text{com}, \text{prf}, \text{mod}\}$.

Proof. Given interpretation v and statement a , it holds that $\Gamma_{D_S}(v)(a) = \mathbf{t}$ iff $\forall w \in [v]_2 : w(a) = \mathbf{t}$ iff $\forall (B, a) \in X \exists a' \in B : v(a') = \mathbf{f}$ iff a is acceptable wrt. v and $\Gamma_{D_S}(v)(a) = \mathbf{f}$ iff $\forall w \in [v]_2 : w(a) = \mathbf{f}$ iff $\exists (B, a) \in X \forall a' \in B : v(a') = \mathbf{t}$ iff a is unacceptable wrt. v . Hence $\sigma(S) = \sigma(D_S)$ for $\sigma \in \{\text{adm}, \text{com}, \text{prf}, \text{mod}\}$. □

Realizability A set $V \subseteq \mathcal{V}$ of interpretations is *realizable* in a formalism \mathcal{F} under a semantics σ if and only if there exists a knowledge base $\text{kb} \in \mathcal{F}$ having exactly $\sigma(\text{kb}) = V$. Pührer [26] characterized realizability for ADFs under various three-valued semantics.

We will reuse the central notions for capturing the complete semantics in this work.

Definition 1 (Pührer [26]). Let V be a set of interpretations. A function $f : \mathcal{V}_2 \rightarrow \mathcal{V}_2$ is a *com-characterization* of V iff: for each $v \in \mathcal{V}$ we have $v \in V$ iff for each $a \in A$:

- $v(a) \neq \mathbf{u}$ implies $f(v_2)(a) = v(a)$ for all $v_2 \in [v]_2$ and
- $v(a) = \mathbf{u}$ implies $f(v'_2)(a) = \mathbf{t}$ and $f(v''_2)(a) = \mathbf{f}$ for some $v'_2, v''_2 \in [v]_2$. ▲

Intuitively, a *com-characterization* f assigns the Boolean value $f(v)(a)$ to a statement a that the acceptance condition of a would have under v in an ADF that has V as its complete semantics. From a function of this kind we can build a corresponding ADF by the following construction. For $f : \mathcal{V}_2 \rightarrow \mathcal{V}_2$, we define D_f as the ADF where the acceptance formula for each statement a is given by

$$\varphi_a^f = \bigvee_{w \in \mathcal{V}_2, f(w)(a) = \mathbf{t}} \phi_w \quad \text{with} \quad \phi_w = \bigwedge_{w(a') = \mathbf{t}} a' \wedge \bigwedge_{w(a') = \mathbf{f}} \neg a'$$

Observe that for any $v \in \mathcal{V}_2$ we have $v(\phi_w) = \mathbf{t}$ iff $v = w$ by definition. Intuitively, the acceptance condition φ_a^f is constructed such that v is a model of φ_a^f if and only if we find $f(v)(a) = \mathbf{t}$.

Proposition 2 (Pührer [26]). Let $V \subseteq \mathcal{V}$ be a set of interpretations. (1) For each ADF D with $\text{com}(D) = V$, there is a *com-characterization* f_D for V ; (2) for each *com-characterization* $f : \mathcal{V}_2 \rightarrow \mathcal{V}_2$ for V we have $\text{com}(D_f) = V$.

The result shows that V can be realized under complete semantics if and only if there is a *com-characterization* for V .

3 A General Framework for Realizability

The underlying idea of our framework is that all abstract argumentation formalisms introduced in the previous section can be viewed as subclasses of ADFs. This is clear for ADFs themselves and for BADFs by definition; for (SET)AFs it is fairly easy to see. However, knowing that these formalisms can be recast as ADFs is not enough. To employ this knowledge for realizability, we must precisely characterize the corresponding subclasses in terms of restricting the ADFs' acceptance functions. Fortunately, this is also possible and paves the way for the framework we present in this section. Most importantly, we will make use of the fact that different formalisms and different semantics can be characterized modularly, that is, independently of each other.

Towards a uniform account of realizability for ADFs under different semantics, we start with a new characterization of realizability for ADFs under admissible semantics that is based on a notion similar in spirit to *com-characterizations*.

Definition 2. Let V be a set of interpretations. A function $f : \mathcal{V}_2 \rightarrow \mathcal{V}_2$ is an *adm-characterization* of V iff: for each $v \in \mathcal{V}$ we have $v \in V$ iff for every $a \in A$:

- $v(a) \neq \mathbf{u}$ implies $f(v_2)(a) = v(a)$ for all $v_2 \in [v]_2$. ▲

Similar as for a *com-characterization*, an *adm-characterization* f assigns the value $f(v)(a)$ to a statement a that the acceptance condition of a would evaluate to under v in an ADF that has V as its admissible semantics. Note that the only difference to Definition 1 is dropping the second condition related to statements with truth value \mathbf{u} . While,

the two conditions in Definition 1 capture the relation $\Gamma_{D_f}(v) = v$, the remaining one in Definition 2 boils down to $v \leq_i \Gamma_{D_f}(v)$ that defines the admissible semantics.

Proposition 3. *Let $V \subseteq \mathcal{V}$ be a set of interpretations. (1) For each ADF D such that $\text{adm}(D) = V$, there is an adm -characterization f_D for V ; (2) for each adm -characterization $f : \mathcal{V}_2 \rightarrow \mathcal{V}_2$ for V we have $\text{adm}(D_f) = V$.*

Proof. (1) We define the function $f_D : \mathcal{V}_2 \rightarrow \mathcal{V}_2$ as $f_D(v_2)(a) = v_2(\varphi_a)$ for every $v_2 \in \mathcal{V}_2$ and $a \in A$ where φ_a is the acceptance formula of a in D . We will show that f_D is an adm -characterization for $V = \text{adm}(D)$. Let v be an interpretation. Consider the case $v \in \text{adm}(D)$ and $v(a) \neq u$ for some $a \in A$ and some $v_2 \in [v]_2$. From $v \leq_i \Gamma_D(v)$ we get $v_2(\varphi_a) = v(a)$. By definition of f_D it follows that $f_D(v_2)(a) = v(a)$. Now assume $v \notin \text{adm}(D)$ and consequently $v \not\leq_i \Gamma_D(v)$. There must be some $a \in A$ such that $v(a) \neq \mathbf{u}$ and $v(a) \neq \Gamma_D(v)(a)$. Hence, there is some $v_2 \in [v]_2$ with $v_2(\varphi_a) \neq v(a)$ and $f_D(v_2)(a) \neq v(a)$ by definition of f_D . Thus, f_D is an adm -characterization for V .

(2) Observe that for every two-valued interpretation v_2 and every $a \in A$ we have $f(v_2)(a) = v_2(\varphi_a^f)$. (\subseteq): Let $v \in \text{adm}(D_f)$ be an interpretation and $a \in A$ a statement such that $v(a) \neq \mathbf{u}$. Let v_2 be a two-valued interpretation with $v_2 \in [v]_2$. Since $v \leq_i \Gamma_{D_f}(v)$ we have $v(a) = v_2(\varphi_a^f)$. Therefore, by our observation it must also hold that $f(v_2)(a) = v(a)$. Thus, by Definition 2, $v \in V$. (\supseteq): Consider an interpretation v such that $v \notin \text{adm}(D_f)$. We show that $v \notin V$. From $v \notin \text{adm}(D_f)$ we get $v \not\leq_i \Gamma_{D_f}(v)$. There must be some $a \in A$ such that $v(a) \neq \mathbf{u}$ and $v(a) \neq \Gamma_{D_f}(v)(a)$. Hence, there is some $v_2 \in [v]_2$ with $v_2(\varphi_a^f) \neq v(a)$ and consequently $f(v_2)(a) \neq v(a)$. Thus, by Definition 2 we have $v \notin V$. \square

When listing sets of interpretations in examples, for the sake of readability we represent three-valued interpretations by sequences of truth values, tacitly assuming that the underlying vocabulary is given and has an associated total ordering. For example, for the vocabulary $A = \{a, b, c\}$ we represent the interpretation $\{a \mapsto \mathbf{t}, b \mapsto \mathbf{f}, c \mapsto \mathbf{u}\}$ by the sequence **tfu**.

Example 2. Consider the sets $V_1 = \{\mathbf{uuu}, \mathbf{tff}, \mathbf{ftu}\}$ and $V_2 = \{\mathbf{tff}, \mathbf{ftu}\}$ of interpretations over $A = \{a, b, c\}$. The mapping $f = \{\mathbf{ttt} \mapsto \mathbf{ftt}, \mathbf{ttf} \mapsto \mathbf{tft}, \mathbf{tft} \mapsto \mathbf{ttt}, \mathbf{ttf} \mapsto \mathbf{tff}, \mathbf{ftt} \mapsto \mathbf{ftf}, \mathbf{ftf} \mapsto \mathbf{ftt}, \mathbf{fft} \mapsto \mathbf{tff}, \mathbf{fff} \mapsto \mathbf{ftf}\}$ is an adm -characterization for V_1 . Thus, the ADF D_f has V_1 as its admissible interpretations. Indeed, the realizing ADF has the following acceptance conditions:

$$\begin{aligned}\varphi_a^f &\equiv (a \wedge b \wedge \neg c) \vee (a \wedge \neg b) \vee (\neg a \wedge \neg b \wedge c) \\ \varphi_b^f &\equiv (a \wedge c) \vee (\neg a \wedge b) \vee (\neg a \wedge \neg b \wedge \neg c) \\ \varphi_c^f &\equiv (a \wedge b) \vee (\neg a \wedge b \wedge \neg c) \vee (\neg b \wedge c)\end{aligned}$$

For V_2 no adm -characterization exists because $\mathbf{uuu} \notin V_2$, but the implication of Definition 2 trivially holds for a, b , and c . \blacksquare

We have seen that the construction D_f for realizing under complete semantics can also be used for realizing a set V of interpretations under admissible semantics. The only difference is that we here require f to be an adm -characterization instead of a com -characterization for V . Note that admissible semantics can be characterized by properties that are easier to check than existence of an adm -characterization (see the work of Pührer [26]). However, using the same type of characterizations for different semantics allows for a unified approach for checking realizability and constructing a realizing ADF in case one exists.

For realizing under the model semantics, we can likewise present an adjusted version of com -characterizations.

Definition 3. Let $V \subseteq \mathcal{V}$ be a set of interpretations. A function $f : \mathcal{V}_2 \rightarrow \mathcal{V}_2$ is a mod -characterization of V if and only if: (1) f is defined on V (that is, $V \subseteq \mathcal{V}_2$) and (2) for each $v \in \mathcal{V}_2$, we have $v \in V$ iff $f(v) = v$. \blacktriangle

As we can show, there is a one-to-one correspondence between mod -characterizations and ADF realizations.

Proposition 4. *Let $V \subseteq \mathcal{V}$ be a set of interpretations. (1) For each ADF D such that $\text{mod}(D) = V$, there is a mod -characterization f_D for V ; (2) vice versa, for each mod -characterization $f : \mathcal{V}_2 \rightarrow \mathcal{V}_2$ for V we find $\text{mod}(D_f) = V$.*

A related result was given by Strass [28, Proposition 10]. The characterization we presented here fits into the general framework of this paper and is directly usable for our realizability algorithm. The next result summarizes how ADF realizability can be captured by different types of characterizations for the semantics we considered so far.

Theorem 5. *Let $V \subseteq \mathcal{V}$ be a set of interpretations and consider $\sigma \in \{\text{adm}, \text{com}, \text{mod}\}$. There is an ADF D such that $\sigma(D) = V$ if and only if there is a σ -characterization for V .*

The preferred semantics of an ADF D is closely related to its admissible semantics as, by definition, the preferred interpretations of D are its \leq_i -maximal admissible interpretations. As a consequence we can also describe preferred realizability in terms of adm -characterizations. We use the lattice-theoretic standard notation $\max_{\leq_i} V$ to denote the \leq_i -maximal elements of a given set V .

Corollary 6. *Let $V \subseteq \mathcal{V}$ be a set of interpretations. There is an ADF D with $\text{prf}(D) = V$ iff there is an adm -characterization for some $V' \subseteq \mathcal{V}$ with $V \subseteq V'$ and $\max_{\leq_i} V' = V$.*

Finally, we give a result on the complexity of deciding realizability for the mentioned formalisms and semantics. We assume here that the representation of an interpretation-set V over vocabulary A has size $\Theta(3^{|A|})$, that is, the size grows asymptotically in the order of $3^{|A|}$. A possible encoding could be a bit string of length $3^{|A|}$ where the presence (or absence) of each $v \in V$ is encoded by a 1 (or 0) at a particular position in the string. There might be specific V with smaller possible representations, but we have no grounds to presume a representation that is exponentially better in the general case.

Proposition 7. *Let $\mathcal{F} \in \{\text{AF}, \text{SETAF}, \text{BADF}, \text{ADF}\}$ be a formalism and $\sigma \in \{\text{adm}, \text{com}, \text{prf}, \text{mod}\}$ be a semantics. The decision problem “Given a vocabulary A and a set $V \subseteq \mathcal{V}$ of interpretations over A , is there a $\text{kb} \in \mathcal{F}$ such that $\sigma(\text{kb}) = V$?” can be decided in nondeterministic time that is polynomial in the size of V .*

Proof. Roughly, we guess a function $f : \mathcal{V}_2 \rightarrow \mathcal{V}_2$ and verify that it is a σ -characterization. Such a function f can be represented in size $O(2^{|A|} \cdot |A|)$, that is, at most polynomial in the input of size $O(3^{|A|})$: the fact that $n \cdot 2^n \in o(3^n) \subseteq O(3^n)$ follows from

$$\lim_{n \rightarrow \infty} \frac{n \cdot 2^n}{3^n} = \lim_{n \rightarrow \infty} \frac{n}{\left(\frac{3}{2}\right)^n} \stackrel{*}{=} \lim_{n \rightarrow \infty} \frac{1}{\ln \frac{3}{2} \cdot \left(\frac{3}{2}\right)^n} = 0$$

where the starred equality holds by L'Hôpital's rule.

To verify that the guessed f is indeed a σ -characterization, we check (some of) the properties of Definition 1. For $\sigma = \text{com}$, this can be done in polynomial time as follows: for each $v \in \mathcal{V}$ and $a \in A$, we

look at the set $[v]_2 \subseteq \mathcal{V}_2$ (which is at most polynomial in the input) and check for the respective witness interpretations (if $v(a) = \mathbf{u}$) or their absence (if $v(a) \neq \mathbf{u}$). For $\sigma = \text{adm}$, there are even less conditions to check. For $\sigma = \text{mod}$, we compute the set F of fix-points of f (by going through \mathcal{V} once and checking $f(v) = v$ for each $v \in \mathcal{V}$) and verify that $F = V$. For $\sigma = \text{prf}$, we guess the V' (with $V \subseteq V' \subseteq \mathcal{V}$) from Corollary 6 alongside f and verify that f is an adm -characterization for V' and that $\max_{\leq_i} V' = V$. \square

3.1 Deciding Realizability: Algorithm 1

Our main algorithm for deciding realizability is a propagate-and-guess algorithm in the spirit of the DPLL algorithm for deciding propositional satisfiability [19]. It is generic with respect to (1) the formalism \mathcal{F} and (2) the semantics σ for which should be realized. To this end, the propagation part of the algorithm is kept exchangeable and will vary depending on formalism and semantics. Roughly, in the propagation step the algorithm uses the desired set V of interpretations to derive certain necessary properties of the realizing knowledge base (line 2). This is the essential part of the algorithm: the derivation rules (*propagators*) used there are based on characterizations of realizability with respect to formalism and semantics. (Propagators will be explained in detail in the next two subsections.) Once propagation of properties has reached a fixed point (line 7), the algorithm checks whether the derived information is sufficient to construct a knowledge base. If so, the knowledge base can be constructed and returned (line 9). Otherwise (no more information can be obtained through propagation and there is not enough information to construct a knowledge base yet), the algorithm guesses another assignment for the characterization (line 11) and calls itself recursively.

The main data structure that Algorithm 1 operates on is a set of triples (v, a, \mathbf{x}) consisting of a two-valued interpretation $v \in \mathcal{V}_2$, an atom $a \in A$ and a truth value $\mathbf{x} \in \{\mathbf{t}, \mathbf{f}\}$. This data structure is intended to represent the σ -characterizations introduced in Definitions 1 to 3. There, a σ -characterization is a function $f : \mathcal{V}_2 \rightarrow \mathcal{V}_2$ from two-valued interpretations to two-valued interpretations. However, as the algorithm builds the σ -characterization step by step and there might not even be a σ -characterization in the end (because V is not realizable), we use a set F of triples (v, a, \mathbf{x}) to be able to represent both partial and incoherent states of affairs. The σ -characterization candidate induced by F is partial if we have that for some v and a , neither $(v, a, \mathbf{t}) \in F$ nor $(v, a, \mathbf{f}) \in F$; likewise, the candidate is incoherent if for some v and a , both $(v, a, \mathbf{t}) \in F$ and $(v, a, \mathbf{f}) \in F$. If F is neither partial nor incoherent, it gives rise to a unique σ -characterization that can be used to construct the knowledge base realizing the desired set of interpretations. The correspondence to the characterization-function is then such that $f(v)(a) = \mathbf{x}$ iff $(v, a, \mathbf{x}) \in F$.

In our presentation of the algorithm we focused on its main features, therefore the guessing step (line 11) is completely “blind”. It is possible to use techniques known from constraint satisfaction problems, such as shaving (removing guessing possibilities that directly lead to inconsistency). Finally, we remark that the algorithm can be extended to enumerate all possible realizations of a given interpretation set – by keeping all choice points in the guessing step and thus exhaustively exploring the whole search space.

In the case where the constructed relation F becomes functional at some point, the algorithm returns a realizing knowledge base $kb_\sigma^\mathcal{F}(F)$. For ADFs, this just means that we denote by f the σ -characterization represented by F and set $kb_\sigma^{\text{ADF}}(F) = D_f$. For the remaining formalisms we will introduce the respective constructions

Algorithm 1 $\text{realize}(\mathcal{F}, \sigma, V, F)$

Input: • a formalism \mathcal{F}
• a semantics σ for \mathcal{F}
• a set V of interpretations $v : A \rightarrow \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$
• a relation $F \subseteq \mathcal{V}_2 \times A \times \{\mathbf{t}, \mathbf{f}\}$, initially empty

Output: a $\text{kb} \in \mathcal{F}$ with $\sigma(\text{kb}) = V$ or “no” if none exists

- 1: **repeat**
- 2: **set** $F_\Delta := \bigcup_{p \in P_\sigma^\mathcal{F}} p(V, F) \setminus F$
- 3: **set** $F := F \cup F_\Delta$
- 4: **if** $\exists v \in \mathcal{V}_2, \exists a \in A : \{(v, a, \mathbf{t}), (v, a, \mathbf{f})\} \subseteq F$ **then**
- 5: **return** “no”
- 6: **end if**
- 7: **until** $F_\Delta = \emptyset$
- 8: **if** $\forall v \in \mathcal{V}_2, \forall a \in A, \exists x \in \{\mathbf{t}, \mathbf{f}\} : (v, a, x) \in F$ **then**
- 9: **return** $kb_\sigma^\mathcal{F}(F)$
- 10: **end if**
- 11: **choose** $v \in \mathcal{V}_2, a \in A$ with $(v, a, \mathbf{t}) \notin F, (v, a, \mathbf{f}) \notin F$
- 12: **if** $\text{realize}(\mathcal{F}, \sigma, V, F \cup \{(v, a, \mathbf{t})\}) \neq \text{“no”}$ **then**
- 13: **return** $\text{realize}(\mathcal{F}, \sigma, V, F \cup \{(v, a, \mathbf{t})\})$
- 14: **else**
- 15: **return** $\text{realize}(\mathcal{F}, \sigma, V, F \cup \{(v, a, \mathbf{f})\})$
- 16: **end if**

in later subsections.

The algorithm is parametric in two dimensions, namely with respect to the formalism \mathcal{F} and with respect to the semantics σ . These two aspects come into the algorithm via so-called *propagators*. A propagator is a formalism-specific or semantics-specific set of derivation rules. Given a set V of desired interpretations and a partial σ -characterization F , a propagator p derives new triples (v, a, \mathbf{x}) that must necessarily be part of any total σ -characterization f for V such that f extends F . In what follows, we present semantics propagators for admissible, complete and two-valued model (in (SET)AF terms stable) semantics, and formalism propagators for BADFs, AFs, and SETAFs.

3.2 Semantics Propagators

The semantics propagators are defined in Figure 1. They are directly derived from the properties of σ -characterizations presented in Definitions 1 to 3. While the definitions provide exact conditions to check whether a given function is a σ -characterization, the propagators allow us to derive definite values of partial characterizations that are necessary to fulfill the conditions for being a σ -characterization.

For admissible semantics, the condition for a function f to be an adm -characterization of a desired set of interpretations V (cf. Definition 2) can be split into a condition for desired interpretations $v \in V$ and two conditions for undesired interpretations $v \notin V$. Propagator p_{adm}^∞ derives new triples by considering interpretations $v \in V$. Here, for all two-valued interpretations v_2 that extend v , the value $f(v_2)$ has to be in accordance with v on v 's Boolean part, that is, the algorithm adds $(v_2, a, v(a))$ whenever $v(a) \neq \mathbf{u}$. On the other hand, p_{adm}^\neq derives new triples for $v \notin V$ in order to ensure that there is a two-valued interpretation v_2 extending v where $f(v_2)$ differs from v on a Boolean value of v . Note that while p_{adm}^∞ immediately allows us to derive information about F for each desired interpretation $v \in V$, propagator p_{adm}^\neq is much weaker in the sense that it only derives a triple of F if there is no other way to meet the conditions for an undesired interpretation. Special treatment is required for the interpretation $v_{\mathbf{u}}$ that maps all statements to \mathbf{u} and is admissible for every

$$\begin{aligned}
p_{adm}^{\infty}(V, F) &= \{(v_2, a, v(a)) \mid v \in V, v_2 \in [v]_2, v(a) \neq \mathbf{u}\} \\
p_{adm}^{\infty}(V, F) &= \{(v_2, a, \neg v(a)) \mid v \in \mathcal{V} \setminus V, v_2 \in [v]_2, \\
&\quad v(a) \neq \mathbf{u}, \forall b \in A \setminus v^{-1}(\mathbf{u}), \forall v'_2 \in [v]_2 : \\
&\quad (a, v_2) \neq (b, v'_2) \rightarrow (v'_2, b, v(b)) \in F\} \\
p_{adm}^{\ddagger}(V, F) &= \{(v, a, \mathbf{t}), (v, a, \mathbf{f}) \mid v \in \mathcal{V}_2, a \in A, v_{\mathbf{u}} \notin V\} \\
p_{mod}^{\infty}(V, F) &= \{(v, a, v(a)) \mid v \in V, a \in A\} \\
p_{mod}^{\infty}(V, F) &= \{(v, a, \neg v(a)) \mid v \in \mathcal{V}_2 \setminus V, a \in A, \\
&\quad \forall c \in A \setminus \{a\} : (v, c, v(c)) \in F\} \\
p_{mod}^{\ddagger}(V, F) &= \{(v, a, \mathbf{t}), (v, a, \mathbf{f}) \mid v \in \mathcal{V}_2, a \in A, V \not\subseteq \mathcal{V}_2\}
\end{aligned}$$

Figure 1: Semantics propagators for the complete ($P_{com}^{ADF} = \{p_{com}^{\infty, \mathbf{tf}}, p_{com}^{\infty, \mathbf{u}}, p_{com}^{\ddagger, \mathbf{tf}}, p_{com}^{\ddagger, \mathbf{u}}\}$ with $p_{com}^{\infty, \mathbf{tf}}(V, F) = p_{adm}^{\infty}(V, F)$), admissible ($P_{adm}^{ADF} = \{p_{adm}^{\infty}, p_{adm}^{\ddagger}, p_{adm}^{\ddagger}\}$), and model semantics ($P_{mod}^{ADF} = \{p_{mod}^{\infty}, p_{mod}^{\ddagger}, p_{mod}^{\ddagger}\}$).

ADF. This is not captured by p_{adm}^{∞} and p_{adm}^{\ddagger} , as these deal only with interpretations that have Boolean mappings. Thus, propagator p_{adm}^{\ddagger} serves to check whether $v_{\mathbf{u}} \in V$. If this is not the case, the propagator immediately makes the relation F incoherent and the algorithm correctly answers “no”.

For complete semantics and interpretations $v \in V$, propagator $p_{com}^{\infty, \mathbf{tf}}$ derives triples just like in the admissible case. Propagator $p_{com}^{\infty, \mathbf{u}}$ deals with statements $a \in A$ having $v(a) = \mathbf{u}$ for which there have to be at least two $v_2, v'_2 \in [v]_2$ having $f(v_2)(a) = \mathbf{t}$ and $f(v'_2)(a) = \mathbf{f}$. Hence $p_{com}^{\infty, \mathbf{u}}$ derives triple $(v_2, a, \neg \mathbf{x})$ if for all other $v'_2 \in [v]_2$ we find a triple (v'_2, a, \mathbf{x}) . For interpretations $v \notin V$ it must hold that there is some $a \in A$ such that (i) $v(a) \neq \mathbf{u}$ and $f(v_2)(a) \neq v(a)$ for some $v_2 \in [v]_2$ or (ii) $v(a) = \mathbf{u}$ but for all $v_2 \in [v]_2$, $f(v_2)$ assigns the same Boolean truth value \mathbf{x} to a . Now if neither (i) nor (ii) can be fulfilled by any statement $b \in A \setminus \{a\}$ due to the current contents of F , propagators $p_{com}^{\ddagger, \mathbf{tf}}$ and $p_{com}^{\ddagger, \mathbf{u}}$ derive triple $(v_2, a, \neg v(a))$ for $v(a) \neq \mathbf{u}$ if needed for a to fulfill (i) and $(v_2, a, \neg \mathbf{x})$ for $v(a) = \mathbf{u}$ if needed for a to fulfill (ii), respectively.

Example 3. Consider the set $V_3 = \{\mathbf{uuu}, \mathbf{fuu}, \mathbf{uuf}, \mathbf{ftf}\}$. First, we consider a run of $realize(\text{ADF}, adm, V_3, \emptyset)$. In the first iteration, propagator p_{adm}^{∞} ensures that F_{Δ} in line 2 contains $(\mathbf{fff}, a, \mathbf{f})$, $(\mathbf{ftf}, a, \mathbf{f})$, $(\mathbf{ftf}, c, \mathbf{f})$, and $(\mathbf{fff}, c, \mathbf{f})$. Based on the latter three tuples and $\mathbf{fuf} \notin V_3$, propagator p_{adm}^{\ddagger} derives $(\mathbf{fff}, a, \mathbf{t})$ in the second iteration which together with $(\mathbf{fff}, a, \mathbf{f})$ causes the algorithm to return “no”. Consequently, V_3 is not adm -realizable. A run of $realize(\text{ADF}, com, V_3, \emptyset)$ on the other hand returns com -characterization f for V_3 that maps \mathbf{ttf} to \mathbf{ttf} , \mathbf{ftt} to \mathbf{ftt} , \mathbf{ftf} and \mathbf{fff} to \mathbf{ftf} and all other $v_2 \in \mathcal{V}_2$ to \mathbf{fff} . Hence, ADF D_f , given by the acceptance conditions $\varphi_a^f = a \wedge b \wedge \neg c$, $\varphi_b^f = (\neg a \wedge b \wedge \neg c) \vee (\neg a \wedge \neg b \wedge \neg c)$, and $\varphi_c^f = \neg a \wedge b \wedge c$, has V_3 as its complete semantics. ■

Finally, for two-valued model semantics, propagator p_{mod}^{∞} derives new triples by looking at interpretations $v \in V$. For those, we must find $f(v) = v$ in each mod -characterization f by definition. Thus the algorithm adds $(v, a, v(a))$ for each $a \in A$ to the partial characterization F . Propagator p_{mod}^{\ddagger} looks at interpretations $v \in \mathcal{V}_2 \setminus V$, for which it must hold that $f(v) \neq v$. Thus there must be a statement $a \in A$ with $v(a) \neq f(v)(a)$, which is exactly what this propagator derives whenever it is clear that there is only one statement candidate left. This, in turn, is the case whenever all $b \in A$ with the opposite truth value $\neg v(a)$ and all $c \in A$ with $c \neq a$ cannot coherently become the necessary witness any more. The propagator p_{mod}^{\ddagger} checks whether $V \subseteq \mathcal{V}_2$, that is, the desired set of interpret-

Algorithm 2 $realizePrf(\mathcal{F}, V)$

Input: • a formalism \mathcal{F}

• a set V of interpretations $v : A \rightarrow \{\mathbf{t}, \mathbf{f}, \mathbf{u}\}$

Output: Return some $\text{kb} \in \mathcal{F}$ with $prf(\text{kb}) = V$ if one exists or “no” otherwise.

```

1: if  $\max_{\leq_i} V \neq V$  then
2:   return “no”
3: end if
4: set  $V^{<i} := \{v \in \mathcal{V} \mid \exists v' \in V : v <_i v'\}$ 
5: set  $X := \emptyset$ 
6: repeat
7:   choose  $V' \subseteq V^{<i}$  with  $V' \notin X$ 
8:   set  $X := X \cup \{V'\}$ 
9:   set  $V^{adm} := V \cup V'$ 
10:  if  $realize(\mathcal{F}, adm, V^{adm}, \emptyset) \neq \text{“no”}$  then
11:    return  $realize(\mathcal{F}, adm, V^{adm}, \emptyset)$ 
12:  end if
13: until  $\forall V' \subseteq V^{<i} : V' \in X$ 
14: return “no”

```

tations consists entirely of two-valued interpretations. In that case this propagator makes the relation F incoherent, following a similar strategy as p_{adm}^{\ddagger} .

The Special Case of Preferred Semantics Realizing a given set of interpretations V under preferred semantics requires special treatment. We do not have a σ -characterization function for $\sigma = prf$ at hand to directly check realizability of V but have to find some $V' \subseteq \{v \in \mathcal{V} \mid \exists v' \in V : v <_i v'\}$ such that $V \cup V'$ is realizable under admissible semantics (cf. Corollary 6). Algorithm 2 implements this idea by guessing such a V' (line 7) and then using Algorithm 1 to try to realize $V \cup V'$ under admissible semantics (line 11). If $realize$ returns a knowledge base kb realizing $V \cup V'$ under adm we can directly use kb as solution of $realizePrf$ since it holds that $prf(\text{kb}) = V$, given that V is an \leq_i -antichain (line 2).

3.3 Formalism Propagators

When constructing an ADF realizing a given set V of interpretations under a semantics σ , the function $kb_{\sigma}^{ADF}(F)$ makes use of the σ -characterization given by F in the following way: v is a model of the acceptance condition φ_a if and only if we find $(v, a, \mathbf{t}) \in F$. Now as bipolar ADFs, SETAFs and AFs are all subclasses of ADFs by restricting the acceptance conditions of statements, these restrictions also carry over to the σ -characterizations. The propagators defined

$$\begin{aligned}
p^{\text{SETAF}}(V, F) &= \{(v_{\mathbf{f}}, a, \mathbf{t}) \mid a \in A\} \cup \{(w, a, \mathbf{t}) \mid (v, a, \mathbf{t}) \in F, w \in \mathcal{V}_2, w <_t v\} \cup \{(w, a, \mathbf{f}) \mid (v, a, \mathbf{f}) \in F, w \in \mathcal{V}_2, v <_t w\} \\
p^{\text{AF}}(V, F) &= p^{\text{SETAF}}(V, F) \cup \{(v_1 \sqcup_t v_2, a, \mathbf{t}) \mid (v_1, a, \mathbf{t}) \in F, (v_2, a, \mathbf{t}) \in F\} & L^+ &= \{(b, a) \mid (v, a, \mathbf{f}) \in F, v(b) = \mathbf{f}, (v|_t^b, a, \mathbf{t}) \in F\} \\
p^{\text{BADF}}(V, F) &= \{(v|_t^b, a, \mathbf{x}) \mid (v, a, \mathbf{x}) \in F, (w, a, \neg \mathbf{x}) \in F, w(b) = \mathbf{f}, (w|_t^b, a, \mathbf{x}) \in F\} & L^- &= \{(b, a) \mid (v, a, \mathbf{t}) \in F, v(b) = \mathbf{f}, (v|_t^b, a, \mathbf{f}) \in F\}
\end{aligned}$$

Figure 2: Formalism propagators. For formalism $\mathcal{F} \in \{\text{AF}, \text{SETAF}, \text{BADF}\}$ and any $\sigma \in \{\text{adm}, \text{com}, \text{prf}, \text{mod}\}$, we set the respective propagator for \mathcal{F} to $P_\sigma^\mathcal{F} = P_\sigma^{\text{ADDF}} \cup \{p^\mathcal{F}\}$ with $p^\mathcal{F}$ as defined above. L^+ and L^- define link polarities for kb_σ^{BADF} .

in Figure 2 use structural knowledge on the form of acceptance conditions of the respective formalisms to reduce the search space or to induce incoherence of F whenever V is not realizable.

Bipolar ADFs For bipolar ADFs, we use the fact that each of their links must have at least one polarity, that is, must be supporting or attacking. Therefore, if a link is not supporting, it must be attacking, and vice versa. For canonical realization, we obtain the polarities of links, that is, the sets L^+ and L^- , as defined in Figure 2.

AFs To explain the AF propagators, we first need some more definitions. On the two classical truth values, we define the truth ordering $\mathbf{f} <_t \mathbf{t}$, whence the operations \sqcup_t and \sqcap_t with $\mathbf{f} \sqcup_t \mathbf{t} = \mathbf{t}$ and $\mathbf{f} \sqcap_t \mathbf{t} = \mathbf{f}$ result. These operations can be lifted pointwise to two-valued interpretations as usual, i.e., $(v_1 \sqcup_t v_2)(a) = v_1(a) \sqcup_t v_2(a)$ and $(v_1 \sqcap_t v_2)(a) = v_1(a) \sqcap_t v_2(a)$. Again, the reflexive version of $<_t$ is denoted by \leq_t . The pair (\mathcal{V}_2, \leq_t) of two-valued interpretations ordered by the truth ordering forms a complete lattice with glb \sqcap_t and lub \sqcup_t . This complete lattice has the least element $v_{\mathbf{f}} : A \rightarrow \{\mathbf{f}\}$, the interpretation mapping all statements to false, and the greatest element $v_{\mathbf{t}} : A \rightarrow \{\mathbf{t}\}$ mapping all statements to true, respectively.

Acceptance conditions of AF-based ADFs have the form of conjunctions of negative literals. In the complete lattice (\mathcal{V}_2, \leq_t) , the model sets of AF acceptance conditions correspond to the lattice-theoretic concept of an *ideal*, a subset of \mathcal{V}_2 that is downward-closed with respect to \leq_t and upward-closed with respect to \sqcup_t . The propagator directly implements these closure properties: application of p^{AF} ensures that when a σ -characterization F that is neither incoherent nor partial is found in line 8 of Algorithm 1, then there is, for each $a \in A$, an interpretation v_a such that $(v_a, a, \mathbf{t}) \in F$ and $v \leq_t v_a$ for each $(v, a, \mathbf{t}) \in F$. Hence v_a is crucial for the acceptance condition, or in AF terms the attacks, of a and we can define $kb_\sigma^{\text{AF}}(F) = (A, \{(b, a) \mid a, b \in A, v_a(b) = \mathbf{f}\})$.

SETAFs The propagator for SETAFs, p^{SETAF} , is a weaker version of that of AFs, since we cannot presume upward-closure with respect to \sqcup_t . In SETAF-based ADFs the acceptance formula is in *conjunctive normal form* containing only negative literals. By a transformation preserving logical equivalence we obtain an acceptance condition in *disjunctive normal form*, again with only negative literals; in other words, a *disjunction* of AF acceptance formulas. Thus, the model set of a SETAF acceptance condition is not necessarily an ideal, but a union of ideals. For the canonical realization we can make use of the fact that, for each $a \in A$, the set $V_a^{\mathbf{t}} = \{v \in \mathcal{V}_2 \mid (v, a, \mathbf{t}) \in F\}$ is downward-closed with respect to \leq_t , hence the set of models of $\bigvee_{v \in \max_{\leq_t} V^{\mathbf{t}}} \bigwedge_{v(b)=\mathbf{f}} \neg b$ is exactly $V_a^{\mathbf{t}}$. The clauses of its corresponding CNF-formula exactly coincide with the sets of arguments attacking a in $kb_\sigma^{\text{SETAF}}(F)$.

3.4 Correctness

For a lack of space, we could not include a formal proof of soundness and completeness of Algorithm 1, but rather present arguments for termination and correctness.

Termination With each recursive call, the set F can never decrease in size, as the only changes to F are adding the results of propagation in line 3 and adding the guesses in line 11. Also within the until-loop, the set F can never decrease in size; furthermore there is only an overall finite number of triples that can be added to F . Thus at some point we must have $F_\Delta = \emptyset$ and leave the until-loop. Since F always increases in size, at some point it must either become functional or incoherent, whence the algorithm terminates.

Soundness If the algorithm returns $kb_\sigma^\mathcal{F}(F)$ as a realizing knowledge base, then according to the condition in line 8 the relation F induced a total function $f : \mathcal{V}_2 \rightarrow \mathcal{V}_2$. In particular, because the until-loop must have been run through at least once, there was at least one propagation step (line 2). Since the propagators are defined such that they enforce everything that must hold in a σ -characterization, we conclude that the induced function f indeed is a σ -characterization for V . By construction, we consequently find that $\sigma(kb_\sigma^\mathcal{F}(F)) = V$.

Completeness If the algorithm answers “no”, then the execution reached line 5. Thus, for the constructed set F , there must have been an interpretation $v \in \mathcal{V}_2$ and a statement $a \in A$ such that $\{(v, a, \mathbf{t}), (v, a, \mathbf{f})\} \subseteq F$, that is, F is incoherent. Since F is initially empty, the only way it could get incoherent is in the propagation step in line 2. (The guessing step cannot create incoherence, since exactly one truth value is guessed for v and a .) However, the propagators are defined such that they infer only assignments (triples) that are necessary for the given F . Consequently, the given interpretation set V is such that either there is no realization within the ADF fragment corresponding to formalism \mathcal{F} (that is, the formalism propagator derived the incoherence) or there is no σ -characterization for V with respect to general ADFs (that is, the semantics propagator derived the incoherence). In any case, V is not σ -realizable for \mathcal{F} .

4 Implementation

As Algorithm 1 is based on propagation, guessing, and checking it is perfectly suited for an implementation using answer set programming (ASP) [24, 21] as this allows for exploiting conflict learning strategies and heuristics of modern ASP solvers. Thus, we developed ASP encodings in the `gringo` language [17] for our approach. Similar as the algorithm, our declarative encodings are modular, consisting of a main part responsible for constructing set F and separate encodings for the individual propagators. If one wants, e.g., to compute an AF realization under admissible semantics for a set V of interpretations, an input program encoding V is joined with the main encoding, the propagator encoding for admissible semantics as well as the propagator encoding for AFs. Every answer set of such a program encodes a respective characterization function. Our ASP encoding for preferred semantics is based on the admissible encoding and guesses further interpretations following the essential idea of Algorithm 2. For constructing a knowledge base with the desired semantics, we also provide two ASP encodings that transform the output to an ADF in the syntax of

the DIAMOND tool [14], respectively an AF in ASPARTIX syntax [13, 15]. Both argumentation tools are based on ASP themselves. The encodings for all the semantics and formalisms we covered in the paper can be downloaded from <http://www.dbai.tuwien.ac.at/research/project/adf/unreal/>.

5 Expressiveness Results

In this section we briefly present some results that we have obtained using our implementation. We first introduce some necessary notation to describe the relative expressiveness of knowledge representation formalisms [18, 28]. For formalisms \mathcal{F}_1 and \mathcal{F}_2 with semantics σ_1 and σ_2 , we say that \mathcal{F}_2 under σ_2 is at least as expressive as \mathcal{F}_1 under σ_1 and write $\mathcal{F}_1^{\sigma_1} \leq_e \mathcal{F}_2^{\sigma_2}$ if and only if $\Sigma_{\mathcal{F}_1}^{\sigma_1} \subseteq \Sigma_{\mathcal{F}_2}^{\sigma_2}$, where $\Sigma_{\mathcal{F}}^{\sigma} = \{\sigma(\text{kb}) \mid \text{kb} \in \mathcal{F}\}$ is the *signature of \mathcal{F} under σ* . As usual, we define $\mathcal{F}_1 <_e \mathcal{F}_2$ if and only if $\mathcal{F}_1 \leq_e \mathcal{F}_2$ and $\mathcal{F}_2 \not\leq_e \mathcal{F}_1$.

We now start by considering the signatures of AFs, SETAFs and (B)ADFs for the unary vocabulary $\{a\}$:

$$\begin{aligned} \Sigma_{\text{AF}}^{\text{adm}} &= \Sigma_{\text{SETAF}}^{\text{adm}} = \{\{\mathbf{u}\}, \{\mathbf{u}, \mathbf{t}\}\} \\ \Sigma_{\text{AF}}^{\text{com}} &= \Sigma_{\text{SETAF}}^{\text{com}} = \{\{\mathbf{u}\}, \{\mathbf{t}\}\} \\ \Sigma_{\text{AF}}^{\text{prf}} &= \Sigma_{\text{SETAF}}^{\text{prf}} = \{\{\mathbf{u}\}, \{\mathbf{t}\}\} \\ \Sigma_{\text{AF}}^{\text{mod}} &= \Sigma_{\text{SETAF}}^{\text{mod}} = \{\emptyset, \{\mathbf{t}\}\} \\ \Sigma_{\text{ADF}}^{\text{adm}} &= \Sigma_{\text{BADF}}^{\text{adm}} = \Sigma_{\text{AF}}^{\text{adm}} \cup \{\{\mathbf{u}, \mathbf{f}\}, \{\mathbf{u}, \mathbf{t}, \mathbf{f}\}\} \\ \Sigma_{\text{ADF}}^{\text{com}} &= \Sigma_{\text{BADF}}^{\text{com}} = \Sigma_{\text{AF}}^{\text{com}} \cup \{\{\mathbf{f}\}, \{\mathbf{u}, \mathbf{t}, \mathbf{f}\}\} \\ \Sigma_{\text{ADF}}^{\text{prf}} &= \Sigma_{\text{BADF}}^{\text{prf}} = \Sigma_{\text{AF}}^{\text{prf}} \cup \{\{\mathbf{f}\}, \{\mathbf{t}, \mathbf{f}\}\} \\ \Sigma_{\text{ADF}}^{\text{mod}} &= \Sigma_{\text{BADF}}^{\text{mod}} = \Sigma_{\text{AF}}^{\text{mod}} \cup \{\{\mathbf{f}\}, \{\mathbf{t}, \mathbf{f}\}\} \end{aligned}$$

The following result shows that the expressiveness of the formalisms under consideration is in line with the amount of restrictions they impose on acceptance formulas.

Theorem 8. For any $\sigma \in \{\text{adm}, \text{com}, \text{prf}, \text{mod}\}$:

1. $\text{AF}^{\sigma} <_e \text{SETAF}^{\sigma}$.
2. $\text{SETAF}^{\sigma} <_e \text{BADF}^{\sigma}$.
3. $\text{BADF}^{\sigma} <_e \text{ADF}^{\sigma}$.

Proof. (1) $\text{AF}^{\sigma} \leq_e \text{SETAF}^{\sigma}$ is clear (by modeling individual attacks via singletons). For $\text{SETAF}^{\sigma} \not\leq_e \text{AF}^{\sigma}$ the witnessing interpretation sets over vocabulary $A = \{a, b, c\}$ are $\{\mathbf{uuu}, \mathbf{ttf}, \mathbf{tft}, \mathbf{ftt}\} \in \Sigma_{\text{SETAF}}^{\sigma} \setminus \Sigma_{\text{AF}}^{\sigma}$ and $\{\mathbf{ttf}, \mathbf{tft}, \mathbf{ftt}\} \in \Sigma_{\text{SETAF}}^{\tau} \setminus \Sigma_{\text{AF}}^{\tau}$ with $\sigma \in \{\text{adm}, \text{com}\}$ and $\tau \in \{\text{prf}, \text{mod}\}$. By each pair of arguments of A being \mathbf{t} in at least one model, a realizing AF cannot feature any attack, immediately giving rise to the model \mathbf{ttt} . The respective realizing SETAF is given by the attack relation $X = \{(\{a, b\}, c), (\{a, c\}, b), (\{b, c\}, a)\}$.

(2) It is clear that $\text{SETAF}^{\sigma} \leq_e \text{BADF}^{\sigma}$ holds (SETAFs are bipolar since all parents are always attacking). For $\text{BADF}^{\sigma} \not\leq_e \text{SETAF}^{\sigma}$ the respective counterexamples can be read off the signatures above: for $\sigma \in \{\text{adm}, \text{com}\}$ we find $\{\mathbf{u}, \mathbf{t}, \mathbf{f}\} \in \Sigma_{\text{BADF}}^{\sigma} \setminus \Sigma_{\text{SETAF}}^{\sigma}$ and for $\tau \in \{\text{prf}, \text{mod}\}$ we find $\{\mathbf{t}, \mathbf{f}\} \in \Sigma_{\text{BADF}}^{\tau} \setminus \Sigma_{\text{SETAF}}^{\tau}$. The realizing bipolar ADF has acceptance condition $\varphi_a = a$.

(3) For $\sigma = \text{mod}$ the result is known [28, Theorem 14]; for the remaining semantics the model sets witnessing $\text{ADF}^{\sigma} \not\leq_e \text{BADF}^{\sigma}$ over vocabulary $A = \{a, b\}$ are

$$\begin{aligned} \{\mathbf{uu}, \mathbf{tu}, \mathbf{tt}, \mathbf{tf}, \mathbf{fu}\} &\in \Sigma_{\text{ADF}}^{\text{adm}} \setminus \Sigma_{\text{BADF}}^{\text{adm}} \\ \{\mathbf{uu}, \mathbf{tu}, \mathbf{tt}, \mathbf{tf}, \mathbf{fu}\} &\in \Sigma_{\text{ADF}}^{\text{com}} \setminus \Sigma_{\text{BADF}}^{\text{com}} \\ \{\mathbf{tt}, \mathbf{tf}, \mathbf{fu}\} &\in \Sigma_{\text{ADF}}^{\text{prf}} \setminus \Sigma_{\text{BADF}}^{\text{prf}} \end{aligned}$$

A witnessing ADF is given by $\varphi_a = a$ and $\varphi_b = a \leftrightarrow b$. \square

Theorem 8 is concerned with the relative expressiveness of the formalisms under consideration, given a certain semantics. Considering different semantics we find that for all formalisms the signatures become incomparable:

Proposition 9. $\mathcal{F}_1^{\sigma_1} \not\leq_e \mathcal{F}_2^{\sigma_2}$ and $\mathcal{F}_2^{\sigma_2} \not\leq_e \mathcal{F}_1^{\sigma_1}$ for all formalisms $\mathcal{F}_1, \mathcal{F}_2 \in \{\text{AF}, \text{SETAF}, \text{BADF}, \text{ADF}\}$ and all semantics $\sigma_1, \sigma_2 \in \{\text{adm}, \text{com}, \text{prf}, \text{mod}\}$ with $\sigma_1 \neq \sigma_2$.

Proof. First, the result for *adm* and *com* follows by $\{\mathbf{u}, \mathbf{t}\} \in \Sigma_{\text{AF}}^{\text{adm}}$, but $\{\mathbf{u}, \mathbf{t}\} \notin \Sigma_{\text{ADF}}^{\text{com}}$ and $\{\mathbf{t}\} \in \Sigma_{\text{AF}}^{\text{com}}$, but $\{\mathbf{t}\} \notin \Sigma_{\text{ADF}}^{\text{adm}}$. Moreover, taking into account that the set of preferred interpretations (resp. two-valued models) always forms a \leq_i -antichain while the set of admissible (resp. complete) interpretations never does, the result follows for $\sigma_1 \in \{\text{adm}, \text{com}\}$ and $\sigma_2 \in \{\text{prf}, \text{mod}\}$. Finally, since a $\text{kb} \in \mathcal{F}$ may not have any two-valued models and a preferred interpretation is not necessarily two-valued, the result for *prf* and *mod* follows. \square

Disregarding the possibility of realizing the empty set of interpretations under the two-valued model semantics, we obtain the following relation for ADFs.

Proposition 10. $(\Sigma_{\text{ADF}}^{\text{mod}} \setminus \{\emptyset\}) \subseteq \Sigma_{\text{ADF}}^{\text{prf}}$.

Proof. Consider some $V \in \Sigma_{\text{ADF}}^{\text{mod}}$ with $V \neq \emptyset$. Clearly $V \subseteq \mathcal{V}_2$ and by Proposition 4 there is a *mod*-characterization $f : \mathcal{V}_2 \rightarrow \mathcal{V}_2$ for V , that is, $f(v) = v$ iff $v \in V$. Define $f' : \mathcal{V}_2 \rightarrow \mathcal{V}_2$ such that $f'(v) = f(v) = v$ for all $v \in V$ and $f'(v)(a) = \neg v(a)$ for all $v \in \mathcal{V} \setminus V$ and $a \in A$. Now it holds that f' is an *adm*-characterization of $V' = \{v \in \mathcal{V} \mid \forall v_2 \in [v]_2 : v_2 \in V\} \cup \{v_u\}$. Since $\max_{\leq_i} V' = V$ we get that the ADF D with acceptance formula $\varphi_a^{f'}$ for each $a \in A$ has $\text{prf}(D) = V$ whence $V \in \Sigma_{\text{ADF}}^{\text{prf}}$. \square

In contrast, this relation does not hold for AFs, which was shown for extension-based semantics by Linsbichler et al. [20, Theorem 5] and immediately follows for the three-valued case.

6 Discussion

We presented a framework for realizability in which AFs, SETAFs, BADFs and general ADFs can be treated in a uniform way. The centerpiece of our approach is an algorithm for deciding realizability of a given interpretation-set in a formalism under a semantics. The algorithm makes use of so-called propagators, by which it can be adapted to the different formalisms and semantics. We also presented an implementation of our framework in answer set programming and several novel expressiveness results that we obtained using our implementation. In unpublished related work, our colleague Sylwia Polberg studied a wide range of abstract argumentation formalisms, in particular their relationship with ADFs [25]. This can be the basis for including further formalisms into our realizability framework: all that remains to do is figuring out suitable ADF fragments and developing propagators for them, just like we did exemplarily for Nielsen and Parsons' SETAFs. For further future work, several semantics whose realizability is yet unstudied could be added to our framework, for example semantics based on conflict-freeness, like three-valued versions of conflict-free, naive, and stage semantics [27, 16, 29].

Acknowledgements This research was supported by the German Research Foundation (DFG) under project BR 1817/7-1 and the Austrian Science Fund (FWF) under projects I1102, I2854 and P25518.

References

- [1] Leila Amgoud and Claudette Cayrol, 'A reasoning model based on the production of acceptable arguments', *Annals of Mathematics and Artificial Intelligence*, **34**(1–3), 197–215, (2002).
- [2] Pietro Baroni, Federico Cerutti, Massimiliano Giacomin, and Giovanni Guida, 'AFRA: Argumentation framework with recursive attacks', *International Journal of Approximate Reasoning*, **52**(1), 19–37, (2011).
- [3] Ringo Baumann, Wolfgang Dvořák, Thomas Linsbichler, Hannes Strass, and Stefan Woltran, 'Compact argumentation frameworks', in *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI 2014)*, eds., Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pp. 69–74. IOS Press, (2014).
- [4] Gerhard Brewka, Stefan Ellmauthaler, Hannes Strass, Johannes P. Wallner, and Stefan Woltran, 'Abstract Dialectical Frameworks Revisited', in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI 2013)*, ed., Francesca Rossi, pp. 803–809. AAAI Press / IJCAI, (2013).
- [5] Gerhard Brewka, Sylwia Polberg, and Stefan Woltran, 'Generalizations of Dung frameworks and their role in formal argumentation', *IEEE Intelligent Systems*, **29**(1), 30–38, (2014). Special Issue on Representation and Reasoning.
- [6] Gerhard Brewka and Stefan Woltran, 'Abstract Dialectical Frameworks', in *Proceedings of the 12th International Conference on Principles of Knowledge Representation and Reasoning (KR 2010)*, eds., Fangzhen Lin, Ulrike Sattler, and Mirosław Truszczyński, pp. 102–111. AAAI Press, (2010).
- [7] Martin Caminada and Dov Gabbay, 'A logical account of formal argumentation', *Studia Logica*, **93**(2–3), 109–145, (2009).
- [8] Claudette Cayrol and Marie-Christine Lagasque-Schiex, 'On the acceptability of arguments in bipolar argumentation frameworks', in *Proceedings of the 8th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2005)*, ed., Lluís Godo, volume 3571 of *Lecture Notes in Computer Science*, pp. 378–389. Springer, (2005).
- [9] 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).
- [10] Paul E. Dunne, Wolfgang Dvořák, Thomas Linsbichler, and Stefan Woltran, 'Characteristics of multiple viewpoints in abstract argumentation', *Artificial Intelligence*, **228**, 153–178, (2015).
- [11] Paul E. Dunne, Wolfgang Dvořák, Thomas Linsbichler, and Stefan Woltran, 'Characteristics of multiple viewpoints in abstract argumentation', in *Proceedings of the Fourth Workshop on Dynamics of Knowledge and Belief (DKB 2013)*, eds., Christoph Beierle and Gabriele Kern-Isberner, pp. 16–30, (2013).
- [12] Sjur K. Dyrkolbotn, 'How to Argue for Anything: Enforcing Arbitrary Sets of Labellings using AFs', in *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR 2014)*, eds., Chitta Baral, Giuseppe De Giacomo, and Thomas Eiter, pp. 626–629. AAAI Press, (2014).
- [13] Uwe Egly, Sarah A. Gaggl, and Stefan Woltran, 'Answer-set programming encodings for argumentation frameworks', *Argument & Computation*, **1**(2), 147–177, (2010).
- [14] Stefan Ellmauthaler and Hannes Strass, 'The DIAMOND system for computing with abstract dialectical frameworks', in *Proceedings of the Fifth International Conference on Computational Models of Argument (COMMA 2014)*, eds., Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, volume 266 of *FAIA*, pp. 233–240. IOS Press, (2014).
- [15] Sarah A. Gaggl, Norbert Manthey, Alessandro Ronca, Johannes P. Wallner, and Stefan Woltran, 'Improved answer-set programming encodings for abstract argumentation', *Theory and Practice of Logic Programming*, **15**(4–5), 434–448, (2015).
- [16] Sarah A. Gaggl and Hannes Strass, 'Decomposing Abstract Dialectical Frameworks', in *Proceedings of the Fifth International Conference on Computational Models of Argument (COMMA 2014)*, eds., Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, volume 266 of *FAIA*, pp. 281–292. IOS Press, (2014).
- [17] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub, *Answer Set Solving in Practice*, Morgan and Claypool Publishers, 2012.
- [18] Goran Gogic, Henry Kautz, Christos Papadimitriou, and Bart Selman, 'The comparative linguistics of knowledge representation', in *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 1995)*, pp. 862–869. Morgan Kaufmann, (1995).
- [19] Carla P. Gomes, Henry A. Kautz, Ashish Sabharwal, and Bart Selman, 'Satisfiability Solvers', in *Handbook of Knowledge Representation*, eds., Frank van Harmelen, Vladimir Lifschitz, and Bruce W. Porter, volume 3 of *Foundations of Artificial Intelligence*, 89–134, Elsevier, (2008).
- [20] Thomas Linsbichler, Christof Spanring, and Stefan Woltran, 'The hidden power of abstract argumentation semantics', in *Theory and Applications of Formal Argumentation – 3rd International Workshop (TFAFA 2015), Revised Selected Papers*, eds., Elizabeth Black, Sanjay Modgil, and Nir Oren, volume 9524 of *Lecture Notes in Computer Science*, pp. 146–162. Springer, (2015).
- [21] Victor W. Marek and Mirosław Truszczyński, 'Stable models and an alternative logic programming paradigm', in *In The Logic Programming Paradigm: a 25-Year Perspective*, eds., Krzysztof R. Apt, Victor W. Marek, Mirosław Truszczyński, and David S. Warren, 375–398, Springer, (1999).
- [22] Sanjay Modgil, 'Reasoning about preferences in argumentation frameworks', *Artificial Intelligence*, **173**(9–10), 901–934, (2009).
- [23] Søren Holbech Nielsen and Simon Parsons, 'A generalization of Dung's abstract framework for argumentation: Arguing with sets of attacking arguments', in *Proceedings of the 3rd International Workshop on Argumentation in Multi-Agent Systems (ArgMAS 2006)*, eds., Nicolas Maudet, Simon Parsons, and Iyad Rahwan, volume 4766 of *Lecture Notes in Computer Science*, pp. 54–73. Springer, (2006).
- [24] Ilkka Niemelä, 'Logic programs with stable model semantics as a constraint programming paradigm', *Annals of Mathematics and Artificial Intelligence*, **25**(3–4), 241–273, (1999).
- [25] Sylwia Polberg. Understanding the Abstract Dialectical Framework (Preliminary Report). Available at <http://arxiv.org/abs/1607.00819>, July 2016.
- [26] Jörg Pührer, 'Realizability of Three-Valued Semantics for Abstract Dialectical Frameworks', in *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*, eds., Qiang Yang and Michael Wooldridge, pp. 3171–3177. AAAI Press, (2015).
- [27] Hannes Strass, 'Approximating operators and semantics for abstract dialectical frameworks', *Artificial Intelligence*, **205**, 39–70, (December 2013).
- [28] Hannes Strass, 'Expressiveness of Two-Valued Semantics for Abstract Dialectical Frameworks', *Journal of Artificial Intelligence Research*, **54**, 193–231, (2015).
- [29] Hannes Strass and Johannes P. Wallner, 'Analyzing the Computational Complexity of Abstract Dialectical Frameworks via Approximation Fixpoint Theory', *Artificial Intelligence*, **226**, 34–74, (2015).

A Scalable Clustering-Based Local Multi-Label Classification Method

Lu Sun¹ and Mineichi Kudo¹ and Keigo Kimura¹

Abstract. Multi-label classification aims to assign multiple labels to a single test instance. Recently, more and more multi-label classification applications arise as large-scale problems, where the numbers of instances, features and labels are either or all large. To tackle such problems, in this paper we develop a clustering-based local multi-label classification method, attempting to reduce the problem size in instances, features and labels. Our method consists of low-dimensional data clustering and local model learning. Specifically, the original dataset is firstly decomposed into several regular-scale parts by applying clustering analysis on the feature subspace, which is induced by a supervised multi-label dimension reduction technique; then, an efficient local multi-label model, meta-label classifier chains, is trained on each data cluster. Given a test instance, only the local model belonging to the nearest cluster to it is activated to make the prediction. Extensive experiments performed on eighteen benchmark datasets demonstrated the efficiency of the proposed method compared with the state-of-the-art algorithms.

1 INTRODUCTION

Originated from traditional single-label classification, multi-label classification (MLC) enables to associate an instance with multiple labels. MLC has been used to tackle a number of real-world applications like text categorization [11], semantic image classification [3], video annotation [16] and music emotions detection [24], etc. Various MLC decomposition methods, such as Binary Relevance [3], Classifier Chains [18, 6], Calibrated Label Ranking [8] and Label Powerset [26], have been proposed by decomposing a multi-label problem into one or a set of single-label classification problems.

As the rapid increasing of web-related applications, more and more recent multi-label datasets emerge in large-scale, whose numbers of instances, features and labels are far from the regular-size. For example, there are millions of videos in the video-sharing website Youtube, while each one can be tagged by some of millions of candidate categories. Such large-scale problems challenge the existing MLC methods. Several methods [38, 5] have been proposed to tackle such a situation by training a multi-label model on the feature or the label subspaces. The common assumption behind these methods is that noisy features exist in the original data and the training label matrix is low-rank. Although these methods achieved much success in a number of MLC applications, further improvement in terms of time complexity and prediction accuracy is recently required.

In this study, we put on two assumptions about the locality in MLC setting: (a) meta-labels, i.e. reasonable and strong label combinations, exist implicitly in the label space; (b) only a fraction of

features and instances are relevant to a meta-label. These assumptions are supported by several observations. For example, in Enron dataset, 53 labels are categorized into only four meta-labels, and in image annotation, an object typically relates to only a few regions in the high-dimensional feature space.

Hence, we presume that MLC can be tackled by decomposing the original large-scale data into several regular-scale datasets, each of which is relevant to only several meta-labels in a feature subspace with a fraction of training instances. Based on this assumption, a Clustering-based Local MLC (CLMLC) method is proposed in this paper. CLMLC consists of two stages, low-dimensional data clustering and local model learning. In the first stage, a supervised dimension reduction is firstly conducted to project the original high-dimensional data into a low-dimensional feature subspace, while preserving feature-label correlation. Then clustering analysis is applied to partition the low-dimensional data into several regular-scale datasets. In the second stage, within each data cluster, meta-labels are mined by saving both label similarity and instance locality, and then classifier chains over meta-labels are built as the local MLC model. Given a test instance, prediction is made on the basis of the local model corresponding to its nearest data cluster. To empirically evaluate the performance of CLMLC, extensive experiments on regular/large-scale datasets from various domains are carried out with the state-of-the-art MLC algorithms.

2 RELATED WORKS

To handle large-scale MLC problems, recently many research efforts have been paid to Feature Space Dimension Reduction (FS-DR) and Label Space Dimension Reduction (LS-DR). In FS-DR, traditional supervised dimension reduction approaches, such as Latent Semantic Indexing, Linear Discriminant Analysis, Canonical Correlation Analysis and Hypergraph Spectral Learning, are specifically extended to match the MLC setting [35, 29, 22, 21]. On the other hand, in order to improve the discriminative ability for each label, LIFT [37] and LLSF [9] are proposed to extract label-specific features. In LS-DR, based on the assumption of low-rank of label matrix, several embedding methods, such as Compressive Sensing [12], CPLST [5] and FaIE [13], encode the sparse label space by preserving label correlations and maximizing predictability of latent label space. By combining FS-DR and LS-DR, several methods have been proposed in recent years. WSABIE [31] learns a low-dimensional joint embedding space by approximately optimizing the precision on the top k relevant labels. By modeling MLC as a general empirical risk minimization problem with a low-rank constraint, LEML [34] scales to very large datasets even with missing labels. To handle the extreme MLC problems with a large number of labels, a tree-based

¹ Hokkaido University, Sapporo 060-0814, Japan, email: {sunlu, mine, kkimura}@main.ist.hokudai.ac.jp

method, FastXML, is proposed in [15] by directly optimizing a specific ranking loss function, nDCG, and by efficiently executing its formulation in light of an alternating minimization algorithm.

The above methods can be categorized as global MLC methods, since they assume that feature-label relationship can be modeled on the whole training data. The global methods probably contradict real-world problems, harming classification accuracy and bringing in high time complexity, especially in large-scale datasets. To relax the assumption, local MLC methods are proposed, aiming to solve a complex problem by dividing it into multiple simpler ones. The local strategy has two advantages. First, simpler problems can be solved by simpler techniques, like transforming a global nonlinear problem into a local linear problem. Second, the training and testing can be more efficient, making the algorithm tractable for large-scale datasets.

As local MLC methods, Hierarchical Multi-Label Classification (HMC) [19, 1, 28] builds a hierarchy of single-label classifiers. Under the hierarchy constraint, the training data for each classifier is restricted so that it contains only the instances associated with parent labels. However, HMC's applications are limited on particular problems in text categorization and genomics analysis. Applying the same strategy of HMC, HOMER [25] breaks the constraint on the predefined label hierarchy. It builds the label hierarchy by recursively conducting balanced k -means on the label space, transforming the original task into a tree-shaped hierarchy of simpler tasks, each one relevant to a subset of labels. The local strategy is also applied by directly finding data clusters. CBMLC [14] partitions the original multi-label datasets into multiple small-scale datasets, on which multi-label classifiers are built individually. Given a test instance, it is feed only to the classifier corresponding to the nearest cluster. To speed up the k NN classification, SLEEC [2] partitions the original training data into several clusters, learning a local nonlinear embedding per cluster and conducting k NN only within the test sample's nearest cluster. On the other hand, in the regression setting, several regression tree based methods, RETIS [10], M5 system [17] and HTL [23], also employ the local strategy. Similar with the classical regression tree algorithm like CART [4], such methods divide the input space into mutually exclusive regions described by propositional assertions on the input features. The difference is that RETIS, M5 and HTL build several alternative regression models in the leaves of a tree to improve predictive accuracy. In [36], Regression Clustering partitions the original dataset into several subsets. Each regression is conducted on its own subset with a simpler distribution, leading to a better generalization ability.

Based on the above survey, we notice that seldom research works focus on local MLC methods. Motivated by the work of CBMLC [14], in this paper, we propose the Clustering-based Local MLC (CLMLC) method. In CLMLC, we assume a large-scale problem can be divided into a number of small or medium-scaled problems without loss of discriminative information. Different with CBMLC, CLMLC is built on a feature subspace and employs different local models for different data clusters.

3 THE CLMLC METHOD

In the scenario of MLC, an instance is typically represented by a pair (\mathbf{x}, \mathbf{y}) , which contains a feature vector $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^D$ and the corresponding label vector $\mathbf{y} \in \mathcal{Y} \subseteq \{0, 1\}^L$, where $y_\ell = 1$ if and only if ℓ -th label is associated with instance \mathbf{x} , and $y_\ell = 0$ otherwise, $\ell \in \{1, \dots, L\}$. Assume that we are given a dataset of N instances $\mathcal{S} = [\mathbf{X}_S, \mathbf{Y}_S]$, where $\mathbf{X}_S = [\mathbf{x}_1, \dots, \mathbf{x}_N]^T$ and $\mathbf{Y}_S = [\mathbf{y}_1, \dots, \mathbf{y}_N]^T$ denote the feature and label matrix, respectively. Given a testing

dataset $\mathcal{T} = [\mathbf{X}_T, \mathbf{Y}_T]$, the task of MLC is to find an optimal classifier $h : \mathcal{X} \rightarrow \mathcal{Y}$ which assigns a label matrix $\hat{\mathbf{Y}}_T$ to test data \mathbf{X}_T such that h minimizes a loss function on $\hat{\mathbf{Y}}_T$ and \mathbf{Y}_T .

Now we present the proposed CLMLC method, which can scale to MLC problems in large N , D and L . CLMLC comprises low-dimensional data clustering and local model learning.

3.1 Low-dimensional data clustering

We assume that a large-scale dataset could be decomposed into several smaller local sets. To this end, clustering analysis is introduced to find the local clusters. However, directly applying cluster analysis would probably produce unstable outputs and suffer from high computational cost, especially when the dimensionality of the original feature space is relatively high. In this sense, a dimensionality reduction approach is necessary as a pre-processing technique before applying clustering analysis.

Let \mathbf{X} and \mathbf{Y} be already centered so as to $\mathbf{X}^T \mathbf{1} = \mathbf{0}$ and $\mathbf{Y}^T \mathbf{1} = \mathbf{0}$. The Partial Least Squares (PLS) [30] finds the directions of maximum covariance between \mathbf{X} and \mathbf{Y} by Singular Value Decomposition (SVD) as follows:

$$\min_{\mathbf{U}, \mathbf{V}} \|\mathbf{X}^T \mathbf{Y} - \mathbf{U} \mathbf{\Lambda}_d \mathbf{V}^T\|_F^2, \quad (1)$$

where $\mathbf{\Lambda}_d$ is a diagonal matrix $(\lambda_1, \lambda_2, \dots, \lambda_d)$ with the largest d singular values of $\mathbf{X}^T \mathbf{Y}$, and $\|\cdot\|_F$ denotes the Frobenius norm. This is also the solution of the maximization problem:

$$\begin{aligned} \max_{\mathbf{U}, \mathbf{V}} \quad & \text{Tr}(\mathbf{U}^T \mathbf{X}^T \mathbf{Y} \mathbf{V}) \\ \text{s.t.} \quad & \mathbf{U}^T \mathbf{U} = \mathbf{V}^T \mathbf{V} = \mathbf{I}_d. \end{aligned} \quad (2)$$

One of limitations of PLS is the lack of invariance to arbitrary linear transformations on \mathbf{X} [32].

To overcome this limitation, Orthonormalized PLS (OPLS) [32] is proposed by orthonormalizing \mathbf{X} to $\mathbf{X}(\mathbf{X}^T \mathbf{X})^{-\frac{1}{2}}$ in (1), and we have

$$\min_{\mathbf{U}, \mathbf{V}} \|\mathbf{X}^T \mathbf{X}^{-\frac{1}{2}} \mathbf{X}^T \mathbf{Y} - \mathbf{U} \mathbf{\Lambda}_d \mathbf{V}^T\|_F^2. \quad (3)$$

Similar with (2), (3) can be also rewritten to a maximization problem:

$$\begin{aligned} \max_{\mathbf{U}} \quad & \text{Tr}(\mathbf{U}^T \mathbf{X}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X} \mathbf{U}) \\ \text{s.t.} \quad & \mathbf{U}^T \mathbf{X}^T \mathbf{X} \mathbf{U} = \mathbf{I}. \end{aligned} \quad (4)$$

The solution \mathbf{U} consists of eigenvectors \mathbf{u} corresponding to the largest d eigenvalues of a generalized eigenvalue problem

$$(\mathbf{X}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X}) \mathbf{u} = \lambda (\mathbf{X}^T \mathbf{X}) \mathbf{u}. \quad (5)$$

To avoid the singularity of $\mathbf{X}^T \mathbf{X}$ and reduce the model complexity, in practice a regularization term $\gamma \mathbf{I}$ with $\gamma > 0$ is commonly introduced to (5), leading to

$$(\mathbf{X}^T \mathbf{Y} \mathbf{Y}^T \mathbf{X}) \mathbf{u} = \lambda (\mathbf{X}^T \mathbf{X} + \gamma \mathbf{I}) \mathbf{u}. \quad (6)$$

In general, directly solving the generalized eigenvalue problem (6) suffers from an expensive cost and thus might not scale to large-scale problems. In this study, we use an efficient two-stage approach [20] to address the problem. In the first stage, a penalized least squares problem is solved by regressing the centered feature matrix \mathbf{X} to the centered label matrix \mathbf{Y} ; after projecting \mathbf{X} into the subspace by the regression, in the second stage, the resulting generalized eigenvalue problem is solved by SVD.

Algorithm 1 Low-dimensional data clustering

Input: \mathbf{X} : centered data matrix, \mathbf{Y} : centered label matrix, d : size of feature subspace, K : number of data clusters

Output: \mathbf{U} : projection matrix, \mathbf{R}, \mathbf{C} : clustering output

1: Solve the least squares problem:

$$\min_{\mathbf{U}_1} \|\mathbf{X}\mathbf{U}_1 - \mathbf{Y}\|_F^2 + \|\mathbf{U}_1\|_F^2;$$

2: $\mathbf{H} = \mathbf{U}_1^\top \mathbf{X}^\top \mathbf{Y}$;

3: Decompose $\mathbf{H} = \mathbf{U}_H \Lambda_d \mathbf{U}_H^\top$ by SVD;

4: $\mathbf{U} = \mathbf{U}_1 \mathbf{U}_2$, where $\mathbf{U}_2 = \mathbf{U}_H \Lambda_d^{-\frac{1}{2}}$;

5: $[\mathbf{R}, \mathbf{C}] \leftarrow k\text{-means}(\mathbf{Z}, K)$ by (7), where $\mathbf{Z} = \mathbf{X}\mathbf{U}$.

Through (6), we find an orthonormal basis $[\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_d]$ to form \mathbf{U} . Therefore we can have a low-dimensional expression $\mathbf{z} \in \mathbb{R}^d$ by projection $\mathbf{z} = \mathbf{U}^\top \mathbf{x}$, $\mathbf{Z} = \mathbf{X}\mathbf{U}$ as well. Then we conduct clustering on $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N$ in the light of elimination of most of noisy features. In this paper, k -means is employed, aiming to approximately solve the following optimization problem:

$$\min_{\mathbf{R}, \mathbf{C}} \sum_{i=1}^N \sum_{j=1}^K r_{ij} \|\mathbf{z}_i - \mathbf{c}_j\|_2^2 \quad (7)$$

s.t. $\forall i, \|\mathbf{r}_i\|_0 = 1, \|\mathbf{r}_i\|_1 = 1,$

where \mathbf{R} represents the $N \times K$ indicator matrix, indicating the assignment from data points to centroids, while the centroid matrix $\mathbf{C} = [\mathbf{c}_1, \dots, \mathbf{c}_K]^\top$, whose $\mathbf{c}_j = \sum_i r_{ij} \mathbf{x}_i / \sum_i r_{ij}$. $\|\cdot\|_0, \|\cdot\|_1$ and $\|\cdot\|_2$ denote the ℓ_0, ℓ_1 and ℓ_2 norm, respectively. In general, k -means is realized as an iterative algorithm. The pseudo code of low-dimensional data clustering is given in Algorithm 1.

3.2 Local model learning

In the second stage, we perform local model learning in each cluster. By expecting the existence of meta-labels, We use Laplacian eigenmap to learn meta-labels within each cluster, and then build classifier chains over meta-labels for local model learning. For each data cluster, we construct a graph $\mathbf{G} = \langle \mathbf{V}, \mathbf{E} \rangle$ in the label space, where \mathbf{V} is the vertex/label set, and \mathbf{E} is the edge set containing edges between each label pair. Given an appropriate affinity matrix \mathbf{A} on \mathbf{E} , meta-label learning can be considered as a graph cut problem: cutting the graph \mathbf{G} into a set of sub-graphs.

For constructing affinity matrix \mathbf{A} , we use two different sources: the label space and the instance space. In this study, we utilize Jacard index and heat kernel affinity to represent the label similarity and instance locality, respectively.

- Label similarity $\mathbf{A}^{(L)} = \{A_{\ell m}^{(L)}\}_{\ell, m=1}^L$,

$$A_{\ell m}^{(L)} := \frac{\sum_{i=1}^N y_{i\ell} \cdot y_{im}}{\sum_{i=1}^N (y_{i\ell} + y_{im} - y_{i\ell} \cdot y_{im})}. \quad (8)$$

- Instance locality $\mathbf{A}^{(I)} = \{A_{\ell m}^{(I)}\}_{\ell, m=1}^L$,

$$A_{\ell m}^{(I)} := e^{-\|\boldsymbol{\mu}_\ell - \boldsymbol{\mu}_m\|_2^2}, \text{ where } \boldsymbol{\mu}_\ell = \frac{\sum_{i=1}^N \mathbf{z}_i \cdot y_{i\ell}}{\sum_{i=1}^N y_{i\ell}}. \quad (9)$$

By combining these two matrices, we obtain the following affinity matrix $\mathbf{A} = \{A_{\ell m}\}_{\ell, m=1}^L$,

$$A_{\ell m} := \frac{1}{2} (A_{\ell m}^{(L)} + A_{\ell m}^{(I)}). \quad (10)$$

Algorithm 2 Local model learning

Input: \mathbf{Z}^c : local data matrix, \mathbf{Y}^c : local label matrix, n : number of meta-labels, \mathcal{L} : meta-label classifier

Output: \mathbf{h}^c : local classifier

1: Compute \mathbf{A} according to (10);

2: $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where $\mathbf{D} = \text{diag}(\sum_\ell A_{\ell m})$;

3: Solve $\mathbf{L}\mathbf{w} = \lambda \mathbf{D}\mathbf{w}$ by n smallest eigenvalues;

4: $\mathbf{R}^c \leftarrow k\text{-means}(\mathbf{W}, n)$, where $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_n]$;

5: **for** $k \in \{1, \dots, n\}$ **do**

6: $id = \text{find}(\mathbf{R}^c == k)$

7: $h_k^c \leftarrow \mathcal{L}(\mathbf{Z}^c, \mathbf{Y}^c(:, id))$;

8: $\mathbf{Z}^c = \mathbf{Z}^c \cup \mathbf{Y}^c(:, id)$;

9: $\mathbf{h}^c \leftarrow \{h_k^c\}_{k=1}^n$.

To cut the graph \mathbf{G} into n sub-graphs (n meta-labels) is equivalent to perform k -means on the n smallest eigenvectors $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_n]$ of the generalized eigenvalue problem:

$$\mathbf{L}\mathbf{w} = \lambda \mathbf{D}\mathbf{w}, \quad (11)$$

where $\mathbf{D} = (D_{\ell\ell}) = (\sum_m A_{\ell m})$, and \mathbf{L} is the Laplacian matrix, $\mathbf{L} = \mathbf{D} - \mathbf{A}$. Thus, the label assignment to n meta-labels is obtained by applying k -means on the rows of \mathbf{W} .

After finding meta-labels, a sophisticated multi-label classifier \mathcal{L} could be applied to capture the strong label correlations within each meta-label. On the other hand, to model relatively weak meta-label correlations, a simple MLC method is also necessary in the meta-label space. In this way, label correlations can be well captured without much time cost. To this end, we introduce an efficient classifier chains method [18] over the meta-label space. In general, for each meta-label within a meta-label chain, we expand its training data by taking previous meta-labels as extra features before feeding the data into \mathcal{L} . The outline of local model learning is given in Algorithm 2.

3.3 Prediction

Given a test instance $\mathbf{x} \in \mathbf{X}_T$, the prediction can be made by two steps. Firstly, \mathbf{x} is encoded into the feature subspace by $\mathbf{z} = \mathbf{U}^\top \mathbf{x}$. Secondly, the local classifier \mathbf{h}^c corresponding to \mathbf{z} 's nearest cluster \mathbf{c} such as,

$$\mathbf{c} = \arg \min_{\mathbf{c} \in \mathbf{C}} \|\mathbf{z} - \mathbf{c}\|_2^2, \quad (12)$$

is activated to predict the label assignment by $\hat{\mathbf{y}} = \mathbf{h}^c(\mathbf{z})$. Note that \mathbf{C} in (12) is the centroid matrix obtained according to (7).

3.4 Remarks

The complete procedure of CLMLC, including training (Steps 1 to 5) and testing (Steps 6 to 8), is outlined in Algorithm 3. It is worth noting that CLMLC is able to serve as a *meta-strategy* for large-scale MLC problems. For example, other dimension reduction or clustering analysis techniques could be used to replace the OPLS or k -means in Algorithm 1, in order to handle specific problem settings or data patterns. Similarly, any MLC method can be directly applied for local model learning in Algorithm 2. It shows the high flexibility of CLMLC to address various MLC problems.

4 EXPERIMENTS

4.1 Datasets and evaluation metrics

In order to evaluate the performance of the proposed CLMLC method and other MLC methods, we conducted experiments on eight

Algorithm 3 CLMLC

Input: \mathbf{X} : centered data matrix, \mathbf{Y} : centered label matrix, \mathbf{x} : test instance, d : size of feature subspace, K : number of data clusters, n : number of meta-labels, \mathcal{L} : meta-label classifier

Output: $\hat{\mathbf{y}}$: predicted label set

Training:

- 1: $[\mathbf{U}, \mathbf{R}, \mathbf{C}] \leftarrow \langle \text{Algorithm 1} \rangle (\mathbf{X}, \mathbf{Y}, d, K)$;
- 2: $\mathbf{Z} = \mathbf{X}\mathbf{U}$;
- 3: **for** $\mathbf{c} \in \mathbf{C}$ **do**
- 4: Find local dataset $[\mathbf{Z}^c, \mathbf{Y}^c]$ by \mathbf{R} ;
- 5: $\mathbf{h}^c \leftarrow \langle \text{Algorithm 2} \rangle (\mathbf{Z}^c, \mathbf{Y}^c, n, \mathcal{L})$;

Testing:

- 6: $\mathbf{z} = \mathbf{U}^T \mathbf{x}$;
- 7: Find \mathbf{z} 's nearest cluster \mathbf{c} by (12);
- 8: $\hat{\mathbf{y}} \leftarrow \mathbf{h}^c(\mathbf{z})$;

teen benchmark datasets in Mulan [27], where nine datasets come from two data sources, Rcv1 and Corel16k. The statistics of the datasets are summarized in Table 1. For the convenience of parameter setting, we treat the first six sets as regular-scale datasets, and last twelve sets as large-scale datasets, respectively.

Table 1: The statistics of experimental multi-label datasets. ‘‘Card.’’, ‘‘Den.’’ and ‘‘Dist.’’ denote the label cardinality, label density and the number of distinct label combinations, respectively.

Dataset	N	D	L	Card.	Den.	Dist.	Domain
Birds	645	260	19	1.014	0.053	133	audio
Genbase	662	1186	27	1.252	0.046	32	biology
Medical	978	1449	45	1.245	0.028	94	text
Enron	1702	1001	53	3.378	0.064	753	text
Scene	2407	294	6	1.074	0.179	15	image
Yeast	2417	103	14	4.237	0.303	198	biology
Corel5k	5000	499	374	3.522	0.009	1453	image
Rcv1s1	6000	944	101	2.880	0.029	837	text
Rcv1s2	6000	944	101	2.634	0.026	800	text
Rcv1s3	6000	944	101	2.614	0.026	783	text
Rcv1s4	6000	944	101	2.667	0.022	629	text
Bibtex	7395	1836	159	2.402	0.015	1654	text
Corel16k1	13766	500	153	2.859	0.019	1791	image
Corel16k2	13761	500	164	2.882	0.018	1782	image
Corel16k3	13760	500	154	2.829	0.018	1718	image
Corel16k4	13837	500	162	2.842	0.018	1760	image
Corel16k5	13847	500	160	2.858	0.018	1784	image
Delicious	16105	500	983	19.020	0.019	3937	text

Given a test dataset $\mathcal{T} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N_T}$, we use four evaluation metrics for the experimental results. Here $\mathbb{1}$ denotes the indicator function.

- **Exact-Match** $:= \frac{1}{N_T} \sum_{i=1}^{N_T} \mathbb{1}_{\hat{\mathbf{y}}_i = \mathbf{y}_i}$,
- **Hamming-Score** $:= \frac{1}{N_T} \sum_{i=1}^{N_T} \frac{1}{L} \sum_{\ell=1}^L \mathbb{1}_{\hat{y}_{i\ell} = y_{i\ell}}$,
- **Macro-F1** $:= \frac{1}{L} \sum_{\ell=1}^L \frac{2 \sum_{i=1}^{N_T} \hat{y}_{i\ell} \cdot y_{i\ell}}{\sum_{i=1}^{N_T} \hat{y}_{i\ell} + \sum_{i=1}^{N_T} y_{i\ell}}$,
- **Micro-F1** $:= \frac{2 \sum_{\ell=1}^L \sum_{i=1}^{N_T} \hat{y}_{i\ell} \cdot y_{i\ell}}{\sum_{\ell=1}^L \sum_{i=1}^{N_T} \hat{y}_{i\ell} + \sum_{\ell=1}^L \sum_{i=1}^{N_T} y_{i\ell}}$.

The above metrics can be cast into two categories, instance-based metrics (Exact-Match and Hamming-Score) and label-based metrics (Macro-F1 and Micro-F1 [33]). Exact-Match is the most stringent measure, since it does not evaluate partial match of labels. In spite of that, it is a good metric to measure how well label correlations are modeled. Hamming-Score emphasizes on the prediction accuracy on label-instance pairs, and is able to evaluate the performance on each single label. However, since Hamming-Score treats equally

false positives and false negatives, it is weak in imbalanced MLC problems. The label-based metrics overcome the limitations of the two instance-based metrics. Macro-F1 computes F1-Score locally over each label, which is more sensitive to the performance on the labels in minority. In contrast, Micro-F1 computes F1-Score globally over all labels, thus it tends to be influenced more by the labels in majority.

4.2 Configuration

The proposed CLMLC method was compared with four state-of-the-art MLC methods:

- **ECC** [18]: an ensemble of classifier chains, where chain orders are generated randomly. Each classifier of a single CC is trained by taking previously assigned labels as extra attributes.
- **MLHSL** [21]: an FS-DR MLC method. A dataset is encoded by mapping features into a subspace, and then an MLC method is built on the basis of the encoded dataset.
- **CPLST** [5]: an LS-DR MLC method. The label space is encoded by a feature-aware principal label space transformation, and the round-based decoding [5] is used to predict the label set.
- **CBMLC** [14]: a first attempt on applying clustering analysis on the dataset before feeding the data to a multi-label classifier.

ECC is adopted due to its superior performance compared with other MLC decomposition methods, such as BR [3] and CC [18], as shown in [18]. As global MLC methods, MLHSL is chosen as a representative of FS-DR methods, while CPLST is chosen by its performance advantage, especially in Hamming-Score, over several LS-DR methods, such as Compressive Sensing, PLST and orthogonally constraint CCA, as shown in [5]. As a local MLC method, CBMLC is selected for comparison in cluster analysis. Note that SLEEC [2] is excluded from the comparing methods, although it employs the similar local strategy with CLMLC. This is because SLEEC focuses on extreme MLC [15], where standard multi-label evaluation metrics like our four metrics are not appropriate.

In the experiments, *5-fold cross validation* was performed to evaluate the classification performance. For fair comparison, CC with ridge regression¹ was used as the baseline classifier for CBMLC, MLHSL, CPLST and CLMLC. In parameter setting, for CLMLC, we set the size of feature subspace d by $\min\{L, 30\}$, and the number of clusters K by 20/100 for regular/large-scale datasets, respectively. For a cluster \mathbf{c} , the number of meta-labels n was set to $\lceil L^c/5 \rceil$. CLMLC employed an ensemble of 2 CCs as the meta-label classifier \mathcal{L} . ECC used an ensemble of 10 CCs. In addition, in order to scale up ECC, random sampling was applied to randomly select 75% of instances and 50% of features for building each CC in ECC, as recommended in [18]. CBMLC and MLHSL shared the same value of K and d with CLMLC, respectively. For CPLST, we set the ratio for LS-DR by 0.8/0.6 for regular/large-scale datasets, respectively. Note that the parameters were chosen for the comparing methods in order to balance the classification accuracy and execution time, according to the experimental results on conducting grid search in the parameter spaces (detailed discussion will be made in Section 4.4). We obtained the MATLAB codes of CPLST¹ and MLHSL² given by the authors, and implemented the MATLAB codes of ECC³, CBMLC³ and CLMLC³ by ourselves. Experiments were performed in a computer configured with an Intel Quad-Core i7-4770 CPU at 3.4GHz with 4GB RAM.

¹ https://github.com/hsuantien/mlc_lsdr

² <http://www.public.asu.edu/~jye02/Software/MLDR/>

³ <https://github.com/futuresun912/CLMLC.git>

Table 2: Experimental results (mean (rank)) on eighteen multi-label datasets in four evaluation metrics.

Method	Exact-Match																	
	Birds	Genbase	Medical	Enron	Scene	Yeast	Corel5k	Rcv1s1	Rcv1s2	Rcv1s3	Rcv1s4	Bibtex	Corel16k1	Corel16k2	Corel16k3	Corel16k4	Corel16k5	Delicious
ECC	0.515 (3)	0.974 (4)	0.640 (3)	0.121 (2)	0.607 (2)	0.197 (2)	0.006 (3)	0.098 (4)	0.214 (4)	0.216 (4)	0.329 (2)	0.157 (3)	0.009 (3.5)	0.007 (4)	0.009 (3)	0.008 (3)	0.008 (3.5)	0.001 (4.5)
MLHSL	0.524 (1.5)	0.982 (1)	0.676 (2)	0.120 (3)	0.596 (3)	0.196 (3)	0.004 (5)	0.114 (3)	0.220 (3)	0.219 (3)	0.320 (4)	0.119 (5)	0.009 (3.5)	0.008 (3)	0.007 (4)	0.007 (4)	0.008 (3.5)	0.002 (3)
CPLST	0.502 (4)	0.980 (2)	0.583 (4)	0.092 (5)	0.479 (5)	0.149 (5)	0.005 (4)	0.063 (5)	0.176 (5)	0.173 (5)	0.290 (5)	0.148 (4)	0.007 (5)	0.006 (5)	0.006 (5)	0.006 (5)	0.007 (5)	0.001 (4.5)
CBMLC	0.375 (5)	0.973 (5)	0.578 (5)	0.101 (4)	0.553 (4)	0.163 (4)	0.012 (2)	0.062 (5)	0.255 (2)	0.248 (2)	0.326 (3)	0.164 (2)	0.016 (2)	0.014 (2)	0.017 (2)	0.017 (2)	0.015 (2)	0.012 (1)
CLMLC	0.524 (1.5)	0.979 (3)	0.688 (1)	0.147 (1)	0.627 (1)	0.205 (1)	0.029 (1)	0.224 (1)	0.316 (1)	0.319 (1)	0.400 (1)	0.172 (1)	0.030 (1)	0.029 (1)	0.030 (1)	0.028 (1)	0.030 (1)	0.004 (2)
avg. rank	CLMLC (1.194) > CBMLC (2.833) > MLHSL (3.194), ECC (3.194) > CPLST (4.583)																	
Method	Hamming-Score																	
	Birds	Genbase	Medical	Enron	Scene	Yeast	Corel5k	Rcv1s1	Rcv1s2	Rcv1s3	Rcv1s4	Bibtex	Corel16k1	Corel16k2	Corel16k3	Corel16k4	Corel16k5	Delicious
ECC	0.951 (3)	0.999 (3)	0.989 (2)	0.933 (3)	0.896 (1)	0.793 (2)	0.990 (2.5)	0.973 (2)	0.977 (2)	0.977 (2)	0.982 (1.5)	0.988 (1.5)	0.981 (2)	0.982 (2.5)	0.982 (2)	0.982 (2.5)	0.982 (2)	0.981 (2.5)
MLHSL	0.954 (1.5)	0.999 (3)	0.990 (1)	0.936 (2)	0.875 (4)	0.786 (3)	0.990 (2.5)	0.973 (2)	0.977 (2)	0.977 (2)	0.981 (3)	0.986 (4)	0.981 (2)	0.982 (2.5)	0.982 (2)	0.982 (2.5)	0.982 (2)	0.981 (2.5)
CPLST	0.950 (4)	0.999 (3)	0.986 (5)	0.911 (5)	0.887 (2)	0.797 (1)	0.991 (1)	0.973 (2)	0.977 (2)	0.977 (2)	0.982 (1.5)	0.988 (1.5)	0.981 (2)	0.982 (2)	0.983 (1)	0.983 (1)	0.982 (2)	0.982 (1)
CBMLC	0.887 (5)	0.999 (3)	0.987 (4)	0.930 (4)	0.869 (5)	0.750 (5)	0.988 (4)	0.966 (5)	0.971 (5)	0.969 (5)	0.976 (5)	0.986 (4)	0.972 (5)	0.976 (5)	0.975 (5)	0.975 (5)	0.975 (5)	0.976 (5)
CLMLC	0.954 (1.5)	0.999 (3)	0.988 (3)	0.940 (1)	0.885 (3)	0.779 (4)	0.986 (5)	0.969 (4)	0.973 (4)	0.973 (4)	0.979 (4)	0.986 (4)	0.977 (4)	0.979 (4)	0.978 (4)	0.979 (4)	0.979 (4)	0.979 (4)
avg. rank	CPLST (2.167), ECC (2.167) > MLHSL (2.417) > CLMLC (3.583) > CBMLC (4.667)																	
Method	Macro-F1																	
	Birds	Genbase	Medical	Enron	Scene	Yeast	Corel5k	Rcv1s1	Rcv1s2	Rcv1s3	Rcv1s4	Bibtex	Corel16k1	Corel16k2	Corel16k3	Corel16k4	Corel16k5	Delicious
ECC	0.290 (3)	0.725 (5)	0.340 (4)	0.196 (1)	0.703 (1)	0.354 (4)	0.014 (4)	0.118 (4)	0.131 (3)	0.108 (3.5)	0.109 (3)	0.193 (3)	0.014 (4.5)	0.016 (4)	0.017 (4)	0.010 (4.5)	0.013 (4)	0.034 (4)
MLHSL	0.302 (2)	0.767 (1)	0.354 (3)	0.160 (4)	0.648 (4)	0.354 (3)	0.010 (5)	0.104 (5)	0.096 (5)	0.091 (5)	0.089 (5)	0.095 (5)	0.014 (4.5)	0.014 (5)	0.014 (5)	0.010 (4.5)	0.012 (5)	0.025 (5)
CPLST	0.287 (4)	0.761 (2.5)	0.374 (1)	0.167 (3)	0.639 (5)	0.351 (5)	0.016 (3)	0.125 (3)	0.110 (4)	0.108 (3.5)	0.108 (4)	0.186 (4)	0.015 (3)	0.018 (3)	0.021 (3)	0.013 (3)	0.015 (3)	0.048 (3)
CBMLC	0.188 (5)	0.738 (4)	0.312 (5)	0.195 (2)	0.655 (3)	0.418 (1)	0.032 (2)	0.204 (2)	0.195 (2)	0.185 (2)	0.176 (2)	0.257 (1)	0.068 (1)	0.063 (1)	0.058 (1)	0.070 (1)	0.060 (1)	0.143 (1)
CLMLC	0.369 (1)	0.761 (2.5)	0.358 (2)	0.153 (5)	0.689 (2)	0.400 (2)	0.038 (1)	0.215 (1)	0.210 (1)	0.198 (1)	0.189 (1)	0.210 (2)	0.056 (2)	0.057 (2)	0.053 (2)	0.051 (2)	0.047 (2)	0.067 (2)
avg. rank	CLMLC (1.861) > CBMLC (2.056) > CPLST (3.333) > ECC (3.528) > MLHSL (4.222)																	
Method	Micro-F1																	
	Birds	Genbase	Medical	Enron	Scene	Yeast	Corel5k	Rcv1s1	Rcv1s2	Rcv1s3	Rcv1s4	Bibtex	Corel16k1	Corel16k2	Corel16k3	Corel16k4	Corel16k5	Delicious
ECC	0.440 (4)	0.990 (3)	0.806 (2)	0.499 (1)	0.694 (1)	0.642 (1)	0.126 (4)	0.325 (4)	0.356 (4)	0.350 (4)	0.430 (3)	0.381 (4)	0.092 (3)	0.079 (4)	0.076 (4)	0.077 (4)	0.073 (4.5)	0.096 (4)
MLHSL	0.452 (2)	0.992 (1.5)	0.812 (1)	0.483 (2)	0.640 (4)	0.627 (4)	0.140 (3)	0.310 (5)	0.330 (5)	0.317 (5)	0.392 (5)	0.279 (5)	0.089 (4)	0.084 (3)	0.065 (5)	0.085 (3)	0.090 (3)	0.063 (5)
CPLST	0.450 (3)	0.992 (1.5)	0.756 (4)	0.414 (5)	0.635 (5)	0.631 (3)	0.106 (5)	0.349 (3)	0.371 (3)	0.365 (3)	0.440 (2)	0.382 (3)	0.070 (5)	0.078 (5)	0.079 (3)	0.070 (5)	0.073 (4.5)	0.194 (3)
CBMLC	0.265 (5)	0.988 (4)	0.740 (5)	0.463 (4)	0.641 (3)	0.581 (5)	0.151 (2)	0.371 (2)	0.387 (2)	0.376 (2)	0.426 (4)	0.393 (2)	0.163 (2)	0.157 (2)	0.154 (2)	0.161 (1)	0.156 (1)	0.268 (1)
CLMLC	0.474 (1)	0.987 (5)	0.782 (3)	0.480 (3)	0.676 (2)	0.632 (2)	0.173 (1)	0.401 (1)	0.423 (1)	0.422 (1)	0.472 (1)	0.396 (1)	0.164 (1)	0.164 (1)	0.160 (1)	0.157 (2)	0.148 (2)	0.214 (2)
avg. rank	CLMLC (1.722) > CBMLC (2.722) > ECC (3.250) > MLHSL (3.639) > CPLST (3.667)																	

4.3 Experimental results

Experimental results of five comparing MLC methods on benchmark datasets are reported in Table 2, where the averaged rank of each method over all datasets is shown in the last row of each metric. For each evaluation metric, the larger the value, the better the performance. Among the five comparing methods, the best performance is highlighted in boldface.

For all the 72 configurations (18 datasets \times 4 evaluation metrics), CLMLC ranked 1st among five comparing MLC methods at 37.8% cases, ranked 2nd at 18.9% cases, and ranked 5th at only 3.3% cases, which was remarkably better than the other methods. Specifically, CLMLC outperformed the other methods in Exact-Match (ranked 1st at 88.9% cases) and Micro-F1 (ranked 1st at 55.6% cases), and was competitive in terms of Macro-F1 (ranked 1st/2nd at 33.3%/61.1% cases). It demonstrates the effectiveness of the clustering-based local strategy adopted in CLMLC. The similar instances with similar label sets can be grouped together by CLMLC, leading to its strong capability on modeling label correlations and thus superior performance in Exact-Match. However, such grouped local data sometimes weaken the influence of minority labels, resulting in the worse performance of CLMLC in Hamming-Score (ranked 4nd at 66.7% cases). CPLST and ECC performed better than the other methods in Hamming-Score (ranked 1st at 38.9% and 16.7% cases, respectively), since it is designed to be optimized in Hamming-Score, according to the theoretical analysis in [5]. In Hamming-Score, MLHSL ranked in 1st/2nd place at 11.1%/61.1% cases, but performed worse in other metrics, especially on large-scale datasets. It is probably because large-scale datasets typically need a sufficient number of instances for training, while FS-DR tends to remove too many features. CBMLC outperformed other methods except CLMLC in Exact-Match (ranked 1st/2nd at 5.6%/55.5% cases), Macro-F1 (ranked 1st/2nd at 44.4%/33.3% cases) and Micro-F1 (ranked 1st/2nd at 16.7%/44.4% cases), but worked worst in Hamming-Score (ranked 5th at 72.2% cases). In addition, CBMLC worked worse than CLMLC on the average in all the four metrics, indicating that cluster analysis should be applied after appropriate feature dimension reduction. Note that the two local MLC methods, CLMLC and CBMLC, worked remarkably better than ECC, MLHSL and CPLST in terms of Exact-Match, Macro-F1 and Micro-F1 on the twelve large-scale datasets, demonstrating the superiority of local MLC strategy on real-world problems.

The execution time on seven large-scale datasets, including both training and prediction time, is reported in Table 3. The least time cost is highlighted in boldface. Among all the methods, MLHSL needed the least execution time on the average due to the low-dimensional feature subspace induced by FS-DR. CLMLC consumed the second least time on the average. Note that, CLMLC paid only slightly higher time cost than MLHSL on the Corel16k datasets. On datasets with large number of labels (large values in L), like delicious, CLMLC consumed more execution time than MLHSL and CPLST. Benefiting from LS-DR, CPLST cost the third least execution time, which was significantly less than ECC and CBMLC. But such superiority of CPLST decreased as the number of features increased (large values in D), like Bibtex. Due to its clustering analysis directly applied on high-dimensional datasets, CBMLC consumed the second largest time on all the datasets. ECC consumed the largest time on all the seven datasets, resulting from the ensemble strategy. In summary, the proposed CLMLC is one of the best choices for MLC in the balance of performance and execution time, especially when Exact-Match or Macro/Micro-F1 is the principal goal and the practical processing speed is required in a large-scale problem.

Table 3: Execution time (10^3 sec) over seven large-scale datasets.

	Corel5k	Rcv1s1	Rcv1s2	Bibtex	Corel16k1	Corel16k2	Delicious
ECC	0.353	0.190	0.187	1.285	0.229	0.252	6.042
MLHSL	0.018	0.004	0.004	0.015	0.008	0.009	0.528
CPLST	0.042	0.045	0.036	0.223	0.042	0.042	0.558
CBMLC	0.097	0.112	0.127	1.002	0.131	0.151	1.916
CLMLC	0.005	0.004	0.004	0.014	0.010	0.010	0.567

To derive a more objective insistence on the experimental results, we conducted *Friedman test* [7] with significance level 0.05 (5 methods, 18 datasets). The results are shown in Table 4. Since the values of the Friedman Statistic F_F in terms of all metrics were higher than the Critical Value, the null hypothesis of equal performance was rejected. Then, we proceeded to a *Nemenyi testing* to confirm the difference between any two methods. According to [7], the performance of two methods is regarded as significantly different if their average ranks differ by at least the Critical Difference (CD). Figure 1 shows the CD diagrams for four evaluation metrics at 0.05 significance level. In each subfigure, the value of CD is given as a rule above the axis, where the averaged rank is marked. In Figure 1, the algorithms which are not significantly different are connected by a

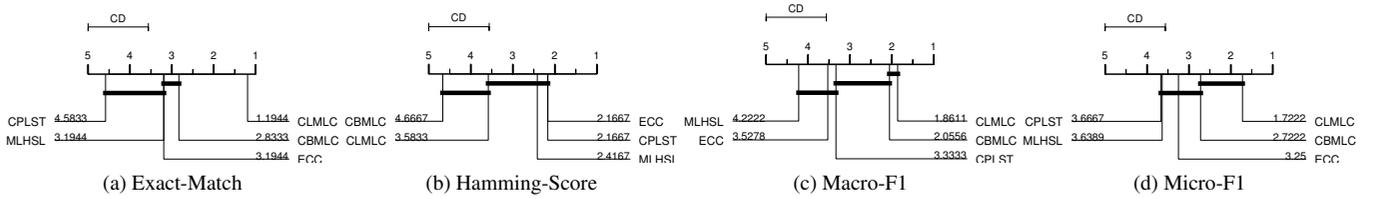


Figure 1: CD diagrams (0.05 significance level) of five comparing methods.

thick line. In summary, among 90 comparisons (5 methods \times 18 datasets), CLMLC achieved statistically superior performance than all the other methods in terms of Exact-Match. In Macro/Micro-F1, CLMLC achieved statistically comparable performances with CBMLC, and statistically superior performances than ECC, MLHSL and CPLST. Such observation demonstrates the competing performance of the proposed CLMLC in Exact-Match and Macro/Micro-F1, compared with the state-of-the-art MLC methods.

Table 4: Results of the Friedman Statistics F_F (5 methods, 18 datasets) and the Critical Value (0.05 significance level). The null hypothesis as the equal performance is rejected, if the values of F_F in terms of all metrics are higher than the Critical Value.

Friedman Test	Exact Match	Hamming Score	Macro F1	Micro F1
F_F	24.166	15.992	11.680	6.051
Critical Value	2.507			

Table 5 reports the reduced sizes of training datasets in CLMLC, which are averaged by 5-fold cross validation. Here “std.” shows the standard deviation of the values from K clusters. As shown in Table 5, consistently with our previous assumptions, there is strong locality in datasets, especially on datasets in text domain, like Medical, Rcv1 and Bibtex, where $\bar{L}^c \ll L$ in each data cluster c . Indeed the problem sizes in terms of N , D and L have been significantly reduced. For example, in Bibtex, the average problem size ($\bar{N}^c \times d \times \bar{L}^c$) in each cluster c has been reduced to nearly 1/30000 by CLMLC compared with the original set, bringing the fastest execution time on Bibtex (Table 3).

Table 5: Problem sizes of training datasets in CLMLC. The values were averaged by 5-fold cross validation. Here “std.” denotes the standard deviation.

Dataset	Original size			Reduced size			
	N	D	L	$\bar{N}^c \pm \text{std.}$	d	$\bar{L}^c \pm \text{std.}$	K
Birds	516	260	19	25.80 \pm 45.36	19	7.24 \pm 2.93	20
Genbase	530	1186	27	26.48 \pm 45.28	27	2.78 \pm 2.72	20
Medical	782	1449	45	39.12 \pm 39.47	30	4.68 \pm 2.97	20
Enron	1362	1001	53	68.08 \pm 66.81	30	23.85 \pm 6.54	20
Scene	1926	294	6	96.28 \pm 30.61	6	4.75 \pm 1.33	20
Yeast	1934	103	14	96.68 \pm 18.49	14	13.31 \pm 0.69	20
Corel5k	4000	499	374	40.00 \pm 18.18	30	54.08 \pm 25.19	100
Rcv1s1	4800	944	101	48.00 \pm 28.92	30	19.54 \pm 12.62	100
Rcv1s2	4800	944	101	48.00 \pm 31.34	30	18.51 \pm 11.78	100
Rcv1s3	4800	944	101	48.00 \pm 30.69	30	18.13 \pm 12.00	100
Rcv1s4	4800	944	101	48.00 \pm 33.80	30	14.36 \pm 9.94	100
Bibtex	5916	1836	159	59.16 \pm 39.44	30	29.95 \pm 20.41	100
Corel16k1	11013	500	164	110.13 \pm 42.59	30	71.31 \pm 22.18	100
Corel16k2	11009	500	164	110.09 \pm 48.86	30	71.32 \pm 24.37	100
Corel16k3	11008	500	154	110.08 \pm 44.93	30	69.11 \pm 22.39	100
Corel16k4	11070	500	162	110.70 \pm 48.46	30	70.73 \pm 22.91	100
Corel16k5	11078	500	160	110.78 \pm 46.55	30	72.82 \pm 23.64	100
Delicious	12884	500	983	128.84 \pm 155.55	30	333.48 \pm 200.60	100

4.4 Parameter sensitivity analysis

To evaluate the potentiality of CLMLC, a parameter sensitivity analysis was conducted. First, the parameters d and K were dealt with the Rcv1s1 and Bibtex datasets, where d controls the dimensionality of the feature subspace, and K is the number of data clusters. In this experiment, we kept the value of n by $\lceil L^c/5 \rceil$, and increased d from 5 to 100 by step 5, and K from 10 to 200 by step 10. Figure 2 shows the experimental results in terms of four evaluation metrics, whose values are averaged by 5-fold cross validation. In Figure 2, the warmer the color, the better the performance. We observe that as the values of d and K increased, its performance in Exact-Match and Macro/Micro-F1 upgraded, and then became stable once d and K reached 30 and 100, respectively. In contrast, as the values of d and K increased, its performance in Hamming-Score degraded, although the change was very slight (within 0.5%). Figure 3 shows the execution time for parameter sensitivity analysis, where $d \in \{10, 30, 50, 70, 90\}$. On both two datasets, the execution time increased as the value of d increased. As the value of K increased, the execution time first decreased, and then increased on Rcv1s1 but became stable on Bibtex. Thus, by considering trade-off between classification accuracy and execution time, we set values of d and K to those as stated in Section 4.2.

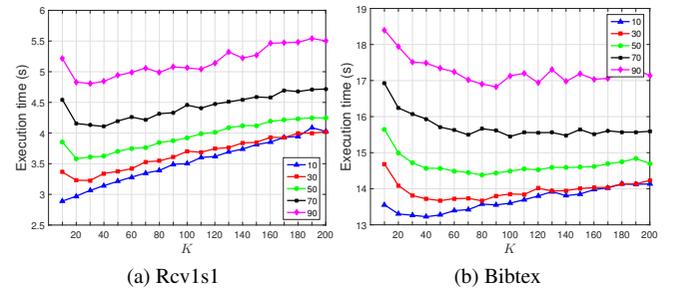


Figure 3: The execution time (sec) over different values of the dimensionality d of feature subspace and the number K of clusters on the Rcv1s1 and Bibtex datasets.

Next, keeping the values of d and K to 30 and 100, we conducted a sensitivity analysis over n , where n is the number of meta-labels for each cluster. Instead of directly varying the value of n , we increased x from 2 to 20 by step 1 as $n = \lceil L^c/x \rceil$. Figure 4 shows the experimental results in four metrics averaged by 5-fold cross validation. For convenience, the values of each metric were normalized by its maximum. As the value of x increased, the performance increased in Macro-F1, but decreased in Exact-Match. Note that performance in Hamming-Score seemed irrelevant to the change of x 's value. Thus, it was suggested to set smaller/larger value of x if the objective is to optimize Exact-Match/Macro-F1.

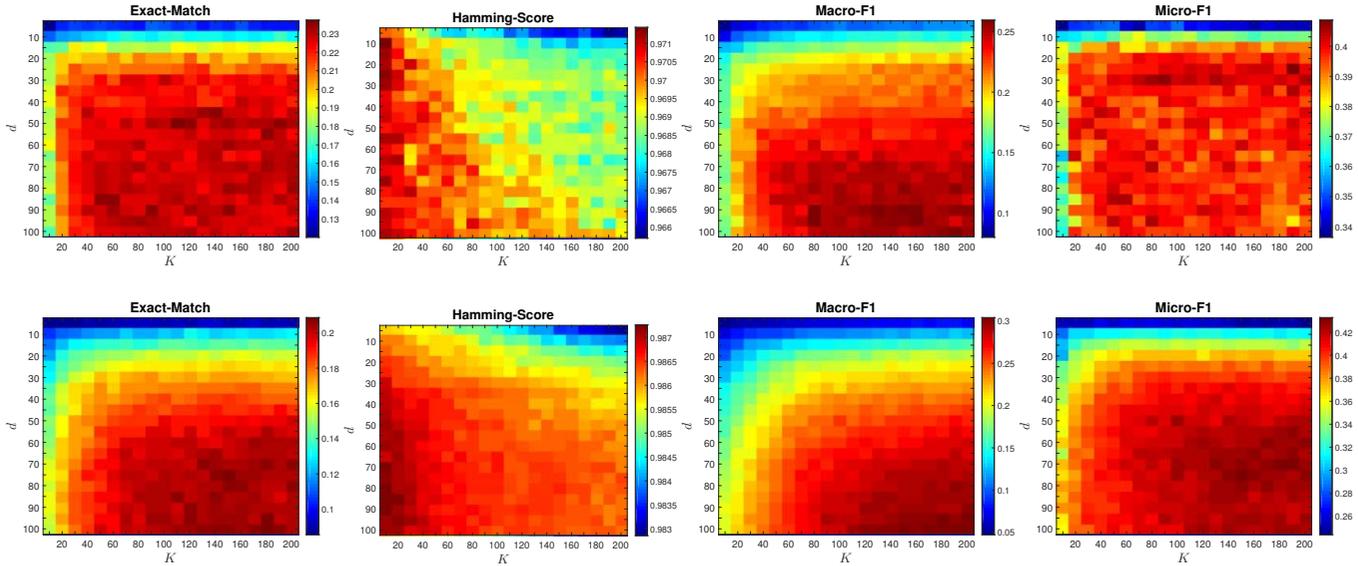


Figure 2: Parameter sensitivity analysis over the dimensionality d of feature subspace and the number K of clusters on the Rcv1s1 (the top row) and Bibtex (the bottom row) datasets ($n = \lceil L^c/5 \rceil$). The size of d/K was increased from 5/10 to 100/200 by step 5/10.

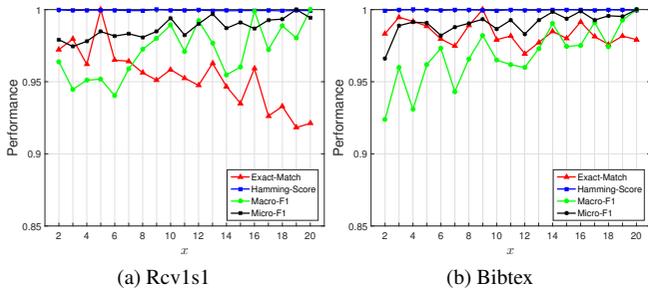


Figure 4: Parameter sensitivity analysis over x ($n = \lceil L^c/x \rceil$) on the Rcv1s1 and Bibtex datasets ($d = 30, K = 100$). The values of each metric were normalized by its maximum.

To optimize the parameters of MLHSL, CPLST and CBMLC, another set of parameter sensitivity analysis has been performed individually. Specifically, for MLHSL, d shared the similar tendency with CLMLC. For CPLST, the ratio of LS-DR remarkably influenced the experimental results. As the ratio increased, its performance upgraded. As the ratio approached 0.8/0.6 on regular/large-scale datasets, the performance became stable, while execution time increased dramatically. For CBMLC, as the number of cluster K increased, the values of evaluation metrics, except Hamming-Score, increased and became stable as K approached 100. Such observations validate the effectiveness of parameter configurations in Section 4.2.

5 CONCLUSION

In this paper, we have proposed a Clustering-based Local Multi-Label Classification (CLMLC) method, relying on the assumption that a multi-label dataset can be decomposed into several datasets of smaller sizes, where meta-labels exist and are relevant to only a fraction of features and training data. In CLMLC, by applying clustering analysis on the feature subspace, similar instances associated with similar labels are grouped together and then fed into local models. Extensive experiments conducted on real-world benchmark datasets

verified the validity of our assumption and demonstrated the efficiency of CLMLC. For the future work, we will seek a more appropriate method for building local models, which is currently a bottleneck for the application of CLMLC on extreme multi-label datasets.

ACKNOWLEDGEMENTS

This work was partially supported by JSPS KAKENHI Grant Numbers 15H02719 and China Scholarship Council.

REFERENCES

- [1] Zafer Barutcuoglu, Robert E. Schapire, and Olga G. Troyanskaya, ‘Hierarchical multi-label prediction of gene function’, *Bioinformatics*, **22**(7), 830–836, (2006).
- [2] K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain, ‘Sparse local embeddings for extreme multi-label classification’, in *Advances in Neural Information Processing Systems 28*, pp. 730–738, (2015).
- [3] M. Boutell, J. Luo, X. Shen, and C. Brown, ‘Learning multi-label scene classification’, *Pattern Recognition*, **37**(9), 1757–1771, (2004).
- [4] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*, Wadsworth and Brooks, Monterey, CA, 1984.
- [5] Y. Chen and H. Lin, ‘Feature-aware label space dimension reduction for multi-label classification’, in *Advances in Neural Information Processing Systems 25*, 1529–1537, (2012).
- [6] Krzysztof Dembczynski, Willem Waegeman, and Eyke Hullermeier, ‘An analysis of chaining in multi-label classification’, in *Proceedings of the 20th European Conference on Artificial Intelligence*, pp. 294–299, (2012).
- [7] Janez Demšar, ‘Statistical comparisons of classifiers over multiple data sets’, *Journal of Machine Learning Research*, **7**, 1–30, (2006).
- [8] J. Fürnkranz, E. Hullermeier, E.L. Mencia, and K. Brinker, ‘Multilabel classification via calibrated label ranking’, *Machine Learning*, **73**(2), 133–153, (2008).
- [9] Jun Huang, Guorong Li, Qingming Huang, and Xindong Wu, ‘Learning label specific features for multi-label classification’, in *2015 IEEE International Conference on Data Mining, 2015*, pp. 181–190, (2015).
- [10] Aram Karalić, ‘Employing linear regression in regression tree leaves’, in *Proceedings of the 10th European Conference on Artificial Intelligence*, pp. 440–441, (1992).

- [11] I. Katakis, G. Tsoumakas, and Vlahavas I., 'Multilabel text classification for automated tag suggestion', in *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases 2008 Discovery Challenge*, (2008).
- [12] J. Langford, T. Zhang, D. Hsu, and S. Kakade, 'Multi-label prediction via compressed sensing', in *Advances in Neural Information Processing Systems 22*, 772–780, (2009).
- [13] Z. Lin, G. Ding, M. Hu, and J. Wang, 'Multi-label classification via feature-aware implicit label space encoding', in *Proceedings of the 31st International Conference on Machine Learning*, pp. 325–333, (2014).
- [14] G. Nasierding, G. Tsoumakas, and A. Kouzani, 'Clustering based multi-label classification for image annotation and retrieval', in *IEEE International Conference on Systems, Man and Cybernetics*, pp. 4514–4519, (2009).
- [15] Y. Prabhu and M. Varma, 'Fastxml: a fast, accurate and stable tree-classifier for extreme multi-label learning', in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 263–272, (2014).
- [16] G. Qi, X. Hua, Y. Rui, J. Tang, T. Mei, and H. Zhang, 'Correlative multi-label video annotation', in *Proceedings of the 15th ACM International Conference on Multimedia*, pp. 17–26, (2007).
- [17] J. R. Quinlan, 'Learning with continuous classes', in *Proceedings of the Australian Joint Conference on Artificial Intelligence*, pp. 343–348. World Scientific, (1992).
- [18] J. Read, B. Pfahringer, G. Holmes, and E. Frank, 'Classifier chains for multi-label classification', *Machine Learning*, **85**(3), 333–359, (2011).
- [19] Juho Rousu, Craig Saunders, Sandor Szedmak, and John Shawe-Taylor, 'Learning hierarchical multi-category text classification models', in *Proceedings of the 22nd International Conference on Machine Learning*, pp. 744–751, (2005).
- [20] L. Sun, B. Ceran, and J. Ye, 'A scalable two-stage approach for a class of dimensionality reduction techniques', in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 313–322, (2010).
- [21] L. Sun, S. Ji, and J. Ye, 'Hypergraph spectral learning for multi-label classification', in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 668–676, (2008).
- [22] L. Sun, S. Ji, and J. Ye, 'Canonical correlation analysis for multilabel classification: A least-squares formulation, extensions, and analysis', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **33**(1), 194–200, (2011).
- [23] Luís Torgo, 'Functional models for regression tree leaves', in *Proceedings of the 14th International Conference on Machine Learning, Nashville, Tennessee, USA, July 8-12, 1997*, pp. 385–393, (1997).
- [24] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas, 'Multi-label classification of music into emotions', in *Proceedings of the 9th International Conference on Music Information Retrieval*, pp. 325–330, (2008).
- [25] G. Tsoumakas, I. Katakis, and I. Vlahavas, 'Effective and efficient multilabel classification in domains with large number of labels', in *Proceedings of ECML/PKDD 2008 Workshop on Mining Multidimensional Data*, (2008).
- [26] G. Tsoumakas, I. Katakis, and L. Vlahavas, 'Random k-labelsets for multilabel classification', *IEEE Transactions on Knowledge and Data Engineering*, **23**(7), 1079–1089, (2011).
- [27] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas, 'Mulan: A java library for multi-label learning', *Journal of Machine Learning Research*, **12**, 2411–2414, (2011).
- [28] Celine Vens, Jan Struyf, Leander Schietgat, Sašo Džeroski, and Hendrik Blockeel, 'Decision trees for hierarchical multi-label classification', *Machine Learning*, **73**(2), 185–214, (2008).
- [29] H. Wang, C. Ding, and H. Huang, 'Multi-label linear discriminant analysis', in *Proceedings of the 11th European Conference on Computer Vision*, volume 6316, 126–139, (2010).
- [30] D. Watkins, *Chemometrics, mathematics and statistics in chemistry*, Reidel Publishing Company, Dordrecht, Netherlands, 1984.
- [31] J. Weston, S. Bengio, and N. Usunier, 'Wsabie: Scaling up to large vocabulary image annotation', in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pp. 2764–2770, (2011).
- [32] K. Worsley, J. Poline, K. Friston, and A. Evans, 'Characterizing the response of PET and fMRI data using multivariate linear models', *Neuroimage*, **6**(4), 305–319, (1997).
- [33] Yiming Yang and Xin Liu, 'A re-examination of text categorization methods', in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '99*, pp. 42–49, New York, NY, USA, (1999). ACM.
- [34] H. Yu, P. Jain, P. Kar, and S. Dhillon, 'Large-scale multi-label learning with missing labels', in *Proceedings of the 31st International Conference on Machine Learning*, pp. 593–601, (2014).
- [35] Kai Yu, Shipeng Yu, and Volker Tresp, 'Multi-label informed latent semantic indexing', in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '05*, pp. 258–265, (2005).
- [36] B. Zhang, 'Regression clustering', in *Proceedings of the 3rd IEEE International Conference on Data Mining*, pp. 451–458, (2003).
- [37] M. Zhang and L. Wu, 'Lift: multi-label learning with label-specific features', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **37**(1), 107–120, (2015).
- [38] Y. Zhang and Z. Zhou, 'Multi-label dimensionality reduction via dependence maximization', in *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, pp. 1503–1505, (2008).

Multiscale Triangular Centroid Distance for Shape-Based Plant Leaf Recognition

Chengzhuan Yang and Hui Wei and Qian Yu¹

Abstract. The shapes of plant leaves are very important to plant ecologists and botanists because these can help distinguish plant species as well as serve as health indicators. In this paper, we present a novel contour-based shape descriptor named multiscale triangular centroid distance (MTCD) for plant leaf recognition. MTCD features at different triangles are extracted from each contour point to provide a compact, multiscale shape descriptor. Both local and global features of a plant leaf are effectively captured by the proposed method. A simple cosine distance is used to calculate the dissimilarity measurement between MTCD descriptors. Therefore, MTCD is a rapid approach for shape matching and is suitable for real-time application. The proposed method has been evaluated using four publicly available plant leaf datasets, including the Swedish Leaf dataset, the Smithsonian Leaf dataset, the Flavia Leaf dataset, and the ImageCLEF2012 Leaf dataset. The experimental results show that this novel approach can achieve high recognition accuracy. Comparisons with other state-of-the-art shape-based plant leaf recognition methods further demonstrate the effectiveness and efficiency of MTCD.

1 INTRODUCTION

There exists a large quantity of plant species on Earth [34]. However, the precise identification of every plant species is a challenging task, particularly for non-expert stakeholders such as land managers, foresters, and agronomists. This is because plant species identification requires specialized knowledge and in-depth training in botany and plant systematics. Therefore, an automatic plant identification system is very important for general use because it can assist in rapidly distinguishing a large number of plant species. This identification system may be helpful even for experienced botanists and plant ecologists.

Plant identification mainly involves examination of various organs such as flowers, leaves, fruit, stem and bark. A review of some existing methods for plant species identification can be found in [7]. Leaves are often used for plant identification because its features are more universal and consistent. Leaves can be characterized based on shape, color, and texture. The color and texture of plants may vary over time and under different environmental conditions, whereas leaf shape has the most discriminative power. Botanists often use the shape of leaves for species identification. In addition, leaf shape is the easiest to extract from images, and the overall shape structure of a leaf may be preserved even though the leaf specimen is damaged by age or by insects.

In this paper, we concentrate on the shape of the leaf and on shape-based methods for plant leaf recognition. Shape is one of the most important features of an object and plays an important role in a diverse range of applications such as content-based image retrieval [29], object detection [10, 20], robot navigation [14], and object tracking [15, 27], to name a few. In shape recognition methods, there are usually two crucial parts: shape representation and shape matching. Shape representation can be roughly divided into two classes of methods: contour-based methods and region-based methods [38]. The contour-based methods have been significantly more popular than the region-based ones in the past decade. This is because human beings can readily distinguish shapes based on contour features. At the same time, we are only interested in the contours of shape in various applications, i.e., contour-based object detection [10], whereas the interior content is less important. In the same way, in the present study, we concentrated on contour-based methods for shape recognition. Recently, several important contour-based approaches have been proposed, which can be classified as either global or local.

Global methods consider the shape as a whole, representing it by a single global descriptor such as Fourier-based descriptors [17], curvature scale space [25] descriptors, polygonal multi-resolution and elastic matching (PMEM) [2], contour points distribution histogram (CPDH) [31], and binary angular pattern [29]. The advantages of global descriptors include generally robust to moderate amounts of noise and the efficiency of shape matching. However, these approaches face major difficulties in capturing the finer details of shape boundaries. Therefore, the accuracy of shape matching is lower.

Local methods represent a shape by a set of local shape descriptors, each corresponding to a certain part of the shape or simply to a sample point along its contour. A classic example in this category is shape context (SC) [3], which describes a shape by a set of two-dimensional (2-D) histograms that capture landmark distributions. The inner distance shape context (IDSC) [21] is an extension of SC, which replaces the Euclidean distance with the inner distance to achieve robustness against articulation. The hierarchical Procrustes matching (HPM) method, which was proposed by McNeill et al. [24], captures shape information across different hierarchies. The shape tree (ST) was proposed by Felzenszwalb et al. [9]; it can capture hierarchical geometric propensities of a shape. In [37], a shape descriptor called contour flexibility (CF) was proposed, which describes each contour point by its deformable potential. Wang et al. [33] described a shape descriptor called height function (HF) for shape matching and retrieval. Subsequently, Liu et al. [22] presented a shape matching framework that utilizes a shape descriptor named metric partition constraint (MPC), in which various metric methods can be included. Recently, Jia et al. [13] presented a novel shape descriptor called hierarchical characteristic number contexts (HCNC),

¹ Laboratory of Cognitive Model and Algorithm, Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University, Shanghai, China, email: {chengzhuanyang13, weihui.yuqian12}@fudan.edu.cn

which is invariant to affine and projective transformations. The advantages of local shape descriptors include superiority in describing the fine details of a shape. At the same time, local shape descriptors can usually achieve a higher accuracy of shape matching than global ones. However, such superiority is typically at the cost of reduced efficiency in terms of computation time. The above local methods usually adopt a point-to-point alignment process, which is then followed by dynamic programming (DP) to find the best alignment between corresponding points of the two shapes. As is known to all, the use of a DP algorithm to find the correspondence points between two shapes is extensively time-consuming. Therefore, these approaches are not suitable for retrieval tasks involving large-scale datasets and the task of real-time application.

In this paper, we propose a novel contour-based shape descriptor called multiscale triangular centroid distance (MTCD) to capture both local and global features of a shape while being invariant to translation, rotation, and scaling. In addition, our shape descriptor uses a multiscale representation that can better capture the geometric propensities of a shape. At the same time, when using MTCD for shape recognition, it is not necessary to use DP to find point wise correspondence, which in turn makes MTCD an efficient shape descriptor. In the experiments, we only use a very simple cosine distance to compute the dissimilarity measurement between MTCD descriptors. Therefore, the proposed shape descriptor is computationally very efficient. We applied the proposed method to the task of plant leaf recognition using experiments involving four datasets: the Swedish Leaf dataset, the Smithsonian Leaf dataset, the Flavia Leaf dataset, and the ImageCLEF2012 Leaf dataset. The experimental results indicate that the proposed method can achieve higher recognition accuracy than the state-of-the-art shape-based plant leaf recognition methods. Comparisons with other shape-based approaches indicate that the computation time of our method is lower, thereby achieving a faster plant leaf recognition speed.

The remainder of this paper is organized as follows. A brief review of related work is presented in Section 2. In Section 3, we describe the details of the proposed MTCD shape descriptor. Experimental results are provided in Section 4. The final section presents our conclusions and future work.

2 RELATED WORK

Plant leaf recognition has recently attracted research efforts in computer vision and related areas [26]. This task can be seen as a particular case of the more general image classification, which has been extensively studied by the computer vision and pattern recognition communities. There are already some studies on plant leaf recognition by using only shape information. Therefore, in the following, we briefly review some important studies that are relate to the subject of this paper.

Most of shape-based leaf recognition approaches use geometric features to characterize the shape of a leaf. The first group of approaches extracts various plant morphological characteristics such as aspect ratio, rectangularity, convex area ratio, and so on. For example, Wu et al. [36] used aspect ratio, rectangularity, and narrow factor as features, and employed a neural network as a classifier for plant leaf classification. Du et al. [8] extracted invariant moment features and geometric features, including aspect ratio, rectangularity, area ratio of convexity, and eccentricity to describe leaf shape. Pahalawatta et al. [28] used a different set of features, namely, stem-to-blade ratio and compactness, for plant species identification. Caballero and Aranda [4] combined geometric features with shape de-

scriptors for effective plant leaf image retrieval. Cerutti et al. [5] presented a method for plant species identification by using high-level geometrical descriptors and an implementation of the system is available in a mobile application. A common disadvantage of these approaches is that it is generally difficult to accurately extract geometric features using imperfect measurements, and these methods usually cannot distinguish a large number of species.

The second group of approaches makes use of shape descriptors for plant leaf recognition. Soderkvist [32] combined curvature scale space, Fourier descriptor, and Hu's moments in building a tree-structured classification system, which was then tested on the Swedish leaf dataset. Ling et al. [21] use the proposed IDSC descriptor for plant leaf classification, and they achieved good experimental results. Subsequently, this methodology was further exploited for developing a working system [26] to assist in the identification of plant species. In [9], shape-tree was also used for plant leaf recognition, which then generated better experiment results using the Swedish leaf dataset [32]. Hearn [12] applied Fourier descriptors to the automated identification of plant leaves. Hu et al. [30] proposed a contour-based shape descriptor named multiscale distance matrix (MDM) for fast plant leaf recognition. They used the matrix of pairwise distances between points sampled on the boundary of a leaf to capture the geometric structure of a shape while being invariant to translation, rotation, scaling, and bilateral symmetry. Laga et al. [19] described a closed planar curves representation by using squared root velocity function (SRVF) for plant leaf shape analysis. Kumar et al. [18] described a mobile app for identifying plant species by using an automatic visual recognition system. Recently, we observed that some methods combined shape features with other characteristics for plant leaf recognition. For example, Chaki et al. [6] presented a novel methodology for characterizing and recognizing plant leaves using a combination of texture and shape features. Kalyoncu and Toygar [16] were the first to propose some new features that may be used to distinguish plant leaf margins. They then used geometric features, MDM, moment invariants, as well as the proposed new features for plant leaf classification.

3 THE PROPOSED MTCD SHAPE DESCRIPTOR

In this section, we will focus on shape feature extraction and dissimilarity measures. First, we introduce the proposed MTCD shape descriptor in detail. Second, a shape dissimilarity measurement based on the proposed descriptor is used for shape matching. Third, we analyze the time complexities of the proposed shape descriptor for plant leaf recognition. Finally, the properties of the proposed shape descriptor are discussed.

3.1 MTCD Shape Descriptor

Let $P_i = (x_i, y_i)$, $(i = 1, 2, \dots, N)$ denote the sequence of equidistant sample points on the outer contour of a given shape S , which is generated by counter-clockwise direction tracing of the boundary at a constant speed, and where x_i and y_i are the coordinates of the point P_i , P_1 is the starting point, and N is the number of boundary point. Because the shape boundary is closed, we have $P_{N+1} = P_1$. For each point $P_i = (x_i, y_i)$ of the shape S , we can find its two adjacent points $P_{i+t} = (x_{i+t}, y_{i+t})$ and $P_{i-t} = (x_{i-t}, y_{i-t})$, where $i \in [1, N]$ and $t \in [1, T]$. T represents the number of scales, which takes $T = \lfloor (N-1)/2 \rfloor$ in the experiment, where $\lfloor (N-1)/2 \rfloor$ is the floor value of $(N-1)/2$. The above three consecutive points can

form a triangle $\triangle P_{i-t}P_iP_{i+t}$. We then compute the coordinates of the centroid point $g_{it} = (x_{g_{it}}, y_{g_{it}})$ of this triangle, which is given by the following expression:

$$\begin{cases} x_{g_{it}} = (x_{i-t} + x_i + x_{i+t})/3 \\ y_{g_{it}} = (y_{i-t} + y_i + y_{i+t})/3 \end{cases} \quad (1)$$

For each point P_i of the contour of an object, we can obtain T triangles by the above ways. We then compute the distances between this point and the centroid points g_{it} , ($i \in [1, N], t \in [1, T]$) of all triangles. Thus, we can obtain the MTCD shape descriptor of this point P_i , which can be expressed by using the following expression:

$$MTCD(P_i) = (TCD(P_i, g_{i1}), \dots, TCD(P_i, g_{iT})). \quad (2)$$

Here, $TCD(P_i, g_{it}) = \sqrt{(x_i - x_{g_{it}})^2 + (y_i - y_{g_{it}})^2}$, where (x_i, y_i) and $(x_{g_{it}}, y_{g_{it}})$ represent the coordinates of point P_i and the centroid point g_{it} , respectively. Therefore, we can obtain the MTCD descriptor of shape S , as expressed by the following equation:

$$\begin{aligned} MTCD(S) &= (MTCD(P_1), \dots, MTCD(P_N)) \\ &= \begin{pmatrix} TCD(P_1, g_{11}) & \dots & TCD(P_N, g_{N1}) \\ \vdots & \ddots & \vdots \\ TCD(P_1, g_{1T}) & \dots & TCD(P_N, g_{NT}) \end{pmatrix}. \end{aligned} \quad (3)$$

We observe that $MTCD(S)$ is an $T \times N$ matrix with column i being the shape descriptor $MTCD(P_i)$ of the sample point P_i of shape S .

Figure 1 shows an example of MTCD extracted from a leaf shape. For the leaf shape depicted in Figure 1(a), we sampled 128 points on the contour of the shape, which is shown in Figure 1(b). The MTCD descriptor of this leaf shape is shown in Figure 1(e), and the size of this descriptor is 63×128 . In this feature matrix, blue entries represent small values, whereas red entries represent large ones. The rows of this descriptor represent the shape descriptor in one scale. The columns of this descriptor represent each point of the shape descriptor in all scales. Figure 1(c), (d), and (f) depict rows 1, 20, and 63 of the MTCD, respectively. It can be seen that the first row depicts the finest level of the shape and the last row presents the coarsest level of the shape. Therefore, this descriptor can capture the local and global features of the leaf shape.

From the definition of the MTCD descriptor, we can easily prove that the MTCD descriptor has intrinsic invariance to translation of the shape contour. For the case of rotating the shape contour, the position of the starting point of the contour will be changed and make the MTCD descriptor shift by l , i.e., $MTCD(P_i) = MTCD(P_{i+l})$, where l is the displacement of the starting point. By checking the properties of this descriptor, we can find that the MTCD descriptor is not scale invariant. To make our shape descriptor scale invariant, we row wise normalize $MTCD(S)$ by dividing by the maximal absolute value of each row as follows:

$$TCD(P_i, g_{it}) = \frac{TCD(P_i, g_{it})}{\max_{i=1}^N \{TCD(P_i, g_{it})\}}. \quad (4)$$

To distinguish the usual normalization process by some global measure for the shape contour, the normalization process according to Eq. (4) can be considered as local normalization. The advantage of this kind of normalization is analyzed in detail in [1]. Consequently, the value of each entry in the matrix $MTCD(S)$ after normalization is in the interval $[0, 1]$.

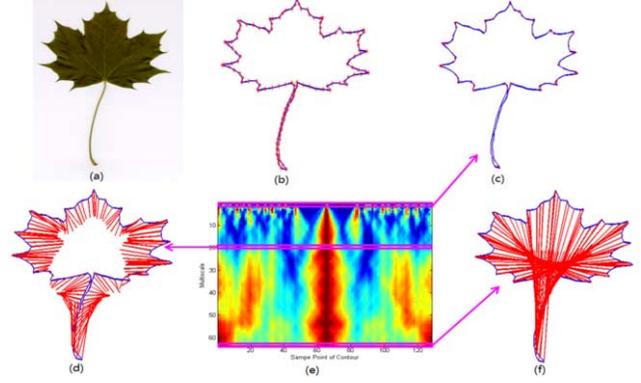


Figure 1. An example of MTCD extracted from a leaf shape. (a) The leaf image; (b) the contour shape of the leaf (a), and the sample point is painted in red color; (c) The extracted MTCD of the leaf; The red lines in (c), (d), and (f) correspond to the first, twenty, and last rows of the MTCD, respectively.

Here, we apply Fourier transforms to each row of the MTCD descriptor, and discard the phase information to obtain invariance to rotation of the shape contour. At the same time, the dimensionality of the proposed shape descriptor can be further reduced through the Fourier transforms, which can improve the efficiency and effectiveness of the shape descriptor for shape matching. To explain the Fourier transform to each row of the MTCD descriptor conveniently. Next, we use the d_t , $t \in [1, T]$ that represents each row of the MTCD descriptor. The discrete Fourier transforms for the d_t is given by

$$FD_t(i) = \frac{1}{N} \sum_{u=1}^N d_t(u) \exp\left(\frac{-j2\pi(u-1)i}{N}\right), i = 1, 2, \dots, N. \quad (5)$$

where $j^2 = -1$, and the $abs(FD_t(i))$ is the absolute value of $FD_t(i)$ and represents the magnitudes of the discrete Fourier transform coefficients $FD_t(i)$. It is not difficult to prove that $abs(FD_t(i))$ is invariant to the rotation. Therefore, we use the magnitudes of the Fourier transform coefficients to describe the shape. To make the generated shape descriptor robust and compact, the lowest M order coefficients are used, where $M \ll N$. The resulting final MTCD descriptor is used for plant leaf recognition, which is defined as follows:

$$MTCD = \{abs(FD_t(v)) | t = 1, 2, \dots, T; v = 1, 2, \dots, M\}. \quad (6)$$

From the definition of the final MTCD descriptor, we can observe that the size of this descriptor becomes small. The dimensionality of this descriptor ranges from $T \times N$ to $T \times M$, where $M \ll N$. Therefore, the descriptor can further improve the efficiency and reduce the storage of space for plant leaf recognition.

3.2 Shape Dissimilarity Measure

In this subsection, we will introduce how to measure the shape dissimilarity between the two MTCD descriptors. Let $MTCD_A = \{abs(FD_t^A(v)) | t = 1, 2, \dots, T; v = 1, 2, \dots, M\}$ and $MTCD_B = \{abs(FD_t^B(v)) | t = 1, 2, \dots, T; v = 1, 2, \dots, M\}$ denote the MTCD descriptors that are extracted from shapes A and B, respectively. To compare conveniently, we convert $MTCD_A$ and $MTCD_B$ descriptors to a row vector. Thus, we obtain $MTCD_A = \{abs(FD_1^A), \dots, abs(FD_S^A) | S = 1, 2, \dots, T \times M\}$

and $MTCD_B = \{abs(FD_1^B), \dots, abs(FD_S^B) | S = 1, 2, \dots, T \times M\}$, respectively. The shape dissimilarity between these descriptors can be calculated by using the following equation:

$$Dist(A, B) = 1 - \frac{\sum_{k=1}^S abs(FD_k^A)abs(FD_k^B)}{\sqrt{\sum_{k=1}^S (abs(FD_k^A))^2} \sqrt{\sum_{k=1}^S (abs(FD_k^B))^2}} \quad (7)$$

We use the cosine distance to measure the difference degree of the two MTCD descriptors. The smaller the cosine distance, the more similar the two shapes. It can be seen from Eq. (7) that the MTCD features, $abs(FD_i^A(v))$ and $abs(FD_i^B(v))$, of two shapes in each scale are compared, respectively. Thus, the computation of cosine distance is very simple and is also very efficient for shape matching in plant leaf recognition.

3.3 Time Complexity Analysis

The elapsed time of shape recognition can be divided into two parts. One is the time of calculating the shape descriptor, the other is the time of shape matching. During the extraction of the shape features for each scale level $t = 1, 2, \dots, T$, calculating the triangular centroid distance for all the contour points requires time $O(N)$. Therefore, the total time of calculating the MTCD shape descriptor is $O(NT)$. At the same time, in order to obtain invariance to rotation of the shape contour and reduce the dimensionality of the proposed shape descriptor. We apply the Fourier transform to the MTCD descriptor for each scale level $t = 1, 2, \dots, T$. The complexities of calculating the Fourier transform coefficients $FD_t(i)$ is $O(TN \log_2 N)$ for all scale levels T (using the fast algorithm for calculating the discrete Fourier transform). Therefore, the total time of calculating the MTCD descriptor is $O(NT) + O(TN \log_2 N) = O(TN \log_2 N) = O(\lfloor (N-1)/2 \rfloor N \log_2 N)$.

In the shape matching stage, the time of computing Eq. (7) is $O(S) = O(MT) = O(M(\lfloor (N-1)/2 \rfloor))$, where N is the number of sample points in the shape contour, and $M \ll N$ is the number of Fourier coefficients used for the MTCD descriptor. In the experiment, because M and T are small, the computation of cosine distance is very fast. Our current implementation takes about 0.5 milliseconds to compute a matching in a 3.1 Ghz computer. Thus, the proposed method is highly efficient in shape-based plant leaf recognition.

3.4 The Summary of the Characteristics of the Proposed MTCD

The proposed MTCD shape descriptor has the following properties that render it highly suitable for large-scale image dataset recognition task. We will also explain the advantages of using the MTCD descriptor in detail.

Invariance to similarity transforms: Similarity transforms, including translation, rotation, and uniform scaling invariance, are a fundamental criteria required by the MPEG-7 standard [23]. The proposed MTCD shape descriptor has intrinsic invariance to translation, and at the same time, the local normalization is adopted to obtain the scale invariant. In the end, we apply the discrete Fourier transforms to obtain the rotation invariance descriptor.

Compactness of the MTCD descriptor: The number of features from each scale level $t = 1, 2, \dots, T$ is M . Thus, the total number of features of the MTCD descriptor is MT . In our experiment, we set $N = 128$ and $M = 16$, making the proposed descriptor very compact, with only 1,008 features, thus requiring less memory to

store the descriptor. On the other hand, the famous inner distance shape descriptor (IDSC) [21] uses $N \times N_d \times N_\theta$ features, where N is the number of sample points on the boundary of shape, N_d is the number of inner-distance bins, and N_θ is the number of inner-angle bins. In [21], these parameters are set to $N = 128$, $N_d = 8$, and $N_\theta = 12$ in the experiments. Hence, the features of the IDSC descriptor are $128 \times 8 \times 12 = 12,288$, which is far greater than the proposed method.

Multiscale representation structure: The proposed MTCD descriptor can capture the local and global features of a leaf shape. From the definition of the MTCD descriptor, we can find that the large scale can embody the coarse level information of a shape and the small scale can embody the finest level information of the shape. From the coarse scale to the fine scale, a multiscale representation structure can be formed. It can be proved by experiments that this multiscale representation structure can contain rich information about the shape of an object and can more efficiently capture the geometric properties of a shape.

4 EXPERIMENTAL RESULTS

In this section, we test our method on four leaf datasets: the Swedish Leaf dataset [32], the Smithsonian Leaf dataset [21], the Flavia Leaf dataset [36], and the ImageCLEF2012 Leaf dataset [11]. In the experiments, the same parameters ($N = 128, M = 16$) are used for the proposed approach. The complete algorithm was implemented by MATLAB. The experimental platform employed a PC with a Quad-Core 3.1 GHZ CPU and 4GB memory.

4.1 The Swedish Leaf dataset

The Swedish Leaf dataset is a well-known public dataset generated by a leaf classification project at Linköping University and the Swedish Museum of Natural History [32]. This dataset consists of 15 species of leaves, with 75 images per species for a total of 1,125 images. Some example images from this dataset are shown in Figure 2. The Swedish leaf dataset is very challenging due to its high inter-species similarity. Note that some species are quite similar from the view of leaf shape such as the first, third, and ninth species.

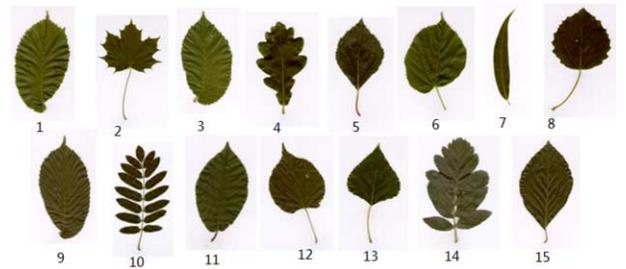


Figure 2. Leaves from the Swedish leaf dataset, one image per species.

To compare our approach with existing methods, the same performance evaluation standard used in [21] and [32] is adopted in our experiment. We randomly selected 25 training images from each species and classified the remaining images using a nearest neighbor approach. Table 1 compares our classification rate to the other shape-based methods that have been tested on this dataset. The results of other methods are directly drawn from the published results.

Table 1. Classification results on the Swedish Leaf dataset

Method	Recognition rate
Soderkvist [32]	82.40%
SC+DP [21]	88.12%
Fourier Descriptor [21]	89.60%
IDSC+DP [21]	94.13%
Shape-tree [9]	96.28%
Spatial PACT [35]	90.61%
MDM [30]	93.60%
Our Method	96.31%

As can be observed, our method achieved the highest classification rate among the eight methods. At the same time, the efficiency of our method is very high. Among the other methods, the shape-tree method achieved a slightly lower rate compared to our method. However, this method is still a point-wise matching method, and is too slow for application in a real-time system.

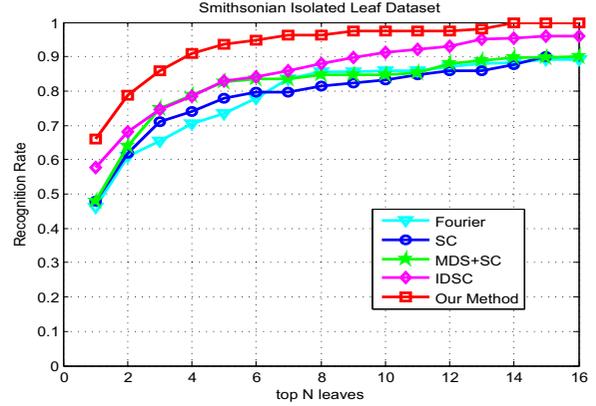
4.2 The Smithsonian Leaf dataset

The Smithsonian Leaf dataset is a collection of isolated leaves from the Smithsonian project [21]. This dataset contains a total of 343 leaf images from 93 species. The number of leaves from different species varies. Figure 3 shows some sample images from this dataset. The leaf images from this dataset are vulnerable to lighting changes and the leaves may not have been well flattened.

**Figure 3.** The Smithsonian leaf dataset contains a total of 343 leaf images from 93 species. One typical image from each species is shown.

To directly compare our approach with existing methods, we adopted the evaluation protocol used in [21]. We randomly selected 187 leaves to form a training set, and performed testing on the remaining 156 ones. The retrieval performance was evaluated using performance curves that show the recognition rate among the top N leaves, where N varies from 1 to 16. The recognition rate versus the top N leaves curve is presented in Figure 4.

The curves in Figure 4 are the recognition rates indicating how often the testing leaves can be correctly classified based on the top N candidates. It can be noticed that such percentage increases monotonically with respect to N , and reduces to the 1-NN recognition rate when $N = 1$. As can be observed, our method achieved the best recognition rate among the five methods. Except for the results generated by our method, the results of the other approaches were derived from [21]. The MDS+SC method represents the combination of multidimensional scaling (MDS) and shape context (SC). It's also

**Figure 4.** Recognition rate on the Smithsonian leaf dataset.

noticeable that the recognition rate of our method is 8% higher than that using the IDSC method that retrieves only one candidate result.

4.3 The Flavia Leaf dataset

The Flavia Leaf dataset contains a total of 1,907 leaf images belonging to 32 different species, with the number of samples in each ranging from 50 to 77. Figure 5 shows some samples from this dataset.

**Figure 5.** Some samples from the Flavia leaf dataset, which contains 1907 leaf images from 32 species. One sample per species is shown.

Several approaches were tested in [19] on the Flavia leaf dataset. To compare our approach with these methods, the same evaluation metrics as those in [19] were used in our experiment. In [19], two evaluation metrics were employed: the Mean Average Precision (MAP) and the precision-recall curve. Precision P is defined as $P = r/n$, and recall R is defined as $R = r/m$, where n is total of number of the retrieved images, r is the number of relevant images to the query image among the n retrieved images, and m is the total number of relevant images in the whole dataset.

The MAP value is measured on a set of queries Q and is defined as follows:

$$MAP = \frac{\sum_{q \in Q} AP(q)}{|Q|}. \quad (8)$$

Here, $|Q|$ is the number of queries. $AP(q)$ is the average precision

score for each query q and is defined as

$$AP(q) = \frac{\sum_{k=1}^M (P(k) \times f(k))}{N}, \quad (9)$$

where $P(k)$ is the precision at cut-off k in the list of retrieved images, and $f(k)$ is equal to 1 when the image at rank k is relevant to query q and 0 otherwise. M is the number of retrieved images and N is the number of retrieved relevant images for query q . The higher the MAP score is, the better is the performance. The MAP scores of our method with the other methods are shown in Table 2. The results of other methods were directly drawn from the published results [19, 30]. Owing to the results of the IDSC method are not provided on this dataset. However, by running the released code of [21], we were able to obtain the MAP values and precision-recall curve of the IDSC method. In Table 2, the GEDT method is short for shape context of the Gaussian Euclidian distance transform of the shape boundaries.

Table 2. Mean Average Precision (MAP) on the Flavia Leaf dataset

Method	MAP
GEDT [19]	48.01%
IDSC [21]	59.32%
MDM [30]	47.91%
Riemannian metric [19]	57.21%
Our Method	61.84%

It can be seen from Table 2 that our method achieved the best MAP scores. Despite the high similarity of the shapes of different species, our method indicated a high capability of distinguishing various species.

Figure 6 shows precision-recall curve, which is often used to evaluate image retrieval performance. Precision measures retrieval accuracy and recall speed, and recall measures the robustness of the retrieval. Figure 6 shows that our method achieved the best precision-recall curve among all of the compared methods.

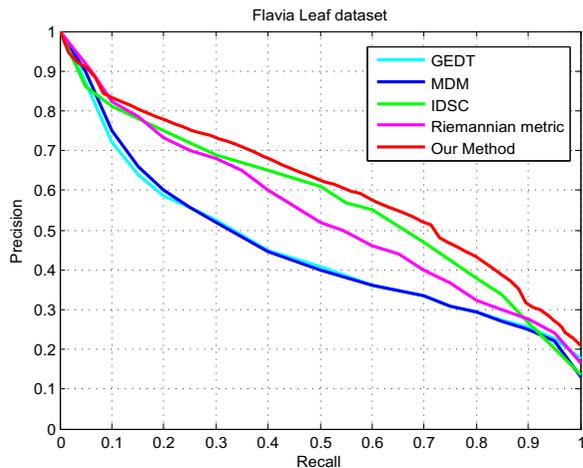


Figure 6. Precision-recall curves on the Flavia leaf dataset.

4.4 The ImageCLEF2012 Leaf dataset

The ImageCLEF2012 Leaf dataset was created by Goëau et al. and was introduced in [11]. To the best of our knowledge, it is the largest dataset currently available for research. This dataset contains a total of 11572 images that are subdivided into three different categories of leaf images: "scans", "scans-like photos" and "photographs". The class of "scan" images contain a white background and minimal shadowing. The class of "scan-like photo" images includes photographs with a white backdrop, but may exhibit heavy shadowing. The images of class of "photographs" contain unconstrained photos with natural background. Some samples of these three categories of images are shown in Figure 7.



Figure 7. Some samples of the three categories of images from the ImageCLEF2012 leaf dataset. The first, second, and third rows represent the class of "scans", "scans-like photos", and "photographs" images, respectively.

We only use the "scans" class images in the experiments, which accounts for 57% of the entire dataset. This is because we only focused on shape-based plant leaf recognition, whereas the shapes of the "scans" class images can be more reliably extracted by using a preprocessing step. There are 6,630 leaf images in this subset, which contains 115 different species, with the number of samples in each leaf ranging from 2 to 249. Some samples from the "scans" category subset of the ImageCLEF2012 dataset are shown in Figure 8.



Figure 8. Some samples from the "scans" subset of the ImageCLEF2012 dataset, which contains a total of 6,630 leaf images from 115 different species. One sample per species is shown.

The "scans" category dataset is more challenging than the other datasets in previous sections because this dataset contains a large number of different species with similarity leaf shapes in inter-class species and dissimilarity leaf shapes in intra-class species. This subset contains 4,870 images for training and 1,760 images for testing. The performance metrics used are the same as for the Smithsonian Leaf dataset, which is described in Subsection 4.2. We compare our method with the methods of IDSC [21] and MDM [30]. Because the results of IDSC and MDM are not provided on this dataset. Thus, we implement the method of MDM and obtain the recognition results on this dataset. At the same time, we also run the released code of IDSC method and get the results on this dataset. The recognition rates are plotted in Figure 9.

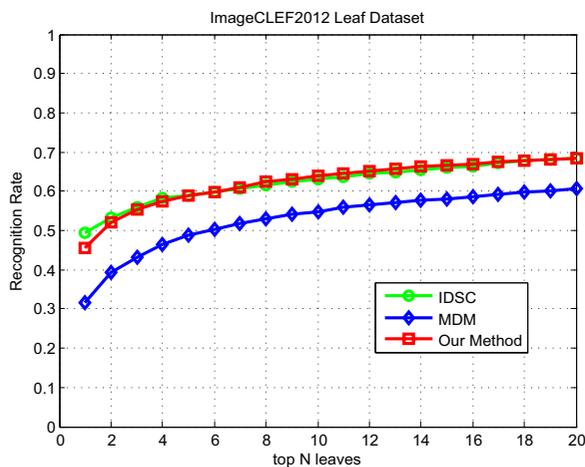


Figure 9. Recognition rate on the ImageCLEF2012 leaf dataset.

It can be seen from Figure 9 that the result of our method is significantly better than the MDM method. At the same time, our method gives a comparable performance to the IDSC method. However, our approach is faster than the IDSC method. Therefore, our approach is more suitable for the large-scale dataset of retrieval task and the task of real-time application.

To illustrate the efficiency of our method, we report the average CPU time of completing one test, including feature extraction and matching, which gives us a rough feeling of the advantage of our method. We compared our method with the IDSC method because the IDSC method is one of the most popular and important methods for shape-based leaf recognition. In addition, the released code of the IDSC method is provided. Therefore, we can directly compare it with our method using similar conditions. Table 3 presents the results of the comparison. Because the final feature of MTCD is a vector, the cosine distance is directly computed for matching, whereas the IDSC method applies DP to match two IDSCs. The computing of cosine distance or DP is regarded as matching, whereas other processes are considered as feature extraction.

Table 3. Comparison of time(s) usage on different situations

Situations	IDSC	Our Method
Swedish leaf dataset	14,342	53
One-to-one Matching	6.8903	0.0005

All algorithms were implemented by Matlab, except for the DP procedure of IDSC method, which is coded in C in the mex form. All default parameters in the IDSC method remained unchanged in the experiments. The first row of Table 3 shows that the time consumption of our approach is far less than that of the IDSC method. Our approach only required 53 seconds to complete one test using the Swedish leaf dataset, whereas the IDSC method needed 14,342 seconds. The time required by our method is less than 0.4% of the IDSC method. Note that the IDSC method uses DP in C language to compute for distance. Thus, it is relatively more rapid to compute for loop operations than the Matlab language. The efficiency of IDSC method can be further decreased when distance is calculated by using the Matlab language.

To demonstrate the efficiency of our method for shape matching, we compared the two methods in a one-to-one matching situation. The second row of Table 3 shows the one-to-one comparison of time utilization with the DP procedure of IDSC also implemented by Matlab instead of C. The time usage of our method was also far less than that using the IDSC method. We only needed 0.5 milliseconds to compute a match, whereas the DP procedure of the IDSC method required about 7 seconds for a match using Matlab. Therefore, the IDSC method needs more time to match two common leaves, and our method is more suitable for a real-time recognition system.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a novel shape description method called MTCD for plant leaf recognition. The proposed MTCD descriptor can better capture the local and global information of a leaf shape, and at the same time, the proposed descriptor has the following desirable properties: invariance, compactness, and multiscale representation structure. The performance of our method has been evaluated on four leaf datasets: the Swedish Leaf dataset, the Smithsonian Leaf dataset, the Flavia Leaf dataset, and the ImageCLEF2012 Leaf dataset. The experimental results demonstrate that the proposed method outperforms the state-of-the-art shape-based plant leaf recognition methods in terms of accuracy, efficiency, and storage requirement. However, we only employed the shapes of plant leaves, and our future work involves using the combined features of shape and texture for plant leaf recognition. We believe that the performance of our method can be further improved when the texture feature of a leaf is incorporated.

ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China (Project No. 61375122), (in part) by Shanghai Science and Technology Development Funds (Project No. 13dz2260200, 13511504300).

REFERENCES

- [1] N. Alajlan, I El Rube, M. S. Kamel, and G. Freeman, 'Shape retrieval using triangle-area representation and dynamic space warping', *Pattern Recognition*, **40**(7), 1911–1920, (2007).
- [2] Emad Attalla and Pepe Siy, 'Robust shape similarity retrieval based on contour segmentation polygonal multiresolution and elastic matching', *Pattern Recognition*, **38**(12), 2229–2241, (2005).
- [3] Serge Belongie, Jitendra Malik, and Jan Puzicha, 'Shape matching and object recognition using shape contexts', *IEEE Trans.PAMI*, **24**(4), 509–522, (2002).
- [4] Carlos Caballero and M. Carmen Aranda, 'Plant species identification using leaf image retrieval', in *Proceedings of the ACM International Conference on Image and Video Retrieval*, pp. 327–334. ACM, (2010).

- [5] Guillaume Cerutti, Laure Tougne, Julien Mille, Antoine Vacavant, and Didier Coquin, 'Understanding leaves in natural images - a model-based approach for tree species identification', *Computer Vision and Image Understanding*, **117**(10), 1482–1503, (2013).
- [6] Jyotismita Chaki, Ranjan Parekh, and Samar Bhattacharya, 'Plant leaf recognition using texture and shape features with neural classifiers', *Pattern Recognition Letters*, **58**(C), 61–68, (2015).
- [7] J. S. Cope, D. Corney, J. Y. Clark, P. Remagnino, and P. Wilkin, 'Plant species identification using digital morphometrics: a review', *Expert Systems with Applications*, **39**, 7562–7573, (2012).
- [8] Ji-Xiang Du, Xiao-Feng Wang, and Guo-Jun Zhang, 'Leaf shape based plant species recognition', *Applied Mathematics and Computation*, **185**(2), 883–893, (2007).
- [9] Pedro Felzenszwalb and Joshua Schwartz, 'Hierarchical matching of deformable shapes', in *In CVPR*, (2007).
- [10] Vittorio Ferrari, Frederic Jurie, and Cordelia Schmid, 'From images to shape models for object detection', *International Journal of Computer Vision*, **87**(3), 284–303, (2010).
- [11] Hervé Goëau, Pierre Bonnet, Alexis Joly, Itheri Yahiaoui, Daniel Barthélémy, Boujemaa Nozha, and Jean-François Molino, 'The ImageCLEF 2012 Plant identification Task', in *CLEF 2012*, Rome, Italie, (September 2012).
- [12] D. J. Hearn, 'Shape analysis for the automated identification of plants from images of leaves', *Taxon*, **58**(3), 934–954, (2009).
- [13] Qi Jia, Xin Fan, Yu Liu, Haojie Li, Zhongxuan Luo, and He Guo, 'Hierarchical projective invariant contexts for shape recognition', *Pattern Recognition*, **52**, 358–374, (2016).
- [14] Hanbyul Joo, Yekeun Jeong, Olivier Duchenne, Seong Young Ko, and In So Kweon, 'Graph-based robust shape matching for robotic application', in *IEEE International Conference on Robotics and Automation*, pp. 1207–1213, (2009).
- [15] Gall Juergen, Yao Angela, Razavi Nima, Van Gool Luc, and Lempitsky Victor, 'Hough forests for object detection, tracking, and action recognition', *IEEE Trans. PAMI*, **33**(11), 2188–2201, (2011).
- [16] Cem Kalyoncu and nsen Toygar, 'Geometric leaf classification', *Computer Vision and Image Understanding*, **133**, 102–109, (2015).
- [17] Hannu Kauppinen, Tapio Seppnen, and Matti Pietikinen, 'An experimental comparison of autoregressive and fourier-based descriptors in 2d shape classification.', *IEEE Trans. Pattern Anal. Mach. Intell.*, **17**(2), 201–207, (1995).
- [18] Neeraj Kumar, Peter N. Belhumeur, Arijit Biswas, David W. Jacobs, W. John Kress, Ida C. Lopez, and Joo V. B. Soares, 'Leafsnap: A computer vision system for automatic plant species identification', in *European Conference on Computer Vision*, pp. 502–516, (2012).
- [19] H. Laga, S. Kurtke, A. Srivastava, M. Golzarian, and S. J. Miklavcic, 'A riemannian elastic metric for shape-based plant leaf classification', in *Digital Image Computing Techniques and Applications (DICTA), 2012 International Conference on*, pp. 1–7, (2012).
- [20] Liang Lin, Xiaolong Wang, Wei Yang, and Jianhuang Lai, 'Learning contour-fragment-based shape model with and-or tree representation', in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 135–142. IEEE, (2012).
- [21] Haibin Ling and David W. Jacobs, 'Shape classification using the inner-distance', *IEEE Trans. Pattern Anal. Mach. Intell.*, **29**, 286–299, (2007).
- [22] Yu Liu, Qi Jia, He Guo, and Xin Fan, 'A shape matching framework using metric partition constraint', in *Image Processing (ICIP), 2013 20th IEEE International Conference on*, (Sept 2013).
- [23] José M. Martinez, 'MPEG-7 Overview (version 9)', Technical Report N5525, International Organization for Standardization - Organization for International Normalization, (2003).
- [24] Graham McNeill and Sethu Vijayakumar, 'Hierarchical procrustes matching for shape retrieval', in *Computer Vision and Pattern Recognition (CVPR), 2006 IEEE Conference on*. IEEE, (2006).
- [25] Farzin Mokhtarian, Sadegh Abbasi, and Josef Kittler, 'Efficient and robust retrieval by shape content through curvature scale space', pp. 35–42, (1996).
- [26] Peter N. Belhumeur, Daozheng Chen, Steven Feiner, David W. Jacobs, W. John Kress, Haibin Ling, Ida C. Lopez, Ravi Ramamoorthi, Sameer Sheorey, Sean White, and Ling Zhang, 'Searching the world's herbaria: A system for visual identification of plant species', in *Computer Vision - ECCV 2008, 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part IV*, pp. 116–129, (2008).
- [27] Khan Naemullah, Algarni Marei, Anthony Yezzi, and Ganesh Sundaramoorthi, 'Shape-tailored local descriptors and their application to segmentation and tracking', in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pp. 3890–3899. IEEE, (2015).
- [28] K.K. Pahalawatta, *Plant Species Biometric Using Feature Hierarchies*, Master's thesis, University of Canterbury, September 2008.
- [29] Hu Rong-Xiang, Jia Wei, Ling Haibin, Zhao Yang, and Gui Jie, 'Angular pattern and binary angular pattern for shape retrieval', *IEEE Transactions on Image Processing*, **23**(3), 1118–1127, (2014).
- [30] Hu Rongxiang, Jia Wei, Ling Haibin, and Huang Deshuang, 'Multiscale distance matrix for fast plant leaf recognition.', *Image Processing IEEE Transactions on*, **21**(11), 4667–4672, (2012).
- [31] Xin Shu and Xiao-Jun Wu, 'A novel contour descriptor for 2d shape matching and its application to image retrieval', *Image and Vision Computing*, **29**(4), 286–294, (2011).
- [32] Oskar J O Söderkvist, *Computer Vision Classification of Leaves from Swedish Trees*, Master's thesis, Linköping University, SE-581 83 Linköping, Sweden, September 2001.
- [33] Junwei Wang, Xiang Bai, Xinge You, Wenyu Liu, and Longin Jan Latecki, 'Shape matching and classification using height functions', *Pattern Recognition Letters*, **33**(2), 134–143, (2012).
- [34] Xiaofeng Wang, De-Shuang Huang, Ji-Xiang Du, Huan Xu, and Laurent Heutte, 'Classification of plant leaf images with complicated background.', *Applied Mathematics and Computation*, **205**(2), 916–926, (2008).
- [35] Jianxin Wu and James M. Rehg, 'Centrist: A visual descriptor for scene categorization', *IEEE Trans. Pattern Anal. Mach. Intell.*, **33**(8), 1489–1501, (2011).
- [36] Stephen Gang Wu, Forrest Sheng Bao, Eric You Xu, Yu xuan Wang, Yi fan Chang, and Qiao liang Xiang, 'A leaf recognition algorithm for plant classification using probabilistic neural network', in *Signal Processing and Information Technology, 2007 IEEE International Symposium on*, pp. 11–16. IEEE, (2007).
- [37] Chunjing Xu, Jianzhuang Liu, and Xiaou Tang, '2D shape matching by contour flexibility', *IEEE Trans. Pattern Anal. Mach. Intell.*, **31**(1), 180–186, (2009).
- [38] Dengsheng Zhang and Guojun Lu, 'Review of shape representation and description techniques', *Pattern Recognition*, **37**(1), 1–19, (2004).

Complexity of Control by Partitioning Veto and Maximin Elections and of Control by Adding Candidates to Plurality Elections

Cynthia Maushagen¹ and Jörg Rothe¹

Abstract. Control by partition refers to situations where an election chair seeks to influence the outcome of an election by partitioning either the candidates or the voters into two groups, thus creating two first-round subelections that determine who will take part in a final round. The model of partition-of-voters control attacks is remotely related to “gerrymandering” (maliciously resizing election districts). While the complexity of control by partition (and other control actions) has been studied thoroughly for many voting systems, there are no such results known for the important veto and maximin voting systems. We settle the complexity of control by partition for veto in a broad variety of models and for maximin with respect to destructive control by partition of candidates. We also observe that a reduction from the literature [8] showing the parameterized complexity of control by adding candidates to plurality elections, parameterized by the number of voters, is technically flawed by giving a counterexample, and we show how this reduction can be fixed.

1 INTRODUCTION

Along with manipulation [2, 9] and bribery [15, 17], electoral control [3, 23] has been the focus of much attention in computational social choice; see the book chapters by Faliszewski and Rothe [18] and Baumeister and Rothe [5] for a survey of the related results. Control scenarios model settings where an external agent, commonly referred to as the *chair*, seeks to influence the outcome of an election by such actions as adding, deleting, or partitioning either the candidates or the voters. We here focus on control by partition.

The above-mentioned chapters and the papers cited therein comprehensively describe applications of voting in artificial intelligence, multiagent systems, ranking algorithms, meta-websearch, etc., and they discuss how computational complexity can be used to provide some protection against manipulation, bribery, and control attacks. In particular, they give real-world examples of the various control types introduced by Bartholdi et al. [3] for the constructive control goal where the chair aims at making a given candidate win and by Hemaspaandra et al. [23] for destructive control where the goal is to prevent a given candidate’s victory.

The complexity of control has been studied for many voting systems, including plurality, Condorcet, and approval voting [3, 23, 7] and its variants [14, 12], Copeland [17, 7], Borda [34, 11, 28, 8], (normalized) range voting [29], and Schulze voting [32, 30] (see the book chapters [18, 5] for an overview). Comparing veto (a.k.a. antiplural-

ity) and plurality, even though these two scoring protocols are defined in similarly simple way, they behave quite differently for constructive coalitional weighted manipulation: While this problem is easy to solve in plurality for any number of candidates, it is NP-complete in veto for three or more candidates [9].² The main motivation of this paper is to find out whether veto similarly parts company from plurality regarding the complexity of control.

Perhaps a bit surprisingly, the important voting systems veto and maximin have not been investigated in terms of their control complexity by partition of either candidates or voters but only with respect to control by adding or deleting candidates or voters: Faliszewski et al. [16] studied maximin and Lin [26] studied veto for these control types in terms of their classical complexity, and their parameterized complexity has been explored by Liu and Zhu [27] for maximin and by Chen et al. [8] for veto and maximin. To the best of our knowledge, complexity results for control by partition have been missing for these two systems to date. This is all the more surprising as control by partition of voters provides a simplified model of gerrymandering (i.e., maliciously resizing election districts), a particularly natural control type known from the real world. One reason why these control scenarios have been neglected so far for veto and maximin may be that proofs for control by partition tend to be technical and challenging. We settle the complexity of control by partition for veto in a broad variety of models and for maximin with respect to destructive control by partition of candidates.

Since most of the known results on control by partition are in the original model as suggested by Bartholdi et al. [3] where the candidates or voters can be partitioned into two sets of arbitrary sizes, we will focus on this model too, in order to allow for comparability of results. However, we suggest to also study these problems for veto and maximin in the more refined models due to Erdélyi et al. [13] that restrict such partitions to sets of roughly the same size and due to Puppe and Tasnádi [33] that take geographical constraints into account when resizing election districts. Note that Bachrach et al. [1] study a related but different aspect of misrepresentation in district voting: Their “misrepresentation ratio” quantifies the deviation from proportional representation in district-based elections and they prove bounds on this ratio for various voting rules including veto.

In addition, we observe that a reduction due to Chen et al. [8, Theorem 1] showing the parameterized complexity of constructive control by adding candidates to plurality elections, where the parameter

¹ Institut für Informatik, Heinrich-Heine-Universität Düsseldorf, 40225 Düsseldorf, Germany, email: Cynthia.Maushagen@uni-duesseldorf.de, rothe@cs.uni-duesseldorf.de

² Indeed, Hemaspaandra and Hemaspaandra [20] proved a dichotomy result saying that plurality is the only nontrivial scoring protocol for which constructive coalitional weighted manipulation is easy (and Conitzer et al. [9] observed this too for the case of three candidates).

is the number of voters, is technically flawed. Specifically, we give a counterexample showing that their reduction maps a no-instance of the problem MULTI-COLORED-CLIQUE to a yes-instance of this control problem, and we show how this reduction can be fixed.

2 PRELIMINARIES

In this section, we define the needed voting systems and control problems and give some background on computational complexity.

2.1 Elections, Plurality, Veto, and Maximin Voting

An election is given by a pair (C, V) , where C is a set of candidates and V a list of the voters' preferences over the candidates (which we will simply refer to as votes). We will consider only preferences that are linear orders (strict rankings) with the left-most candidate being the most preferred one. For example, a preference $d c a b$ means that this voter prefers d to c , c to a , and a to b .

We will consider three well-known voting systems: plurality, veto (a.k.a. antiplurality), and maximin (a.k.a. Simpson). In *plurality*, every voter gives one point to her most preferred candidate, and whoever scores the most points wins. Plurality is the perhaps simplest and still very prominent positional scoring protocol, a class of important voting systems that are based on the candidates' positional scores. In *veto*, every voter vetoes her least preferred candidate, which means that this candidate gets no point while all other candidates receive one point from this voter, and whoever scores the most points wins. Veto is another prominent positional scoring protocol. By contrast, *maximin voting* is based on the pairwise comparisons between the candidates and belongs to the class of Condorcet-consistent voting rules.³ Given an election (C, V) , for any two candidates $c, d \in C$, let $N(c, d)$ denote the number of voters preferring c to d . The *maximin score* of c is $\min_{c \neq d} N(c, d)$, and whoever has the largest maximin score wins the election.

2.2 Control Problems

We consider control by partition of either candidates or voters, as defined by Bartholdi et al. [3] and—for destructive control—by Hemaspaandra et al. [23].⁴ The definitions below have been used in many papers; we refer to the book chapters by Faliszewski and Rothe [18] and Baumeister and Rothe [5] for the formal definitions of all problems studied here and for real-world examples motivating each control scenario we are interested in. In each such control scenario, starting from a given election (C, V) and a distinguished candidate $c \in C$, we form two subelections—either (C_1, V) and (C_2, V) where C is partitioned into C_1 and C_2 (i.e., $C_1 \cap C_2 = \emptyset$ and $C_1 \cup C_2 = C$), or (C, V_1) and (C, V_2) where V is partitioned into V_1 and V_2 (i.e., $V_1 \cap V_2 = \emptyset$ and $V_1 \cup V_2 = V$)—whose winners move forward to a final round if they survive the given tie-handling rule: either *ties-eliminate* (TE) that requires that only unique winners of a first-round subelection move forward, or *ties-promote* (TP) that requires that all winners of a first-round subelection move forward.

³ A (weak) Condorcet winner is a candidate who defeats (ties-or-defeats) every other candidate in pairwise comparison. Condorcet winners do not always exist, but when they do, they are unique, whereas it is possible that there are several weak Condorcet winners. A voting rule is *Condorcet-consistent* if it respects the Condorcet winner whenever one exists.

⁴ Constructive control by adding candidates, also due to Bartholdi et al. [3], will be defined in Section 6 because this control type will be considered only there.

Such a partition of either C or V is the chair's control action, and the chair's goal is either to ensure that the distinguished candidate c wins the final round (in the *constructive* case) or to prevent c 's victory (in the *destructive* case), where the final round is always held with all votes from V . In the case of candidate control, we further distinguish between *run-off partition of candidates*, where the winners of (C_1, V) and (C_2, V) surviving the tie-handling rule face each other in the final run-off, and *partition of candidates*, where the winners of (C_1, V) surviving the tie-handling rule face all candidates of C_2 in the final round.

For each such control scenario, we can define a decision problem. As an example, we formally define the decision problem associated with constructive control by partition of voters in model TE for some given voting system \mathcal{E} :

\mathcal{E} -CONSTRUCTIVE-CONTROL-BY-PARTITION-OF-VOTERS-TE

Given:	An election (C, V) and a distinguished candidate $c \in C$.
Question:	Can V be partitioned into V_1 and V_2 such that c is the unique \mathcal{E} winner of the two-round election where the winners of the two first-round subelections (C, V_1) and (C, V_2) who survive tie-handling rule TE run against each other in a final round (with the votes from V correspondingly restricted)?

The above problem (denoted by \mathcal{E} -CCPV-TE—the shorthands of the other problems to be used later on will be clear from this example) is defined in the *unique-winner model*. We will also consider the *nonunique-winner model* where the question is changed to ask whether c is a winner (possibly among several winners) of the final round, and we will always specify the winner model we are referring to.

For a control type \mathcal{C} (such as constructive control by partition of voters in model TE), an election system \mathcal{E} is said to be *immune to* \mathcal{C} if it is impossible for the chair to reach her control goal (e.g., to make the given candidate c a unique winner in the constructive case for the unique-winner model, or to ensure that c is not a winner in the destructive case for the nonunique-winner model) via exerting control of type \mathcal{C} ; otherwise, \mathcal{E} is said to be *susceptible to* \mathcal{C} . It is easy to observe that the two voting systems we study here, veto and maximin, are susceptible to every type of control (in both winner models) we have defined above; due to space limitations we omit giving detailed examples verifying these claims. If an election system \mathcal{E} is susceptible to some control type \mathcal{C} , it is common to study the computational complexity of the associated control problem: We say \mathcal{E} is *vulnerable to* \mathcal{C} if the control problem corresponding to \mathcal{C} can be solved in polynomial time, and we say \mathcal{E} is *resistant to* \mathcal{C} if \mathcal{C} is NP-hard.

2.3 Computational Complexity

We assume that the reader is familiar with the basic notions of computational complexity, such as the complexity classes P (deterministic polynomial time) and NP (nondeterministic polynomial time) and with the notions of NP-hardness and NP-completeness, based on the polynomial-time many-one reducibility. For more background, we refer to the book by Garey and Johnson [19].

In Section 6, we will also be concerned with *parameterized* complexity. In particular, we consider a result about W[1]-hardness. W[1] is a parameterized complexity class that in some sense corresponds

to the classical complexity class NP, and just as NP-hardness indicates that a problem is infeasible to solve in the sense of classical complexity theory (i.e., has no polynomial-time algorithm unless $P = NP$), $W[1]$ -hardness can be taken as strong evidence that a problem is not even fixed-parameter tractable. For more background on parameterized complexity and fixed-parameter tractability, we refer to the books by Downey and Fellows [10] and Niedermeier [31].

3 CONTROL BY PARTITION OF VOTERS IN VETO ELECTIONS IN MODEL TE

In this section, we show that it is easy to control veto elections by partition of voters in model TE. We start with the constructive case.

3.1 Veto-CCPV-TE

We show that veto is vulnerable to constructive control by partition of voters in model TE, in both winner models. Essentially, the polynomial-time algorithm used to prove Theorem 1 exploits the fact that, due to the TE model, control is impossible only if either there are two candidates and the distinguished candidate is not already a veto winner (in the unique-winner model: is not already the only veto winner) of the given election, or there are more than two candidates and some candidate other than the distinguished candidate is not vetoed by any voter. In all other cases it is easy to find a successful partition that ensures the distinguished candidate's victory.

Theorem 1 *Veto-CCPV-TE is in P in both the unique-winner and the nonunique-winner model.*

Proof. The following polynomial-time algorithm solves the problem. Given an election (C, V) with n votes in V and a candidate $c \in C$, it proceeds as follows: (1) If there are no more than two candidates, then if c already is a winner (in the unique-winner model: the only winner) of (C, V) , control is possible via the trivial partition (V, \emptyset) , so accept; otherwise, control is impossible, so reject. (2) Otherwise (i.e., if $|C| > 2$), if $score(d) = n$ for some $d \in C \setminus \{c\}$, control is impossible, so reject. (3) Otherwise (i.e., if $|C| > 2$ and $score(d) < n$ for all $d \in C \setminus \{c\}$), it is safe to accept, since control is possible via the partition (V_1, V_2) of V that puts all voters who veto c into V_1 and all other voters into V_2 .

The above algorithm runs in polynomial time and is correct. This is obvious for step 1. Further, it is impossible for c to defeat the candidate d with $score(d) = n$ in step 2 (as d scores the maximum number of points in each first-round subelection, no matter how V is partitioned, which makes it impossible for c to win alone in any subelection). And in step 3, no candidate from V_1 can move to the final round, because either V_1 is empty (in case no one vetoes c) or each of the at least two candidates other than c wins subelection (C, V_1) with the same score and, therefore, will be eliminated in model TE. On the other hand, each candidate $d \neq c$ is vetoed by at least one voter ending up in V_2 , whereas c is not vetoed by any voter in V_2 and thus wins subelection (C, V_2) and the final run-off. This argument applies to both the unique-winner and the nonunique-winner model. \square

3.2 Veto-DCPV-TE

A similar algorithm works in the destructive case. Note that Theorem 2 follows immediately from Theorem 1 for the unique-winner

model,⁵ but not for the nonunique-winner model. Therefore, we present a proof (which in fact works for both winner models).

Theorem 2 *Veto-DCPV-TE is in P in both the unique-winner and the nonunique-winner model.*

Proof. Given an election (C, V) and a distinguished candidate c , our algorithm works as follows: (1) If $|C| = 1$, control is impossible, so reject. (2) If $|C| = 2$, determine the set of veto winners. If c wins alone, control is impossible, so reject. Otherwise, control is possible via the trivial partition (V, \emptyset) , so accept. (3) If $|C| > 2$, it is safe to outright accept, since control is always possible: Fix some candidate $d \neq c$ and partition V into (V_1, V_2) such that V_1 contains all voters vetoing d and V_2 contains all remaining voters.

The above algorithm obviously runs in polynomial time and its correctness is straightforward for steps 1 and 2, while it follows for step 3 from the observation that if either c or d is vetoed by everyone then (V_1, V_2) will be trivial (either (\emptyset, V) or (V, \emptyset)) and will thus prevent c from winning, and if neither c nor d is vetoed by everyone then there is a candidate e , $c \neq e \neq d$, who ties for winner with c in (C, V_1) , while d ties-or-defeats c in (C, V_2) ; in either case, c cannot move forward to the final round due to model TE. \square

4 CONTROL BY PARTITION OF CANDIDATES IN VETO ELECTIONS

We now turn to control by partition of candidates in veto elections, considering both constructive and destructive control, both tie-handling models, TE and TP, both the unique-winner and the nonunique-winner model, and the partition problems both with and without run-off.

4.1 Veto-CCRPC-TE, Veto-CCRPC-TP, Veto-CCPC-TE, and Veto-CCPC-TP

We start by showing that veto is resistant to constructive control by run-off partition of candidates.

Theorem 3 *Veto-CCRPC-TE is NP-complete in both the unique-winner and the nonunique-winner model, and Veto-CCRPC-TP is NP-complete in the unique-winner model.*

Proof. Membership of Veto-CCRPC-TE in NP is obvious. To show that it is NP-hard, we reduce from ONE-IN-THREE-POSITIVE-3SAT, an adaption from the well-known NP-complete problem ONE-IN-THREE-3SAT where the clauses of the given boolean formula do not contain any negated variables [19, p. 259]:

ONE-IN-THREE-POSITIVE-3SAT

- Given:** A set X of boolean variables, a set S of clauses over X , each containing exactly three unnegated literals.
- Question:** Does there exist a truth assignment to the variables in X such that exactly one literal is set to true for each clause in S ?
-

⁵ As noted by Hemaspaandra et al. [23, Footnote 5 on p. 257], for voting systems that always have at least one winner (such as veto), any destructive control problem in the unique-winner model disjunctively truth-table reduces to the corresponding constructive control problem in the nonunique-winner model.

Let (X, S) be an instance of ONE-IN-THREE-POSITIVE-3SAT with $X = \{x_1, \dots, x_m\}$ and $S = \{S_1, \dots, S_n\}$. Construct an election (C, V) with distinguished candidate $c \in C$ by defining $C = X \cup \{c, w\}$, where the elements of X from now on will also be viewed as candidates, and the list V of votes as follows:

# votes	preference
$2n^2 + 1$	$w c \cdots x_i$ for each $i \in \{1, \dots, m\}$
$n - 1$	$w \cdots c$
1	$c \cdots w S_j \setminus \{x_i\}$ for each $j \in \{1, \dots, n\}$ and $x_i \in S_j$
$2n$	$w \cdots c S_j$ for each $j \in \{1, \dots, n\}$

There are $m + 2$ candidates and $(2n^2 + 1)m + 2n^2 + 4n - 1$ voters in the election. If a set of candidates occurs in such a vote, we tacitly assume a fixed ordering of its candidates in this preference. The dots in a vote represent all remaining candidates (in an arbitrary, fixed order). In particular, there are $3n$ votes of the form $c \cdots w S_j \setminus \{x_i\}$. If, say, clause S_1 contains the literals x_2, x_5 , and x_7 , then the corresponding three votes are

$$c \cdots w x_2 x_5, \quad c \cdots w x_2 x_7, \quad c \cdots w x_5 x_7.$$

Candidate w alone wins in election (C, V) , since the candidates score the following points:⁶

$$\begin{aligned} \text{score}(c) &= (2n^2 + 1)m + 3n + 2n^2, \\ \text{score}(w) &= (2n^2 + 1)m + 3n + n - 1 + 2n^2, \text{ and} \\ \text{score}(x_i) &\leq (2n^2 + 1)(m - 1) + n - 1 + 3n + 2n^2. \end{aligned}$$

Obviously, the reduction can be computed in polynomial time. It remains to show that (X, S) is a yes-instance of ONE-IN-THREE-POSITIVE-3SAT if and only if (C, V, c) is a yes-instance of Veto-CCRPC-TE (in both winner models).

(\Rightarrow) If (X, S) is a yes-instance of ONE-IN-THREE-POSITIVE-3SAT, then there is a subset $U = \{u_1, \dots, u_k\}$ of X (renaming its elements for convenience) such that $|U \cap S_j| = 1$ for each $j \in \{1, \dots, n\}$. We claim that partitioning C into $C_1 = U \cup \{c, w\}$ and $C_2 = C \setminus C_1$ ensures that c is the only veto winner (and thus, *a fortiori*, c is a veto winner). To see this, note that the candidates in subelection (C_1, V) have the following scores:

$$\begin{aligned} \text{score}(c) &= (2n^2 + 1)m + 3n + 2n^2, \\ \text{score}(w) &= (2n^2 + 1)m + n - 1 + 2n + 2n^2, \text{ and} \\ \text{score}(u_i) &\leq (2n^2 + 1)(m - 1) + n - 1 + 3n + 2n^2. \end{aligned}$$

For c to win (C_1, V) alone, we have to show that $\text{score}(c) > \text{score}(w)$ and $\text{score}(c) > \text{score}(u_i)$ for all $u_i \in U$: First, $\text{score}(c) > \text{score}(w)$ is equivalent to $(2n^2 + 1)m + 3n + 2n^2 > (2n^2 + 1)m + n - 1 + 2n + 2n^2$, which in turn is equivalent to $3n > 3n - 1$; second, $\text{score}(c) > \text{score}(u_i)$ is equivalent to $(2n^2 + 1)m + 3n + 2n^2 > (2n^2 + 1)(m - 1) + n - 1 + 3n + 2n^2$, which in turn is equivalent to $2n^2 + 1 > n - 1$.

Being the only veto winner of subelection (C_1, V) , c will move forward to the final run-off. If more than one candidate wins subelection (C_2, V) (thus TE blocking them all from moving to the final run-off), c 's overall victory is ensured. On the other hand, if some candidate $x_i \in C_2$ is the only veto winner of (C_2, V) , c will face x_i in the run-off. However, since

$$\text{score}(c) \geq (2n^2 + 1)m + 3n > n - 1 + 2n^2 \geq \text{score}(x_i)$$

in the run-off $(\{c, x_i\}, V)$, c wins the run-off and is the only overall veto winner. Thus (C, V, c) is a yes-instance of Veto-CCRPC-TE in the unique-winner model.

(\Leftarrow) Conversely, let (X, S) be a no-instance of ONE-IN-THREE-POSITIVE-3SAT. Then, for each partition of X into X_1 and X_2 , let k_i be the number of clauses containing i literals from X_1 . We have $1 \leq k_0 + k_1 + k_2 + k_3 \leq n$, since we started from a no-instance of ONE-IN-THREE-POSITIVE-3SAT. We will show that for each possible combination of the k_i (corresponding to each possible partition of X), candidate c cannot end up being a veto winner (*a fortiori*, c cannot be the only veto winner). Note that a partition of X induces a partition of $C = X \cup \{c, w\}$ into C_1 and $C_2 = C \setminus C_1$ (assuming, without loss of generality, that $c \in C_1$). It is enough to distinguish the three cases below, and in each case, we will show that c is not a veto winner.

Case 1: $C_1 = \{c, w\}$. Then $\text{score}(c) = 3n$ and $\text{score}(w) = (2n^2 + 1)m + n - 1 + 2n^2 \geq 4n^2 + n$, so w is the only veto winner of this subelection, and since c does not take part in the final run-off, c will not be an overall winner.

Case 2: C_1 contains c but not w . It is enough to show that w is the only winner of the other subelection, (C_2, V) , since if c wins (C_1, V) , then either c is not promoted to the final round due to TE (if there are other winners) or c loses the final round as we have seen in Case 1. In subelection (C_2, V) , for each $x_i \in C_2$, we have

$$\begin{aligned} \text{score}(w) &\geq (2n^2 + 1)m + n - 1 + 2n^2 \\ &> (2n^2 + 1)(m - 1) + n - 1 + 3n + 2n^2 \\ &\geq \text{score}(x_i), \end{aligned}$$

where the ‘‘greater than’’ follows from $2n^2 + 1 > 3n$, which is true for all $n > 1$. (For $n = 1$, however, we would have started from a yes-instance of ONE-IN-THREE-POSITIVE-3SAT, which contradicts our assumption.) Thus w is the only veto winner of (C_2, V) , which precludes c 's overall victory in this case.

Case 3: C_1 contains c, w , and some elements of X . Distinguish the following three subcases.

Case 3.1: $k_0 \geq 2$. In this case, we have

$$\begin{aligned} \text{score}(c) &\leq (2n^2 + 1)m + 3n + (n - k_0)2n \text{ and} \\ \text{score}(w) &\geq (2n^2 + 1)m + n - 1 + 2n^2. \end{aligned}$$

Regardless of the points the elements of X in C_1 score, it suffices to show that $\text{score}(c) \leq \text{score}(w)$. This, however, holds since (for $k_0 \geq 2$) the inequality $2n + 1 \leq 2k_0n$ implies

$$(2n^2 + 1)m + 3n + (n - k_0)2n \leq (2n^2 + 1)m + n - 1 + 2n^2.$$

Case 3.2: $k_0 = 1$. In this case, we have

$$\begin{aligned} \text{score}(c) &\leq (2n^2 + 1)m + 3n + (n - k_0)2n \text{ and} \\ \text{score}(w) &\geq (2n^2 + 1)m + n - 1 + 2(n - 1) + 2n^2. \end{aligned}$$

Now, the inequality $3 \leq 2n$ (which is true for $n > 1$; the case $n = 1$ can again be excluded) implies $\text{score}(c) \leq \text{score}(w)$ also in this case.

Case 3.3: $k_0 = 0$. Since we have a no-instance, at least one clause must contain at least two literals from X_1 , so

$$\begin{aligned} \text{score}(c) &= (2n^2 + 1)m + 3n + 2n^2 \text{ and} \\ \text{score}(w) &\geq (2n^2 + 1)m + n - 1 + 2n + 1 + 2n^2. \end{aligned}$$

The term $2n + 1$ in $\text{score}(w)$ is due to the third row in V . Every clause S_j contains at least one literal corresponding to a candidate

⁶ Here and in the following, we omit a detailed argumentation of why certain candidates score a certain number of points in some election, due to space limitations and since these scores can be determined straightforwardly.

x_i in C_1 , so w gains at least two points per clause. Since at least one clause contains at least two literals corresponding to candidates in C_1 , w receives all three possible points for this clause, which explains the important additional point. Again, it is enough to show $score(c) \leq score(w)$. But this follows since $3n + 2n^2 \leq 2n^2 + 3n$ implies

$$(2n^2 + 1)m + 3n + 2n^2 \leq (2n^2 + 1)m + n - 1 + 2n + 1 + 2n^2.$$

By model TE, c cannot move forward to the final round and thus cannot win the overall election. As we have shown that c is not a veto winner in any partition of the candidates, (C, V, c) is a no-instance of Veto-CCRPC-TE.

The proof (omitted here) that Veto-CCRPC-TP is NP-complete in the unique-winner model works by suitably adapting the above proof. \square

A minor tweak in the construction of the previous proof (namely, by having n instead of $n - 1$ votes of the form $w \cdots c$, all else being equal) works for showing NP-hardness of both Veto-CCPC-TE and Veto-CCPC-TP in the nonunique-winner model. Other minor changes work in the unique-winner case. The proof of Theorem 4 is omitted due to space limitations.

Theorem 4 *Veto-CCPC-TP and Veto-CCPC-TE are NP-complete in both the nonunique-winner and the unique-winner model.*

4.2 Veto-DCRPC-TE and Veto-DCPC-TE

Now we turn to the destructive variant of the previous problem, but now in both winner models. We again show resistance via a reduction from ONE-IN-THREE-POSITIVE-3SAT.

Theorem 5 *Veto-DCRPC-TE is NP-complete in both the unique-winner and the nonunique-winner model.*

Proof. Membership of both problems in NP is again obvious. For showing NP-hardness, let (X, S) be an instance of ONE-IN-THREE-POSITIVE-3SAT with $X = \{x_1, \dots, x_m\}$ and $S = \{S_1, \dots, S_n\}$. Construct an election (C, V) with $C = X \cup \{c, w\}$, $c \in C$ being the distinguished candidate, and the following list of votes:

# votes	preference
$3n + 1$	$c w \cdots x_i$ for each $i \in \{1, \dots, m\}$
$2n + 2$	$c \cdots w S_j$ for each $j \in \{1, \dots, n\}$
n	$c \cdots w$
1	$w \cdots c S_j \setminus \{x_i\}$ for each $j \in \{1, \dots, n\}$ and $x_i \in S_j$

The election contains $m + 2$ candidates and $(3n + 1)m + (2n + 6)n$ voters. The reduction can be computed in polynomial time. It is easy to see that c is the only veto winner of election (C, V) :

$$\begin{aligned} score(c) &= (3n + 1)m + (2n + 2)n + n + 3n, \\ score(w) &= (3n + 1)m + (2n + 2)n + 3n, \text{ and} \\ score(x_i) &\leq (3n + 1)(m - 1) + (2n + 2)n + n + 3n. \end{aligned}$$

We claim that (X, S) is a yes-instance of ONE-IN-THREE-POSITIVE-3SAT if and only if (C, V, c) is a yes-instance of Veto-DCRPC-TE (in both winner models).

(\Rightarrow) If (X, S) is a yes-instance of ONE-IN-THREE-POSITIVE-3SAT, then there is a subset U of X such that $|U \cap S_j| = 1$ for each $j \in \{1, \dots, n\}$. Partitioning C into $C_1 = U \cup \{c, w\}$ and $C_2 = C \setminus C_1$

ensures that c is not a veto winner (*a fortiori*, c is not the only veto winner), since c and w have the same score in subelection (C_1, V) :

$$\begin{aligned} score(c) &= (3n + 1)m + (2n + 2)n + n + 2n \text{ and} \\ score(w) &= (3n + 1)m + (2n + 2)n + 3n, \end{aligned}$$

so, by model TE, c cannot move forward to the final round.

(\Leftarrow) Conversely, let (X, S) be a no-instance of ONE-IN-THREE-POSITIVE-3SAT. As in the proof of Theorem 3, we consider all possible partitions of C into C_1 and C_2 (again assuming, without loss of generality, that $c \in C_1$) and show that c always is the only veto winner (*a fortiori*, c is a veto winner) overall.

Case 1: $C_1 = \{c, w\}$. Then $score(c) = (3n + 1)m + (2n + 2)n + n$ and $score(w) = 3n$, so c moves forward to the final round. If the other subelection, (C_2, V) , has more than one winner, TE blocks them all, so c wins. If (C_2, V) has a unique winner, say x_i , we have $score(c) = (3n + 1)m + (2n + 2)n + n$ and $score(x_i) \leq 3n$ in the final round, $(\{c, x_i\}, V)$, so c wins.

Case 2: C_1 contains c and some elements of X but not w .

$$\begin{aligned} score(c) &= (3n + 1)m + (2n + 2)n + n + 3n \text{ and} \\ score(x_i) &\leq (3n + 1)(m - 1) + (2n + 2)n + n + 3n \end{aligned}$$

then imply that c scores at least $3n + 1$ points more than any x_i and moves forward to the final round. If (C_2, V) has more than one winner, c outright wins; if either w or some x_i wins in (C_2, V) , c wins the run-off as shown in Case 1.

Case 3: C_1 contains c , w , and some elements of X . Rename the elements of $U = C_1 \cap X$ by $U = \{u_1, \dots, u_\ell\}$. Let k be the number of clauses S_j such that $|S_j \cap U| = 0$.

Case 3.1: $k > 0$. Then the scores in (C_1, V) are:

$$\begin{aligned} score(c) &\geq (3n + 1)m + (2n + 2)n + n + 2(n - k), \\ score(w) &= (3n + 1)m + (2n + 2)(n - k) + 3n, \text{ and} \\ score(u_i) &\leq (3n + 1)(m - 1) + (2n + 2)n + n + 3n. \end{aligned}$$

For c to win subelection (C_1, V) alone, we need to show that $score(c) > score(w)$ and $score(c) > score(u_i)$ for each $u_i \in U$. Simplifying the scores of c and w , we get $2n^2 + 5n - 2k > 2n^2 + 5n - 2nk - 2k$, which is equivalent to $2nk > 0$, which is true because $k > 0$ and $n > 0$. Obviously, c also wins out over each $u_i \in U$, since simplifying their scores yields $2n + 1 > 2k$, which is true. In the run-off, c is either alone or faces some x_i (if x_i is the only veto winner of subelection (C_2, V)). By the argument just given, c triumphs over x_i and is the only overall veto winner.

Case 3.2: $k = 0$. Since (X, S) is a no-instance, there is at least one clause S_j with $|S_j \cap U| \geq 2$ in this case. This implies the following scores in (C_1, V) :

$$\begin{aligned} score(c) &\geq (3n + 1)m + (2n + 2)n + n + 2n + 1, \\ score(w) &= (3n + 1)m + (2n + 2)n + 3n, \text{ and} \\ score(u_i) &\leq (3n + 1)(m - 1) + (2n + 2)n + n + 3n. \end{aligned}$$

Thus c is the only veto winner of subelection (C_1, V) and (by the above arguments) wins also the final run-off alone. Hence, (C, V, c) is a no-instance of Veto-DCRPC-TE. \square

In both winner models, the problems DCRPC-TE and DCPC-TE are known to be identical for all voting systems [22, Thm. 8 on p. 386]; the proofs can be found in the related technical report by Hemaspaandra et al. [21]. Thus we have from Theorem 5:

Corollary 6 *Veto-DCPC-TE is NP-complete in both the unique-winner and the nonunique-winner model.*

4.3 Veto-DCRPC-TP and Veto-DCPC-TP

We next turn to the ties-promote model, TP. By slightly modifying the proof of Theorem 5, we will show resistance in both cases for the nonunique-winner model.

Theorem 7 *Veto-DCRPC-TP and Veto-DCPC-TP are NP-complete in the nonunique-winner model.*

Proof. Starting with Veto-DCRPC-TP, we only describe the differences with the construction given in the proof of Theorem 5. The only required change is that the votes of the form $c \cdots w$ (see the third row) occur $n - 1$ instead of n times. The arguments showing the correctness of the construction then need to be adapted to model TP; the details are omitted here due to space limitations. Regarding Veto-DCPC-TP, note that DCRPC-TP and DCPC-TP are known to be identical problems in the nonunique-winner model for all voting systems [22, Thm. 8 on p. 386]. \square

5 DESTRUCTIVE CONTROL BY PARTITION OF CANDIDATES IN MAXIMIN ELECTIONS

Finally, we turn to destructive control by partition of candidates in maximin elections. We start with the ties-eliminate model.

5.1 Maximin-DCRPC-TE and Maximin-DCPC-TE

While veto is vulnerable to both constructive and destructive control by partition of voters but not to the types of candidate control we have studied, maximin voting turns out to be vulnerable to destructive control by partition of candidates.

Theorem 8 *In both the unique-winner and the nonunique-winner model, Maximin-DCRPC-TE is in P.*

Proof. Given an election (C, V) with distinguished candidate $c \in C$ as input, our polynomial-time algorithm for Maximin-DCRPC-TE simply works as follows: If c is the Condorcet winner of (C, V) , control is impossible, so reject; otherwise, accept.

To see that the algorithm is correct, note that control is always possible if c is not a Condorcet winner of (C, V) : In the unique-winner model, we can argue that there is at least one candidate, say $d \in C$, such that $N(d, c) \geq N(c, d)$. Now, partitioning C into $C_1 = \{d\}$ and $C_2 = C \setminus C_1$ ensures that d moves forward to the final run-off, and even if c emerges as the only maximin winner of the other subelection, (C_2, V) , and faces d in the run-off, c will not be the only maximin winner of the overall election. The proof in the nonunique-winner model is similar: If some candidate d defeats c , then again partition C into $C_1 = \{d\}$ and $C_2 = C \setminus C_1$, so d defeats c in the run-off. Otherwise, there must be a candidate e that ties-or-defeats c , so partitioning C into $C_1 = \{c, e\}$ and $C_2 = C \setminus C_1$ makes sure that due to the TE rule, c does not move forward to the run-off and does not win. On the other hand, if c is the Condorcet winner of (C, V) , in both winner models, no partition of C can prevent c from being the only maximin winner of the overall election. \square

Again, we can apply the known result that DCRPC-TE equals DCPC-TE for all voting systems [22, Thm. 8 on p. 386].

Corollary 9 *In both the unique-winner and the nonunique-winner model, Maximin-DCPC-TE is in P.*

5.2 Maximin-DCRPC-TP and Maximin-DCPC-TP

In the ties-promote model, TP, the algorithm used to prove Theorem 8 works as well, though the proof of correctness needs to be slightly adjusted. Note that, unlike in TE, DCRPC-TP and DCPC-TP are not known to coincide in the *unique-winner* model, though DCRPC-TP equals DCPC-TP in the *nonunique-winner* model [22, Thm. 8 on p. 386], as noted in the proof of Theorem 7.

Theorem 10 *In the unique-winner model, both Maximin-DCRPC-TP and Maximin-DCPC-TP are in P.*

Proof. Given an election (C, V) with distinguished candidate $c \in C$ as input, the simple polynomial-time algorithm for Maximin-DCRPC-TE from the proof of Theorem 8 also works here: If c is the Condorcet winner of (C, V) , reject; otherwise, accept.

The proof of correctness is adjusted as follows. If c is the Condorcet winner of (C, V) , our destructive goal can again never be reached: No partition of C can prevent c from being the only maximin winner of the overall election. On the other hand, if c is not a Condorcet winner of (C, V) , we distinguish two cases: First, if c is a weak Condorcet winner of (C, V) , there exists a candidate, say d , such that $N(d, c) = N(c, d)$; partitioning C into $C_1 = \{d\}$ and $C_2 = C \setminus C_1$ ensures that c will not be the only maximin winner of the overall election. Second, if c is not even a weak Condorcet winner of (C, V) , there exists a candidate, say d , such that $N(d, c) > N(c, d)$; partitioning C into $C_1 = \{c, d\}$ and $C_2 = C \setminus C_1$ will ensure that c does not even win subelection (C_1, V) . Obviously, this argument works both with and without run-off, i.e., both for Maximin-DCRPC-TP and Maximin-DCPC-TP. \square

6 CONSTRUCTIVE CONTROL BY ADDING CANDIDATES IN PLURALITY ELECTIONS

In this section, we consider only a single control scenario (constructive control by adding candidates) for the simplest natural voting system, plurality. That plurality is resistant to this control type in the sense of Plurality-CCAC being NP-hard has already been known since the first paper on electoral control, due to Bartholdi et al. [3]. Recently, Chen et al. [8] considered the parameterized complexity of control problems for natural voting systems when there are only few voters. In particular, they proved that the parameterized variant of Plurality-CCAC, parameterized by the number of voters, is W[1]-hard by reducing from the W[1]-hard problem MULTI-COLORED-CLIQUE, parameterized by the clique order [8, Theorem 1].

However, while the proof sketch of this result does provide a very clever reduction, it is technically flawed. In this section, we first briefly present their reduction from the proof sketch of [8, Theorem 1], then give a counterexample showing that it is not correct, and finally fix this flaw by suitably adapting their reduction in order to make it correct. The W[1]-hard parameterized problem Chen et al. [8] reduce from is formally defined as follows.

MULTI-COLORED-CLIQUE	
Given:	An undirected graph $G = (V(G), E(G))$, where $V(G)$ is partitioned into h sets $V_1(G), \dots, V_h(G)$ such that each $V_i(G) = \{v_1^{(i)}, \dots, v_{n'}^{(i)}\}$ consists of exactly n' vertices with color i and G has only edges connecting vertices of distinct colors.
Parameter:	the number h of colors.
Question:	Does there exist a size- h clique containing some vertex for each color?

Given a voting rule \mathcal{E} , the parameterized problem \mathcal{E} -CCAC (parameterized by the number of voters) is defined as follows.

\mathcal{E} -CCAC	
Given:	A set C of registered candidates, a set A of as yet unregistered candidates, $C \cap A = \emptyset$, a list of preferences V over $C \cup A$, a nonnegative integer k , and a distinguished candidate $p \in C$.
Parameter:	the number of votes in V .
Question:	Does there exist a subset $A' \subseteq A$ such that $\ A'\ \leq k$ and p is an \mathcal{E} winner of the election $(C \cup A', V')$ with V' being V restricted to $C \cup A'$?

We now describe the reduction from the proof sketch of Theorem 1 due to Chen et al. [8]. Let $G = (V(G), E(G))$ be a given undirected graph, where $V(G)$ is partitioned into h sets $V_1(G), \dots, V_h(G)$ such that each $V_i(G) = \{v_1^{(i)}, \dots, v_{n'}^{(i)}\}$ consists of exactly n' vertices with color i and G has only edges connecting vertices of distinct colors. Construct the following instance (C, A, V, k, p) of Plurality-CCAC:

- The set of registered candidates is $C = \{p, d\}$, where p is the distinguished candidate the chair wants to see win.
- The set A of unregistered candidates contains
 - a *vertex candidate* v for each $v \in V(G)$ and
 - two *edge candidates* (u, v) and (v, u) for each edge $\{u, v\} \in E(G)$.
- To specify the list V of votes, we adopt the following notation from [8]. Let $E(i, j)$ be the set of all edge candidates (u, v) with $u \in V_i(G)$ being colored i and $v \in V_j(G)$ being colored j . For each vertex $v_z^{(i)} \in V_i(G)$, let $L(v_z^{(i)}, j)$ be the set of all edge candidates $(v_z^{(i)}, v)$ with $v \in V_j(G)$ and $(v_z^{(i)}, v) \in E(G)$. For each $i, j, 1 \leq i \neq j \leq n$, define the following two linear orders:

$$R(i, j) : v_1^{(i)} L(v_1^{(i)}, j) \dots v_{n'}^{(i)} L(v_{n'}^{(i)}, j)$$

$$R'(i, j) : L(v_1^{(i)}, j) v_1^{(i)} \dots L(v_{n'}^{(i)}, j) v_{n'}^{(i)}.$$

Now we are ready to define the following three types of votes:

1. For each i , there is one vote of the form $v_1^{(i)} \dots v_{n'}^{(i)} d \dots$.
 2. For each pair of colors $i, j, 1 \leq i \neq j \leq n$, there are (a) $h-1$ votes of the form $E(i, j) d \dots$, (b) one vote of the form $R(i, j) d \dots$, and (c) one vote of the form $R'(i, j) d \dots$.
 3. There are h votes of the form $d \dots$ and h votes of the form $p \dots$.
- At most $k = h + 2\binom{h}{2}$ candidates can be added.

Chen et al. [8] then argue that p can become a plurality winner by adding at most k candidates from A if and only if graph G has a size- h multi-colored clique (i.e., a clique containing a vertex for each color). However, we now present a counterexample for this claim.

Example 11 Figure 1 shows a graph G corresponding to a no-instance of MULTI-COLORED-CLIQUE. In particular, the vertex set $V(G)$ is partitioned into three sets containing two vertices each:

$$V_1(G) = \{\textcircled{1}, \textcircled{6}\}, \quad V_2(G) = \{\textcircled{2}, \textcircled{5}\}, \quad V_3(G) = \{\textcircled{3}, \textcircled{4}\}$$

but, obviously, G has no clique of size three.

However, we now show that the above construction maps this no-instance of MULTI-COLORED-CLIQUE to a yes-instance of

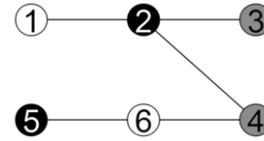


Figure 1: Counterexample for the reduction for [8, Theorem 1]

Plurality-CCAC. Indeed, from G we obtain the set $C = \{p, d\}$ of registered candidates, where p is the distinguished candidate the chair wants to see win. The set of unregistered candidates is

$$A = \left\{ \textcircled{1}, \textcircled{2}, \textcircled{3}, \textcircled{4}, \textcircled{5}, \textcircled{6}, \textcircled{1}, \textcircled{2}, \textcircled{3}, \textcircled{2}, \textcircled{4}, \textcircled{2}, \textcircled{6}, \textcircled{4}, \textcircled{6}, \textcircled{6}, \textcircled{5} \right\}$$

with six vertex candidates and ten edge candidates. Figure 2 gives the list V of votes over $C \cup A$, where the number before a vote indicates how many votes of this type there are according to the above construction. Finally, since $h = 3$, we are allowed to add $k = 3 + 2\binom{3}{2} = 9$ candidates. Since G has no clique of size three, it should be impossi-

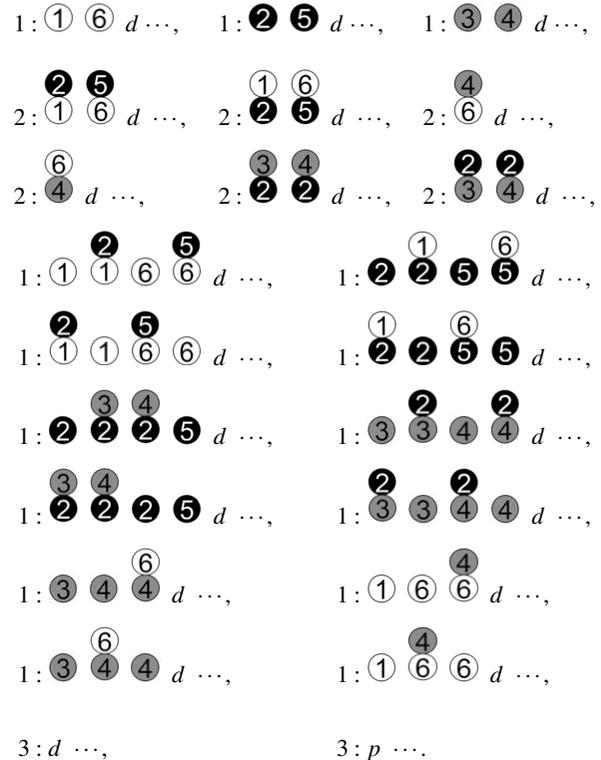


Figure 2: Constructing a yes-instance of Plurality-CCAC from a no-instance of MULTI-COLORED-CLIQUE according to the proof sketch of [8, Theorem 1]

ble to make p a plurality winner by adding at most $k = 9$ candidates.

However, adding the candidates $\textcircled{6}, \textcircled{2}, \textcircled{4}, \textcircled{6}, \textcircled{2}, \textcircled{6}, \textcircled{2}, \textcircled{4}, \textcircled{4}$ to the election implies that each candidate scores exactly three points. In particular, p has become a plurality winner, which shows that a no-instance of MULTI-COLORED-CLIQUE has been mapped to a yes-instance of Plurality-CCAC by the reduction presented in the proof sketch of [8, Theorem 1].

Let us discuss the observation made in Example 11 in general and let us see how the reduction can be adapted so as to work correctly.

First note that p can never score more than h points, no matter how many candidates are added to the election. Thus, for p to be a winner, no other candidate must score more than h points. Candidate d will score at least h points, no matter if any (and how many) candidates are added. To prevent d from scoring more than h points, any size- $(h+2\binom{h}{2})$ set $A' \subseteq A$ of added unregistered candidates must contain exactly one vertex candidate for each color and exactly one edge candidate of each (ordered) pair of colors. In their proof sketch Chen et al. [8, p. 2049] further say that “if A' contains two vertex candidates u, v but not the edge candidate (u, v) then, due to the orders $R(i, j) \succ d \succ \dots$ and $R'(i, j) \succ d \succ \dots$, either u or an edge candidate (u', v') (where $u' \in V_i(G)$, $v' \in V_j(G)$, but $(u', v') \neq (u, v)$) receives too many points, causing p not to win.”⁷ That is, they claim that adding two vertex candidates, $u \in V_i(G)$ and $v \in V_j(G)$, enforces addition of the edge candidate (u, v) because this would be required to ensure that the points from the two votes $R(i, j) d \dots$ and $R'(i, j) d \dots$ are scored by *different* candidates. This, however, is not always true. Indeed, to ensure that different candidates score points from these two votes, it is enough to add with u and v an edge candidate (u, v_j) such that v_j and v have the same color and $(u, v_j) \in E(G)$. Therefore, it in fact is possible to add two edge candidates (u, v) and (v', u') with $u, u' \in V_i(G)$, $v, v' \in V_j(G)$, $i \neq j$, and $(u, v) \neq (u', v')$.

In Example 11, the vertex candidates $\textcircled{2}$ and $\textcircled{4}$ and also the edge candidate $\textcircled{3}$ have been added. The problem is that one is not

forced to also add the edge candidate $\textcircled{3}$. The above argument is only correct if one assumes that the edge candidates (u, v) and (v, u) must always be added together. To enforce this, one can adapt the votes $E(i, j) d \dots$ by requiring each edge candidate (u, v) is followed by the corresponding edge candidate (v, u) . For instance, in

Example 11 that means that the two votes $\textcircled{2} \textcircled{2} d \dots$ are both

changed to $\textcircled{2} \textcircled{3} \textcircled{2} \textcircled{4} d \dots$. Now, if one adds two *unmatching* edge candidates (i.e., (u, v) and (v', u') with $(u, v) \neq (u', v')$, $u, u' \in V_i$, and $v, v' \in V_j$) then, without loss of generality, edge candidate (u, v) receives the points in the now modified votes $E(i, j) d \dots = \dots (u, v) (v, u) \dots (u', v') (v', u') \dots d \dots$ and $E(j, i) d \dots = \dots (v, u) (u, v) \dots (v', u') (u', v') \dots d \dots$. However, if one adds the *matching* edge candidates, say (u, v) and (v, u) , then each of them receives a point only from one of these modified votes. This enforces that only *matching* edge candidates can be added (otherwise, p would not win). The votes $R(i, j) : v_1^{(i)} L(v_1^{(i)}, j) \dots v_n^{(i)} L(v_n^{(i)}, j)$ and $R'(i, j) : L(v_1^{(i)}, j) v_1^{(i)} \dots L(v_n^{(i)}, j) v_n^{(i)}$ then imply that with an edge candidate (u, v) also the candidate u must be added. If some other vertex candidate $u' \neq u$ were added, the above two votes restricted to candidates u' and (u, v) would either both be u' (u, v) or both be $(u, v) u'$, which would give one of these two candidates too many points.

Obviously, matching vertex and edge candidates can be added only if there is a size- h multi-colored clique in the given graph G . If there is no such clique, at least one unmatching candidate has to be added.

7 CONCLUSIONS AND OPEN QUESTIONS

We have studied the complexity of control by partition of either voters or candidates for veto elections and of destructive control by partition of candidates for maximin. Recall that our main goal was to

find out whether veto parts company from plurality regarding the complexity of control by partition. We have seen that, in stark contrast with constructive coalitional weighted manipulation where veto and plurality behave quite differently, the results obtained for control by partition in veto are exactly the same as those known for control by partition in plurality [3, 23]: Control by partition of candidates is hard, whereas control by partition of voters is easy. For veto, the complexity is still open for CCPV-TP and DCPV-TP and in some cases for one of the two winner models. Table 1 gives a detailed overview by comparing our results for veto with the known results for plurality due to Bartholdi et al. [3] and Hemaspaandra et al. [23] (which all were shown only in the unique-winner model), in particular indicating the open questions by question marks. In this table, V stands for vulnerability (i.e., the corresponding control problem is in P) and R for resistance (i.e., the corresponding control problem is NP-hard). By R^* and $=^*$ (respectively, by R^\dagger) we indicate that this result has been shown only in the nonunique-winner (respectively, in the unique-winner) model (and, for our R^* and R^\dagger entries, the question of whether these resistance results hold also in the other winner model is left open), while all other results hold in both the nonunique-winner and the unique-winner model. For maximin, we only obtained (easy) polynomial-time algorithms for destructive candidate control cases—which is similar to the results known for these control types in Copeland elections [17].

Control problem	Veto	Plurality
CCPV-TE	V (Thm. 1)	V
DCPV-TE	V (Thm. 2)	V
CCPV-TP	?	R
DCPV-TP	?	R
CCRPC-TE	R (Thm. 3)	R
DCRPC-TE = DCPC-TE	R (Thm. 5 and Cor. 6)	R
CCPC-TE	R (Thm. 4)	R
CCRPC-TP	R^\dagger	R
DCRPC-TP =* DCPC-TP	R^* (Thm. 7)	R
CCPC-TP	R (Thm. 4)	R

Table 1: Complexity results for control by partition for veto in comparison with plurality

We have also identified and fixed a technical flaw in a very clever reduction due to Chen et al. [8]. Their reduction concerns the parameterized complexity of control by adding candidates to plurality elections, parameterized by the number of voters.

Regarding future work, a quite challenging interesting open question is to completely characterize the class of scoring protocols in terms of control complexity (i.e., to establish dichotomy results for the various control types), as has been done by Hemaspaandra et al. [24] for constructive control by adding voters, by Hemaspaandra and Hemaspaandra [20] for constructive coalitional weighted manipulation, and by Betzler and Dorn [6] and Baumeister and Rothe [4] for the possible winner problem (a generalization of coalitional unweighted manipulation due to Konczak and Lang [25]). Finally, it would also be interesting to study veto with respect to the refined models of control by partition introduced by Erdélyi et al. [13].

ACKNOWLEDGEMENTS

We thank the reviewers for many helpful suggestions. This work has been supported in part by DFG grant RO-1202/15-1.

⁷ We omit the order symbol \succ , so the orders $R(i, j) \succ d \succ \dots$ and $R'(i, j) \succ d \succ \dots$ in this quote are written $R(i, j) d \dots$ and $R'(i, j) d \dots$ here.

REFERENCES

- [1] Y. Bachrach, O. Lev, Y. Lewenberg, and Y. Zick, ‘Misrepresentation in district voting’, in *Proceedings of the 25th International Joint Conference on Artificial Intelligence*. AAAI Press/IJCAI, (July 2016). To appear.
- [2] J. Bartholdi III, C. Tovey, and M. Trick, ‘The computational difficulty of manipulating an election’, *Social Choice and Welfare*, **6**(3), 227–241, (1989).
- [3] J. Bartholdi III, C. Tovey, and M. Trick, ‘How hard is it to control an election?’, *Mathematical and Computer Modelling*, **16**(8/9), 27–40, (1992).
- [4] D. Baumeister and J. Rothe, ‘Taking the final step to a full dichotomy of the possible winner problem in pure scoring rules’, *Information Processing Letters*, **112**(5), 186–190, (2012).
- [5] D. Baumeister and J. Rothe, ‘Preference aggregation by voting’, in *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*, ed., J. Rothe, chapter 4, 197–325, Springer-Verlag, (2015).
- [6] N. Betzler and B. Dorn, ‘Towards a dichotomy for the possible winner problem in elections based on scoring rules’, *Journal of Computer and System Sciences*, **76**(8), 812–836, (2010).
- [7] N. Betzler and J. Uhlmann, ‘Parameterized complexity of candidate control in elections and related digraph problems’, *Theoretical Computer Science*, **410**(52), 5425–5442, (2009).
- [8] J. Chen, P. Faliszewski, R. Niedermeier, and N. Talmon, ‘Elections with few voters: Candidate control can be easy’, in *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pp. 2045–2051. AAAI Press, (January 2015).
- [9] V. Conitzer, T. Sandholm, and J. Lang, ‘When are elections with few candidates hard to manipulate?’, *Journal of the ACM*, **54**(3), Article 14, (2007).
- [10] R. Downey and M. Fellows, *Parameterized Complexity*, Springer-Verlag, 2nd edn., 2013.
- [11] E. Elkind, P. Faliszewski, and A. Slinko, ‘Cloning in elections: Finding the possible winners’, *Journal of Artificial Intelligence Research*, **42**, 529–573, (2011).
- [12] G. Erdélyi, M. Fellows, J. Rothe, and L. Schend, ‘Control complexity in Bucklin and fallback voting: A theoretical analysis’, *Journal of Computer and System Sciences*, **81**(4), 632–660, (2015).
- [13] G. Erdélyi, E. Hemaspaandra, and L. Hemaspaandra, ‘More natural models of electoral control by partition’, in *Proceedings of the 4th International Conference on Algorithmic Decision Theory*, pp. 396–413. Springer-Verlag *Lecture Notes in Artificial Intelligence #9346*, (September 2015).
- [14] G. Erdélyi, M. Nowak, and J. Rothe, ‘Sincere-strategy preference-based approval voting fully resists constructive control and broadly resists destructive control’, *Mathematical Logic Quarterly*, **55**(4), 425–443, (2009).
- [15] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra, ‘How hard is bribery in elections?’, *Journal of Artificial Intelligence Research*, **35**, 485–532, (2009).
- [16] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra, ‘Multimode control attacks on elections’, *Journal of Artificial Intelligence Research*, **40**, 305–351, (2011).
- [17] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe, ‘Lull and Copeland voting computationally resist bribery and constructive control’, *Journal of Artificial Intelligence Research*, **35**, 275–341, (2009).
- [18] P. Faliszewski and J. Rothe, ‘Control and bribery in voting’, in *Handbook of Computational Social Choice*, eds., F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia, chapter 7, 146–168, Cambridge University Press, (2016).
- [19] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, 1979.
- [20] E. Hemaspaandra and L. Hemaspaandra, ‘Dichotomy for voting systems’, *Journal of Computer and System Sciences*, **73**(1), 73–83, (2007).
- [21] E. Hemaspaandra, L. Hemaspaandra, and C. Menton, ‘Search versus decision for election manipulation problems’, Technical Report arXiv:1202.6641 [cs.GT], Computing Research Repository, arXiv.org/corr/, (2012).
- [22] E. Hemaspaandra, L. Hemaspaandra, and C. Menton, ‘Search versus decision for election manipulation problems’, in *Proceedings of the 30th Annual Symposium on Theoretical Aspects of Computer Science*, volume 20 of *LIPICs*, pp. 377–388. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, (February 2013).
- [23] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe, ‘Anyone but him: The complexity of precluding an alternative’, *Artificial Intelligence*, **171**(5–6), 255–285, (2007).
- [24] E. Hemaspaandra, L. Hemaspaandra, and H. Schnoor, ‘A control dichotomy for pure scoring rules’, in *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pp. 712–720. AAAI Press, (July 2014).
- [25] K. Konczak and J. Lang, ‘Voting procedures with incomplete preferences’, in *Proceedings of the Multidisciplinary IJCAI-05 Workshop on Advances in Preference Handling*, pp. 124–129, (July/August 2005).
- [26] A. Lin, *Solving Hard Problems in Election Systems*, Ph.D. dissertation, Rochester Institute of Technology, Rochester, NY, USA, March 2012.
- [27] H. Liu and D. Zhu, ‘Parameterized complexity of control problems in maximin election’, *Information Processing Letters*, **110**(10), 383–388, (2010).
- [28] A. Loreggia, N. Narodytska, F. Rossi, B. Venable, and T. Walsh, ‘Controlling elections by replacing candidates or votes (extended abstract)’, in *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems*, pp. 1737–1738. IFAAMAS, (May 2015).
- [29] C. Menton, ‘Normalized range voting broadly resists control’, *Theory of Computing Systems*, **53**(4), 507–531, (2013).
- [30] C. Menton and P. Singh, ‘Control complexity of Schulze voting’, in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pp. 286–292. AAAI Press/IJCAI, (August 2013).
- [31] R. Niedermeier, *Invitation to Fixed-Parameter Algorithms*, Oxford University Press, 2006.
- [32] D. Parkes and L. Xia, ‘A complexity-of-strategic-behavior comparison between Schulze’s rule and ranked pairs’, in *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pp. 1429–1435. AAAI Press, (July 2012).
- [33] C. Puppe and A. Tasnádi, ‘Optimal redistricting under geographical constraints: Why “pack and crack” does not work’, *Economics Letters*, **105**(1), 93–96, (2009).
- [34] N. Russel, *Complexity of Control of Borda Count Elections*, Master’s thesis, Rochester Institute of Technology, 2007.

Agent-Based Refinement for Predicate Abstraction of Multi-Agent Systems

Francesco Belardinelli¹ and Alessio Lomuscio¹ and Jakub Michaliszyn²

Abstract. We put forward an agent-based refinement methodology for the verification of infinite-state Multi-Agent Systems by predicate abstraction. We use specifications defined in a three-valued variant of the temporal epistemic logic ATLK. We define “failure states” as candidates for refinement, and provide a sound automatic procedure for their identification. Further, we introduce a methodology based on Craig’s interpolants for the refinement of the agent-specific predicates upon which the abstraction is built. We illustrate the refinement technique on an infinite-state auction scenario, and show that specifications of interest, that could not be checked by plain abstraction, can now be verified on the refined models.

1 Introduction

Over the past 15 years, considerable research has taken place in the area of verification of *finite state* Multi-Agent Systems (MAS). This includes symbolic model checking methods [13, 25], SAT-based methods [31], partial-order reductions [23], and symmetry reduction [8]. Considerably less attention has so far been paid to devising techniques for verifying *infinite state* MAS. Since, like standard programs, MAS typically denote infinite models, devising techniques for verifying infinite state MAS remains of considerable interest.

Predicate abstraction [9, 19] is a successful approach to the verification of infinite state programs. In predicate abstraction finite state models, representing under- and over-approximations of the system, are generated automatically by Boolean programs built on predicates derived from the program’s and the system’s specifications. If the truth value of the specification cannot immediately be determined on the initial Boolean program, the list of predicates is updated automatically and a new Boolean program is generated and checked. While this procedure cannot be complete due to the undecidability of the underlying problem, by checking several refinements in succession it is often possible to determine the truth value of the specification on the infinite-state program. A key aspect of this approach is the actual derivation of the refined abstractions.

While predicate abstraction is an established technique in software verification, considerable challenges need to be overcome before it can be applied to MAS. These include the fact that MAS semantics are modular in the agents and that agent-based specifications are considerably richer than those traditionally used in software engineering. Any predicate abstraction technique for MAS ought to support these aspects.

In this paper we introduce a refinement technique for verifying MAS against specifications defined in the agent-based logic ATLK.

A key aspect of the approach we take is the particular setup we consider when combining ATL [2] and epistemic logic [12, 30] to form the logic ATLK that we use to specify MAS. ATL is most often used in its original variant that assumes systems with perfect recall and complete information. This setup is attractive from a verification perspective as the corresponding model checking problem is PTIME, like CTL and CTLK [11, 26]. In contrast, epistemic logic is defined on the basis of incomplete information. This creates a tension in combinations such as ATEL [15], where ATL and epistemic modalities do not share the same underlying information model for the agents in the system. Proposals have been made to overcome this modelling difficulty. The natural setting involves assuming incomplete information for both the strategic ATL modalities and the epistemic operators [17]. If memoryless, uniform strategies [18] are assumed, this leads to a decidable model checking problem; the resulting complexity is however Δ_2^P -complete [16]. In turn this makes the model checking problem exponential against implicit structures given by modelling languages. Given the difficulty of model checking large state spaces, any practical prospect of verifying MAS is unfeasible under this assumption.

To solve the difficulty above we here work with a variant of ATLK which is defined on incomplete information and memoryless, non-uniform strategies. Under non-uniform strategies, agents do not have to play the same action in the same local state, as long as the action is allowed by their protocol. This set up has been proposed in [27] and used in a number of applications [25]. Under this setting ATLK retains a PTIME model checking problem and verification can be performed via fixed-point characterisations of the ATL operators. The semantics of non-uniform strategies is considerably more convoluted and was formally presented in [20, 21, 22]. In this paper we adopt this framework, which we recall in the next section. However, we refer to these papers for more motivations, discussions of these features, as well as relationship with alternative assumptions, including plain ATEL (see [1]). We stress that the framework here proposed entirely subsumes CTLK, for which no predicate refinement methodology has been proposed yet.

Related Work. Other than the contributions hereafter, we are aware of no work addressing the verification of infinite-state MAS by predicate abstraction. In [32, 3] the authors define a predicate abstraction methodologies supporting CTL and Alternating Modal Logic (AML) specifications. This work differs from the present one in several respects. Firstly, their specification language does not support epistemic modalities. Secondly, the AML semantics assumes complete information and perfect recall; instead we only assume incomplete information and no memory. Thirdly, no method is given for the refinement of predicates. In contrast, we here put forward an algorithm based on Craig’s interpolants that is used to generate suc-

¹ Department of Computing, Imperial College London, UK

² Institute of Computer Science, University of Wrocław, Poland

cessive refinements from the agents' models.

Closer to our work are [21, 22] where a three valued logic ATLK is defined and a procedure for an agent-based state and action abstraction is given. However, no solution is proposed there for performing refinement on the abstract models. So if a specification is initially undefined, no conclusion can be drawn. We here follow that approach, but extend it by identifying so called "failure pairs", which we exploit to build a refined model. This enables us to solve the verification problem in several cases of interest where the original technique fails.

Predicate abstraction for the verification of MAS against temporal-epistemic specifications was proposed in [14]. Our contribution differs from that work as we support ATL specifications; the underlying semantics is different; also while [14] addresses the specific case of GSM programs, here we deal with generic MAS; lastly, in common with [22], [14] cannot deal with predicate refinement, which constitutes our main contribution here.

Scheme of the paper. The paper is structured as follows. In Sections 2 we summarise the methodology from [22] for initial abstraction on MAS. In Section 3 we define the concept of *failure pair*; define an algorithm for their identification, and study its properties. We adopt these in Section 4 to derive Craig's interpolants that we use to revise the list of predicates in the initial abstraction. We exemplify the technique in Section 5 and conclude in Section 6 by pointing to future work.

2 Predicate Abstraction for MAS

In this paper we assume that agents have imperfect information and use memoryless (positional) strategies [4, 27]; this is in contrast with previous approaches [32, 3] that assume perfect information. We initially follow the three-valued abstraction methodology introduced by [20, 21], that we summarise hereafter. In the following $Ag = \{1, \dots, m\}$ is a set of agents and \mathcal{V} a set of propositions. Given a set U , \bar{U} denotes its complement (w.r.t. some $V \supseteq U$).

We first define the notion of *interpreted system* [12], to represent formally the execution of a multi-agent system.

Definition 1 (IS) An interpreted system is a tuple $M = (\{L_i, Act_i, P_i, t_i\}_{i \in Ag}, I, \Pi)$ such that:

- for each agent $i \in Ag$,
 - L_i is the (possibly infinite) set of local states;
 - Act_i is the set of actions;
 - $P_i : L_i \rightarrow (2^{Act_i} \setminus \{\emptyset\})$ is the local protocol;
 - $t_i \subseteq L_i \times ACT \times L_i$ is the local transition relation, where $ACT = Act_1 \times \dots \times Act_{|Ag|}$ is the set of joint actions;
- $I \subseteq L_1 \times \dots \times L_{|Ag|}$ is the set of global initial states;
- $\Pi : L_1 \times \dots \times L_{|Ag|} \times \mathcal{V} \rightarrow \{\text{tt}, \text{ff}, \text{uu}\}$ is the labelling function.

By Def. 1 each agent i in an interpreted system is assumed to perform the actions in Act_i , according to protocol P_i . Differently from the standard notion of IS [12], here the transition function is local [24], and propositional atoms can receive three truth values: true tt, false ff, and undefined uu. We say that a truth value t is *defined* whenever $t \neq \text{uu}$. Also, $v.i$ denotes the $i + 1$ -th element of tuple v .

Given an IS M , we introduce the *global transition relation* $T \subseteq S \times ACT \times S$ such that $T(s, a, s')$ holds iff for all $i \in Ag$,

- $t_i(s.i, a, s'.i)$, and

- $a.i \in P_i(s.i)$.

Then, $S \subseteq L_1 \times \dots \times L_{|Ag|}$ denotes the set of *global states*, reachable by the global transition relation T from the set I of initial states. Finally, for every $i \in Ag$, $\sim_i \subseteq S^2$ is the *epistemic indistinguishability relation* defined as $s \sim_i s'$ iff $s.i = s'.i$ [12]. Given a set $\Gamma \subseteq Ag$ of agents, the relation \sim_Γ is the transitive closure of $(\bigcup_{i \in \Gamma} \sim_i)$. In the following we assume that our models are *non-terminating*, i.e., for every $s \in S$ and enabled joint action $a \in ACT$, $T(s, a, s')$ holds for some state $s' \in S$.

In this paper we analyse two logics built on the same syntax, but with different semantics: the two-valued logic ATLK^{2v} and the three-valued logic ATLK^{3v}.

Definition 2 (ATLK) Formulas in the logics ATLK^{2v} and ATLK^{3v} are defined as follows:

$$\varphi ::= q \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle\Gamma\rangle\rangle X\varphi \mid \langle\langle\Gamma\rangle\rangle(\varphi U \varphi) \mid \langle\langle\Gamma\rangle\rangle G\varphi \mid C_{\Gamma'}\varphi$$

where $q \in \mathcal{V}$, $i \in Ag$, $\Gamma, \Gamma' \subseteq Ag$, and $\Gamma' \neq \emptyset$.

The logic ATLK is an epistemic extension of Alternating-time Temporal Logic [17, 27], including the common knowledge operator $C_{\Gamma'}$. We refer to [27, 21] for the reading of modalities and derived operators in the context of the present semantics. We use abbreviations to introduce $\langle\langle\Gamma\rangle\rangle F\varphi$, the remaining propositional connectives, and ATLK operators. In particular, for every agent $i \in Ag$, we define the individual knowledge operator K_i as $C_{\{i\}}$.

In order to provide a semantics to ATLK by means of interpreted systems, we introduce the notion of a *memoryless strategy* for agent $i \in Ag$ as a function $f_i : L_i \rightarrow (2^{Act_i} \setminus \emptyset)$ such that for every local state $l \in L_i$, $f_i(l) \subseteq P_i(l)$. Given a path $p = s_0 s_1 \dots, p^i$ denotes the $i + 1$ -th element s_i in p . Given a set $F_\Gamma = \{f_i \mid i \in \Gamma\}$ of strategies, one for each agent $i \in \Gamma$, a set X of paths is F_Γ -*compatible* if it is a minimal, non-empty set of paths such that for every $p \in X$, position $j \geq 0$, joint actions a, a' , and state s' , if $T(p^j, a, p^{j+1})$, $T(p^j, a', s')$, and for every $i \in \Gamma$, $a'.i = a.i \in f_i(p^j.i)$, then there exists some path $p' \in X$ starting with p^0, \dots, p^j, s' . Let $out(s, F_\Gamma)$ be the family of all F_Γ -compatible sets of paths starting from s .

We briefly comment on the notions of strategy and compatible path just introduced. Specifically, we assume that strategies are non-uniform in the sense of [27], i.e., agents can execute different actions at different global states in which their own local state is the same. This is in contrast with both the original semantics for ATL [2], which stipulates complete information of the global state, and with successive proposals to accommodate imperfect information [17]. However, it can be proved that the present formulation and the perfect information account of [2] are logically equivalent in the sense that an ATL formula is true in the setting we here adopt if and only if the formula is true in the semantics adopted in [2]. It follows that, for the two-valued fragment, an ATLK formula holds in an interpreted system under the present semantics if it holds in the ATEL logic in [15]. In particular, the fixed point characterisations of ATL operators hold in the present setting.

Finally, we report the three-valued satisfaction relation \models^3 from [21]. We assume the Kleene semantics for the standard boolean connectives. For the ATL and knowledge modalities, the semantic is defined as follows.

Definition 3 (Satisfaction) The 3-valued satisfaction relation \models^3

for an IS M , state $s \in S$, and formula φ is defined as follows:

$$M, s \models^3 \langle\langle \Gamma \rangle\rangle X\varphi = \begin{cases} \text{tt} & \text{iff for some strategy } F_\Gamma, \text{ some } X \in \text{out}(s, F_\Gamma) \\ & \text{and all } p \in X, \text{ we have } (M, p^1 \models^3 \varphi) = \text{tt} \\ \text{ff} & \text{iff for some strategy } F_{\bar{\Gamma}}, \text{ some } X \in \text{out}(s, F_{\bar{\Gamma}}) \\ & \text{and all } p \in X \text{ we have } (M, p^1 \models^3 \varphi) = \text{ff} \end{cases}$$

$$M, s \models^3 \langle\langle \Gamma \rangle\rangle \varphi_1 U \varphi_2 = \begin{cases} \text{tt} & \text{iff for some strategy } F_\Gamma, \text{ some } X \in \text{out}(s, F_\Gamma) \\ & \text{and all } p \in X, \text{ there is } k \geq 0 \text{ s.t. } (M, p^k \models^3 \\ & \varphi_2) = \text{tt} \text{ and for all } j < k, (M, p^j \models^3 \varphi_1) = \text{tt} \\ \text{ff} & \text{iff for some strategy } F_{\bar{\Gamma}}, \text{ some } X \in \text{out}(s, F_{\bar{\Gamma}}) \\ & \text{and all } p \in X, k \geq 0 \text{ we have } (M, p^k \models^3 \varphi_2) = \\ & \text{ff or there is } j < k \text{ s.t. } (M, p^j \models^3 \varphi_1) = \text{ff} \end{cases}$$

$$M, s \models^3 \langle\langle \Gamma \rangle\rangle G\varphi = \begin{cases} \text{tt} & \text{iff for some strategy } F_\Gamma, \text{ some } X \in \text{out}(s, F_\Gamma) \text{ and} \\ & \text{all } p \in X, i \geq 0 \text{ we have } (M, p^i \models^3 \varphi) = \text{tt} \\ \text{ff} & \text{iff for some strategy } F_{\bar{\Gamma}}, \text{ some } X \in \text{out}(s, F_{\bar{\Gamma}}) \text{ and} \\ & \text{all } p \in X, \text{ there is } i \geq 0 \text{ s.t. } (M, p^i \models^3 \varphi) = \text{ff} \end{cases}$$

$$M, s \models^3 C_\Gamma \varphi = \begin{cases} \text{tt} & \text{iff } (M, s' \models^3 \varphi) = \text{tt} \text{ for all } s' \sim_\Gamma s \\ \text{ff} & \text{iff } (M, s \models^3 \varphi) = \text{ff} \end{cases}$$

In all other cases, the value of formula φ is undefined (uu).

The two-valued satisfaction relation \models^2 can be derived from \models^3 by considering clauses for tt only, as well as classic negation. An IS M satisfies property φ in ATLK, or $M \models^2 \varphi$, iff for all initial states $s \in I$, $(M, s) \models^2 \varphi$. Similarly, $(M \models^3 \varphi) = \text{tt}$ (resp. ff) iff for all (resp. some) $s \in I$, $((M, s) \models^3 \varphi) = \text{tt}$ (resp. ff). Otherwise, $(M \models^3 \varphi) = \text{uu}$.

In [21] it is shown that that for every φ in ATLK, $(M \models^3 \varphi) = \text{tt}$ implies $M \models^2 \varphi$ and $(M \models^3 \varphi) = \text{ff}$ implies $M \not\models^2 \varphi$. That is, defined truth values are preserved.

Since interpreted systems might have a possibly infinite state space, abstraction techniques have been developed to make verification feasible [7, 6]. In this section we describe the agent-based abstraction techniques put forward in [21, 22] that uses predicates derived from the system description and the specification to be checked.

Assume an IS M and a list $(\vec{p}_1, \dots, \vec{p}_{|Ag|})$ of tuples of predicates, where intuitively each predicate represents a condition on an agent's protocol, or transition relation. The satisfaction of conjunctions c of literals (predicates and their negation), called *cubes*, can naturally be given at an agent's local state, denoted as $l_i \models c$. A cube is *satisfiable* iff it is satisfied by some local state. By using predicates, agent descriptions can be abstracted as follows.

Definition 4 (Abstract Agent) Given an agent $i \in Ag$ and a list \vec{p}_i of predicates, the abstract agent is a tuple $i^A = \langle L_i^A, Act_i, P_i^{\text{may}}, P_i^{\text{must}}, t_i^{\text{may}}, t_i^{\text{must}} \rangle$ such that:

- L_i^A is the set of all satisfiable cubes;
- the may protocol P_i^{may} is such that $a \in P_i^{\text{may}}(c)$ iff for some $l \in L_i, l \models c$ and $a \in P_i(l)$;
- the may relation t_i^{may} is such that $t_i^{\text{may}}(c, a, c')$ iff for some local states $l, l' \in L_i, l \models c, l' \models c'$, and $t_i(l, a, l')$;

- the must protocol P_i^{must} is such that $a \in P_i^{\text{must}}(c)$ iff for every $l \in L_i, l \models c$ implies $a \in P_i(l)$;
- the must relation t_i^{must} is such that $t_i^{\text{must}}(c, a, c')$ iff for all $l \in L_i$, if $l \models c$ then $t_i(l, a, l')$ for some l' satisfying c' .

We say that a global state $s \in S$ satisfies a tuple $b = (c_1, \dots, c_{|Ag|})$ of cubes, denoted as $s \models b$, if each $s.i$ satisfies c_i .

Definition 5 (Abstract IS) The predicate abstraction of an IS M w.r.t. predicates $(\vec{p}_1, \dots, \vec{p}_{|Ag|})$ is the IS $M^A = (Ag^A, I^A, \Pi^A)$, where:

- Ag^A is the set of abstract agents i^A w.r.t. \vec{p}_i , for $i \in Ag$;
- for every state $b \in S^A$ (where $S^A = L_1^A \times \dots \times L_{|Ag|}^A$), $q \in \mathcal{V}$ and $t \in \{\text{tt}, \text{ff}\}$, $\Pi^A(b, q) = t$ iff $\Pi(s, q) = t$ for all states s satisfying b ;
- $I^A = \{b \mid \text{for some } s \in I, s \models b\}$.

Furthermore, for every $\Gamma \subseteq Ag$, the abstract transition relation $T_\Gamma^A(b, a, b')$ holds iff

- for all $i \in \Gamma, a.i \in P_i^{\text{must}}(b.i)$ and $t_i^{\text{must}}(b.i, a, b'.i)$;
- for all $i \notin \Gamma, a.i \in P_i^{\text{may}}(b.i)$ and $t_i^{\text{may}}(b.i, a, b'.i)$.

Intuitively, the *may* and *must* components of abstract IS can be seen respectively as over- and under-approximations of the strategic abilities of agents. In the following we use the notion of (*immediate*) *successor* according to relations T_Γ^A and $T_{\bar{\Gamma}}^A$.

An abstraction M^A can be used to interpret the language ATLK according to the three-valued semantics. In particular, the following preservation result applies [21].

Theorem 6 Let M be an IS with predicate abstraction M^A . For every ATLK property φ ,

$$\begin{aligned} (M^A \models^3 \varphi) = \text{tt} & \text{ implies } M \models^2 \varphi; \\ (M^A \models^3 \varphi) = \text{ff} & \text{ implies } M \not\models^2 \varphi. \end{aligned}$$

In [22] the result above is exploited to give a methodology for verifying infinite-state MAS. Starting from the infinite-state agents' descriptions and specifications, the relevant predicates, which are then used to construct the abstract, finite-state interpreted system are derived. The MAS specifications are then evaluated on it. If the truth value is defined, it can be deduced whether or not the specification holds on the original MAS. If the specification is undefined, no conclusion can be drawn. Indeed, [22] provides such an example where the technique is unable to determine the value of a specification.

In what follows we put forward a methodology for iteratively refining the agent-specific predicates so that finer and finer abstractions can be constructed and the truth value of the specification may be determined.

3 Identifying Failure Pairs

In this section, inspired by [3], we define a refinement procedure based on Craig's interpolants. Specifically, given an ATLK formula φ , undefined in some state c of an abstract IS M , we investigate the reason for the undefinedness of φ . To do so, we introduce the notion of failure pair and provide an algorithm for their identification. Differently from [3], which considers in detail only the sublanguage of ATLK containing operator $\langle\langle A \rangle\rangle X$ (i.e., Alternating Modal Logic),

here we account for ATLK, including epistemic operators. The procedure we define is agent-based and therefore modular, whereas in [3] the abstraction is defined at the system level.

Since in this section we only work on abstract models, for convenience we will denote them simply as M .

Definition 7 (Relevant Pair) Given a (abstract) state c and a formula φ , the function R , which returns the set of pairs relevant for the truth of φ in c , is the smallest function (w.r.t. the Lorenz order) satisfying the following conditions for each c, ψ and ψ' .

- $R(c, p) = \{(c, p)\}$, for $p \in AP$
- $R(c, \neg\psi) = \{(c, \psi)\} \cup R(c, \psi)$
- $R(c, \psi \wedge \psi') = \{(c, \psi), (c, \psi')\} \cup R(c, \psi) \cup R(c, \psi')$
- $R(c, \langle\Gamma\rangle X\psi) = \{(c', \psi) \mid T_\Gamma(c, a, c') \text{ or } T_{\bar{\Gamma}}(c, a, c'), \text{ for some joint action } a \in ACT\} \cup \bigcup_{c'} R(c', \psi)$
- $R(c, K_i\psi) = \{(c', \psi) \mid c' \sim_i c\} \cup \bigcup_{c'} R(c', \psi)$
- $R(c, \langle\Gamma\rangle G\psi) = \{(c, \psi), \langle\Gamma\rangle X\psi\} \cup R(c, \psi) \cup R(c, \langle\Gamma\rangle X\psi) \cup R(c, \langle\Gamma\rangle X\langle\Gamma\rangle G\psi)$
- $R(c, \langle\Gamma\rangle(\psi U \psi')) = \{(c, \psi), (c, \psi')\} \cup R(c, \psi) \cup R(c, \psi') \cup R(c, \langle\Gamma\rangle X\psi) \cup R(c, \langle\Gamma\rangle X\psi') \cup R(c, \langle\Gamma\rangle X\langle\Gamma\rangle(\psi U \psi'))$
- $R(c, C_\Gamma\psi) = \{(c, \psi)\} \cup R(c, \psi) \cup \bigcup_{i \in \Gamma} R(c, K_i C_\Gamma\psi) \cup R(c, \psi) \cup \bigcup_{i \in \Gamma} R(c, K_i C_\Gamma\psi)$

Observe that $R(c, \varphi)$ is well defined and can be computed by using standard fix-point algorithms, which are indeed validities in the proposed semantics. Specifically, the clauses for G -, U -, and C -formulas make use of the following fixed-point characterisations:

$$\begin{aligned} \langle\Gamma\rangle G\psi &\equiv \psi \wedge \langle\Gamma\rangle X \langle\Gamma\rangle G\psi \\ \langle\Gamma\rangle(\psi U \psi') &\equiv \psi' \vee (\psi \wedge \langle\Gamma\rangle X \langle\Gamma\rangle(\psi U \psi')) \\ C_\Gamma\psi &\equiv \psi \wedge \bigwedge_{i \in \Gamma} K_i C_\Gamma\psi \end{aligned}$$

Example 8 Consider an abstract interpreted system IS with two agents 1, 2, both having two states 0, 1 and two actions A, B , whose model is depicted on Figure 1.

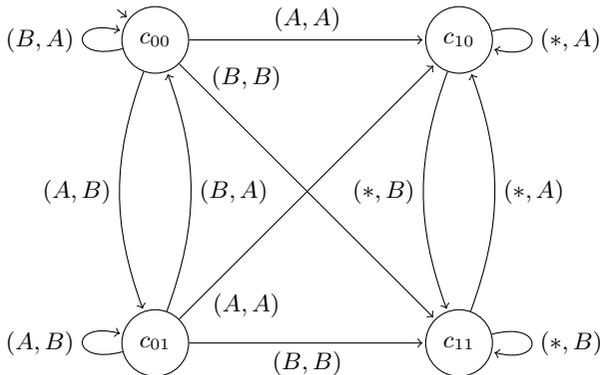


Figure 1. A model for Example 8. We assume that $T_\emptyset = T_{\{1\}} = T_{\{2\}} = T_{\{1,2\}}$ are all as depicted.

This may be interpreted as follows: both agents start in a local state 0. Agent 1 stays in the state 0 until both agents play the same actions; when this happens agent 1 moves to state 1 where it stays

for the rest of the run. The second agent changes its state only on the basis of its action: if it performs action A it then moves to 0; if it performs action B , then it moves to state 1.

Assume that the labelling is such that the only state labelled with p is c_{11} . Consider the formula $\varphi = \langle\{1, 2\}\rangle Gp$.

By definition $R(c_{00}, \varphi)$ contains elements of $\{(c_{00}, \varphi)\}$, $R(c_{00}, p)$, $R(c_{00}, \langle\{1, 2\}\rangle X \langle\{1, 2\}\rangle Gp)$, and for all i, j , $R(c_{ij}, \langle\{1, 2\}\rangle Xp)$.

Then, $R(c_{00}, \langle\{1, 2\}\rangle X \langle\{1, 2\}\rangle Gp)$ contains $(c_{00}, \langle\{1, 2\}\rangle X \langle\{1, 2\}\rangle Gp)$ and all the pairs of $R(c_{ij}, \langle\{1, 2\}\rangle Gp)$, for all i, j . For every j , $R(c_{1j}, \langle\{1, 2\}\rangle Gp)$ contains $(c_{1j}, \langle\{1, 2\}\rangle Gp)$ and elements of $R(c_{10}, p)$, $R(c_{10}, \langle\{1, 2\}\rangle X \langle\{1, 2\}\rangle Gp)$ and for all j , $R(c_{1j}, \langle\{1, 2\}\rangle Xp)$.

The minimal function R satisfying the above conditions is as follows, for each $i, j \in \{0, 1\}$

$$R(c_{ij}, p) = \{(c_{ij}, p)\}$$

$$R(c_{1j}, \langle\{1, 2\}\rangle Xp) = \{(c_{1j}, \langle\{1, 2\}\rangle Xp), (c_{10}, p), (c_{11}, p)\}$$

$$R(c_{0j}, \langle\{1, 2\}\rangle Xp) = \{(c_{0j}, \langle\{1, 2\}\rangle Xp), (c_{10}, p), (c_{11}, p), (c_{00}, p), (c_{01}, p)\}$$

$$R(c_{1j}, \varphi) = \{(c_{1k}, \varphi), (c_{1k}, \langle\{1, 2\}\rangle G\varphi), (c_{1k}, \langle\{1, 2\}\rangle Gp), (c_{1k}, p) \mid k \in \{0, 1\}\}$$

$$R(c_{0j}, \varphi) = \{(c_{lk}, \varphi), (c_{lk}, \langle\{1, 2\}\rangle G\varphi), (c_{lk}, \langle\{1, 2\}\rangle Gp), (c_{lk}, p) \mid l, k \in \{0, 1\}\}$$

The significance of relevant pairs is given by the following immediate lemma.

Lemma 9 If the truth value of φ in c is defined, then truth values for all relevant pairs in $R(c, \varphi)$ are also defined.

Notice that because of loops and the expansions above, for a G -, U -, or a C -formula φ and state c it might be that (c, φ) belongs to $R(c, \varphi)$. Moreover, we show below that the cases for these formulas can be reduced to those for X - and K -formulas. As a consequence, we will be able to focus on refining single steps in a temporal or epistemic transition in the model.

Next, we introduce a notion of failure pair, inspired by [3]. Intuitively, for an abstract state c and formula φ , (c, φ) is a failure pair iff φ is undefined at c , albeit $IS M$ has definite values for all relevant pairs for (c, φ) different from (c, φ) itself.

Definition 10 (Failure Pair) A tuple (c, φ) is a failure pair iff $((M, c) \models^3 \varphi) = \text{uu}$ and for all relevant pairs $(c', \psi) \in R(c, \varphi)$, where ψ is a strict subformula of φ , we have that $((M, c') \models^3 \psi) \in \{\text{tt}, \text{ff}\}$.

A failure pair (c, φ) singles out a formula φ whose undefined value in state c is due to the structural features of the abstract IS itself. Hence, to provide a defined value to φ we have to refine the abstraction by using the information in (c, φ) .

Clearly, propositional formulas are defined whenever all relevant pairs are. Hence, failure pairs can only be determined by atomic propositions and ATL and epistemic operators, as detailed in the following lemma.

Lemma 11 A tuple (c, φ) is a failure pair iff $((M, c) \models^3 \varphi) = \text{uu}$ and one of the following cases hold

- $\varphi = p \in \mathcal{V}$

- $\varphi = \langle\langle\Gamma\rangle\rangle X\psi$ and for all states c' , if $T_{\Gamma}^A(c, a, c')$ or $T_{\bar{\Gamma}}^A(c, a, c')$ for some joint action a , then $((M, c') \models^3 \psi) \in \{\text{tt}, \text{ff}\}$
- $\varphi = K_i\psi$ and $((M, c') \models^3 \psi) \in \{\text{tt}, \text{ff}\}$ for all states $c' \sim_i c$
- φ is a G - or U -formula, in which case there is also a failure pair (c', ψ) , where c' is reachable from c and ψ is a X -formula.
- φ is a C -formula, in which case there is also a failure pair (c', ψ) , where c' is (epistemically) reachable from c and ψ is a K -formula.

Proof sketch. The first part of the lemma follows from Def. 7 and 10. As an example, by contraposition suppose that $\varphi = \langle\langle\Gamma\rangle\rangle X\psi$, $((M, c) \models^3 \varphi) = \text{uu}$, but for some state c' , $T_{\Gamma}^A(c, a, c')$ or $T_{\bar{\Gamma}}^A(c, a, c')$ for some joint action a , and $((M, c') \models^3 \psi) = \text{uu}$. In particular, (c', ψ) is a relevant pair for (c, φ) . Hence, we derive that (c, φ) is not a failure pair. The result follows by contraposition.

For the second part, we show that failure pairs for G -formulas can be reduced to the case for X -formulas. Hence, suppose that

- (i) $((M, c) \models^3 \langle\langle\Gamma\rangle\rangle G\varphi) = \text{uu}$
- (ii) $((M, c) \models^3 \varphi) \in \{\text{tt}, \text{ff}\}$
- (iii) $((M, c') \models^3 \langle\langle\Gamma\rangle\rangle X\varphi) \in \{\text{tt}, \text{ff}\}$ for every reachable c' and for $c' = c$.

From (ii) and (iii) it follows that the truth value of ψ is defined in c and in all of its successors. So, $\langle\langle\Gamma\rangle\rangle G\psi$ is indeed defined in c against (i), which is a contradiction.

The cases for the U - and C -formulas are similar. \square

As a consequence of Lemma 11, we can focus the search for failure pairs and the refinement procedure on X - and K -formulas only. The following result follows from Lemma 11, where a_{Γ} (resp. $a_{\bar{\Gamma}}$) are vectors of actions for the agents in Γ (resp. $\bar{\Gamma}$).

- Lemma 12** (i) If $(c, \langle\langle\Gamma\rangle\rangle X\psi)$ is a failure pair, then for every $a_{\Gamma} \in P_{\Gamma}^{\text{must}}(c)$, for some $a_{\bar{\Gamma}} \in P_{\bar{\Gamma}}^{\text{may}}(c)$, $T_A(c, a_{\Gamma} \cdot a_{\bar{\Gamma}}, c')$ implies $((M, c') \models^3 \psi) = \text{ff}$, and for every $a_{\bar{\Gamma}} \in P_{\bar{\Gamma}}^{\text{must}}(c)$, for some $a_{\Gamma} \in P_{\Gamma}^{\text{may}}(c)$, $T_{\bar{A}}(c, a_{\Gamma} \cdot a_{\bar{\Gamma}}, c')$ implies $((M, c') \models^3 \psi) = \text{tt}$.
- (ii) If $(c, K_i\psi)$ is a failure pair, then for some $c' \neq c$, $c'_i = c_i$ and $((M, c') \models^3 \psi) = \text{ff}$.

Proof sketch. To derive a contradiction, suppose that $(c, \langle\langle\Gamma\rangle\rangle X\psi)$ is a failure pair, but some $a_{\Gamma} \in P_{\Gamma}^{\text{must}}(c)$ is such that for every $a_{\bar{\Gamma}} \in P_{\bar{\Gamma}}^{\text{may}}(c)$, $T_A(c, a_{\Gamma} \cdot a_{\bar{\Gamma}}, c')$ implies $((M, c') \models^3 \psi) \neq \text{ff}$. Since $(c, \langle\langle\Gamma\rangle\rangle X\psi)$ is a failure pair, by Lemma 11 the truth value of ψ at c' has to be defined, and therefore $((M, c') \models^3 \psi) = \text{tt}$. But then, by the semantics in Def. 3 we obtain that $((M, c) \models^3 \langle\langle\Gamma\rangle\rangle X\psi) = \text{tt}$, against the hypothesis that $(c, \langle\langle\Gamma\rangle\rangle X\psi)$ is a failure pair. \square

By building on the preliminaries results illustrated above, we now introduce the algorithm *FRFP* to find relevant failure pairs. The procedure, presented as Algorithm 1, takes as input a finite abstract IS M , a state c and a formula φ such that $((M, c) \models^3 \varphi) = \text{uu}$. As we show hereafter, the algorithm returns a relevant failure pair for (c, φ) .

Lemma 13 The algorithm *FRFP* terminates provided that the interpreted system is finite. Moreover, the algorithm is sound, that is, if *FRFP*(c, φ) returns (c', φ') , then $(c', \varphi') \in R(c, \varphi)$ is a failure pair relevant for (c, φ) .

Proof sketch. Since abstract IS are finite, the algorithm terminates in the case of X -, or K -formulas. Termination in the other cases follows from the fact that φ is finite. The output of the algorithm are failure pairs in view of Lemma 11. For instance, consider

Algorithm 1 The algorithm *FRFP*.

INPUT: Model M ; state c ; formula φ s.t. $((M, c) \models^3 \varphi) = \text{uu}$.
OUTPUT: (c', φ') s.t. $(c', \varphi') \in R(c, \varphi)$.

```

1: procedure FRFP( $c, \varphi$ )
2:   if  $\varphi = p \in \mathcal{V}$  then
3:     return ( $c, p$ )
4:   else if  $\varphi = \neg\varphi'$  then
5:     return FRFP( $c, \varphi'$ )
6:   else if  $\varphi = \varphi_1 \vee \varphi_2$  then
7:     let  $i$  be the minimum s.t.  $((M, c) \models^3 \varphi_i) = \text{uu}$ ;
8:     return FRFP( $c, \varphi_i$ )
9:   else if  $\varphi = \langle\langle\Gamma\rangle\rangle X\varphi'$  then
10:    if for all  $c', T_{\Gamma}(c, a, c')$  or  $T_{\bar{\Gamma}}(c, a, c')$  for some joint ac-
11:    tion  $a \in ACT$  implies  $((M, c') \models^3 \varphi') \in \{\text{tt}, \text{ff}\}$  then
12:      return ( $c, \varphi$ )
13:    else
14:      let  $c'$  be a successor of  $c$  s.t.  $((M, c') \models^3 \varphi') = \text{uu}$ ;
15:      return FRFP( $c', \varphi'$ )
16:    end if
17:   else if  $\varphi = K_i\varphi'$  then
18:     if for every  $c' \sim_i c$ ,  $((M, c') \models^3 \varphi') \in \{\text{tt}, \text{ff}\}$  then
19:       return ( $c, \varphi$ )
20:     else
21:       let  $c'$  be s.t.  $c' \sim_i c$  and  $((M, c') \models^3 \varphi') = \text{uu}$ ;
22:       return FRFP( $c', \varphi'$ )
23:     end if
24:   else if  $\varphi = \langle\langle\Gamma\rangle\rangle G\varphi'$  then
25:     if  $((M, c) \models^3 \varphi') = \text{uu}$  then
26:       return FRFP( $c, \varphi'$ );
27:     else
28:       let  $c'$  be  $c$  or a successor of  $c$  s.t.  $((M, c') \models^3$ 
29:        $\langle\langle\Gamma\rangle\rangle X\varphi') = \text{uu}$ 
30:       return FRFP( $c', \langle\langle\Gamma\rangle\rangle X\varphi'$ ).
31:     end if
32:   else if  $\varphi = \langle\langle\Gamma\rangle\rangle(\varphi_1 U \varphi_2)$  then
33:     if  $((M, c) \models^3 \varphi_2) = \text{uu}$  then
34:       return FRFP( $c, \varphi_2$ );
35:     else if  $((M, c) \models^3 \varphi_1) = \text{uu}$  then
36:       return FRFP( $c, \varphi_1$ );
37:     else
38:       let  $c'$  be  $c$  or a successor of  $c$  s.t.  $((M, c') \models^3 \varphi_1 \wedge$ 
39:        $\langle\langle\Gamma\rangle\rangle X\varphi_2) = \text{uu}$ 
40:       return FRFP( $c', \varphi_1 \wedge \langle\langle\Gamma\rangle\rangle X\varphi_2$ ).
41:     end if
42:   else if  $\varphi = C_{\Gamma}\varphi'$  then
43:     if  $((M, c) \models^3 \varphi') = \text{uu}$  then
44:       return FRFP( $c, \varphi'$ );
45:     else
46:       let  $c' \sim_{\Gamma} c$  and  $i \in Ag$  be s.t.  $((M, c') \models^3 K_i\varphi') =$ 
47:       uu
48:       return FRFP( $c', K_i\varphi'$ ).
49:     end if
50:   end if
51: end procedure

```

$\varphi = \langle\langle\Gamma\rangle\rangle G\psi$. If $((M, c) \models^3 \varphi) = \text{uu}$, then by Lemma 11 either $((M, c) \models^3 \psi) = \text{uu}$ or for some successor c' or $c' = c$, $((M, c) \models^3 \langle\langle\Gamma\rangle\rangle X\psi) = \text{uu}$. In the former case, *FRFP*(c, φ) = *FRFP*(c, ψ) and the algorithm is sound by the inductive step. In the latter case, *FRFP*(c, φ) = *FRFP*($c', \langle\langle\Gamma\rangle\rangle X\psi$), and once again

the result follows by the inductive hypothesis. The other cases are similar. \square

As a consequence, Algorithm 1 together with Lemma 13, defines a procedure to identify failure pairs, which will be used in the next section to refine the list of predicates, and therefore the abstraction.

4 Refining Abstractions

In this section we introduce and analyse a methodology for refining an abstract model on which a specification is initially evaluated as undefined. The method is based on the iterative application of Algorithm 2 below, which takes as input the present list of predicates, upon which the abstraction is built, and a failure pair, and returns as output the revised predicate list upon which a further abstraction can be built.

A key aspect in the derivation of the updated list of predicates is the use of *Craig's interpolants* [28]. Recall that the Craig's interpolant of formulas A and B , whose conjunction $A \wedge B$ is unsatisfiable, is a formula I such that

- $A \rightarrow I$ is valid;
- $I \wedge B$ is unsatisfiable; and
- I contains only non-logical symbols appearing in both A and B .

Craig's interpolants have previously been proven effective in refining abstractions in the context of different semantics and less expressive specification languages [28, 29]. Intuitively, the refinement methodology can be summarised as follows. Assume that formulas A and B represent witnesses for the current and the successive state in the abstract model. If the transition from A to B is spurious, that is, the transition in the abstract model does not correspond to a transition in the concrete system, the conjunction $A \wedge B$ is unsatisfiable. The interpolant I for $A \wedge B$ typically gives useful evidence as regards the reasons of the transition's spuriousness and can usefully provide guidance to refine the model [6].

Hereafter we describe the interpolation procedure we use to generate new predicates. Since the specification of interpreted systems includes first-order features, namely, linear arithmetic over the integers, in the following we adapt the approach originally put forward in [29] by applying concepts from [5] to account for the particular setting.

To begin, recall from Lemma 11 that all failure pairs can be reduced to the cases of atomic, X -, or K -formulas. Therefore, we present the refinement procedure via Craig's interpolations for these three cases in Algorithm 2, and discuss its rationale in the following. Algorithm 2 takes as input a failure pair (c, φ) and a tuple $(\vec{p}_1, \dots, \vec{p}_{|Ag|})$ consisting of vectors of predicates and it returns an updated tuple $(\vec{p}'_1, \dots, \vec{p}'_{|Ag|})$ of vectors of (possibly new) predicates. We assume Algorithm 2 operates on the abstract model under analysis and that φ is either an X - or a K -formula. We will address the atomic case later in the section.

X -Formulas (lines 2-7). The procedure takes as input the failure pair $(c, \langle\langle\Gamma\rangle\rangle X\varphi)$, as provided by Algorithm 1. By Lemma 12, we have that (i) for every $a_\Gamma \in P_\Gamma^{must}(c)$, for some $a_{\overline{\Gamma}} \in P_{\overline{\Gamma}}^{may}(c)$, $T_\Gamma(c, a_\Gamma \cdot a_{\overline{\Gamma}}, c')$ implies $((M, c') \models^3 \varphi) = \text{ff}$, and (ii) for every $a_{\overline{\Gamma}} \in P_{\overline{\Gamma}}^{must}(c)$, for some $a'_\Gamma \in P_\Gamma^{may}(c)$, $T_{\overline{\Gamma}}(c, a'_\Gamma \cdot a_{\overline{\Gamma}}, c'')$ implies $((M, c'') \models^3 \varphi) = \text{tt}$, which corresponds to line 3 in Algorithm 2.

In both cases we check whether the abstract transitions $T_\Gamma(c, a_\Gamma \cdot a_{\overline{\Gamma}}, c')$ and $T_{\overline{\Gamma}}(c, a'_\Gamma \cdot a_{\overline{\Gamma}}, c'')$ correspond to actual transitions in the

Algorithm 2 The algorithm Refine.

INPUT: Failure pair (c, φ) ; $(\vec{p}_1, \dots, \vec{p}_{|Ag|})$.

OUTPUT: $(\vec{p}'_1, \dots, \vec{p}'_{|Ag|})$.

```

1: procedure Refine( $(c, \varphi), (\vec{p}_1, \dots, \vec{p}_{|Ag|})$ )
2:   if  $\varphi = \langle\langle\Gamma\rangle\rangle X\varphi'$  then
3:     let  $c'$  be s.t.  $T_\Gamma(c, a_\Gamma \cdot a_{\overline{\Gamma}}, c')$  and  $((M, c') \models^3 \varphi') = \text{ff}$ ,
       or  $T_{\overline{\Gamma}}(c, a_\Gamma \cdot a_{\overline{\Gamma}}, c')$  and  $((M, c') \models^3 \varphi') = \text{tt}$ ;
4:     if there is  $i \in Ag$  and  $l, l' \in L_i$  s.t.  $l \models c.i, l' \models c'.i$ 
       and  $t_i(l, a_\Gamma \cdot a_{\overline{\Gamma}}, l')$  does not hold then
5:       let  $I$  be an interpolant for  $(l \wedge a_\Gamma \cdot a_{\overline{\Gamma}}) \wedge l'$ 
6:       return  $(\vec{p}_1, \dots, \vec{p}'_i \cdot I, \dots, \vec{p}_{|Ag|})$ 
7:     else return  $(\vec{p}_1, \dots, \vec{p}_{|Ag|})$ 
8:   else if  $\varphi = K_i\varphi'$  then
9:     let  $c'$  be s.t.  $c' \neq c, c'.i = c.i$  and  $((M, c') \models^3 \varphi) = \text{ff}$ 
10:    if there are  $l, l' \in L_i$  s.t.  $l \models c.i, l' \models c'.i$  and  $l \neq l'$ 
11:      let  $I$  be an interpolant for  $l \wedge l'$ 
12:      return  $(\vec{p}_1, \dots, \vec{p}'_i \cdot I, \dots, \vec{p}_{|Ag|})$ 
13:    else return  $(\vec{p}_1, \dots, \vec{p}_{|Ag|})$ 
14:   else return  $(\vec{p}_1, \dots, \vec{p}_{|Ag|})$ 
15:   end if
16: end procedure

```

concrete IS; that is, whether there exists concrete states $s, s', s'' \in S$ such that $s \models c, s' \models c', s'' \models c''$, and both $T(s, a_\Gamma \cdot a_{\overline{\Gamma}}, s')$ and $T(s, a'_\Gamma \cdot a_{\overline{\Gamma}}, s'')$. This check is performed modularly, on the various agents $i \in Ag$. We comment on the case for $T_\Gamma(c, a_\Gamma \cdot a_{\overline{\Gamma}}, c')$; the other case is similar.

By definition of the predicate abstraction, $T_\Gamma(c, a_\Gamma \cdot a_{\overline{\Gamma}}, c')$ holds iff $t_i^{must}(c.i, a_\Gamma \cdot a_{\overline{\Gamma}}, c'.i)$ for $i \in A$ and $t_i^{may}(c.i, a_\Gamma \cdot a_{\overline{\Gamma}}, c'.i)$ for $i \notin A$. Now consider an agent $i \in Ag$ and witnesses $l, l' \in L_i$, that is, $l \models c.i$ and $l' \models c'.i$. Depending on $i \in Ag$, we refine either $t_i^{must}(c.i, a_\Gamma \cdot a_{\overline{\Gamma}}, c'.i)$ or $t_i^{may}(c.i, a_\Gamma \cdot a_{\overline{\Gamma}}, c'.i)$. If $t_i(l, a_\Gamma \cdot a_{\overline{\Gamma}}, l')$ holds, then l and l' witness indeed the abstract transition (line 7 in Algorithm 2). Otherwise, we consider the conjunction $\theta = l \wedge a_\Gamma \cdot a_{\overline{\Gamma}} \wedge l'$, where the atoms and variables in l' are primed, while actions are interpreted as equalities between variables and their primed versions (line 4 in Algorithm 2). If $t_i(l, a_\Gamma \cdot a_{\overline{\Gamma}}, l')$ does not hold, then θ is unsatisfiable, and we can make use of interpolation to refine the abstract transition. To do this, we need to find two new abstract states $d.i$ and $d'.i$ such that $l \models d.i, l' \models d'.i$, and either $t_i^{may}(d.i, a_\Gamma \cdot a_{\overline{\Gamma}}, d'.i)$ or $t_i^{must}(d.i, a_\Gamma \cdot a_{\overline{\Gamma}}, d'.i)$ does not hold (depending on whether $i \in Ag$ or $i \notin Ag$), where t_i^{may} and t_i^{must} are intuitively the new, refined transitions.

As a result, the new transition t_i^{may} is “finer” than t_i^{may} , or t_i^{must} is “coarser” than t_i^{must} ; that is, t_i^{may} relates fewer concrete local states than t_i^{may} , while t_i^{must} relates more concrete local states than t_i^{must} . More specifically, by interpolation we obtain an interpolant I such that $l \wedge a_\Gamma \cdot a_{\overline{\Gamma}} \rightarrow I$ is valid and $l' \wedge I$ is unsatisfiable (line 5 in Algorithm 2).

We now show how I can be used as a predicate to eliminate the spurious transition from l to l' . In particular, I is built on non-logical symbols appearing in both l and l' . Hence, it is local to agent i and can be introduced as a new predicate. The updated list $(\vec{p}_1, \dots, \vec{p}'_i, \dots, \vec{p}_{|Ag|})$ of predicates, where $\vec{p}'_i = \vec{p}_i \cdot I$ (line 6 in Algorithm 2) is then returned and used to construct a further abstracted model M'^A . Notice that M'^A does not contain the state $c.i$, but it includes at least one of the new states $c.i \wedge I$ and $c.i \wedge \neg I$. We now have that either $l \models c.i \wedge I$ or $l \models c.i \wedge \neg I$. Then, let $d.i$ be the state satisfied by l . Now, if $l \models d.i$ and $t_i(l, a_\Gamma \cdot a_{\overline{\Gamma}}, l')$ for some local state $l' \in L_i$ such that $l' \models c'.i$, then $l' \models c'.i \wedge I = d'.i$ as well, as

$l \wedge a_{\Gamma} \cdot a_{\overline{\Gamma}} \rightarrow I$ is a validity. On the other hand, $l' \models c'.i \wedge \neg I = d'.i$, and therefore the successors l'' and l' of l belong to different abstract states $d''.i$ and $d'.i$. As a result, the transition t_i^{may} in M^{A} is finer than t_i^{may} , or t_i^{must} is coarser than t_i^{must} (depending on whether $i \in Ag$ or $i \notin Ag$). This terminates the procedure for X -formulas.

K -formulas (lines 8-13). The case of K -formulas is similar to the previous one. By Lemma 12, for some $c' \neq c$, $c'.i = c.i$ and $((M, c') \models^3 \varphi) = \text{ff}$ (line 9 in Algorithm 2). Now consider witnesses $l, l' \in L_i$ such that $l \models c.i$, $l' \models c'.i$, but $l \neq l'$, if any (line 10 in Algorithm 2). This means that l and l' are spurious witnesses for the i -indistinguishability of abstract states c and c' . It follows that the conjunction $l \wedge l'$ is unsatisfiable, as l and l' are two different assignments of values to the variables and propositional atoms of agent i . Hence, we can find a Craig's interpolant I such that $l \rightarrow I$ is valid and $l' \wedge I$ is unsatisfiable (line 11 in Algorithm 2). As in the case of X -formulas, we use I as a predicate to refine the spurious indistinguishability relation. Specifically, I can be used as a new predicate, as it is built on non-logical symbols appearing in both l and l' . The revised list $(\vec{p}_1, \dots, \vec{p}'_i, \dots, \vec{p}_{|Ag|})$ of predicates can now be returned (line 12 in Algorithm 2), where $\vec{p}'_i = \vec{p}_i \cdot I$; this can be used to generate a refinement M^{A} . On the refined model M^{A} we will obtain $l \models c.i \wedge I = d.i$, while $l' \models c.i \wedge \neg I = d'.i$. Therefore, the states $d.i$ and $d'.i$, refining the states $c.i$ and $c'.i$ respectively, are different and therefore not i -indistinguishable. By considering all $c' \neq c$ such that $c'.i = c.i$, $((M, c') \models^3 \theta) = \text{ff}$, and $l \models c.i$, $l' \models c'.i$ for some $l \neq l'$, we can eventually decide the truth value of failure formula $K_i\varphi$. This terminates the procedure for K -formulas.

Since Algorithm 2 returns an updated list of predicates, the abstraction M^{A} built on it is also an abstraction of the concrete IS M . Hence, Theorem 6 applies, and therefore formulas defined in M^{A} are preserved in M . Moreover, the initial abstraction M^A can be thought of as an abstraction of M^{A} as well, where state c' abstracts state c iff $c'.i \models c.i$, for every $i \in Ag$. We summarise these remarks in the next immediate result.

Theorem 14 *Given an abstraction M^A of an IS M , the refinement M^{A} is also an abstraction of M and it is abstracted by M^A . Thus, the refinement procedure defines a sequence M, \dots, M^{A}, M^A of IS such that any element in the sequence is an abstraction of its predecessors.*

Algorithm 2 does not address the case of atomic propositions that were also identified in Lemma 11 as possible components of failure pairs. Observe that if (c, p) is indeed a failure pair, for p atomic, then p refers to more than one agent. This follows immediately from the fact that the list of predicates for an agent i contains all atoms referring to i itself. Hence, the truth value of such atoms is always immediately defined in the abstraction. As a consequence, an agent-based refinement procedure cannot be given for atomic predicates, as their satisfaction cannot be established by evaluating local states only. This limitation does not appear to be significant as in most cases of interest we expect to be able to resolve the value of the specification of interest by refining the temporal and epistemic transitions. Observe that any abstraction procedure is in any case incomplete as the verification problem is undecidable in general.

Furthermore, note that the complexity of the refinement procedure is determined by lines 6, 7, 12, and 13 in Algorithm 2. These return the states satisfying a given constraint or interpolants. Both these problems can be reduced to solving linear inequalities; therefore the

complexity of the procedure is bounded by the complexity of linear programming.

We conclude by remarking that, since M can be an infinite-state IS and the refinement procedure adds finitely-many states only, the sequence M, \dots, M^{A}, M^A in Theorem 14 is infinite in principle. Hence, as in other predicate abstraction approaches, the refinement procedure is not guaranteed to terminate. Nonetheless, in many cases of interest the methodology can resolve the truth value of specifications that cannot be evaluated on the initial abstraction. We consider one such case in the following section.

5 Verification of an Infinite-state English Auction Protocol

We now illustrate the methodology presented in the paper on an example of an ascending English auction [10]. Consider an auction with an auctioneer A and a finite number of bidders B_1, \dots, B_n . Each bidder B_i has a fixed amount of resources (e.g., money) $m_i > 2$; they follow a protocol of the form “if the latest bid was less than m_i , then non-deterministically decide to bid or not; otherwise do nothing”. We assume that the protocols that the agents run are commonly known. All the bidders and the auctioneer have an integer value to store the latest highest bid. We assume that bids start from 0 and each bid increases the previous bid by 1.

We would like to establish whether it is a common knowledge that the auction terminates and whether the bidders have a strategy to buy the item for 1 resource in two rounds.

We formalise this auction as an infinite-state interpreted system M . Each bidder B_i has a variable lb of type integer representing the latest bid, initially set to 0, and two actions: bid and $skip$. The protocol of B_i is such that: $P_i(lb) = \{bid, skip\}$ when $lb \leq m_i$; $P_i(lb) = \{skip\}$ when $lb > m_i$. We can further encode that the transition function t_i is such that lb remains unchanged if bidder B_i uses the action $skip$; lb is increased by 1 otherwise.

We model the auctioneer A by considering an integer variable lb initially set to 0, a variable $top_bidder \in \{0, \dots, n\}$ initially set to 0 and a boolean variable $sold$, initially set to false, as well as the actions $skip$ and $sold_i$, where $i \in \{0, \dots, n\}$. The transition function for A is such that when all the bidders perform the action $skip$, the variable $sold$ is set to true. At that time the auction is over and the winner is announced; from that point onwards A loops in the same state announcing winner i using action $sold_i$. If any of the bidders uses the action bid , then the top_bidder is chosen non-deterministically among all the bidders who bid. The variable lb is updated as in the case of bidders. We assume a labelling function with atoms $A.sold$ and $A.lb = 1$ depending on A 's local variables.

We investigate the properties specified above by evaluating the formulas

$$\varphi = C_{\{B_1, \dots, B_n\}} \langle \langle \emptyset \rangle \rangle F(A.sold)$$

$$\rho = \langle \langle \{B_1, \dots, B_n\} \rangle \rangle X \langle \langle \{B_1, \dots, B_n\} \rangle \rangle X (A.sold \wedge A.lb = 1)$$

on the infinite-state interpreted system described above.

By conducting the initial abstraction as in [22] we obtain a model M^A based on the predicates $sold$, $top_bidder = 0$, and $lb = 0$ for A (from the definition of the initial state) and the predicates $lb = 0$, $lb < m_i$ for agent B_i (the first from B_i 's initial state, the second from B_i 's protocol).

The may and must transition relations for auctioneer A in M^A coincide. Specifically, from the initial state denoted by $\neg sold \wedge (top_bidder = 0) \wedge (lb = 0)$, if all bidders perform skip, then the next state is $\neg sold \wedge (top_bidder = 0) \wedge (lb = 0)$; otherwise it

becomes $\neg sold \wedge (top_bidder \neq 0) \wedge (lb \neq 0)$. From the latter state, A loops whenever at least one bidder bids, or moves to the state $sold \wedge (top_bidder \neq 0) \wedge (lb \neq 0)$ otherwise, where it can only loop.

The initial state for agent B_i is $(lb = 0) \wedge (lb < m_i)$, where B_i stays if all bidders skip, or moves to $(lb \neq 0) \wedge (lb < m_i)$ otherwise (recall that $m_i > 2$). These are both may and must transitions. From the latter state, B_i has no must transition, but has two may transitions: a loop and a transition to $(lb \neq 0) \wedge (lb \not< m_i)$ over any action where one of the agents bids. In $(\neg lb = 0) \wedge (lb \not< m_i)$, B_i loops for both the may and must transitions.

It can be checked that the initial abstraction built on these predicates is such that ρ is evaluated to true; it follows that ρ is satisfied in the infinite state system. However, φ is undefined in the initial abstraction. We now show how the refinement procedure introduced in this paper enables us to determine the truth value of φ .

By using algorithm *FRFP* from Section 3, we obtain the failure pair $(\neg sold \wedge (top_bidder \neq 0) \wedge (lb \neq 0), (c_1, \dots, c_n), \langle\langle\emptyset\rangle\rangle X(A.sold))$, where for each i , $c_i = (lb \neq 0) \wedge (lb < m_i)$. This enables us to apply the refinement procedure given in Algorithm 2 for the case of X -formulas.

We illustrate this by considering bidder 1 with $m_1 = 10$. Consider a transition from $c = (lb \neq 0) \wedge (lb < 10)$ to $c' = (lb \neq 0) \wedge (lb \not< 10)$, which is the result of abstracting the transition encoded by the condition $lb' = lb + 1$. Let l be the state $lb = 1$, and l' be the state $lb = 10$. Clearly, $l \models c$ and $l' \models c'$. Therefore, we find an interpolant for $l \wedge lb' = lb + 1$ and l' . We can derive a refutation as shown in Figure 2 (see [29]).

$$\frac{\frac{lb = 1}{0 \leq -lb + 1} \text{ LE} \quad \frac{0 = -lb' + lb + 1}{0 \leq -lb' + lb + 1} \text{ LE}}{0 \leq -lb' + 2} \quad \frac{0 = lb' - 10}{0 \leq lb' - 10} \text{ LE}}{0 \leq -8} \text{ C}$$

Figure 2. Rule LE allows to derive disequalities from equalities, while C returns $0 \leq t + t'$ from the premises $0 \leq t$ and $0 \leq t'$.

From the refutation of Figure 2 we can obtain an interpolant [5], simply by setting all disequalities in branches for ψ' to $0 \leq 0$, as shown in Figure 3.

$$\frac{\frac{lb = 1}{0 \leq -lb + 1} \text{ LE} \quad \frac{0 = -lb' + lb + 1}{0 \leq -lb' + lb + 1} \text{ LE}}{0 \leq -lb' + 2} \quad \frac{0 = 0}{0 \leq 0} \text{ LE}}{0 \leq -lb' + 2} \text{ C}$$

Figure 3. Obtaining an interpolant from a refutation.

By doing so, we obtain the formula $lb' \leq 2$, which is indeed an interpolant for the pair (ψ, ψ') , and can be used in the refinement procedure. Therefore, the revised list of predicates for B_i is $[lb = 0, lb \leq 2, lb < m_i]$.

This refinement step results in an abstraction that is still insufficient to decide the value of φ . However, by conducting a number of further refinement steps for B_1 bounded by m_1 , we may derive the list of predicates that fully characterise all the possible values of lb below m_1 . When this is done for all the bidders, the abstract interpreted system is such that the states of bidder B_i correspond to numbers $0, \dots, m_i$. On such a system, it can be checked that the

property φ holds, and therefore it is satisfied in the original infinite-state system.

6 Conclusions

Little attention has so far been devoted to the practical verification of infinite-state MAS. A key requirement of any predicate abstraction technique is not only the initial generation of the predicates, but also their refinement to produce a sequence of abstractions approximating the concrete system. As we discussed in Section 1, present approaches for MAS against ATL specifications fall short in this respect.

In this paper we have put forward a refinement methodology for MAS abstractions to be verified against ATLK specifications. The proposed approach uses state-of-the-art automatic deduction techniques based on interpolants via SMT calls, as pioneered by [29] in the context of purely temporal logic. We showed that the method is sound and illustrated its potential on an infinite state MAS implementation of a simple auction protocol.

A noteworthy feature of the approach lies in the choice and development of both the semantics and the specification language, which are both oriented towards MAS. In terms of semantics we use and extended interpreted systems, that have long been used as a formal model to reason about MAS. In particular, as discussed in the Introduction and differently from [3], we here adopt incomplete information and memoryless strategies. In terms of specifications we follow our previous work in this line [20, 21, 22] by combining strategic concepts given in a weaker form of ATL with an epistemic language. The branching-time temporal-epistemic logic CTLK is entirely subsumed in the approach should this be found to be preferable in some applications.

A further aspect of the work concerns its potential applicability. The choice of adopting non-uniform strategies keeps the decision problem against explicit models in PTIME, which is important in practical verification. It is well known that this comes at the cost of expressivity and it is reflected in the reading of the ATL modalities.

In future work, we intend to implement the technique and algorithms introduced here. We anticipate this will be challenging given the complexity of devising efficient heuristics resolving the non-determinism of some of the refinement steps here described. Since the abstraction methodology is necessarily incomplete, this will also require a considerable amount of tuning of the heuristics against several benchmarks, so that any resulting tool offers the concrete possibility of solving actual MAS programs.

Acknowledgements

This research was funded by the EPSRC under grant EP/I00520X.

REFERENCES

- [1] T. Ågotnes, V. Goranko, W. Jamroga, and M. Wooldridge, ‘Knowledge and ability’, in *Handbook of Logics for Knowledge and Belief*, College Publications, (2015).
- [2] R. Alur, T. A. Henzinger, and O. Kupferman, ‘Alternating-time temporal logic’, *Journal of the ACM*, **49**(5), 672–713, (2002).
- [3] 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).
- [4] 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).

- [5] A. Cimatti, A. Griggio, and R. Sebastiani, 'Efficient generation of Craig interpolants in satisfiability modulo theories', *ACM Transactions of Computational Logic*, **12**(1), 7:1–7:54, (2010).
- [6] E. M. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith, 'Counterexample-guided abstraction refinement', in *Proceedings of the 12th International Conference on Computer Aided Verification (CAV00)*, volume 1855 of *Lecture Notes in Computer Science*, pp. 154–169. Springer, (2000).
- [7] E. M. Clarke, O. Grumberg, and D. Long, 'Model checking and abstractions', *ACM Transactions on Programming Languages and Systems*, **16**(5), 1512–1542, (1994).
- [8] 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).
- [9] S. Das, D. Dill, and S. Park, 'Experience with predicate abstraction', in *Proceedings of the 11th International Conference on Computer Aided Verification (CAV99)*, pp. 160–171. Springer, (1999).
- [10] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*, Cambridge University Press, New York, NY, USA, 2010.
- [11] E. A. Emerson and E. M. Clarke, 'Using branching-time temporal logic to synthesize synchronization skeletons', *Science of Computer Programming*, **2**(3), 241–266, (1982).
- [12] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi, *Reasoning about Knowledge*, MIT Press, Cambridge, 1995.
- [13] P. Gammie and R. van der Meyden, 'MCK: Model checking the logic of knowledge', in *Proceedings of 16th International Conference on Computer Aided Verification (CAV04)*, volume 3114 of *Lecture Notes in Computer Science*, pp. 479–483. Springer, (2004).
- [14] 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).
- [15] W. van der Hoek and M. Wooldridge, 'Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications', *Studia Logica*, **75**(1), 125–157, (2003).
- [16] 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).
- [17] W. Jamroga and W. van der Hoek, 'Agents that know how to play', *Fundamenta Informaticae*, **62**, 1–35, (2004).
- [18] G. Jonker, *Feasible Strategies in Alternating-time Temporal Epistemic Logic*, Master's thesis, University of Utrecht, The Netherlands, 2003.
- [19] S. Lahiri, R. Nieuwenhuis, and A. Oliveras, 'Smt techniques for fast predicate abstraction', in *Proceedings of the 18th International Conference on Computer Aided Verification (CAV06)*, pp. 424–437. Springer, (2006).
- [20] A. Lomuscio and J. Michaliszyn, 'An abstraction technique for the verification of multi-agent systems against ATL specifications', in *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR14)*, pp. 428–437. AAAI Press, (2014).
- [21] A. Lomuscio and J. Michaliszyn, 'Verifying multi-agent systems by model checking three-valued abstractions', in *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS15)*, pp. 189–198, (2015).
- [22] 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).
- [23] A. Lomuscio, W. Penczek, and H. Qu, 'Partial order reduction for model checking interleaved multi-agent systems', *Fundamenta Informaticae*, **101**(1–2), 71–90, (2010).
- [24] A. Lomuscio, H. Qu, and F. Raimondi, 'MCMAS: A model checker for the verification of multi-agent systems', in *Proceedings of the 21th International Conference on Computer Aided Verification (CAV09)*, volume 5643 of *Lecture Notes in Computer Science*, pp. 682–688. Springer, (2009).
- [25] 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>.
- [26] A. Lomuscio and F. Raimondi, 'The complexity of model checking concurrent programs against CTLK specifications', in *DALT*, volume 4327 of *Lecture Notes in Computer Science*, pp. 29–42. Springer, (2006).
- [27] A. Lomuscio and F. Raimondi, 'Model checking knowledge, strategies, and games in multi-agent systems', in *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS06)*, pp. 161–168. ACM Press, (2006).
- [28] K. L. McMillan, 'Interpolation and sat-based model checking', in *Proceedings of the 15th International Conference on Computer Aided Verification (CAV03)*, pp. 1–13, (2003).
- [29] K. L. McMillan, 'An interpolating theorem prover', *Theoretical Computer Science*, **345**(1), 101–121, (2005).
- [30] J.-J. Ch. Meyer and W. van der Hoek, *Epistemic Logic for AI and Computer Science*, volume 41 of *Cambridge Tracts in Theoretical Computer Science*, Cambridge University Press, 1995.
- [31] W. Penczek and A. Lomuscio, 'Verifying epistemic properties of multi-agent systems via bounded model checking', *Fundamenta Informaticae*, **55**(2), 167–185, (2003).
- [32] S. Shoham and O. Grumberg, 'Monotonic abstraction-refinement for CTL', in *Proceedings of the 10th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS04)*, volume 2988 of *Lecture Notes in Computer Science*, pp. 546–560. Springer, (2004).

Combining Deterministic and Nondeterministic Search for Optimal Journey Planning Under Uncertainty

Akihiro Kishimoto, Adi Botea and Elizabeth Daly¹

Abstract. Optimal multi-modal journey planning under uncertainty is a challenging problem, due in part to an increased branching factor generated by nondeterministic actions. Deterministic search, which ignores all uncertainty, can be much faster, but deterministic plans lack correctness and optimality guarantees in the uncertainty-aware domain.

We present a novel approach that combines the strengths of both deterministic and nondeterministic search in order to achieve superior performance. Initially, an A* search is used checking whether the resulting deterministic plan remains correct and optimal under uncertainty. When the plan is invalid, a backpropagation step through the A*'s search graph improves the initial heuristic while preserving its admissibility. After the backpropagation, an AO* search is run with the new improved heuristic. A theoretical analysis proves that our approach is sound and optimal. Our backpropagation correctly handles a subtle issue arising in the presence of state-dominance pruning. This supports the use of these two powerful speedup techniques in combination, for a better overall performance.

We empirically evaluate our solution in multi-modal journey planning under uncertainty, with realistic data from three European cities. Our results show that our approach brings a significant performance improvement over a state-of-the-art, highly optimized journey planning engine based on AO* search.

1 Introduction

Both domain-independent and dependent-specific planning have been extensively studied for decades in the AI research community. Despite recent improvements in domain-independent planning, domain-specific planning is still necessary, due to a strong demand for efficiency. Additionally, ideas developed in domain-specific planning can often be generalized and transferred to domain-independent planning. Conversely, domain-independent planning algorithms can be specialized into high-performance domain-specific methods.

Multi-modal journey planning has garnered increased attention in recent years, for several reasons. First off, this is driven by an increasing practical need, as multi-modal travel can potentially contribute to reducing pollution, congestion and carbon emissions. Secondly, an increasing availability of input data, such as public transport schedules, and smart phone technology facilitate the development and the use of journey planning systems.

Traditional multi-modal journey planning systems are deterministic, implicitly assuming that the input data is accurate which makes journey planning relatively straightforward. However, in real life, a transport network can have a great deal of uncertainty, such as significant differences between published schedules and the actual arrival

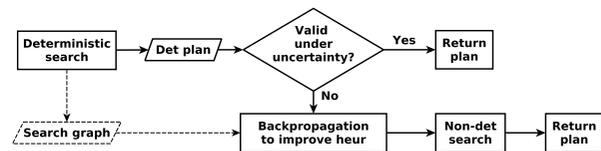


Figure 1. Architecture of our approach.

and departure times of buses, trams or trains due to unforeseen delays. This can result in missed connections, which can in turn cause other segments in the journey to become unrealistic. Recent work has investigated multi-modal journey planning under uncertainty [8, 29], as an approach to provide more reliable journey plans [7].

Given the high volume and time dependent nature of such an application, the speed performance for optimal multi-modal journey planning is extremely important. As a result, the current rational choice of practitioners is the use of domain-specific planning specialized to journey planning. The need for speeding up optimal multi-modal journey planning is particularly important when uncertainty is modeled as part of the problem, for two reasons. First off, deterministic multi-modal journey planning has benefited from more research efforts than the uncertainty-aware problem, being thus a more mature field (see e.g., [3]). Secondly, planning under uncertainty is inherently more computationally difficult.

We present an optimal approach to journey planning under uncertainty, that combines the strengths of deterministic and nondeterministic search, as illustrated in Figure 1. A deterministic solver, based on A* [13] search with pruning enhancements, provides a deterministic plan. This is followed by a so-called validity test, to decide whether the deterministic plan remains correct and optimal in the nondeterministic domain. If the test is positive, the nondeterministic problem has been solved with a standard deterministic search, which is typically faster than nondeterministic search. On the other hand, when the validity test fails, a backpropagation step through the A*'s search graph computes new (improved) heuristic estimations for the states visited in the search. Then, a nondeterministic search, based on AO* [27, 28] search with pruning enhancements, is run using the more accurate heuristic. The new heuristic improves the performance due to the knowledge gathered during the original deterministic search.

The high-level idea summarized in the previous paragraph and illustrated in Figure 1 is domain-independent. However, we develop it and study its use in an application-specific context, as our motivation was advancing the state-of-the-art in multi-modal journey

¹ IBM Research, Ireland

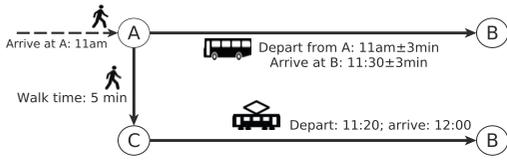


Figure 2. Toy example.

planning under uncertainty. Creating components such as an effective backpropagation technique is not a trivial task. In addition, combining our contributions with a state-of-the-art AO*-based solver introduces a number of challenges to address. Part of the reason is that the highly-optimized A* and AO* search engines use pruning rules, including state-dominance pruning [8]. As discussed in the paper, there are important differences between deterministic and nondeterministic search regarding the conditions under which state-dominance pruning can safely be applied. Transferring informed knowledge from a deterministic domain relaxation to the nondeterministic domain, as done with our backpropagation algorithm, needs a special attention, to handle a subtle issue that may otherwise affect the optimality.

We give a theoretical analysis which proves that our approach is sound and optimal. The proposed solution makes it feasible to harness two powerful speedup techniques (i.e., our approach and dominance pruning) in tandem, for a better overall performance.

We use Botea et al.’s domain modeling [8], a state-of-the-art approach to multi-modal journey planning under uncertainty. As discussed later, the domain is represented as a state space that allows nondeterministic transitions. A plan is a probabilistic contingent plan (or policy in MDP terms). Figure 2 shows a toy problem. A user arrives (e.g., with another connection) at a point *A* by 11am. There are two options to continue to the destination *B*: take a fast bus directly; or walk to a nearby tram stop *C*, and take a tram from there. The first option is faster but, due to uncertainty in the bus departure time, it may or it may not be possible to catch the bus route. As such, the take-bus action has two possible nondeterministic outcomes (branches): “success” and “missed bus” (failure). In general, the probability of each possible outcome is computed from the probabilistic times of the user arrival and the bus departure [8]. In our example, the optimal contingent plan is as follows: At location *A*, attempt to take the bus. If the bus is missed, walk to the tram stop *C* and take the tram.

We experimentally evaluate our proposed solution demonstrating the contribution with realistic multi-modal transportation data from three European cities, a testbed reused from the literature [7]. Our approach achieves significantly improved performance, compared to a state-of-the-art approach based on AO* search and on search enhancements such as those introduced by Botea et al. [8].

2 Related Work

We start with an overview of research in nondeterministic planning, followed by surveying previous research in deterministic search.

2.1 Nondeterministic Planning

Variations of planning under uncertainty include conformant planning [11, 32], contingent planning [30], and probabilistic contingent planning, with differences stemming from the assumptions on the

properties of the actions and the observability (sensing capabilities). See e.g., [4] for a discussion.

In conformant planning, actions can be nondeterministic, the initial state is not fully known, and no sensing is available when executing the plan. Thus, a conformant plan should guarantee reaching the goal under such conditions. In contingent planning, sensing allows to detect the resulting state when a nondeterministic action is applied. Plans can have a tree structure, to ensure that the goal will be reached from any potential nondeterministic successor. Probabilistic contingent planning allows probabilistic actions and sensing. As Bonet and Geffner remark [4], probabilistic contingent planning can generally be modeled as Partially Observable Markov Decision Processes (POMDPs). However, when full observability (full sensing) is assumed, we fall into the framework of MDPs.

While approaches such as AO* and dynamic programming can return optimal plans (policies), computing optimal plans is challenging due to the potentially many states that need to be processed during planning. For instance, decision-theoretic algorithms based on dynamic programming can suffer from large memory requirements, that can be exponential in the domain feature size [2]. Adding heuristic enhancements to AO*, such as an informed admissible heuristic or effective pruning rules can reduce the size of the state space explored in a search for an optimal plan, which is the approach taken by Botea et al. [8]. Hansen and Zilberstein present LAO* [12], an optimal algorithm based on AO* that is capable of finding solutions with loops, and apply LAO* to solve MDPs. We note that the search space of our multi-modal journey planning has no loops, one main reason being that a state includes the time among its components.

Due to the computational difficulty of optimal nondeterministic planning, research has often focused on suboptimal approaches, for a better speed and scalability. Hoffmann and Brafman [18] use Weighted AO*. Bonet and Geffner take an approach based on a greedy, real-time action selection [4, 5, 6]. One of their heuristics, called the min-min state relaxation, is related to our approach of using the results of a deterministic relaxation to obtain an admissible heuristic in the nondeterministic domain.

Some approaches rely on using deterministic planning as part of solving a nondeterministic planning problem [22, 23, 35, 24, 21, 34, 25, 26]. For example, Kuter et al. [24] run multiple deterministic planning rounds, with a deterministic relaxation of the original domain, to gradually add branches to a contingent plan under construction. FF-Replan [35] invokes a deterministic planner for the initial state and for the states in the plan where unexpected outcomes are observed in the nondeterministic scenario.

Thus, the high-level idea of using deterministic planning as part of the process to solve a nondeterministic planning problem appears in both previous work and our work. There are important differences, however, stemming in part from the fact that we ensure solution optimality, whereas such previous work does not. We run exactly one deterministic search. If the result is a correct and optimal nondeterministic plan, we are done. Otherwise, we improve the available heuristic function with the backpropagation and run full-scale AO*. To our knowledge, our approach is the first to combine deterministic and nondeterministic search in this way with evidence that this approach solves realistic, difficult journey planning problems fast and optimally.

2.2 Deterministic Search

Using a deterministic relaxation of a nondeterministic problem to compute an admissible heuristic can be used as a form of domain

abstraction. Using state-space abstractions to build an admissible heuristic is broadly used in planning and heuristic search. See e.g., Hierarchical A* (HA*) [19], pattern databases [10], and merge-and-shrink abstractions [14, 15]. Among these, HA* is more closely related to our method. HA* builds a hierarchy of abstractions, with the lowest level being the original state space. A* search in one abstract space can be used to build an admissible heuristic for the lower abstraction level. HA* features caching techniques to reduce duplicate search effort across multiple levels in the hierarchy.

We incorporate Holte et al.’s P-g heuristic [19]. In HA*, P-g is the heuristic $h(n) = P - g(n)$, where P is an optimal solution cost in the next abstraction level, and $g(n)$ is the g -cost in the next abstraction level of the abstract node corresponding to n . There are multiple notable differences between our approach and HA*. First, we consider domains with multiple types of “consumable resources”, such as the a maximum walking time and a maximum number of legs in a trip. This makes state dominance an important pruning mechanism. Ensuring the correctness and optimality of a hybrid A*–AO* approach, especially in the presence of state dominance, is a non-trivial contribution. Secondly, unlike HA*, our approach backpropagates information through the graph search of A*, for a stronger improvement of the heuristic. Furthermore, our approach runs at most two searches, one deterministic and one nondeterministic. That is, our approach defines exactly two “hierarchical levels”, and employs only one search in the “abstracted” (i.e., deterministic) state space. The number of searches can be larger in HA*, due to a potentially larger number of levels, as well as potentially multiple searches at a given abstraction level. In HA* all search spaces are deterministic, whereas we report a contingent planning system.

Incremental heuristic search aims at efficiently reusing previous search results to solve “similar” new problems. It has mainly been applied to real-time path-planning with dynamic domains. There are two common principles for reusing information in incremental search. One is to start a new A* search with the open and closed lists adapted from the previous A* search [33, 20]. The idea is effective when small changes are observed in the problem from one search to another. On the other hand, in our setting, differences between deterministic and nondeterministic search are more fundamental. In particular, the two types of problems require very different search algorithms, such as A* and AO*. The second common principle in incremental search is to make a heuristic more informed based on previous search results [16, 17]. Caching techniques used for this purpose are similar to those featured in HA*, reviewed earlier.

3 Preliminaries

We start with a description of the multi-modal journey planning problem. Then we discuss how the problem is converted into a nondeterministic planning problem. We give definitions, optimization criteria and assumptions needed in the formal analysis. **ND** and **D** refer to the nondeterministic and deterministic domains, respectively.

3.1 Input Data in Multi-modal Journey Planning

A multi-modal journey planner takes as input a user request (“the instance”) and a network transport snapshot (“the domain”).

The request includes the origin, the destination, the departure or the arrival time,² and the transport modes acceptable in the journey at hand. It also includes so-called *quotas*, which are max acceptable

² In this work we focus on scenarios where the user request specifies a departure time rather than an arrival time.

values for the walking time, the cycling time, and the number of legs (segments) in the trip.

The network snapshot contains information available about a multi-modal transportation network and includes the following data:

- *Relevant locations* include public-transport stops, bike stations in a city’s shared-bike network along with the location of the origin and destination of the request. All relevant locations have their lat and lon coordinates specified, besides their type (e.g., stop or bike station), name and id.
- A *public-transport route* can be represented as an ordered sequence of stops. Each route is served by several *trips* during the day. A trip has an arrival and departure time associated with every stop along the route. Traditionally, these times are deterministic. However, we allow a more general representation, where a departure or arrival time is a probability distribution.
- *Road map data* is represented as a graph with nodes and segments. Each segment is labeled to reflect its direction (e.g., one-way or bi-directional), access to cyclists, access to pedestrians, access to cars and driving speed.

Next we discuss how such input data is converted into a nondeterministic state space.

3.2 Nondeterministic Domain Modeling

We adopt the formalization introduced by Botea et al. [8] for multi-modal journey planning under uncertainty. For a self-contained presentation, we summarize how the problem is formalized here as a nondeterministic state space, defining states and transitions.

A state s is a tuple $(p_s, t_s, q_s, \alpha_s)$ where:

- p_s is the position of the traveler, which is either a relevant location on the map, or the id of a trip, for those states where the user is aboard a trip (e.g., a bus);
- t_s is a probability distribution of the time when the user has reached position p_s ;
- q_s is a vector which lists quotas left in this state (e.g., max walking time and max number of legs in the rest of the trip);
- α_s is a subset of additional variables which are skipped for brevity.

The initial state s_0 is constructed with the components $p_{s_0}, t_{s_0}, q_{s_0}$ taken from the user request, representing the origin, the departure time and the quotas acceptable for the entire trip respectively. A valid state s is not allowed to have negative values on any component of the quota vector q_s . A state s_G is a goal state if it is valid and p_{s_G} corresponds to the destination specified in the user request. Depending on the user preferences, other conditions may be added to the goal state, such as not having a hired bike in possession in a goal state (i.e., if the user hires a bike, the bike should be eventually returned to a bike station). For clarity, we assume that this condition is part of the goal definition as well.

It is common in nondeterministic planning research to group regular states together into a *belief state*, using for instance decision diagrams to represent belief states. This choice seen in related work is motivated by two reasons: to cover domains with partial observability, where a belief state could contain all possible current states; and to mitigate state explosion, obtaining a more compact state space definition. Botea et al.’s modeling [8], which we adopt in this research, works with regular, not belief states, one of the reasons being that states are assumed to be fully observable during the plan execution.

Transitions include Walk, TakeTrip (i.e., boarding a public transport vehicle), GetOffTrip, HireBike, Cycle and

ParkBike. Except for TakeTrip actions, all transitions are deterministic, in the sense that they always succeed and thus each of them has exactly one successor. As illustrated earlier in Figure 2, a TakeTrip action can succeed or fail, depending on the possibly uncertain arrival times of the vehicle and the traveler. Botea et al. show how to compute the probability of each outcome [8]. Obviously, when the probability of success is 1, the TakeTrip action at hand becomes deterministic, with only the successful branch defined. When the probability is 0, no TakeTrip action is defined for that particular trip in that particular state.

On the successful branch, the successor state s of a TakeTrip action has the position p_s set to the id of the trip just boarded. The time t_s is the (stochastic) departure time of the trip from the stop at hand. On the failed branch, the position and the time remain as in the parent state. However, in the successor state, a new flag is set to true, indicating that the trip at hand has just been missed. As such, the parent and the successor state are different, and therefore the failed branch is not a self-loop transition. This property, together with the fact that the time is a state component, ensures that there are no cycles in this state space. In the search, the space is modeled as a tree, rather than a directed acyclic graph.

For brevity, we skip discussing how other transitions generate their successor states. Such details are not difficult to infer based on the discussion provided, and the interested reader can refer to the original work [8].

A state s dominates a state s' , i.e., $s \prec s'$, iff $p_s = p_{s'}$, $q_s \geq q_{s'}$ component-wise, and $P(T_s \leq T_{s'}) = 1$. T_s is the random variable with the density function t_s . In other words, the position is the same, and one state is no worse than the other in terms of available quotas and time. The relation is not symmetric, apart from the obvious exception that every state dominates itself.

$A_{nd}(s)$ is the set of all actions applicable to a state s in **ND**. For a state s and an action $a \in A_{nd}(s)$, $B(s, a)$ is the set of all possible branches. I.e., $|B(s, a)| = 1$ for deterministic actions, and $|B(s, a)| = 2$ for nondeterministic actions with just two outcomes (i.e., succeed and fail). Each branch b uniquely determines a transition to a successor state. It has a probability p_b as illustrated earlier, a cost $c_b(s, a)$ associated with the transition,³ and a successor state $\gamma_b(s, a)$. As the deterministic action has a unique branch, in their case the notations get simplified into $c(s, a)$ and $\gamma(s, a)$. Thus, deterministic actions are a particular type of successful branches.

3.3 Deterministic Domain

We obtain the deterministic domain as a relaxation from **ND**. Specifically, we convert nondeterministic actions into deterministic actions, by keeping only the successful branch. Actions that were deterministic in **ND** are preserved unchanged. $A_{det}(s)$ is the set of actions applicable to a state s in **D**. Clearly, $|A_{det}(s)| = |A_{nd}(s)|$.

This relaxation is optimistic in the sense that, even when the probability of the successful branch is small (but strictly positive), we still consider that as the only outcome of the action in **D**. Such an optimistic approach is one of the conditions that we need to ensure the optimality of plans in our approach, as discussed later in the paper. In particular, we will make use of the following straightforward observation.

Observation 1. *All successful branches in **ND** are available as deterministic actions in **D**.*

³ The cost can depend on both action a and state s , not just on action a . This allows for instance to integrate waiting into the cost of a transition from “arrived at a stop” to “boarded a bus”.

3.4 Optimality Criteria

The cost function $c(n, m)$ returns a non-negative value as a state transition cost from state n to state m . Additionally, $f(n)$, $g(n)$ and $h(n)$ represent an f -value, a g -value, and a h -value (or heuristic value) at state n , respectively. The f -value is defined as $f(n) = g(n) + h(n)$, where $g(n)$ is the sum of the transition costs from the initial state to reach n , and $h(n)$ is a value estimating a cost to reach the destination from n . In this work, the cost is set to the travel time.

The optimal cost $v_{det}(n)$ of a state n in **D**, the optimal expected cost $v_{exp}(n)$ in **ND**, and the optimal worst-case cost $v_{wst}(n)$ in **ND** are defined as:

1. If n is a goal state, $v_{det}(n) = v_{exp}(n) = v_{wst}(n) = 0$.
2. If $|A_{det}(n)| = 0$, $v_{det}(n) = v_{exp}(n) = v_{wst}(n) = \infty$.
3. Otherwise,
 - $v_{det}(n) = \min_{a \in A_{det}(n)} (c(n, a) + v_{det}(\gamma(n, a)))$
 - $v_{exp}(n) = \min_{a \in A_{nd}(n)} \sum_{b \in B(n, a)} p_b (c_b(n, a) + v_{exp}(\gamma_b(n, a)))$
 - $v_{wst}(n) = \min_{a \in A_{nd}(n)} \max_{b \in B(n, a)} (c_b(n, a) + v_{wst}(\gamma_b(n, a)))$.

Note that there are many goal states for one goal, since the time and quotas are part of the state definition.

We use Botea et al.’s objective function [8], which minimizes the worst case, and break ties in favor of better expected costs (i.e., take (v_{wst}, v_{exp}) in the lexicographic order):

Definition 1. *The cost in **ND** of a plan π is the pair $v_{opt}(\pi) = v_{opt}(r) = (v_{wst}(r), v_{exp}(r))$, where r is the root node of the plan.*

We write $(p_1, p_2) \leq_{lex} (q_1, q_2)$ iff $(p_1 < q_1) \vee (p_1 = q_1 \wedge p_2 \leq q_2)$. The inequality is strict when the two pairs are not identical. We can use other objectives (e.g., swap the lexicographic order [7]) with minor changes, but this is beyond our focus.

We list a few additional facts relevant to our analysis:

Proposition 1 ([8]). *In an optimal plan, the cost when following the failed branch of an action cannot beat the cost along the successful branch.*

Observation 2 ([8]). *Let a be an action with two nondeterministic outcomes in a correct plan π . Cutting action a from π , plus the entire subtree along the successful branch, results in a correct plan.*

The example shown in Figure 2 helps see the intuition behind these two claims. The part of the plan under the failed branch of the take-bus action is “walk to C and then take the tram”, as these are the actions taken after failing to catch the bus. Regarding Proposition 1, if this part of the plan under the failed branch had a better cost (i.e., better arrival time) than the bus trip, there would be no point to even attempt to take the bus. Regarding Observation 2, indeed, we can eliminate the attempt to take the bus, together with the part under the successful branch (i.e., ride the bus to B), and still obtain a perfectly valid (but not necessarily optimal) plan. This plan would be “...arrive at A , walk to C and take the tram to B .” More formally, this stems from the fact that we compute *strong (acyclic) plans* [9], meaning every pathway in a contingent plan ends up at the destination. See the original work for more formal proofs of these two claims.

Assumption 1. *We assume that the initial heuristic available is admissible in **D**.*

In particular, this holds for Botea et al.’s heuristic [8], which we reuse in experiments as the initial heuristic available.

4 Cost and Heuristic Relations in D and ND

We present a few results that draw a connection between deterministic and nondeterministic search. Bonet and Geffner have observed similar properties in their work on MDPs and probabilistic planning [4, 6]. The discussion presented in this section is important to ensure the optimality of our hybrid approach.

Theorem 1. For any state n , $v_{det}(n) \leq v_{exp}(n) \leq v_{wst}(n)$.

Proof Sketch (for $v_{det}(n) \leq v_{exp}(n)$). Let π be a plan in **ND**, rooted at n , that is optimal on v_{exp} (i.e., $v_{exp}(\pi) = v_{exp}(n)$). Let p be a smallest-cost pathway in π . Consider all nondeterministic actions having their failed branch within p . Remove these actions from π , together with the subtree under their successful branch, obtaining a plan π' that is correct in **ND**, cf. Observation 2. In π' , p has lost zero or more failed branches (let p' be the new pathway). As p' has only successful branches, it is a correct plan in **D**, cf. Observation 1. Clearly, $v_{det}(p') \leq v_{exp}(\pi) = v_{exp}(n)$, since p was a best-cost pathway in π . Furthermore, $v_{det}(n) \leq v_{det}(p')$, since the former is the optimal value in **D**, and the latter is the cost of one correct plan in **D** (which may be optimal or not). \square

Theorem 2. Any $h(n)$ admissible in **D** is admissible in **ND**.

Proof Sketch Let $h_{nd}(n) = (h(n), h(n))$ and $v_{opt}(n) = (v_{wst}(n), v_{exp}(n))$. It follows from Theorem 1 that $h_{nd}(n) = (h(n), h(n)) \leq_{lex} (v_{det}(n), v_{det}(n)) \leq_{lex} (v_{wst}(n), v_{exp}(n)) = v_{opt}(n)$. \square

Corollary 1. The perfect heuristic in **D** $h^*(n) = v_{det}(n)$ is admissible in **ND**.

These results show that we can use an optimal cost in **D** or a lower-bound to admissibly guide the search in **ND**. We emphasize that the previous cost definitions are specific to a given goal condition (i.e., they are initialized to 0 at goal states), but the initial state is irrelevant. As such, the results derived in this section are specific to a goal condition, but make no assumption about the initial state.

5 Hybrid Approach

Algorithm 1 Hybrid Deterministic Nondeterministic Search

Require: Initial state $root$

- 1: $O = C = H = \phi$
- 2: $(v_{det}, \pi_1) = A^*(root, O, C, h)$
- 3: **if** (all actions in π_1 are deterministic in **ND**) **then**
- 4: **return** π_1
- 5: **else**
- 6: Update($root, O, C, H, v_{det}$)
- 7: **return** AO*($root, \max(H, h)$)

We now introduce our new algorithm, whose main steps are illustrated in Algorithm 1 and Figure 1. Our approach first runs A* with an open list O and a closed list C , returning an optimal plan π_1 in **D**, as well as its cost v_{det} . Unlike standard A*, our A* takes into account the state dominance (see Algorithm 2). Assume that A* generates a successor s and detects that there is a state u in $O \cup C$ such that $u \prec s$. Then, A* discards s , since both $v_{det}(u) \leq v_{det}(s)$ and $g(u) < g(s)$ always hold in the deterministic scenario. That is, A*

does not explore further the search space rooted at s . This plays a crucial role in significantly reducing A*'s search space.

Similarly to A*, AO* performs pruning with state dominance. However, Botea et al. [8] have pointed out that correctly applying state dominance pruning in a nondeterministic search requires additional conditions to satisfy, and they presented sufficient conditions for this purpose. Our AO* implementation observes these.

Algorithm 2 A* with state dominance pruning

Require: State n , open list O , closed list C and consistent heuristic function h

- 1: Enqueue($O, n, h(n)$)
- 2: **while** $O \neq \emptyset$ **do**
- 3: $t = \text{Dequeue}(O)$
- 4: **if** t is a goal **then**
- 5: **return** $(f(t), \text{path}(n, t))$
- 6: Save(t, C)
- 7: **for each** of t 's successors s **do**
- 8: **if** $s \notin C \wedge$ no state $u \in C \cup O$ dominates s **then**
- 9: Enqueue($O, s, g(t) + c(t, s) + h(s)$)
- 10: **return** (∞, \emptyset)

Theorem 3. Consider a deterministic plan π_1 computed in the deterministic search. If all π_1 's actions correspond to a deterministic action in **ND** (as opposed to a being the relaxation of a nondeterministic action in **ND**), then π_1 is correct and optimal in **ND**.

Proof Sketch The correctness follows from the fact that actions, being deterministic in **ND**, will behave as in the deterministic domain, being applicable in the corresponding state and succeeding with probability 1. For optimality, assume there is a nondeterministic plan π_2 with a better score $v_{opt}(\pi_2) <_{lex} v_{opt}(\pi_1)$. Let r_i be the root of π_i , $i = 1, 2$. Applying Theorem 1 to r_2 , we obtain $v_{det}(r_2) \leq v_{exp}(r_2) \leq v_{wst}(r_2)$. At the same time, $v_{det}(r_1) = v_{exp}(r_1) = v_{wst}(r_1)$, since π_1 is linear and thus has no multiple branches. As both π_1 and π_2 are valid plans in **D**, and π_1 is optimal in **D**, it follows that $v_{det}(r_1) \leq v_{det}(r_2)$. Putting all these together, it follows that $v_{opt}(r_1) = (v_{wst}(r_1), v_{exp}(r_1)) = (v_{det}(r_1), v_{det}(r_1)) \leq_{lex} (v_{det}(r_2), v_{det}(r_2)) \leq_{lex} (v_{wst}(r_2), v_{exp}(r_2)) = v_{opt}(r_2)$, which is a contradiction.

The previous result is important because it formalizes the validation step in our approach (line 3 of Algorithm 1). When the validity test fails, our approach updates the heuristic, after which it runs AO*.

The Update method shown in Algorithm 3 implements the back-propagation idea, computing more informed h-values that are stored in a table H . As in related previous work [31, 1], Update traverses the search graph backwards and sets $H(n) = \min_{s_i \in C(n)} (f(s_i) - g(n))$, where $C(n)$ is the set of not-expanded frontier states reachable from n . However, unlike previous approaches, Update does not perform iterative deepening, and its depth-first search is limited to the threshold initialized to v_{det} at the initial state (see Algorithm 1), and to the set of states examined by A*.

As mentioned, both A* and AO* apply dominance pruning. To ensure the plan optimality, our Update procedure correctly handles a subtle but important detail stemming from the use of dominance pruning. Assume a new state s is dominated by an older state $u \in O \cup C$. Both A* and Update skip examining s due to the pruning scheme with state dominance (see line 8 in Algorithm 2 and line 5 in Algorithm 3). However, while regarding s as a deadend in **D**

Algorithm 3 Update

Require: State n , open list O , closed list C , hash table H , and threshold θ

- 1: $r = h(n)$
- 2: **if** $n \in H$ **then**
- 3: $t = \text{GetValue}(n, H)$
- 4: $r = \max(r, t)$
- 5: **if** $\theta \geq r \wedge n$ is not a goal $\wedge n \notin O \wedge$ no state $m \in O \cup C$ dominates n **then**
- 6: /* Improve heuristic values of successors */
- 7: $v = \infty$
- 8: **for each of** n 's successor s **do**
- 9: $v = \min(v, c(n, s) + \text{Update}(s, O, C, H, \theta - c(n, s)))$
- 10: /* Increase r based on an improved heur. value v */
- 11: $r = \max(r, v)$
- 12: /* Increase r based on the optimal solution cost θ */
- 13: $r = \max(r, \theta)$
- 14: /* Save an improved heuristic value */
- 15: Save(n, r, H)
- 16: **return** r

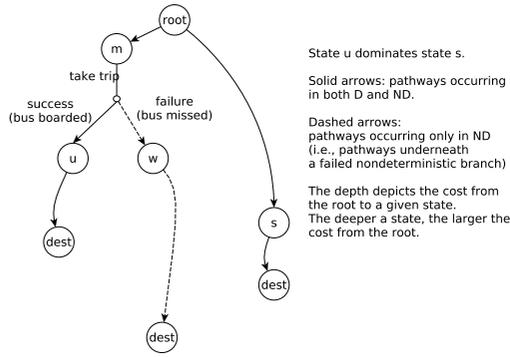


Figure 3. A simple example where a dominated state in **D** belongs to an optimal plan in **ND**

is correct in **A***, setting $H(s) = \infty$ in Update would be an error. Instead, in such cases we set $H(s) = h(s)$, where h is the initial heuristic (line 1 in Algorithm 3). We illustrate why aggressively setting $H(s) = \infty$ in Update would be a mistake with the following example.

Example 1. Figure 3 illustrates a scenario where a state s dominated in **D** belongs to an optimal plan in **ND**. Assume that u dominates s . Recall that in **D** pathways starting with a failed branch are not generated. Thus, in **D** (solid arrows only), there are two plans: the sequential pathway through u and the sequential pathway through s . The one containing u is an optimal plan, and s can safely be pruned in **D** as a state dominated by u .

In **ND** (solid plus dashed arrows), there are two plans: the contingent plan with a branching point under m , and the sequential pathway through s . Here, the latter is an optimal plan, as we assume that the nondeterministic branch in the former plan has a very large cost.

Therefore, setting $H(s) = \infty$ in Update would result in ignoring the plan through s in **ND** and returning a suboptimal solution. To bypass this, our Update method propagates back a conservative admissible heuristic value $h(s)$ for s .

Line 13 in Algorithm 3 is related to Holte et al.'s P-g heuristic [19],

discussed in the related work section. Backpropagation (without line 13) provides better estimates than P-g (line 13 alone) in many cases. Interestingly, while experiments presented later show that P-g alone is not effective in this domain, we found that P-g is helpful in combination with our backpropagation. Specifically, we found line 13 to be particularly useful in states pruned away with state dominance. As such pruned states are not expanded further, their exploration subtree is empty. On the other hand, backpropagation (without line 13) works by increasing the heuristic of a state based on the heuristic of its children. Clearly, when a state has no children (as it happens for states pruned away), backpropagation cannot increase the heuristic of that state. In such cases the P-g rule (line 13) is the only mechanism that can improve the heuristic, which of course can trigger further improvements up in the tree.

We argue that the P-g heuristic preserves the admissibility of H in **D** as follows: 1) When a state n is processed in Update, we have $\theta = v_{det} - g(n)$. This follows easily from the initialization of θ to v_{det} (line 6 in Algorithm 1), and from the way it is updated in each recursive call (line 9 in Algorithm 3); 2) It is known that, at the end of an **A*** search that returns an optimal cost v_{det} , $v_{det} - g(n)$ is an admissible heuristic for every state n with an optimal g-cost.

Our heuristic update strategy implements a few additional ideas. As shown in Algorithm 1, we take the $\max(h(n), H(n))$ as the heuristic to use in **AO***. The reason is that, for those states n not visited in **A***, $H(n)$ returns 0. Secondly, when $u = (l, t_u, \dots)$ dominates $s = (l, t_s, \dots)$ in **D**, $t_s + H(s) = f(s) \geq f(u) = t_u + H(u)$, which further allows us to increase $H(s)$ to $H(u) + t_u - t_s$ within method GetValue in Algorithm 3. Thirdly, we added to **AO*** (both as a module of our method and as a standalone benchmark) an extra heuristic update rule. Let s_s and s_f be the successful and the failed successors of a nondeterministic action. The heuristic of s_f can admissibly be increased to the heuristic of s_s , a property that follows from Proposition 1.

Theorem 4. The hybrid method returns optimal solutions in **ND**.

Proof Sketch It is sufficient to prove $H(n) \leq v_{det}(n), \forall n$. Then H is admissible in **ND** cf. Theorem 2. With no generality loss, assume that $H(n) = h(n)$ if n is not found in H . Here we only show that $H(n) = \min_{a \in A_{det}(n)} (c(n, a) + H(\gamma(n, a))) \leq v_{det}(n)$. Other rules (e.g., increasing $H(n)$ to θ) were discussed earlier. The proof is related to Akagi et al.'s work [1]. Let H_k be the table H after k updates. If $k = 0$, the theorem holds, as nothing new is saved in H , and $h(n) \leq v_{det}(n)$ cf. Assumption 1. Assume that the result holds for k (i.e., $H_k(s) \leq v_{det}(s)$), and we are about to save a new result for node n at step $k + 1$. We have: $H_{k+1}(n) = \min_{a \in A_{det}(n)} (c(n, a) + H_k(\gamma(n, a))) \leq \min_{a \in A_{det}(n)} (c(n, a) + v_{det}(\gamma(n, a))) = v_{det}(n)$. \square

6 Experimental Evaluation

We implemented our ideas on top of DIJA [8], a state-of-the-art engine for multi-modal journey planning under uncertainty. It is a highly-optimized, **AO***-based planning engine, implementing techniques described by Botea et al. [8]. For a fair comparison, the **AO*** algorithm is the same in both DIJA and our approach, except for the heuristic used. As described earlier in the paper, our approach first attempts to solve the problem with deterministic **A*** search. If the resulting plan is not valid in the nondeterministic domain, a backpropagation step provides an improved heuristic to the **AO*** engine.

The optimality criterion is minimizing the worst-case travel time, with ties broken in favor of a better expected travel time. Botea et

al. [7] handle a linear combination between the travel time and the number of legs. However, optimizing purely on the travel time is the most computationally challenging scenario, which is why we focus on improving the performance in this case.

We have used testbed data from the literature [7]. This is transportation data from three European cities, Montpellier, Dublin, and Rome. The Dublin data contain 4,739 stops, 120 routes, and 7,308 trips per day. The road network has 301,638 nodes and 319,846 segments. In Montpellier, bus and tram data amount to 1,297 stops, 36 routes and 3,988 trips per day. The road network has 152,949 nodes and 161,768 links. In Rome, buses, trams, subways and light trains sum up to 391 routes, with 8,896 stops and 39,422 trips per day. The road map contains 522,529 nodes and 566,400 segments.

The original data is deterministic. This was extended with a stochastic noise assigned to the original deterministic arrival and departure times. The noise follows a Normal distribution, truncated to a confidence interval of 99.7%. We report results with two distinct levels of noise. The smaller level has $\sigma^2 = 1600$ seconds, equal roughly to ± 2 minutes around the original deterministic arrival or departure times. In the larger noise level we set $\sigma^2 = 6400$, which roughly corresponds to a ± 4 -minute uncertainty interval.

We used 1,000 journey plan requests (instances) for each city. The origins and the destinations are picked at random, and the departure time is 11AM. Quotas are set to at most 20 minutes of walking, and at most 5 segments per trip. Combined with two levels of noise and two solvers, this sums up to $3,000 \times 2 \times 2 = 12,000$ runs in total.

Figure 4 shows the CPU time in our method, denoted by “Hybrid”, compared to the benchmark state-of-the-art approach DIJA. Dots under the main diagonal are cases where our method is faster, and dots above the diagonal show the opposite. The three parallel lines correspond to the main diagonal $y = x$, $y = 3x$ (above the main diagonal) and $y = \frac{1}{3}x$ (below the main diagonal). These will help better understand the results.

A main conclusion from Figure 4 is that, when our method is faster, it can be faster by a large margin, especially in solving difficult instances. The speedup can significantly exceed one order of magnitude and, in a few cases, two orders of magnitude. On the other hand, when our method is slower, its slowdown is bounded by a factor of 3 in most but not all cases (see the $y = 3x$ line above the main diagonal line). We call this the *asymmetric speedup behavior*.

This (not strict) bounding factor of 3 is present because, in our method, each run invokes at most three time-consuming operations: the A^* search (“Hybrid A^* ”), the backpropagation, and the AO^* search (“Hybrid AO^* ”). Each of these is typically smaller than the standalone AO^* search (“DIJA AO^* ”), as follows. The A^* search has a smaller space to explore, being a deterministic search. Figure 5 (left) shows differences between A^* and AO^* search, both with the initial heuristic in use. The backpropagation traverses A^* ’s search space, being thus comparable with A^* as CPU time. The AO^* of our method is typically faster than the DIJA AO^* , thanks to the better heuristic obtained through the backpropagation phase. Figure 5 (middle) shows the impact of the improved heuristic on AO^* search. The impact of backpropagation will be discussed in more detail later in this section.

Table 1 complements the observations drawn from Figure 4 with summary statistics. For each σ^2 level, Part A reports the total CPU time ratio between DIJA and our method. Our method is consistently better on this metric, as all values reported are greater than 1. This is an important result, showing that the new approach is faster in average on all 6 combinations of a city and a noise level. The average speed is important, for instance, in a server implementation, where

Table 1. Summary statistics. Easy instances, requiring less than 1 second with the baseline approach, are skipped.

City		Mon	Rom	Dub
Uncertainty level: $\sigma^2 = 1600$				
A	CPU time ratio DIJA/Hybrid	5.97	1.07	1.71
B	Max speedup factor Hybrid	166.01	48.77	101.81
	Max speedup factor DIJA	1.51	1.44	1.74
C	Det plan valid %	58.10	27.70	35.80
D	Instances Hybrid faster %	66.66	40.32	49.26
Uncertainty level: $\sigma^2 = 6400$				
A	CPU time ratio DIJA/Hybrid	2.14	1.26	1.18
B	Max speedup factor Hybrid	269.56	82.18	158.07
	Max speedup factor DIJA	5.12	3.28	1.54
C	Det plan valid %	39.70	10.40	15.40
D	Instances Hybrid faster %	57.6	65.85	25.26

a journey planning engine has to answer several queries in a short amount of time. Part B shows the maximum speedup of each system, showing a consistent advantage in favor of the new hybrid approach. Part C reports the percentage of instances where the deterministic plan is valid under uncertainty, and Part D presents the percentage of instances where our method is faster (i.e., dots below the main diagonal in Figure 4). Due to the asymmetric speedup behavior, our method is faster in terms of average solving time (Part A) even in those cases where Part D shows a value lower than 50%.

Savings in speed come from both the ability to avoid backpropagation and AO^* search, when the deterministic search is sufficient, and from the better informed AO^* heuristic when the deterministic search is not sufficient. Note that, given a σ^2 level, Part D shows substantially higher percentages than Part C, confirming that our method is faster substantially more often than just the cases when our method stops after A^* .

We have also evaluated a partial version of Hybrid, with backpropagation turned off. In other words, when the deterministic plan is invalid in the nondeterministic domain, the heuristic is updated using only the P-g rule, without any backpropagation. Comparing the middle and the rightmost charts in Figure 5 convincingly illustrates the benefits of backpropagation.

In the rightmost chart, we observe that, in our problems, the P-g rule alone does not change the AO^* performance significantly. In particular, this implies that, for the subset of instances where A^* alone is not sufficient, a hybrid approach without backpropagation would consistently be slower than a pure AO^* solver. Indeed, in such cases, the system has the additional overhead of running A^* .

On the other hand, the full heuristic update method, with both P-g and backpropagation switched on, performs significantly better than the original heuristic, as shown in the middle chart in Figure 5. Even when the system has to perform A^* , backpropagation and AO^* , the speedups are obtained in AO^* due to a better-informed heuristic offsets and even exceed the overhead of the A^* and backpropagation steps on average. In particular, this explains why values in Part D are higher than values in Part C in Table 1. Thus, backpropagation plays a significant role in the performance of our system.

7 Conclusions

We presented an approach to nondeterministic planning combining A^* and AO^* search. We focused on multi-modal journey planning under uncertainty, an application domain where speeding up optimal solving approaches is an important task. Our theoretical analysis shows that our approach creates optimal plans even in the presence of

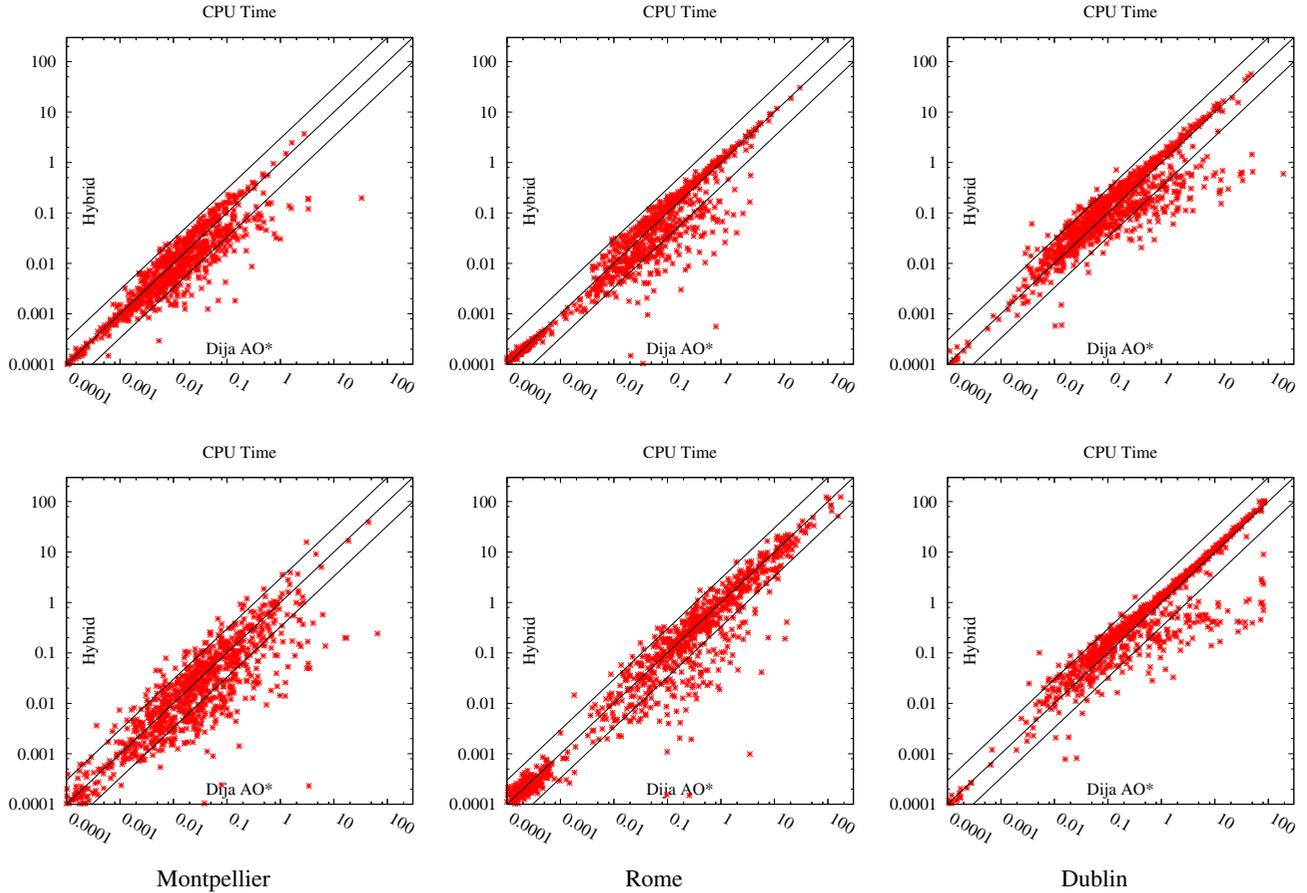


Figure 4. CPU time in our method and DIJA AO* on a logarithmic scale. Top row: $\sigma^2 = 1600$; bottom row: $\sigma^2 = 6400$.

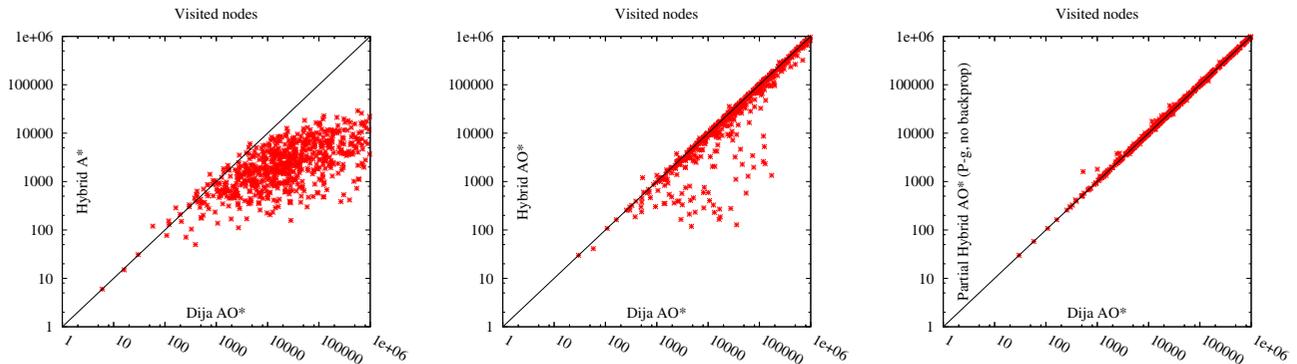


Figure 5. Left: A* in our method vs DIJA AO*; middle: AO* in our method vs DIJA AO*; right: AO* with a partial heuristic update strategy vs AO* with the original heuristic. Data shown for Rome with $\sigma^2 = 6400$. Other combinations (city, noise level) lead to a similar conclusion and are skipped to save room.

dominance relations between states. Empirical results in multi-modal journey planning under uncertainty demonstrate that our approach brings a significant performance improvement over a state-of-the-art journey planner based on AO*.

Assumptions such as modeling nondeterminism with successful and failed attempts are not limited to journey planning. We plan to

develop our method in domain-independent planning, especially in domains with state dominance (e.g., planning under uncertainty with consumable resources). Another interesting direction is to further improve the heuristic function with backpropagation, extending the initial A* search with localized additional explorations.

REFERENCES

- [1] Y. Akagi, A. Kishimoto, and A. Fukunaga, 'On transposition tables for single-agent search and planning: Summary of results', in *Proceedings of the 3rd Symposium on Combinatorial Search (SOCS)*, pp. 1–8, (2010).
- [2] A. G. Barto, S. J. Bradtko, and S. P. Singh, 'Learning to act using real-time dynamic programming', *Artificial Intelligence*, **72**(1-2), 81–138, (1995).
- [3] Hannah Bast, Erik Carlsson, Arno Eigenwillig, Robert Geisberger, Chris Harrelson, Veselin Raychev, and Fabien Viger, 'Fast routing in very large public transportation networks using transfer patterns', in *Algorithms - ESA 2010, 18th Annual European Symposium. Proceedings, Part I*, pp. 290–301, (2010).
- [4] B. Bonet and H. Geffner, 'Planning with incomplete information as heuristic search in belief space', in *Proceedings of the 5th International Conference on Artificial Intelligence Planning Systems*, pp. 52–61, (2000).
- [5] B. Bonet and H. Geffner, 'Faster heuristic search algorithms for planning with uncertainty and full feedback', in *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1233–1238, (2003).
- [6] B. Bonet and H. Geffner, 'mGPT: A probabilistic planner based on heuristic search', *Journal of Artificial Intelligence Research*, **24**, 933–944, (2005).
- [7] A. Botea and S. Braghin, 'Contingent versus deterministic plans in multi-modal journey planning', in *Proceedings of the 25th International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 268–272, (2015).
- [8] A. Botea, E. Nikolova, and M. Berlingerio, 'Multi-modal journey planning in the presence of uncertainty', in *Proceedings of the 23rd International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 20–28, (2013).
- [9] A. Cimatti, M. Pistore, M. Roveri, and P. Traverso, 'Weak, strong, and strong cyclic planning via symbolic model checking', *Artificial Intelligence*, **147**(12), 35 – 84, (2003). Planning with Uncertainty and Incomplete Information.
- [10] J. Culberson and J. Schaeffer, 'Pattern databases', *Computational Intelligence*, **14**(3), 318–334, (1998).
- [11] Robert P. Goldman and Mark S. Boddy, 'Expressive planning and explicit knowledge', in *Proceedings of the 3rd International Conference on Artificial Intelligence Planning Systems (AIPS)*, pp. 110–117, (1996).
- [12] E. A. Hansen and S. Zilberstein, 'LAO*: A heuristic search algorithm that finds solutions with loops', *Artificial Intelligence*, **129**, 35–62, (2001).
- [13] P. E. Hart, N. J. Nilsson, and B. Raphael, 'A formal basis for the heuristic determination of minimum cost paths', *IEEE Transactions on Systems Science and Cybernetics*, **4**(2), 100–107, (1968).
- [14] M. Helmert, P. Haslum, and J. Hoffmann, 'Flexible abstraction heuristics for optimal sequential planning', in *Proceedings of the 17th International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 176–183, (2007).
- [15] M. Helmert, P. Haslum, J. Hoffmann, and R. Nissim, 'Merge-and-shrink abstraction: A method for generating lower bounds in factored state spaces', *Journal of the ACM*, **61**(3), (2014). Article 16.
- [16] C. Hernández, P. Meseguer, X. Sun, and S. Koenig, 'Path-adaptive A* for incremental heuristic search in unknown terrain', in *Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 358–361, (2009).
- [17] C. Hernández, X. Sun, S. Koenig, and P. Meseguer, 'Tree adaptive A*', in *Proceedings of the 10th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 123–130, (2011).
- [18] J. Hoffmann and R. Brafman, 'Contingent planning via heuristic forward search with implicit belief states', in *Proceedings of the 15th International Conference on Automated Planning and Scheduling (ICAPS-05)*, eds., Susanne Biundo, Karen Myers, and Kanna Rajan, pp. 71–80, Monterey, CA, USA, (2005).
- [19] R.C. Holte, M.B. Perez, R.M. Zimmer, and A. J. MacDonald, 'Hierarchical A*: Searching abstraction hierarchies efficiently', Technical report, Department of Computer Science, University of Ottawa, (1995).
- [20] S. Koenig, M. Likhachev, and D. Furcy, 'Lifelong planning A*', *Artificial Intelligence*, **155**(1-2), 93–146, (2004).
- [21] A. Kolobov, Mausam, and D. S. Weld, 'ReTrASE: Integrating paradigms for approximate probabilistic planning', in *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJ-CAI)*, pp. 1746–1753, (2009).
- [22] J. Kurien, P. Nayak, and D. Smith, 'Fragment-based conformant planning', in *Proceedings of the 6th International Conference on Artificial Intelligence Planning Systems (AIPS)*, pp. 153–162, (2002).
- [23] U. Kuter and D. S. Nau, 'Forward-chaining planning in nondeterministic domains', in *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI)*, pp. 513–518, (2004).
- [24] U. Kuter, D. S. Nau, E. Reissner, and R. P. Goldman, 'Using classical planners to solve nondeterministic planning problems.', in *Proceedings of the 18th International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 190–197, (2008).
- [25] C. Muise, S. A. McIlraith, and C. Beck, 'Improved non-deterministic planning by exploiting state relevance', in *Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 172–180, (2012).
- [26] H.-K. Nguyen, D.-V. Tran, T. C. Son, and E. Pontelli, 'On computing conformant plans using classical planners: A generate-and-complete approach.', in *Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 190–198, (2012).
- [27] N. J. Nilsson, 'Searching problem-solving and game-playing trees for minimal cost solutions.', in *IFIP Congress (2)*, pp. 1556–1562, (1968).
- [28] N. J. Nilsson, *Principles of Artificial Intelligence*, Tioga Publishing Co, Palo Alto, CA, 1980.
- [29] T. Nonner, 'Polynomial-time approximation schemes for shortest path with alternatives', in *Proceedings of the European Symposium on Algorithms, ESA*, pp. 755–765, (2012).
- [30] M. A. Peot and D. E. Smith, 'Conditional nonlinear planning', in *Proceedings of the 1st International Conference on Artificial Intelligence Planning Systems (AIPS)*, pp. 189–197, (1992).
- [31] A. Reinefeld and T. A. Marsland, 'Enhanced iterative-deepening search', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **16**(7), 701–710, (1994).
- [32] D. E. Smith and D. S. Weld, 'Conformant Graphplan', in *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI)*, pp. 889–896, (1998).
- [33] A. Stentz, 'The focussed D* algorithm for real-time replanning', in *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1652–1659, (1995).
- [34] F. Teichteil-Königsbuch, T. Cedex, and F. U. Kuter, 'Incremental plan aggregation for generating policies in MDPs', in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 1231–1238, (2010).
- [35] S. Yoon, A. Fern, and R. Givan, 'FF-Replan: A baseline for probabilistic planning', in *Proceedings of the 17th International Conference on Automated Planning and Scheduling (ICAPS)*, pp. 352–359, (2007).

Robust Real-Time Human Perception with Depth Camera

Guyue Zhang¹, Luchao Tian¹, Ye Liu², Jun Liu¹, Xiang An Liu¹, Yang Liu¹ and Yan Qiu Chen^{1*}

Abstract. Perception of the presence and position of human is crucial for many kinds of Artificial Intelligence (AI) applications. In this paper, we have developed a novel two-staged method for real-time human detection in depth image. The first stage is to quickly scan through the image to detect possible head-top locations in order to ensure all the candidate locations are included. The second stage is to use a novel head-shoulder descriptor (HSD) which jointly encodes the One-hot Depth Difference information and local geometric characteristics of human upper body to filter the detections so as to keep the genuine human locations and discard false positives. The results show that our approach using only depth data is superior to other methods using color and depth images on four datasets. In addition, our method performs well under weak illumination conditions or even total darkness. Moreover, our system is also able to run in real-time on conventional PC without GPU acceleration.

1 INTRODUCTION

Human detection is an important task due to its wide application in human-computer interaction, intelligent vehicles, autonomous indoor mobile robots, etc. It is also a critical technology in building smart rooms in which intellectual sensors should be aware of users' presence and locations [7, 8, 9]. However, human detection is still a challenging problem especially in occasion of occlusion, posture variations, dynamic and heavily cluttered background or crowd, etc.

Existing methods [6, 24, 28] for conventional video cameras are reported as able to work in well-illuminated environments with relatively simple and stationary background. However, their performance declines quickly if the illumination conditions deteriorate or the background becomes dynamic and complicated.

With the recent rapid development of depth cameras, such as time-of-flight camera and Kinect, human detection becomes more manageable as depth image is relatively insensitive to scene textures, and the depth information acquired by the sensors using actively emitted near infra-red medium is robust against illumination variation of the environment.

There have been methods for detecting human beings with depth cameras [2, 1, 31, 15, 23, 27, 32, 26]. The work reported in [4] adopts a graph-based segmentation algorithm combined with randomized subsampling for depth image segmentation and a set of parameterized heuristics to reduce candidate segments for classification. Wojek et al. [29] combine a full object detector and multiple object part detectors in a mixture of experts based on their expected visibility.

Spinello et al. [22] take inspiration from HOG (Histogram of Oriented Gradients) detector which is mainly for color/grayscale image and design HOD (Histogram of Oriented Depths) descriptor for depth data, and then achieve promising human detection result. These methods mentioned above using full-body detectors show their effectiveness in many environments, but encounter challenges in crowded environments where occlusions occur frequently and people are often partially visible.

An upper-body detector is a good choice for robust human detection, since the upper part of human body is less likely to be occluded and less deformable. The approach proposed by Xia et al. [30] combines a 2D head contour model and a 3D head surface model to detect people in indoor environments. Ikemura et al. [12] introduce the notion of Relational Depth Similarity Features (RDSF) based on depth information, which is derived from a similarity of depth histograms and represents the relationship between two local regions. The method presented in [13] uses a continuous normalized-depth template as an upper-body detector for close range and a full-body detector for farther range. Choi et al.'s system [5] integrates multi-hypothesis (including human upper-body shape, human face, human skin, as well as human motion) and shows interesting results for locating people in 3D space. Liu et al. [18] combine a Ring-wedge Mask (RWM) and 2D Joint Histogram of Color and Height (JHCH) information to classify plausible human head candidates. Munaro et al. [19] combines depth-based and color-based techniques in a cascade algorithm to detect people.

In this paper, we propose a novel two-staged approach which fully utilizes the unique characteristics of the human's upper body from depth images only. We first quickly localize all candidate head-tops based on the contour information of human head. Excessive numbers of possible human head candidates are extracted in this stage, which contain true human head regions in the scene (with very low miss rate) as well as false positives. Although the false positive rate is still relative high, we are able to reduce the search space at a very low computational cost. These false positives in the detections from the first stage are further filtered out in the second stage by training a classifier with an effective upper-body head-shoulder descriptor (HSD) which consists of a One-hot Depth Difference Descriptor (ODDD) to describe the occlusion relationship among humans and their surrounding environments, and a Binarized Local Surface Descriptor (BLSD) to characterize the local surface geometrical properties of humans.

Our contributions are in the following aspects:

- We propose a fast head-top candidate extractor by localizing extreme points along the depth discontinuities and try to ensure all true human head-tops are included, which reduces searching space to obtain high speed.
- We propose a novel upper-body head-shoulder descriptor (HSD), which jointly encodes the information of One-hot depth differ-

¹ School of Computer Science, Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, China. Emails: {guyuezhang13, lctian14, ljun, xaliu13, yliud13, chenyyq}@fudan.edu.cn. * Corresponding author.

² College of Automation, Nanjing University of Posts and Telecommunications, China. Email: yeliu@njupt.edu.cn.

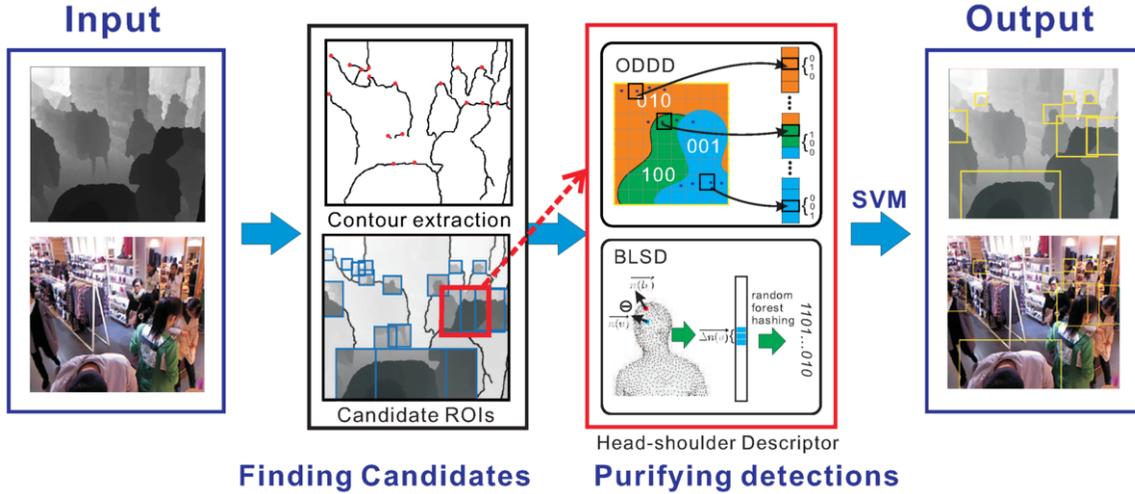


Figure 1: Workflow of the proposed method.

ence and local surface statistics to effectively and efficiently classify detections, aiming at further eliminating false positives while preserving the true detections. Both of the two descriptors are discriminative and compact, and they encode different kinds of information of the upper-body in depth images.

Experimental results have demonstrated the effectiveness of our approach on four available datasets. Although only depth data is employed, the proposed method outperforms existing methods using both depth and RGB data. Using only depth data is also advantageous in environments with dark or volatile illuminations.

2 THE PROPOSED APPROACH

Our method follows a two-stage cascaded structure. The first stage uses extreme points in edge map to detect candidate head-top locations and try to include all the probable locations. The second stage uses a novel head-shoulder descriptor (HSD) to verify the candidates so as to keep the genuine ones and discard false ones. We try to achieve a very low miss rate and expect to efficiently locate the candidates in the first stage, while the subsequent more computationally expensive verification stage only needs to deal with a limited number of candidates rather than on all image pixels. An overview of the proposed detection framework is given in Figure 1.

2.1 Finding Possible Head-top Points

Several existing works also tried to locate the head regions as ROI (Region of Interests) as a preliminary detection stage. For example, [30] and [13] use a depth template to localize the head positions as ROI to reduce the search space. However, the template matching often gets corrupted due to occlusions. [13] and [17] project the 3D point cloud to the ground plane and then use the height information to locate the probable head positions. But these methods require the prior knowledge of the ground plane which is either time consuming or even not possible to estimate.

Our motivation is to quickly and directly find possible head-tops in the depth image without the assist of any point clouds or depth templates, so that the more computationally intensive verification process needs to be applied to only a limited number of candidates rather than all pixels, thus substantially reducing the computation load. We

try to ensure all genuine head-top points are included in the responses while allowing some false positives. There are two successive modules in this stage: depth based contour extraction and head-top candidates localization.

2.1.1 Depth based Contour Extraction

Depth data remains continuous within the same object and varies greatly across distinct objects or parts of objects. So depth discontinuities usually indicate true boundaries between two non-touching objects. In real-world scenes, a standing person’s head is always sufficiently far away from its surrounding background. This inspires us to first extract the contour of human head based on depth discontinuities and then look for further cues in these contours. This cue makes it possible for us to obtain a set of less noisy contours corresponding to human head boundaries from depth data much easier than from RGB images.

Since depth data generated by depth camera may contain some noise and holes, we use a depth image inpainting technique [20] to reconstruct missing data, then the inpainted depth image is smoothed by a Gaussian filter. Gradient magnitude $M(x, y)$ and gradient orientation $\phi(x, y)$ are calculated with a Canny operator. Canny operator is employed as its outputs are not only isolated edge points but a set of contours with points linked which facilitates our further analysis of the contour. Then we locate every possible edge point by the non-maxima suppression (NMS) and extract contours by double-threshold edge linking scheme[3]. With conventional RGB image, the contours output by Canny can be noisy and fragmented, but as we have discussed above, with depth data we can always obtain clearer and more complete human head contours as shown in Figure 2(a).

2.1.2 Head-top Candidates Localization

In most scenarios, the camera is positioned to make the image y -axis inversely aligned with the gravity direction. Now that we have obtained relatively clean and complete contours of human heads, which protrude towards the ceiling of the image, we can find extreme points along the contours in which head-tops are contained as shown in Figure 2 (a). Given a contour consisted of a chain of points $C = \{(x_i, y_i)\}_{i=1}^n$, we define the extreme point l_k as the point which

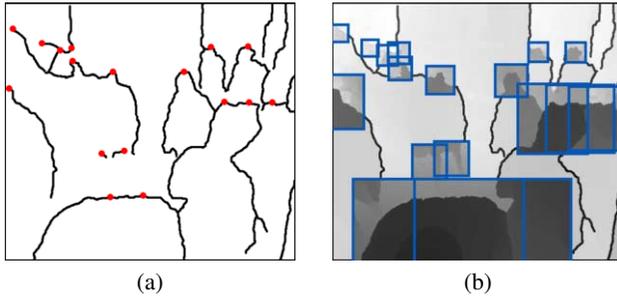


Figure 2: (a) shows an example of depth based contour extraction and extreme point localizations. The head-tops are in red color with contours in black. In (b), each head-top has its own corresponding ROI illustrated in blue box.

has a local maxima y value:

$$l_k = \{(x_k, y_k) \mid y_k \geq y_j, j \in N(k)\} \quad (1)$$

where $N(k) = \{k-s, \dots, k+s\}$ is a neighborhood of point (x_k, y_k) . We set $s = 3$ in all the experiments. Searching along all the contours extracted, we can obtain a set of extreme points as $L = \{l_k\}_{k=1}^m$, which are considered to be head-top candidates. The head-top candidates can be extracted ultra fast since that 1) we do not have to estimate the ground plane; 2) the extreme points are searched on the extracted contours rather than the whole image. Even in highly crowded environments, it makes sure that people’s head-top positions are included in the resultant responses L (see in Section 3.2).

It is a novel idea to extract the candidate points information from edge maps for subsequent processing, which dramatically reduces the searching space and gains high detection speed. It can be easily extended to other vision tasks in depth image, such as object recognition, human activity analysis, and hand gesture analysis.

2.2 Purifying the Detections

We have obtained the probable head-top detection candidates, but the results are over-detected and the false positives should be further filtered out. We train a classifier using a novel head-shoulder descriptor (HSD) combining two features: One-hot Depth Difference Descriptor (ODDD) and Binarized Local Surface Descriptor (BLSD).

2.2.1 Scale Invariant ROI Selection

Conventional object detection in RGB images often involves testing detection windows with different scales to detect objects with different sizes on image, which is time-consuming. With the help of depth information, we can select the ROIs adaptively according to the depth of a 3D point as shown in Figure 2 (b).

We denote an ROI as $r_k = (l_k, w_k, h_k)$ around the head-top point to cover the whole head. Here l_k is a head-top candidate, w_k and h_k are the width and height of selected ROI for l_k , they are adjusted adaptively according to the corresponding depth value d_k . We select the ROI whose left-top corner is $l_k - (0.25h_k, 0.5w_k)$. This is set empirically to make the ROI cover the upper-body and some context. We prefer slightly larger ROIs to diminish the effect of inaccuracy in localizing head-top points. Since human head is roughly spherical, we let ρ denote the head radius in physical quantity while ρ_{d_k} denote the projected radius and $w_k = 3\rho_{d_k}$, $h_k = 4\rho_{d_k}$. The relationship between physical quantity and projected quantity is $\rho_{d_k} = \lambda \frac{\rho}{d_k}$,

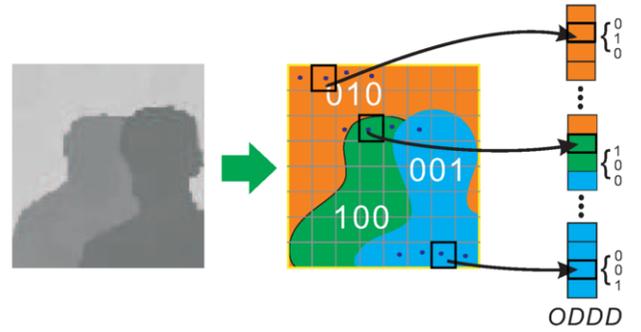


Figure 3: One-hot Depth Difference Descriptor (ODDD). We classify each pixel into three categories: detectee (100), background (010) and overlapper (001). The codes of picked points are concatenated to build a feature vector.

where λ is a constant factor obtained with camera’s intrinsic parameters [11, 18].

2.2.2 Head-shoulder Descriptor (HSD)

The pattern of human head-shoulder in depth images is distinctive from other objects, so the key problem here is to design a head-shoulder descriptor (HSD) to describe this distinctive property. We design an HSD that encodes two different aspects of information: the relative positions among human upper-body and their surrounding environments and the local surface geometrical characteristics of human upper-body. With these two kinds of information compactly encoded, the HSD is highly discriminative in keeping the true human locations and rejecting false positives.

One-hot Depth Difference Descriptor (ODDD) Depth difference has been explored in human pose estimation in [21]. But it is not suitable for describing the pattern of human head-shoulder due to complex relationships between humans and their surrounding background. Depth difference between human and background varies from less than $1m$ to more than $10m$, making the resulted feature badly scaled and may cause misleading classification results. If we directly utilize depth difference as a feature, it is possible to acquire misleading results or may influence the classification results significantly. So directly employing depth differences to detect human is not a wise choice.

In order to make depth difference more suitable for our classification task, we propose a novel One-hot Depth Difference Descriptor (ODDD). Inspired by [18], we classify each pixel in an ROI into three categories: *detectee* (pixels belonging to the region of human to be detected), *background* (pixels regarded as background) and *overlapper* (pixels that can be considered as objects that occlude the human to be detected) by depth difference values (with threshold σ). We assign a three-bit one-hot code for each category, so for each pixel $u = (x, y)$ the feature $f(u)$ is computed as (Figure 3):

$$f^d(u) = \begin{cases} 100, & |d(u) - d(l_k)| \leq \sigma \\ 010, & d(u) - d(l_k) > \sigma \\ 001, & d(u) - d(l_k) < -\sigma \end{cases} \quad (2)$$

We divide the ROI into $\alpha \times \beta$ cells, and to obtain fast computation speed, one point rather than all points is randomly picked from each cell, the one-hot code of the picked points in all the cells are concatenated to build a feature vector.

The feature extracted in above manner is able to shield the large variation of depth differences among pixels while retaining the occlu-

sion relationships (the category) between humans and their surrounding environments. Also, using one-hot code can facilitate processing for many classification methods. It should be noted that [18] also seek to threshold depth difference value, however, they used a hard template with strong prior knowledge to perform classification. Our work is significantly different as we turn the category information of pixels into features and train a classifier with them, which leverages the advantage of large amount of data to account for various kinds of occlusions and view changes.

Binarized Local Surface Descriptor (BLSD) One-hot Depth Difference Descriptor (ODDD) focuses on describing the complex depth pattern formed by humans and their surrounding environments, but it is ineffective in characterizing the local geometrical property of objects' surfaces, which has been proven to be important in RGBD object recognition tasks [25]. With depth information, the 2D pixels can be reprojected into 3D space as point cloud, which represent the 3D surfaces of the scene. For a pixel location $u = (x, y)$, the 3D surface normal vector can be approximated as $\vec{n}(u) = \left(\frac{\partial d(u)}{\partial x}, \frac{\partial d(u)}{\partial y}, -1 \right)^T$ [25]. Instead of building a histogram of normal vectors which loses the spatial information of points, we propose a binarized local surface descriptor (BLSD) which encodes the local surface smoothness in different spatial locations (Figure 4). The feature is also extracted from the ROI in Sec. 2.2.1.

For a pixel location $u = (x, y)$, we first compute the Normal Vector Difference (NVD):

$$f^n(u) = \overline{\Delta n(u)} = \overline{n(u)} - \overline{n(l_k)} \quad (3)$$

where $\overline{n(u)}$ and $\overline{n(l_k)}$ are normalized normal vectors at u and l_k . NVD is much more robust to view point change than normal vectors since the normal vector at head-top l_k is subtracted. And we also divide the ROI into cells, one point is randomly picked in each cell and its NVD is computed. Concatenating the NVDs of all the sampled points in all the cells will form a feature vector F_k which is highly redundant. Instead of using this feature vector for classification, we convert it to a binary code which is much more compact and can be more effectively processed.

We follow the approach of random forest based hashing to learn the compact binary codes [14]. A set of random forests $\{T_i\}_{i=1}^M$ is trained from the training data. Each $T_i = \{t_i^1, t_i^2, \dots, t_i^N\}$ is trained using a randomly selected subset of training data and a randomly selected subset of features of F_k , which serves as a binary test for generating one bit of the binary code (i.e. it generates 1 if F_k is classified as positive and 0 otherwise). In this manner, we are able to obtain a compact M bit BLSD which is combined with ODDD as our final binarized features for classification.

2.2.3 Training and classification

For training the classifier, we use 7,600 positive and 25,060 negative training samples. These samples are obtained in the following way: firstly detections are generated by the first stage of our method on 23 RGB-D video sequences, then 7,600 true human head-tops and 25,060 false positives are manual selected. Finally, ROIs are determined and from which HSDs are extracted.

In the classification procedure, we use a linear Support Vector Machine (SVM) to classify the Head-shoulder Descriptor (HSD) (by concatenating ODDD and BLSD) for an ROI to decide whether the region contains a human head or not.

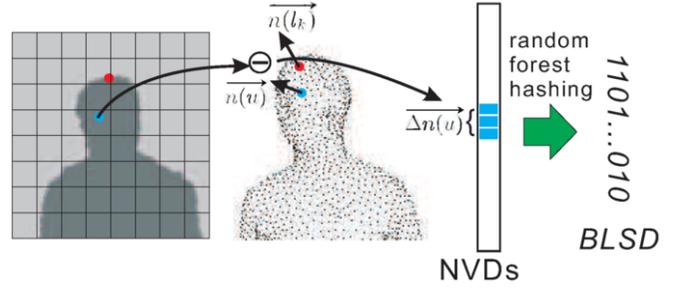


Figure 4: Binarized Local Surface Descriptor (BLSD). Head-top point l_k is in red color and randomly sampled points u are in blue color. Normal vector differences (NVDs) are computed and concatenated, and then are converted into a compact binary code by random forest hashing.

3 EXPERIMENTS AND DISCUSSIONS

We evaluate the detection accuracy and computational efficiency of the proposed method and compare it with other state-of-the-art approaches on four available datasets. These datasets are captured with Kinect at 640×480 resolution.

3.1 Datasets and Metrics for Evaluation

The first dataset is the CLOTHING STORE [16, 17]. It contains two video sequences of 45 minutes length each. The scene is cluttered with pillars, hangers, clothes, cabinets and shoe racks. People take on various poses such as walking, sitting and bending, and they interact with each other frequently.

The second and third datasets named OFFICE and MOBILE PLATFORM are provided by Choi et al. [5]. The former contains 17 video sequences and was captured in an office room. The environment is cluttered, and people in this dataset face different directions and take various poses, such as standing, walking, and sitting on chairs. The latter was collected with a Kinect mounted on a PR2 robot driving around in a building. It contains 18 video sequences with different scenes. This dataset includes various illumination conditions and cluttered backgrounds.

To the best of our knowledge, there is no publicly available RGB-D dataset captured under dark illumination for person detection. In order to comprehensively assess the performance of our method in such environments, we collected a challenging dataset named DARK. This dataset is captured at night with dark illumination which makes the persons indistinguishable from RGB images, as shown in Figure 7 (d). We will show that our method works well under such weak illumination or totally dark conditions. This new dataset is available at <http://www.cv.fudan.edu.cn/humandetection.htm>.

Also, we evaluate the performance via false-positive-per-image (FPPI) vs. miss-rate in our experiments. FPPI is computed in a standard way, as total number of false positives divided by frame numbers. Four images per second from OFFICE and MOBILE PLATFORM [5], one image every three seconds from CLOTHING STORE and DARK are selected to evaluate the performance [17]. A successful detection is counted if the overlap ratio between the annotated bounding box and the detected bounding box is above 0.5.

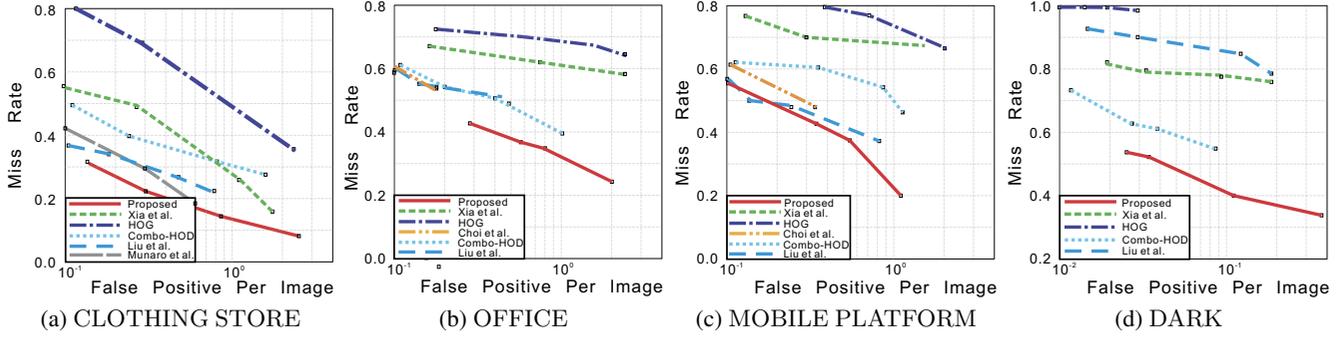


Figure 5: Comparison results with other approaches on four datasets CLOTHING STORE, OFFICE, MOBILE PLATFORM and DARK.

3.2 Analysis and Evaluation

We evaluate the detection accuracy and computational efficiency of the proposed method, with the overall system performance against other methods for a quantitative evaluation, and then we provide a comprehensive evaluations respectively on the computational times, the role of first stage, the contribution of each feature, and the effect of different distance.

Computational Times The proposed system was measured on a desktop PC with i5-2500 CPU and 8GB RAM, and runs at 40 fps without GPU acceleration, which is faster than the recording speed of Kinect (30 fps).

Overall system performance We compare the proposed system against a conventional HOG detector [6], a depth-based detector proposed by Xia et al. [30], a Combo-HOD detector [22], a color-depth detector proposed by Choi et al. [5], an RWM human locator [18], and cascade classifier proposed by Munaro et al. [19] on four datasets illustrated in Figure 5 (a)-(d). The results shown in Fig. 5 are obtained by using the codes from original authors [18, 19] and by our implementation [6, 22, 30]. The performance of [5] is only evaluated on OFFICE and MOBILE PLATFORM (the performance is reported by the authors), as the source code is not available.

The experiments show that our algorithm outperforms state-of-the-art detectors. In the results, the performance of HOG detector is limited due to the clutter of background and various people’s poses in color images. Xia et al.’s method uses a 2D head contour model and a 3D head surface model, which is strongly dependent on human shape, it may fail when in side-view cases. The Combo-HOD detector and Munaro et al.’s method work well in spacious environments, but the performance decreases in our test scenes where people are occluded and posing variedly such as sitting or bowing. The approach proposed by Choi et al. combines multiple cues based on color and depth data. But it may fail because the depth information is not fully exploited. The RWM locator has a strong assumption of the parameter in different categories so that it is easy to fail in divergent scenes of occlusion and tilt.

The proposed method using only depth information provides a more reliable result than the method HOG using RGB data only and the method of Xia et al. using depth only. Moreover, our method even yields higher accuracy than the methods [22, 5, 18, 19], which utilize both color and depth information. Especially in the DARK dataset, where RGB information is limited and many detector with RGB can not work, our approach is significantly superior to others. Some de-

tecting examples are shown in Figure 7. This demonstrates that our method is quite robust in dealing with real-world challenging tasks including occlusion, variations in postures and clutter, and also dark illumination.

Table 1: Average of miss rate and FPPI in first stage.

Dataset	Miss Rate	FPPI
CLOTHING STORE	0.044	41.12
OFFICE	0.049	50.32
MOBILE PLATFORM	0.027	31.45
DARK	0.057	44.32
Average	0.044	41.80

Contribution of the first stage We evaluate it on four datasets with the results in Table 1 with average miss rate at 4.4% and average FPPI at 41.80. The results show that only a few false positives are contained in the responses. This indicates that the finding possible head-top points stage is very effective for search space reduction. And in this stage, the average run time for one frame is around 10 ms, which is fast enough to ensure the real-time processing. On average, about 25 points from the three hundred thousand pixels in a frame are detected as candidate head-top points (1/12,000).

Contribution of the two features We compare each feature used in the proposed descriptor separately. In this experiment, we test each feature at a time to compare the detection results on CLOTHING STORE. As illustrated in Figure 6 (a), using only one feature (ODDD or BLS) may dramatically decrease the performance than combining ODDD and BLS together. In addition, we compare the influence between depth difference only and ODDD in Figure 6 (a). It indicates that the detection performance of ODDD can be improved in average precision much higher than directly using depth differences.

Impact of the distance We evaluate the effect of the distance from camera on detection performance. As the Microsoft Kinect sensor has a practical ranging limit of 0.8m – 3.5m distance [10], the long distance out of the practical range may be more fragmentary and noisier than the short distance. Figure 6 (b) depicting the analysis of the effects of long range (> 3.5m) and short range ($\leq 3.5m$) shows that the proposed method can provide higher detection accuracy for nearer humans.



Figure 7: Examples of detection results on (a) CLOTHING STORE, (b) OFFICE, (c) MOBILE PLATFORM, and (d) DARK.

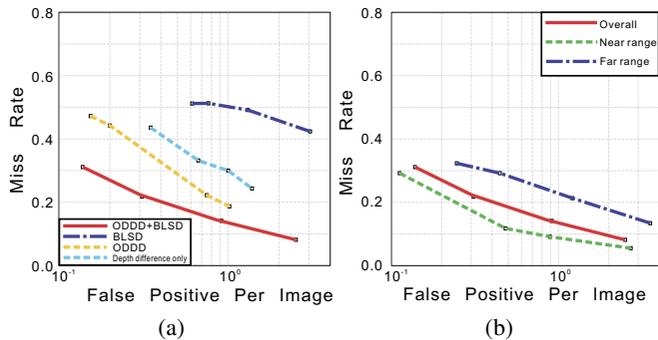


Figure 6: (a) Contribution analysis of each detection feature. The red curve represents the results of the method with both ODDD and BLSD features. Purple and yellow curves show the results of a specific feature either BLSD or ODDD. Blue curve represents depth difference feature. (b) presents the detection performance for different distance ranges ($>3.5\text{m}$ or $<3.5\text{m}$).

4 CONCLUSION

We have presented in this paper a novel two-staged method for detecting humans in depth images. The possible human head-top points are extracted in the edge map by the first stage. These candidates are then fed to the second verification stage to output the final detection results. Experiment results show that the proposed method (without RGB information) can reliably detect people in complex, dynamic and even dark environments in real time with high accuracy, and even outperforms state-of-the-art approaches that use RGB-D data.

ACKNOWLEDGEMENTS

The work presented in this paper is supported by National Natural Science Foundation of China, under Grant No. 61175036.

REFERENCES

- [1] Z. Cai, J. Han, L. Liu, and L. Shao, 'Rgb-d datasets using microsoft kinect or similar sensors: a survey', *Multimedia Tools and Applications*, 1–43, (2016).
- [2] M. Camplani, A. Paiement, M. Mirmehdi, D. Damen, S. Hannuna, T. Burghardt, and L. Tao, 'Multiple human tracking in rgb-d data: A survey', *arXiv*, (2016).
- [3] J. Canny, 'A computational approach to edge detection', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6), 679–698, (1986).
- [4] B. Choi, C. Meriçli, J. Biswas, and M. Veloso, 'Fast human detection for indoor mobile robots using depth images', in *IEEE International Conference on Robotics and Automation*, pp. 1108–1113. IEEE, (2013).
- [5] W. Choi and S. Pantofaru, C. and Savarese, 'A general framework for tracking multiple people from a moving camera', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35**(7), 1577–1591, (2013).
- [6] N. Dalal and B. Triggs, 'Histograms of oriented gradients for human detection', in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pp. 886–893. IEEE, (2005).
- [7] D. Focken and R. Stiefelwagen, 'Towards vision-based 3-d people tracking in a smart room', in *International Conference on Multimodal Interfaces*, pp. 400–405. IEEE, (2002).
- [8] S. A. Guomundsson, R. Larsen, H. Aanæs, M. Pardas, and J. R. Casas, 'ToF imaging in smart room environments towards improved people tracking', in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–6. IEEE, (2008).
- [9] J. Han, E. J Pauwels, P. M De Zeeuw, and P. HN De With, 'Employing a rgb-d sensor for real-time tracking of humans across multiple re-entries in a smart environment', *IEEE Transactions on Consumer Electronics*, **58**(2), 255–263, (2012).
- [10] J. Han, L. Shao, D. Xu, and J. Shotton, 'Enhanced computer vision with microsoft kinect sensor: A review', *IEEE Transactions on Cybernetics*, **43**(5), 1318–1334, (2013).
- [11] C Herrera, J. Kannala, J. Heikkilä, et al., 'Joint depth and color camera calibration with distortion correction', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **34**(10), 2058–2064, (2012).
- [12] S. Ikemura and H. Fujiyoshi, 'Real-time human detection using relational depth similarity features', in *Asian Conference on Computer Vision*, 25–38, Springer, (2011).
- [13] O. H. Jafari, D. Mitzel, and B. Leibe, 'Real-time rgb-d based people detection and tracking for mobile robots and head-worn cameras', in *IEEE International Conference on Robotics and Automation*, pp. 5636–5643. IEEE, (2014).
- [14] X. Li, C. Shen, A. Dick, and A. Hengel, 'Learning compact binary codes for visual tracking', in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2419–2426, (2013).
- [15] T. Linder and K. O Arras, 'People detection, tracking and visualization using ros on a mobile service robot', in *Robot Operating System (ROS)*, 187–213, Springer, (2016).
- [16] J. Liu, Y. Liu, Y. Cui, and Y. Q. Chen, 'Real-time human detection and tracking in complex environments using single rgb-d camera', in *Image*

- Processing (ICIP)*, 2013 20th IEEE International Conference on, pp. 3088–3092. IEEE, (2013).
- [17] J. Liu, Y. Liu, G. Zhang, P. Zhu, and Y. Q. Chen, ‘Detecting and tracking people in real time with rgb-d camera’, *Pattern Recognition Letters*, **53**, 16–23, (2015).
- [18] J. Liu, G. Zhang, Y. Liu, L. Tian, and Y. Q. Chen, ‘An ultra-fast human detection method for color-depth camera’, *Journal of Visual Communication and Image Representation*, **31**, 177–185, (2015).
- [19] M. Munaro, C. Lewis, D. Chambers, P. Hvas, and E. Menegatti, ‘Rgb-d human detection and tracking for industrial environments’, in *Intelligent Autonomous Systems*, 1655–1668, Springer, (2016).
- [20] F. Qi, J. Han, P. Wang, G. Shi, and F. Li, ‘Structure guided fusion for depth map inpainting’, *Pattern Recognition Letters*, **34**(1), 70–76, (2013).
- [21] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, ‘Real-time human pose recognition in parts from single depth images’, *Communications of the ACM*, **56**(1), 116–124, (2013).
- [22] L. Spinello and K. O. Arras, ‘People detection in rgb-d data’, in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3838–3843. IEEE, (2011).
- [23] S-Z Su, Z-H Liu, S-P Xu, S-Z Li, and R. Ji, ‘Sparse auto-encoder based feature learning for human body detection in depth image’, *Signal Processing*, **112**, 43–52, (2015).
- [24] B. Tan, J. Zhang, and L. Wang, ‘Semi-supervised elastic net for pedestrian counting’, *Pattern Recognition*, **44**(10), 2297–2304, (2011).
- [25] S. Tang, X. Wang, X. Lv, T. X. Han, J. Keller, Z. He, M. Skubic, and S. Lao, ‘Histogram of oriented normal vectors for object recognition with a depth sensor’, in *Asian Conference on Computer Vision*, 525–538, Springer, (2013).
- [26] L. Tian, G. Zhang, M. Li, J. Liu, and Y. Q. Chen, ‘Reliably detecting humans in crowded and dynamic environments using rgb-d camera’, in *Proceedings of the IEEE International Conference on Multimedia and Expo*, (2016).
- [27] X-T Truong, V. N. Yoong, and T-D Ngo, ‘Rgb-d and laser data fusion-based human detection and tracking for socially aware robot navigation framework’, in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 608–613. IEEE, (2015).
- [28] X. Wang, T. X. Han, and S. Yan, ‘An hog-lbp human detector with partial occlusion handling’, in *IEEE International Conference on Computer Vision*, pp. 32–39. IEEE, (2009).
- [29] C. Wojek, S. Walk, S. Roth, and B. Schiele, ‘Monocular 3d scene understanding with explicit occlusion reasoning’, in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1993–2000. IEEE, (2011).
- [30] L. Xia, C-C Chen, and JK Aggarwal, ‘Human detection using depth information by kinect’, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pp. 15–22. IEEE, (2011).
- [31] H. Xue, Y. Liu, D. Cai, and X. He, ‘Tracking people in rgb-d videos using deep learning and motion clues’, *Neurocomputing*, **204**, 70–76, (2016).
- [32] G. Zhang, J. Liu, L. Tian, and Y. Q. Chen, ‘Reliably detecting humans with rgb-d camera with physical blob detector followed by learning-based filtering’, in *Acoustics, Speech and Signal Processing (ICASSP)*, *IEEE International Conference on*, pp. 2004–2008, (2016).

A New Kernelized Associative Memory and Some of Its Applications

Matthew Saltz and Lluís A. Belanche¹

Abstract. The classical Bidirectional Associative Memory (BAM) allows for the storage of pairs of vectors, such that when either member of the pair is presented to the BAM, the other member may be successfully recalled. This work presents a novel BAM, improved with respect to its capacity and noise performance through the use of the kernel trick, a common technique in machine learning for transforming linear methods into nonlinear methods. By kernelizing the BAM's energy function directly and defining new methods for recall, the kernel BAM shows improved performance compared to both the original BAM as well as a previously existing nonlinear BAM. This is demonstrated with thorough experimentation on synthetic datasets, and several practical applications are given on real data.

1 INTRODUCTION

Associative memories (AMs), designed to store items (known as *patterns*) and then recall them even under noisy conditions, have been studied for decades [1, 2, 3]. The kind of recall performed by an AM is trivial for the human brain, and many AMs are biologically inspired and typically represented as artificial neural networks. Thus, AMs are often interesting from both a biological modeling as well as a machine learning perspective –see, e.g., [4, 5, 6, 7]. AMs are judged based on the number of patterns they can store without sacrificing their recall abilities (known as the *capacity*), as well as the average amount of noise that can be added to a stored pattern while still maintaining effective recall (known as the *noise performance*).

A classical example of an AM neural network is the Bidirectional Associative Memory, or BAM, which stores *pairs* of associated items [2]. When one item from a pair stored in the BAM is presented to the network, the network retrieves and returns its associated item, even if the input has been corrupted in some way (*i.e.*, it contains noise). An interesting property of the BAM is that recall may occur in both directions, either recalling the first or second item of a pair.

One limitation of the BAM (and other AM networks as well) is that when the patterns to be stored are linearly dependent, recall performance suffers; and as more patterns are added to the network, they will inevitably become linearly dependent, and so the capacity of the BAM is ultimately limited [2, 8, 7]. In order to alleviate this problem, it is possible to project the input patterns into a higher dimensional space, where their representations are more likely to be orthogonal. The larger the dimensionality, the higher the capacity of the network becomes. Recall could be performed in this space, and the result could be mapped back to the input space.

In this work, a novel, improved BAM is presented that takes advantage of the kernel trick to improve capacity and noise performance. While others have used the kernel trick for a variety of au-

toassociative memories (which store only a set of patterns rather than a set of pairs of patterns), the focus has been primarily on kernelizing the recall method directly [9, 10, 4, 11] or, if not, on using the kernel AM for another purpose, such as classification [8]. The kernel BAM presented here focuses on maintaining as closely as possible the analogy with the original BAM by kernelizing the associated *energy function* of the model, and then deriving novel methods for recall from there. Thorough experimentation demonstrates the superiority of the kernel BAM in a variety of cases over the original BAM as well as the most analogous nonlinear BAM found in the literature.

2 ENERGY-BASED MODELS

Given a set of p pairs $\mathcal{S} = \{(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \dots, (\mathbf{x}^{(p)}, \mathbf{y}^{(p)})\}$ with $\mathbf{x}^{(i)} \in \mathcal{X}$, $\mathbf{y}^{(i)} \in \mathcal{Y}$, the goal of an AM is to store the pairs in such a way that, when presented with a vector, known as a *pattern*, $\mathbf{x}^{(i)}$, its corresponding pattern $\mathbf{y}^{(i)}$ is correctly recalled, even if the input pattern $\mathbf{x}^{(i)}$ has been corrupted in some way. An example of this would be to store (*name, telephone*) pairs so that, when a user wants to recall the telephone of someone \mathbf{x} , the name of this someone \mathbf{x} could be presented to the memory, which would recall the correct telephone number \mathbf{y} even if the name \mathbf{x} –as presented to the AM– is corrupted by background noise. The two primary metrics for the quality of an AM are *capacity*, which describes the number of patterns (or pairs) that can be stored in the AM with respect to the dimensionality of the patterns, and *noise performance*, which is how well the AM can recover memories in the presence of noise.

The scenario above, where the AM stores pairs of patterns, is known as an *heteroassociative* memory. The special case where the AM stores only a set of single patterns to remember, rather than a set of pairs, is known as the *autoassociative* case. Some AMs are specially designed for this case, like the well-known Hopfield network [1]. The autoassociative case is equivalent to having $\mathbf{x}^{(i)} = \mathbf{y}^{(i)}$ for all $i \in \{1, \dots, p\}$ in the formulation above.

In order to memorize the set \mathcal{S} of associated pairs, we assume there is some general dependency between the components x_i of \mathbf{x} and the components y_j of \mathbf{y} . We then want to discover the form of this dependency so that we can use it to map an input vector $\mathbf{x}^{(i)}$ from \mathcal{S} to its corresponding vector $\mathbf{y}^{(i)}$ (and potentially vice versa). One way to do this is to define first the believed structure, or *architecture*, of the dependencies, and then introduce an associated function, known as the *energy function*, that takes \mathbf{x} and \mathbf{y} as inputs and evaluates how well \mathbf{x} and \mathbf{y} satisfy the dependency. Finally, we need a way to *infer* one from the other. This process of defining an architecture, an associated energy function, a training method/loss function, and an inference method is known as *energy-based modeling*, and several kinds of AMs can be described naturally in this framework, including

¹ Technical University of Catalonia, Spain, email: belanche@cs.upc.edu

the BAM –see [12] for a review on energy-based learning models.

The most basic form of an AM is the *linear* AM (LAM). When \mathbf{x} and \mathbf{y} are bipolar vectors, with $\mathbf{x} \in \{-1, +1\}^m$ and $\mathbf{y} \in \{-1, +1\}^n$, one defines the associated energy function

$$E(\mathbf{x}, \mathbf{y}) = -\frac{1}{2} \mathbf{y}^T \mathbf{W} \mathbf{x}, \quad \mathbf{W} \in \mathbb{R}^{n \times m} \quad (1)$$

where lower energy values are associated with more compatible vectors; assuming \mathbf{x}_0 as input, we may infer the most compatible \mathbf{y} by taking

$$\mathbf{y}^* = \arg \min_{\mathbf{y} \in \{-1, +1\}^n} E(\mathbf{x}_0, \mathbf{y}) = \text{sgn}[\mathbf{W} \mathbf{x}_0] \quad (2)$$

where $\text{sgn}[\cdot]$ returns the vector of signs of its argument. In classical *Hebbian learning*, \mathbf{W} is computed as the sum of correlation matrices between the $\mathbf{x}^{(i)}$ and their corresponding $\mathbf{y}^{(i)}$, that is,

$$\mathbf{W} = \sum_{i=1}^p \mathbf{y}^{(i)} (\mathbf{x}^{(i)})^T = \mathbf{Y} \mathbf{X}^T \quad (3)$$

being $\mathbf{X} \in \mathbb{R}^{m \times p}$ the matrix where the i -th column is $\mathbf{x}^{(i)}$, and $\mathbf{Y} \in \mathbb{R}^{n \times p}$ the matrix where the i -th column is $\mathbf{y}^{(i)}$. Another approach solves the equation $\mathbf{Y} = \mathbf{W} \mathbf{X}$ to get $\mathbf{W} = \mathbf{Y} \mathbf{X}^\dagger$, where $\mathbf{X}^\dagger = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$ is the Moore-Penrose pseudoinverse. This method has a better capacity and noise performance in general, since Hebbian learning suffers when vectors are correlated [7, 11].

3 KERNEL BIDIRECTIONAL ASSOCIATIVE MEMORIES

The LAM above is limited in modeling capability and in its ability to accurately recover memories with noisy inputs [7]. One way to improve this is to introduce a recurrent structure, feeding the output of the network in one step back into the network as input in the next step. In this way, the network can iteratively improve on its guess for the output. One such recurrent AM is the Bidirectional Associative memory (BAM). The BAM may use the same energy function as in Eq. (1), but in contrast with the LAM, which can only recall in one direction (retrieving the \mathbf{y} for a given \mathbf{x}), the BAM can recall in both directions. It accepts \mathbf{x} or \mathbf{y} as input and performs consecutive updates until the state of the network no longer changes.

The goal of the update process is to start at the input and continually decrease the value of the energy function until it finds a local minimum. The recurrent update process is as follows, where $\mathbf{s}_x(t)$ and $\mathbf{s}_y(t)$ indicate the state of the \mathbf{x} and \mathbf{y} portions of the network at update step t , respectively. Assuming that the network is initially presented with an input for \mathbf{x} :

$$\mathbf{s}_x(0) = \text{input } \mathbf{x}$$

$$\mathbf{s}_y(1) = \arg \min_{\mathbf{y} \in \{-1, +1\}^n} E(\mathbf{s}_x(0), \mathbf{y}) = \text{sgn}[\mathbf{W} \mathbf{s}_x(0)]$$

$$\mathbf{s}_x(2) = \arg \min_{\mathbf{x} \in \{-1, +1\}^m} E(\mathbf{x}, \mathbf{s}_y(1)) = \text{sgn}[\mathbf{s}_y(1)^T \mathbf{W}]$$

...

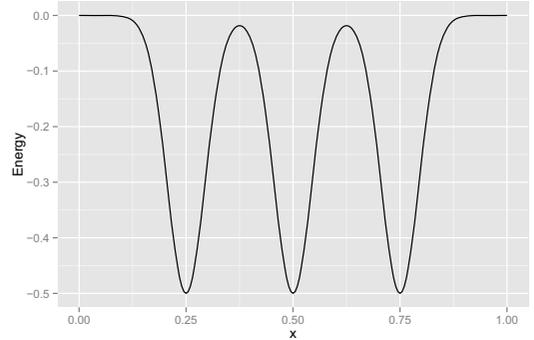
$$\mathbf{s}_y(t+1) = \arg \min_{\mathbf{y} \in \{-1, +1\}^n} E(\mathbf{s}_x(t), \mathbf{y}) = \text{sgn}[\mathbf{W} \mathbf{s}_x(t)]$$

$$\mathbf{s}_x(t+2) = \arg \min_{\mathbf{x} \in \{-1, +1\}^m} E(\mathbf{x}, \mathbf{s}_y(t+1)) = \text{sgn}[\mathbf{s}_y(t+1)^T \mathbf{W}]$$

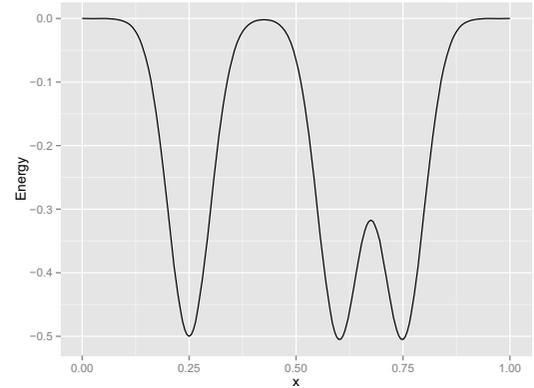
Because there is only a finite number of possible bipolar vectors, and because the energy decreases after every update, this process

must eventually *converge* to a local minimum of the energy function, where convergence means that the state of the network no longer changes state after successive iterations [2].

The BAM network together with the recurrent update process defines a *dynamical system*, and the local minima of the energy function are known as the *attractors* of the dynamical system. Since our goal is to recall memories, we want to train \mathbf{W} such that the stored patterns correspond to the attractors of the system. The space around an attractor in which all vectors will converge to that attractor is known as the *basin of attraction*. Furthermore, we also want to be able to recover memories in conditions of noise. The maximum amount of noise (measured in number of flipped bits) for which an AM performs effectively is known as its *radius of attraction*.



(a) Evenly-spaced memories, at 0.25, 0.5, and 0.75. This illustrates an ideal situation for the radius of attraction, where each memory is an equal distance from the next.



(b) Unevenly-spaced memories, at 0.25, 0.6, and 0.75. The radius of attraction of a BAM with this energy function is limited by the separation between the memories at 0.6 and 0.75.

Figure 1: Two energy functions in the autoassociative case.

An example of two different energy functions with different stored patterns and different radii of attraction is shown in Fig. 1. In this example, we consider the autoassociative case where only \mathbf{x} patterns are stored, with $m = 1$ (so $x \in \mathbb{R}$). To visualize recall for any given x value, one may imagine dropping a ball on the energy function at that x location and watching it roll down to the nearest local minimum. The radius of attraction of a stored pattern describes how close the ball must fall to the pattern in order to successfully converge to that pattern in the worst case. The radius of attraction of an AM is the minimum of all of the radii of attraction of its stored patterns. In Fig. 1a, the memories are evenly spaced and the energy function is such that each memory has an equal radius of attraction

r , which is ideal: the radius of attraction of the AM is also r . In contrast, Fig. 1b illustrates a case where the radius of attraction is limited by two memories which are closer to each other than the rest.

3.1 Kernelizing the energy function

Recall the normal BAM energy function in Eq. (1), being $\mathbf{W} \in \mathbb{R}^{n \times m}$ the BAM weight matrix. Let $k_x : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ and $k_y : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ be kernel functions, where $k_x(\mathbf{r}, \mathbf{t}) = \langle \phi_x(\mathbf{r}), \phi_x(\mathbf{t}) \rangle_{\mathcal{H}_x}$ for some $\phi_x : \mathcal{X} \rightarrow \mathcal{H}_x$, and $k_y(\mathbf{r}, \mathbf{t}) = \langle \phi_y(\mathbf{r}), \phi_y(\mathbf{t}) \rangle_{\mathcal{H}_y}$ for some $\phi_y : \mathcal{Y} \rightarrow \mathcal{H}_y$. Here \mathcal{H}_x and \mathcal{H}_y are the (potentially) different Hilbert spaces associated with the kernels k_x and k_y , and \mathcal{X} and \mathcal{Y} are the domains of \mathbf{x} and \mathbf{y} , respectively.

Now let $\mathbf{W} = \mathbf{U}\mathbf{V}^T$, where $\mathbf{U} \in \mathbb{R}^{n \times l}$ and $\mathbf{V} \in \mathbb{R}^{m \times l}$ are matrices with column vectors \mathbf{u}_i and \mathbf{v}_i respectively. Then we have:

$$\begin{aligned} E(\mathbf{s}_x, \mathbf{s}_y) &= -\frac{1}{2} \mathbf{s}_y^T \mathbf{U} \mathbf{V}^T \mathbf{s}_x \\ &= -\frac{1}{2} (\langle \mathbf{s}_y, \mathbf{u}_1 \rangle, \dots, \langle \mathbf{s}_y, \mathbf{u}_l \rangle)^T (\langle \mathbf{s}_x, \mathbf{v}_1 \rangle, \dots, \langle \mathbf{s}_x, \mathbf{v}_l \rangle) \\ &= -\frac{1}{2} \sum_{i=1}^l \langle \mathbf{s}_y, \mathbf{u}_i \rangle \langle \mathbf{s}_x, \mathbf{v}_i \rangle \end{aligned} \quad (4)$$

Now replacing \mathbf{s}_x by $\phi_x(\mathbf{s}_x)$, \mathbf{v}_i by $\phi_x(\mathbf{v}_i)$, \mathbf{s}_y by $\phi_y(\mathbf{s}_y)$ and \mathbf{u}_i by $\phi_y(\mathbf{u}_i)$, we have the kernelized energy function:

$$\begin{aligned} E(\mathbf{s}_x, \mathbf{s}_y) &= -\frac{1}{2} \sum_{i=1}^l \langle \phi_y(\mathbf{s}_y), \phi_y(\mathbf{u}_i) \rangle_{\mathcal{H}_y} \langle \phi_x(\mathbf{s}_x), \phi_x(\mathbf{v}_i) \rangle_{\mathcal{H}_x} \\ &= -\frac{1}{2} \sum_{i=1}^l k_y(\mathbf{s}_y, \mathbf{u}_i) k_x(\mathbf{s}_x, \mathbf{v}_i) \end{aligned} \quad (5)$$

Letting $\Phi_y(\mathbf{U})$ be the matrix with columns $\phi_y(\mathbf{u}_i)$ and $\Phi_x(\mathbf{V})$ be the matrix with columns $\phi_x(\mathbf{v}_i)$ we see that

$$E(\mathbf{s}_x, \mathbf{s}_y) = -\frac{1}{2} \phi_y(\mathbf{s}_y)^T \Phi_y(\mathbf{U}) \Phi_x(\mathbf{V})^T \phi_x(\mathbf{s}_x) \quad (6)$$

This corresponds to a fully kernelized BAM, where both vectors in an association are projected into their respective feature spaces, and where the weights exist in feature space as well. Figure 2 shows the BAM as it could be seen in feature space. For certain kernels, like the RBF kernel, this corresponds to orthogonalizing the vectors in feature space, so that the performance of Hebbian learning is improved.

Using now Hebbian learning, setting $\mathbf{U} = \mathbf{Y}$ and $\mathbf{V} = \mathbf{X}$ (in which case $l = p$) we obtain:

$$E(\mathbf{s}_x, \mathbf{s}_y) = -\frac{1}{2} \sum_{i=1}^p k_y(\mathbf{s}_y, \mathbf{y}^{(i)}) k_x(\mathbf{s}_x, \mathbf{x}^{(i)}) \quad (7)$$

In the autoassociative case this corresponds to the energy function of the kernel Hopfield network proposed by Caputo et al. [8], though they do not provide mechanisms for inference and use the energy function as a means for classification only. Though it is beyond the scope of this work, it would be possible to find \mathbf{U} and \mathbf{V} through loss-based training using a loss function like the hinge loss or the negative log-likelihood. This would allow the user to choose the number of vectors in the sum by defining the dimensions of \mathbf{U} and \mathbf{V} . However, using Hebbian learning allows direct comparison of the standard BAM to the kernel BAM.

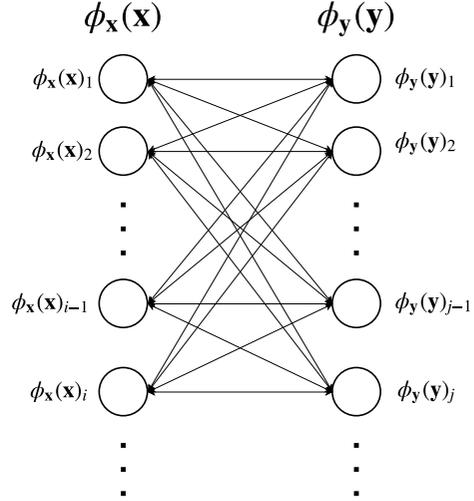


Figure 2: A visualization of the BAM projected into feature space. Note that $\phi_x(\mathbf{x})$ and $\phi_y(\mathbf{y})$ could be infinite-dimensional.

3.2 Inference

Inference (*i.e.*, computing the output for a given input) in the kernel BAM is nontrivial. In a normal BAM update step, we would take

$$\mathbf{s}_y(t+1) = \arg \min_{\mathbf{y} \in \{-1, +1\}^n} E(\mathbf{s}_x(t), \mathbf{y}) = \text{sgn}[\mathbf{W}\mathbf{s}_x(t)] \quad (8)$$

Our first method comes from a comparison with the original BAM update process. Recalling the standard BAM energy function in Eq. (1), we note that $\frac{\partial E}{\partial \mathbf{s}_y} = -\mathbf{W}\mathbf{s}_x$ and we can write the update step as

$$\mathbf{s}_y(t+1) = \text{sgn} \left(-\frac{\partial E}{\partial \mathbf{s}_y}(\mathbf{s}_x(t), \mathbf{0}) \right) \quad (9)$$

Since $\frac{\partial E}{\partial \mathbf{s}_y}$ does not depend on \mathbf{y} , and \mathbf{y} is bipolar,

$$\mathbf{s}_y(t+1) = \arg \min_{\mathbf{y} \in \{-1, +1\}^n} E(\mathbf{s}_x(t), \mathbf{y}) \quad (10)$$

as desired (the math works analogously when updating \mathbf{s}_x). To adapt to the kernel energy function in Eq. (5), we depart from Eq. (9), where

$$\frac{\partial E}{\partial \mathbf{s}_y}(\mathbf{s}_x, \mathbf{s}_y) = -\frac{1}{2} \sum_{i=1}^l \frac{\partial k_y}{\partial \mathbf{s}_y}(\mathbf{s}_y, \mathbf{u}_i) k_x(\mathbf{s}_x, \mathbf{v}_i) \quad (11)$$

which with Hebbian learning finally becomes

$$\mathbf{s}_y(t+1) = \text{sgn} \left(\sum_{i=1}^p \frac{\partial k_y}{\partial \mathbf{s}_y}(\mathbf{0}, \mathbf{y}^{(i)}) k_x(\mathbf{s}_x(t), \mathbf{x}^{(i)}) \right) \quad (12)$$

The intuition behind taking the sign of the negative gradient at $\mathbf{y} = \mathbf{0}$ as an approximation for the minimizing vector of the energy function is that $\mathbf{0}$ is the centroid of all possible bipolar vectors. In this way we approximate the energy function by a hyperplane with the slope of the gradient going through $\mathbf{0}$ and find the minimum bipolar vector on that hyperplane. To see how this plays out with a specific example, consider the RBF kernel $k_y(\mathbf{r}, \mathbf{t}) = \exp(-\gamma \|\mathbf{r} - \mathbf{t}\|^2)$, $\gamma > 0$. Taking the gradient, we obtain

$$\frac{\partial k_{\mathbf{y}}(\mathbf{r}, \mathbf{t})}{\partial \mathbf{r}} = -2\gamma(\mathbf{r} - \mathbf{t}) \exp(-\gamma \|\mathbf{r} - \mathbf{t}\|^2) \quad (13)$$

So then $\mathbf{s}_{\mathbf{y}}(t+1)$ is equal to

$$\text{sgn} \left(\sum_{i=1}^p -2\gamma(\mathbf{y} - \mathbf{y}^{(i)}) \exp(-\gamma \|\mathbf{y} - \mathbf{y}^{(i)}\|^2) k_{\mathbf{x}}(\mathbf{s}_{\mathbf{x}}(t), \mathbf{x}^{(i)}) \right) \quad (14)$$

At $\mathbf{y} = \mathbf{0}$,

$$\mathbf{s}_{\mathbf{y}}(t+1) = \text{sgn} \left(\sum_{i=1}^p \mathbf{y}^{(i)} \exp(-\gamma \|\mathbf{y}^{(i)}\|^2) k_{\mathbf{x}}(\mathbf{s}_{\mathbf{x}}(t), \mathbf{x}^{(i)}) \right) \quad (15)$$

Since the $\mathbf{y}^{(i)}$ are bipolar, and the norm of all bipolar vectors of a given dimension n is \sqrt{n} , it turns out that $\exp(-\gamma \|\mathbf{y}^{(i)}\|^2) = \exp(-\gamma n)$ is a positive constant for all i and we can simplify to

$$\mathbf{s}_{\mathbf{y}}(t+1) = \text{sgn} \left(\sum_{i=1}^p \mathbf{y}^{(i)} k_{\mathbf{x}}(\mathbf{s}_{\mathbf{x}}(t), \mathbf{x}^{(i)}) \right) \quad (16)$$

With the specific choice $k_{\mathbf{x}}(\mathbf{r}, \mathbf{t}) = \alpha^{\mathbf{r}^T \mathbf{t}}$, $\alpha > 1$, this is the update equation in the exponential BAM (eBAM) [13]; in the autoassociative case, it is equivalent to the kernel formalization of the RCAM as presented in [14]. This is interesting because it gives a unifying perspective on the kernelized recall functions of these other works, and a baseline for comparison against the other inference methods proposed here.

The previously described update step can be improved upon in a simple manner by noting that $\mathbf{0}$ is only one of many possible start points for taking the gradient. In what we call the *stochastic* BAM one considers several random starting points with components in $[-1, +1]$, taking the sign of the gradient as before, and then choosing the resulting vector that gives the lowest value for the energy function. To ensure that this gives a vector with a lower energy than would result from the previous method, one must always include $\mathbf{0}$ as one of the possible starting vectors. In practice, this performs better than the naive method, which shows the advantage of kernelizing the energy function rather than the inference method directly.

Other approaches entail treating the problem directly as a discrete optimization problem. The start point chosen is the sign of the gradient of the energy function at $\mathbf{0}$. From here, one iterates over each component of the vector in a random order, flipping its value from $+1$ to -1 or vice versa if this flipping improves the overall energy; one continues to do this until no bit can be flipped. A similar approach is to do *steepest descent* hill-climbing: rather than flipping any bit that improves the energy, one checks every component and changes the component that *most* decreases the energy.

4 EXPERIMENTAL WORK

The experimentation part of this work is divided into two sections, with the first exploring the associative memory properties of the kernel BAM with various inference methods and the second displaying additional practical applications of the kernel BAM.

4.1 A study in capacity and noise

The first study focuses on the associative memory properties of the kernel BAM, namely *capacity* and *noise performance*. Each of these

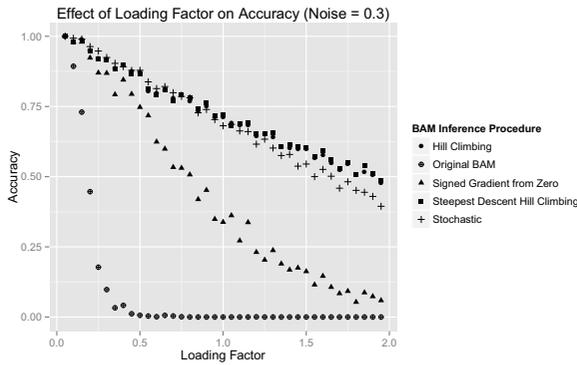
properties is tested with three different kernels: polynomial kernels of degree 2 and 3, and the RBF kernel. For the latter, $\gamma = 0.5$. In each chart the four different inference procedures presented in Section 3.2 are compared. In the case of the stochastic inference procedure, 10 random start points are performed. ‘Loading factor’ signifies the fraction of patterns stored with respect to the dimension of the patterns. Noise is also written as a fraction of the dimension of the data. For the experiments shown, the dimensionality of both $\mathbf{x}^{(i)}$ and $\mathbf{y}^{(i)}$ was $m = n = 32$. For space reasons, only a sample of the results will be shown. The tests were performed as follows:

For each loading factor $\psi \in \{0.05, 0.1, 0.15, \dots, 1.95\}$:

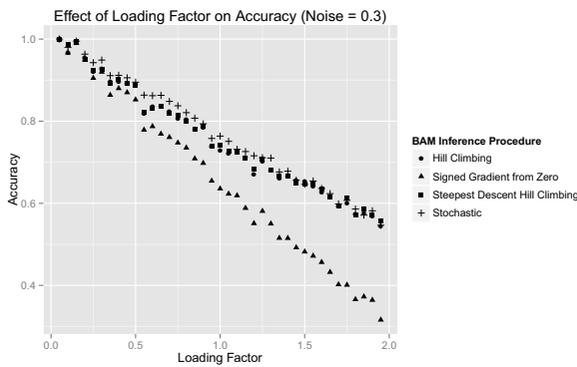
- Generate five datasets $\mathcal{S}_1, \dots, \mathcal{S}_5$ of size $p = n \cdot \psi$ where n is the dimension of the input patterns. Each dataset consists of p randomly generated unique pairs $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ of bipolar vectors.
- For each noise amount $\eta \in \{0, 0.05, 0.1, \dots, 0.5\}$, generate new test datasets \mathcal{S}'_j by, for each vector $\mathbf{x}^{(i)}$ in \mathcal{S}_j , adding 10 corresponding noisy vectors $\hat{\mathbf{x}}_q^{(i)}$, $q \in \{1, \dots, 10\}$ by randomly flipping exactly $\eta \cdot n$ bits of $\mathbf{x}^{(i)}$ 10 times. Thus, for all $j \in \{1, \dots, 5\}$, $\mathcal{S}'_j = \{(\hat{\mathbf{x}}_q^{(i)}, \mathbf{y}^{(i)}) \mid q \in \{1, \dots, 10\}, i \in \{1, \dots, p\}\}$
- For each *bam* in the BAMs to test; for $j \in \{1 \dots 5\}$, train *bam* with \mathcal{S}_j ; for each $(\hat{\mathbf{x}}_q^{(i)}, \mathbf{y}^{(i)}) \in \mathcal{S}'_j$, use *bam* to recall $\hat{\mathbf{y}}^{(i)}$ with the noisy $\hat{\mathbf{x}}_q^{(i)}$ as the input. If $\hat{\mathbf{y}}^{(i)} = \mathbf{y}^{(i)}$ exactly, the recall is considered correct
- Compute the average accuracies as `correct_recalls/total_tests` for each *bam* over all five datasets

Capacity. We first compare how the various recall methods perform with respect to the loading factor and the kernel. The performance of the original, non-kernelized BAM [2] is included in the figures of the degree-2 polynomial kernel for reference. Figure 3 displays the effect of the loading factor on recall performance when noise is fixed at 0.3. It is first noticeable that the biggest difference in performance comes with the polynomial kernels as seen in Figs. 3a and 3b, and also that the general performance of the BAMs increases from the degree-2 to the degree-3 polynomial and from the degree-3 polynomial to the RBF –Fig. 3c. This can be explained by the fact that the corresponding feature spaces are increasing in dimension (with the one in the RBF kernel being infinite-dimensional), doing a better job at orthogonalizing the input patterns in feature space. Note that in most cases, the ‘signed gradient from zero’ method performs worse than the other optimization-based inference methods, especially for the polynomial kernels. This shows the advantage of kernelizing the energy function and operating the BAM in feature space rather than kernelizing the inference procedure directly. It also appears that –among the more advanced inference methods– those based on hill-climbing work best for the degree-2 polynomial, while the stochastic one seems to work better for the other kernels. This could be due to the fact that hill-climbing is likely to get caught in local minima, so when the energy surface is more bumpy, as might be the case with the RBF kernel, the stochastic procedure can avoid the local minima that trap the hill-climbing procedures.

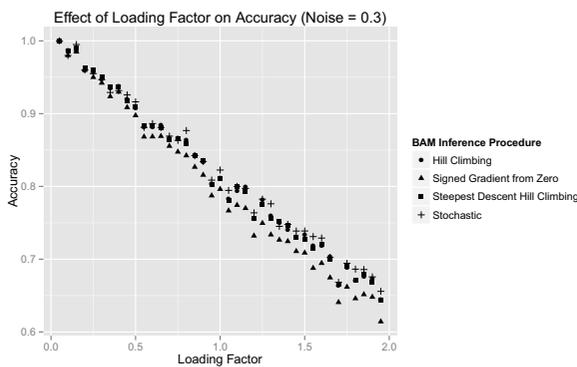
In contrast, with a smoother energy surface, like that of the degree-2 polynomial, hill-climbing is more effective in finding a global minimum than the stochastic procedure. It is also notable that accuracy seems to decrease approximately linearly with an increase in loading factor for all four proposed inference algorithms, with the accuracy of the original BAM decreasing at a rate that is exponential in appearance. The original BAM performs much worse than all other methods in virtually all cases.



(a) Polynomial kernel (degree 2)



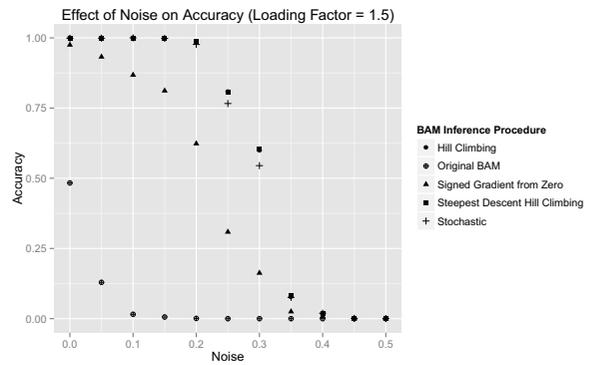
(b) Polynomial kernel (degree 3)



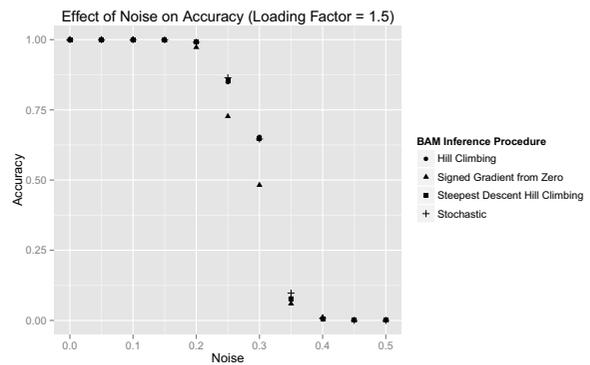
(c) RBF kernel

Figure 3: Effect of loading factor on performance ($\eta = 0.3$).

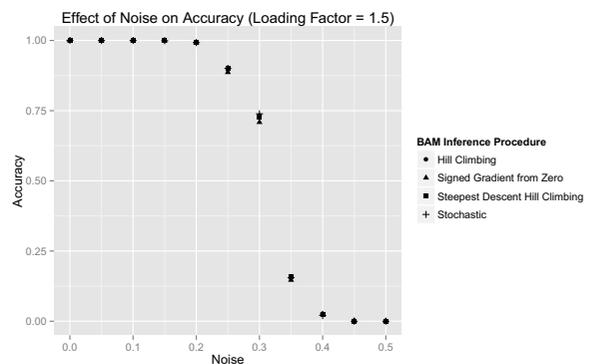
Noise performance. The comments on performance with respect to capacity apply to performance with respect to noise as well, as seen in Fig. 4. With a loading factor of 1.5, the performance difference between inference methods is again most noticeable with the degree-2 polynomial. However, the difference between the inference algorithms is less marked on the endpoints, where accuracy is either very low or very high for all of them. Again, the RBF kernel performs better in general than the other two kernels. Up to 15% noise, all kernel algorithms yield perfect accuracy, even with a loading factor of 1.5. In contrast, even with 0% noise, the original BAM only obtains an accuracy of around 45%. With a loading factor of 0.2 (not shown), all proposed algorithms have a near perfect accuracy with up to 25% noise.



(a) Polynomial kernel (degree 2)



(b) Polynomial kernel (degree 3)

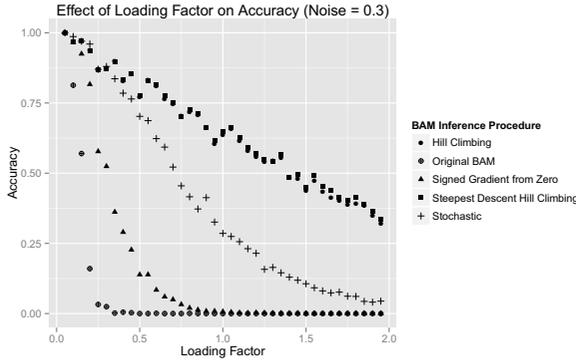


(c) RBF kernel

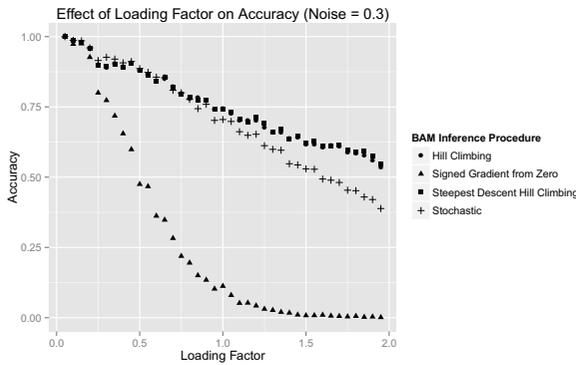
Figure 4: Effect of noise on performance ($\psi = 1.5$).

4.2 A study in one-shot behavior

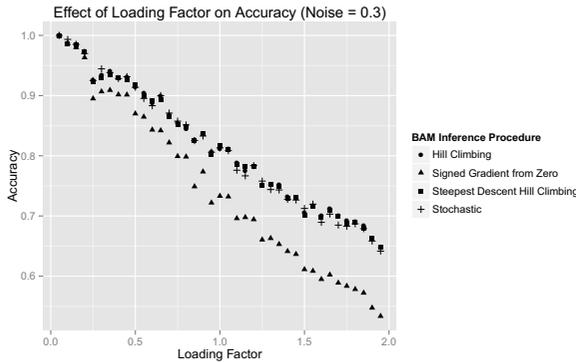
This second study focuses on the “one-shot” performance of the kernel BAM: how it performs when only *one* update step is allowed. The one-shot performance has consequences for applications like classification, where a vector is associated with a class, and only one update step is used to predict it. It also gives an idea of the effectiveness of one update step for a given recall method.



(a) Polynomial kernel (degree 2)



(b) Polynomial kernel (degree 3)

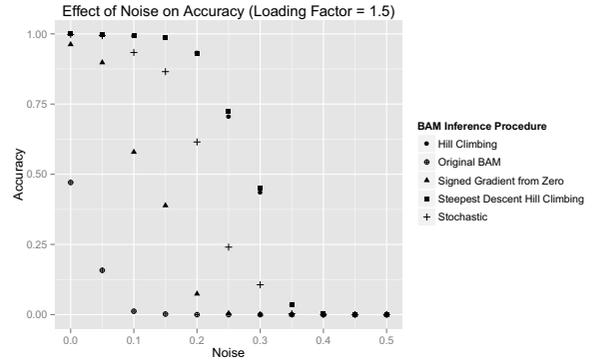


(c) RBF kernel

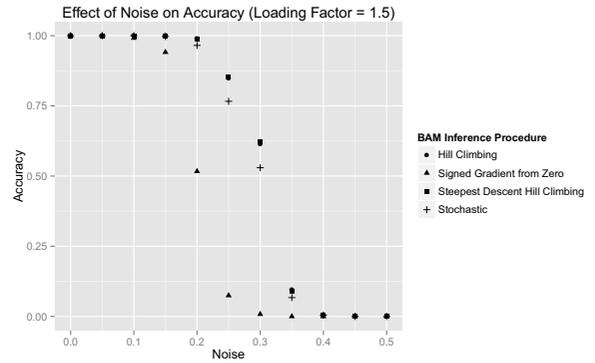
Figure 5: Effect of loading factor on one-shot performance ($\eta = 0.3$).

Figs. 5 and 6 show the accuracy of the various recall methods when only one iteration is allowed. Other than that, the experiments were performed in exactly the same manner as before. In this one-shot scenario, it turns out that differences between the new optimization-based recall methods and the simplest gradient-based method are

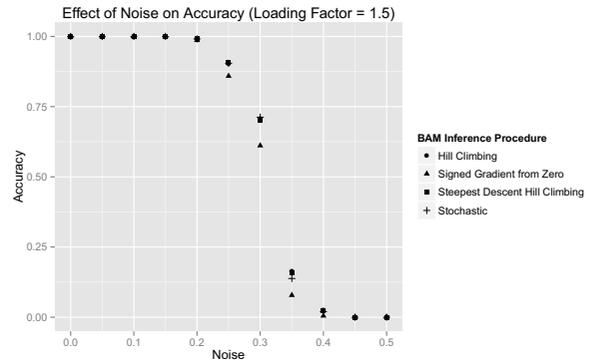
even more pronounced. For example, in Fig. 5a, the accuracy of hill-climbing methods seem to only decline linearly with respect to the increasing loading factor, whereas the simple method seems to decline exponentially, reaching an accuracy of 0% at a loading factor of 1.0 while the hill-climbing methods still have an accuracy of around 63%. Even with the RBF kernel, the difference is significant. The stochastic method is also markedly worse than the hill-climbing methods in the degree-2 polynomial, where in the case with unlimited iterations the performance difference is not particularly noteworthy. Similar comments can be made with respect to the noise performance in Fig. 6.



(a) Polynomial kernel (degree 2)



(b) Polynomial kernel (degree 3)



(c) RBF kernel

Figure 6: Effect of noise on one-shot performance ($\psi = 1.5$).

This demonstrates further that recall techniques that specifically target the minimization of the energy function and thus better main-

tain the analogy with the original BAM are much more effective in each update step, even in the case of the RBF kernel where after many iterations the performance is more similar.

5 APPLICATIONS

This section presents some more practical applications of the kernel BAM. First, the performance of the kernel BAM is tested on associations between images (heteroassociation). For this, we used the MNIST dataset², which consists of 60,000 training images and 10,000 test images of size 28x28 pixels, each valued between 0 and 255. The images were linearly rescaled to the interval $[-1, +1]$. This made selecting the RBF parameter easier, since the parameter $\gamma = 0.5$ seems to work well for bipolar vectors in general³.

Image association. First we created an association list between images without any a priori relation. We randomly selected one example of each digit 0-9, binarized each image, and created a kernel BAM with the RBF kernel for both k_x and k_y (with $\gamma = 0.5$) to store associations between each digit and the digit following it. In other words, the associations stored were $\{('0', '1'), ('1', '2'), \dots, ('9', '0')\}$ –see Fig. 7a. The simplest inference procedure was used to show the recovery of images under two types of noise. In one test, salt and pepper noise was added by randomly flipping 20% of the bits in the input image –Fig. 7b. Since all of these images are correlated (white in the middle, black around the borders), recall is made more difficult than with randomly chosen bipolar vectors. Still, all of these patterns were recovered perfectly. With more than 20% noise, recall performance began to suffer, even with the other more advanced inference methods.

Image reconstruction. In the next test, the top half of the input image was blacked out –Fig. 7c. All associated images were correctly recalled, despite some of the noisy inputs looking very similar. For example, the ‘4’ and the ‘9’ look very similar with the top half blacked out, as do the ‘1’ and the ‘7’. This could have interesting applications in image completion.

Image generation. It is perfectly possible to create an association between an image and a “class” that is also an image. Since there are far fewer classes (the numbers 0–9) than images, performing one-shot recall using a class as input will output the image giving the lowest energy for that class, even if it is not an actual stored image; in other words, the most ‘typical’ image as learned from the data in the BAM. To test this, we randomly sampled 200 images from each class, and used them to train the kernel BAM with the simple, stochastic, and hill-climbing recall procedures. Class labels were encoded by using the binary representation of their ASCII character. The results can be seen in Fig. 8. The obvious comment is that the stochastic BAM is wrong 5 out of 10 times in what it draws –Fig. 8b. Why it draws the incorrect digits is unclear, but it certainly reveals the random nature of the algorithm. It could be that the stochastic BAM retrieves an actual stored memory, albeit one of the wrong class, that happens to have a lower energy than the image displayed by the simple BAM, which seems to be a kind of aggregate of other stored memories. The simple and the hill-climbing inference procedure both correctly draw every digit. The digits written with the hill-climbing technique tend to be more clear than those of the simple inference procedure (e.g. the hole is present in the bottom of the ‘8’).

Image classification. Because the kernel BAM is heteroassociative, it may be used for classification problems, storing associations between individuals and their classes. One-shot recall may then be

used to predict the class of an input. In this case it would be possible for the kernel BAM to work on continuous input data spaces.

In this task, we also used the USPS Handwritten Digits dataset, in which the original scanned digits are binary and of different sizes and orientations; the images here have been deslanted and size normalized, resulting in 16x16 grayscale images. We tested classification not only with the simple kernel BAM and the stochastic kernel BAM but also with a few other algorithms for comparison. First, we compare with a nearest-neighbor classifier (NNC) that uses Euclidean distance. This is a standard classifier to compare to, as it is easy to implement and performs well despite its simplicity. Next, since there are just a few classes, it is possible to simply pick the class that gives the lowest energy for the input image. This is what the inference methods are trying to do in the first place and corresponds to the best possible case for recall using a kernel BAM; we call this algorithm the Lowest-energy BAM or LeBAM. Lastly, we compare to the method presented by Caputo et al. [8] because their kernel energy function is the same in the autoassociative case as that of the kernel BAM. They propose to build a kernel *autoassociative* memory for each class, each using the images for that class only. The class of an input is predicted by computing the energy of each AM for that input and then choosing the class that gives the lowest energy.

The results of classification on both the binarized and continuous versions of the input may be seen in Table 1. For all of the BAMs as well as for Caputo’s method, an RBF kernel was used with $\gamma = 0.5$, for both k_x and k_y . In almost all cases, the kernel AMs performed comparatively well against NNC. On both datasets, the LeBAM, which can be considered as the ideal kernel BAM, gives the globally best results, though it performs the same on the USPS dataset as the other two BAM methods. The fact that all kernel AMs obtain competitive accuracies in a supervised classification task for which they were not conceived is also remarkable.

Table 1: Test accuracies for the MNIST and USPS data. The value on the left corresponds to the accuracy when using binarized images, whereas the one on the right corresponds to the accuracy when using continuous values for the pixels (all values are percentages).

	NNC	BAM	Stoch. BAM	LeBAM	ref. [8]
MNIST	96.14 - 96.91	96.13 - 96.95	96.16 - 96.95	96.16 - 96.96	96.16 - 96.95
USPS	92.92 - 94.37	93.07 - 94.57	93.07 - 94.57	93.07 - 94.57	93.07 - 94.52

6 CONCLUSIONS AND FUTURE WORK

In this work, a new bidirectional associative memory (BAM) has been presented that uses the kernel trick to improve capacity and noise performance. Unlike other kernel associative memories, many of which kernelize the inference method of the associative memory directly, the kernel BAM presented here kernelizes the energy function, which allows for the creation of more effective recall procedures. Experimental work on synthetic data strongly suggests the effectiveness of the new recall procedures, and several practical use cases have been demonstrated using synthetic and real data.

The kernel BAM yields many different directions for future work, some of which have been mentioned previously. One opportunity for improvement would be to adapt the kernel BAM to work on continuous data, which could be achieved by replacing the discrete optimization methods presented by continuous ones, though it would be still necessary to constrain the domain of the input vectors (to vectors of length 1, for example, or with components in the interval $[-1, +1]$).

² <http://yann.lecun.com/exdb/mnist/>

³ See the Appendix for an explanation of this fact.

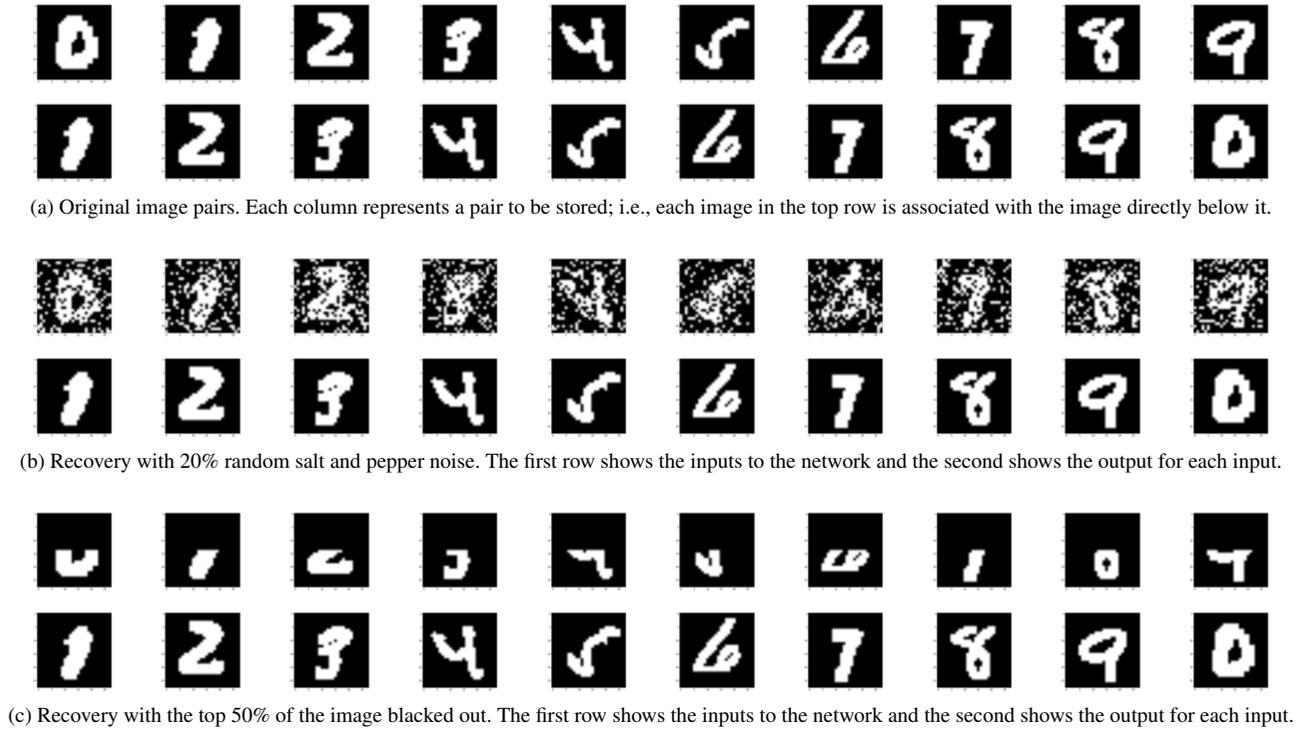


Figure 7: Associating a number with the number following it.

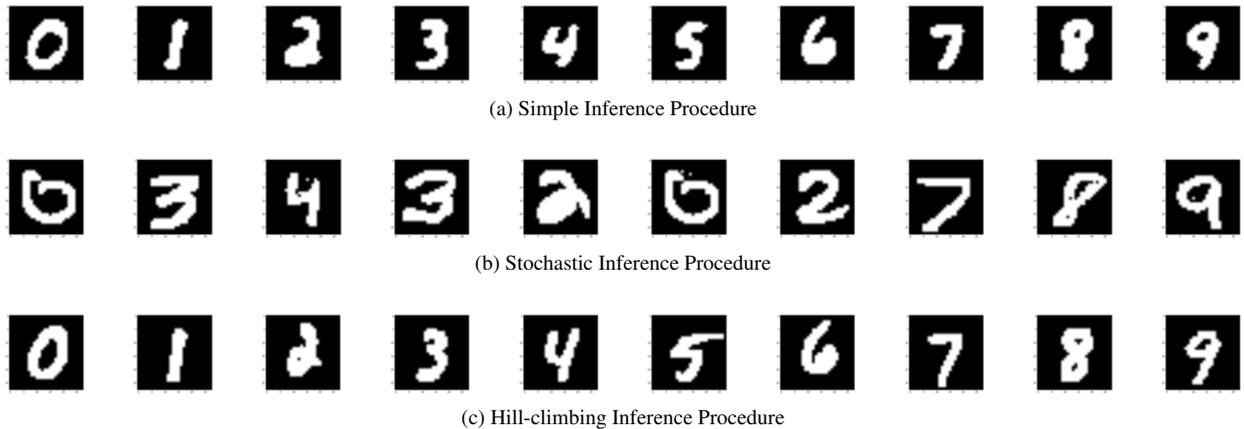


Figure 8: Image generation with the kernel BAM. The RBF kernel was used for both k_x and k_y , with $\gamma = 0.5$.

One interesting thing to note about the hill-climbing approaches is that, since they do not necessarily require the gradient of the energy function (barring the choice of the start point, which could be done differently), it would be possible to use non-differentiable kernel functions for k_x and k_y . This would allow for the use of the kernel BAM to store non-numeric data types, like text or graphs, for which working kernels exist [15, 16, 17]. The difficulty in this would be to find better optimization methods for the update step that would accommodate different kinds of data. However, classification using the described LeBAM approach would already be straightforward with kernels on non-numeric data, since only the energy function is required and not an actual update step, demonstrating another benefit of kernelizing the energy function.

A Choosing the RBF parameter for the Kernel BAM

Noting that we can write $\gamma = 0.5 \cdot \sigma^{-2}$, where σ^2 is the variance of a Gaussian, we see that $\gamma = 0.5$ corresponds to $\sigma = 1$. Looking at the plot of the energy function in Fig. 1a (which happens to be the real energy function of a kernel BAM with an RBF kernel), we can see that each basin of attraction can be put in correspondence to a Gaussian curve, and the basins of attraction of two adjacent memories meet around two standard deviations away from the center of their respective Gaussians. When we consider that—in a bipolar vector—the smallest possible change is to flip one component from ± 1 to ∓ 1 , taking σ to be somewhere between 0.5 and 2 makes sense and we chose to fix $\sigma = 1$ as the geometric mean of these values.

REFERENCES

- [1] J.J. Hopfield, 'Neural networks and physical systems with emergent collective computational abilities', *Proceedings of the national academy of sciences*, **79**(8), 2554–2558, (1982).
- [2] B. Kosko, 'Bidirectional associative memories', *Systems, Man and Cybernetics, IEEE Transactions on*, **18**(1), 49–60, (1988).
- [3] M.E. Acevedo-Mosqueda, C. Yáñez-Márquez, and M.A. Acevedo-Mosqueda, 'Bidirectional associative memories: Different approaches', *ACM Computing Surveys*, **45**(2), 18:1–18:30, (2013).
- [4] D. Nowicki and H. Siegelmann, 'Flexible kernel memory', *PLoS One*, **5**(6), (2010).
- [5] S. Chartier, M. Boukadoum, and M. Amiri, 'BAM learning of nonlinearly separable tasks by using an asymmetrical output function and reinforcement learning', *Neural Networks, IEEE Transactions on*, **20**(8), 1281–1292, (2009).
- [6] S. Chartier, G. Giguere, P. Renaud, J.M. Lina, and R. Proulx, 'FEBAM: A feature-extracting bidirectional associative memory', in *Neural Networks, 2007. IJCNN 2007. International Joint Conference on*, pp. 1679–1684, (2007).
- [7] R. Rojas, *Neural Networks: A Systematic Introduction*, Springer Berlin Heidelberg, 1996.
- [8] B. Caputo and H. Niemann, 'From markov random fields to associative memories and back: Spin glass markov random fields', Proc. of IEEE Workshop on Statistical and Computational Theories of Vision, 101–102, (2001).
- [9] C. García and J.A. Moreno, 'The Hopfield associative memory network: Improving performance with the kernel "trick"', in *Advances in Artificial Intelligence—IBERAMIA 2004*, 871–880, Springer, (2004).
- [10] S. Chen, L. Chen, and Z.H. Zhou, 'A unified SWSI–KAMS framework and performance evaluation on face recognition', *Neurocomputing*, **68**, 54–69, (2005).
- [11] B. L. Zhang, H. Zhang, and S.S. Ge, 'Face recognition by applying wavelet subband representation and kernel associative memory', *Neural Networks, IEEE Transactions on*, **15**(1), 166–177, (2004).
- [12] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang, 'A tutorial on energy-based learning', Bakir et al. (eds), *Predicting Structured Outputs*, MIT Press (2006).
- [13] Y.J. Jeng, C.C. Yeh, and T.D. Chiueh, 'Exponential bidirectional associative memories', *Electronics Letters*, **26**(11), 717–718, (1990).
- [14] R. Perfetti and E. Ricci, 'Recurrent correlation associative memories: a feature space perspective', *Neural Networks, IEEE Transactions on*, **19**(2), 333–345, (2008).
- [15] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, 'Text classification using string kernels', *The Journal of Machine Learning Research*, **2**, 419–444, (2002).
- [16] T. Gärtner, P. Flach, and S. Wrobel, 'On graph kernels: Hardness results and efficient alternatives', in *Learning Theory and Kernel Machines*, 129–143, Springer, (2003).
- [17] T. Gärtner, 'A survey of kernels for structured data', *ACM SIGKDD Explorations Newsletter*, **5**(1), 49–58, (2003).

Strategical Argumentative Agent for Human Persuasion

Ariel Rosenfeld¹ and Sarit Kraus²

Abstract. Automated agents should be able to persuade people in the same way people persuade each other - via dialogs. Today, automated persuasion modeling and research use unnatural assumptions regarding persuasive interaction, which creates doubt regarding their applicability for real-world deployment with people. In this work we present a novel methodology for persuading people through argumentative dialogs. Our methodology combines theoretical argumentation modeling, machine learning and Markovian optimization techniques that together result in an innovative agent named SPA. Two extensive field experiments, with more than 100 human subjects, show that SPA is able to persuade people significantly more often than a baseline agent and no worse than people are able to persuade each other.

1 Introduction

Persuasion is designed to influence others by modifying their beliefs or actions. People often engage in persuasive interactions through dialog in which parties who hold (partially) conflicting points of view can exchange arguments. Automated agents should be able to persuade people in the same manner; namely, by presenting arguments during a dialog.

Persuasive technologies offer various techniques for an automated agent (the *persuader*) to convince a human (the *persuadee*) to change how she thinks or what she does [17]. Some of these techniques use argumentative dialogs as their persuasion mechanism. However, strategical aspects of argumentative persuasive dialogs are still under-developed (see [48] for a review). Argumentation Theory has recently investigated the challenge of finding optimal persuasion strategies in dialogs [25, 23]. In particular, the proposed approaches do not assume that the opponent will play optimally, which is a common assumption in game theoretical analysis of persuasion dialogs [19, 38], and do not assume perfect knowledge of the persuadees' characteristics. The proposed methods have yet to be investigated with people, mainly due to their assumed strict protocols for the dialog which make their implementation with people very challenging.

In this paper we present a novel methodology for designing automated agents for human persuasion through argumentative dialogs without assuming a predefined protocol. Our methodology is based on a newly designed argumentation framework called the *Weighted Bipolar Argumentation Framework (WBAF)* which we introduce in this paper and for which we suggest a semantic. The framework and semantic are aimed at modeling the initial beliefs held by a reasoner (in our case, the persuadee) as well as the fuzzy nature in which arguments and opinions within the framework affect each other. Unlike classic semantics which label each argument in the framework as justified or not, our suggested semantic allows each argument to

carry a continuous value representing its justification level within the framework. Then, we formally define the persuasion task given the assumption that the *persuadee* acts stochastically. The persuasion task's goal is to maximize the probability that the persuadee will take the desired action or change her views on a given matter by presenting arguments. We reduce the persuasion task to a Partially Observable Markov Decision Process (POMDP) [27] and approximate its solution using the prediction of the persuadee's argumentation framework and argumentative behavior. This prediction is done using Machine Learning (ML) techniques based on collected human argumentative data. Using the obtained policy, which approximates the optimal policy for the corresponding POMDP, our agent presents arguments to its human interlocutor during a dialog.

In two field experiments, with a total of more than 100 human subjects³, we show that our agent, which we named SPA, was able to persuade subjects to change their opinions and take a desired action significantly more often than when interacting with a baseline agent and no worse than when subjects attempted to persuade each other. To the best of our knowledge, this is the first work within the context of strategical argumentation to consider the optimization of persuasive dialogs with people.

The remainder of the paper is organized as follows. In Section 2 we survey related work. In Section 3 we present the theoretical argumentation framework used in this work and suggest a semantic for it. In Section 4 we formally define the persuasion task and in Section 5 we describe our methodology and solution using a novel arguments provision agent (named SPA). In Sections 6 and 7 we evaluate our methodology in two real-world domains. Finally, in Section 8 we provide a summary and list future directions for this line of study.

2 Related Work and Background

Theoretical modeling and strategical studies of agents' behavior in persuasion interactions, within both argumentation theory and multi-agent systems, have presented logics, protocols and policies which enable agents to engage each other in a meaningful manner [29, 34, 30, 7, 35, 13]. In this realm, studies rely on the assumption that the engaging agents adhere to strict protocols and logics or that the agents are given unrealistic prior knowledge on their opponent's knowledge and beliefs [48]. Furthermore, strategic persuasion is inherently NP-complete [20].

The literature on the optimization of persuasive strategies in argumentative dialogs can be divided into 2 broad approaches:

1. *Game Theory* – in which the agent assumes that its counter-part maximizes expected utility acts optimally.
2. *Heuristic-Based* – in which the agent uses a strategy following some rule-of-thumb notion.

¹ Bar Ilan University, Israel, email: arielros1@gmail.com

² Bar Ilan University, Israel.

³ All experiments were authorized by the corresponding IRB.

In the Game Theory approach, theoretically founded methods and guarantees are provided for computing optimal argumentative strategies (e.g., [38]). However, it has been shown that people often do not adhere to the optimal, monolithic strategies that can be derived analytically in the argumentative context [39, 40, 41]. Therefore, this work is suited to the Heuristic-Based approach.

In the Heuristic-Based approach, the persuadee is neither assumed to be strategic nor is she assumed to act optimally. Several heuristics for persuasive dialog policies have been proposed in the literature, for example, the heuristic of selecting arguments supporting the agent’s most important values is proposed in [5], revealing as little information as possible [33] or presenting arguments which have a high success rate from past experiences [49]. As observed in [22], a history of previous dialogues can be used to predict the arguments that the persuadee might put forward. Naturally, this prediction (sometimes called persuadee or opponent modeling) is a key component in designing persuasive arguments; a recent example is presented in [28]. In this realm, the persuadee is usually assumed to act *stochastically*, an assumption we also make in this work. However, the persuader is not assumed to have perfect knowledge of the persuadee’s characteristics. To address this shortcoming, we propose a methodology for predicting people’s argumentative choices during a dialog using Machine Learning (ML) techniques. Rosenfeld and Kraus [41] have recently showed that ML can be extremely useful in predicting human argumentative behavior in human discussions. We use the authors’ suggested methodology in this work such that given a certain state of the dialog, the persuader can estimate the persuadee’s next argument using ML.

Hadoux et al. [23] have suggested a variation of a Markovian model to optimize persuasive behavior in dialogs. As in previously suggested modelings, the authors impose restrictive assumptions on the persuader’s and persuadee’s behavior which are relaxed in this work. Hunter [25] also presented a probabilistic modeling of the persuasive dialog using *asymmetrical* dialog procedure, in which only the persuader can posit arguments. In this work, we assume a symmetric dialog in which both parties can posit arguments. Neither of the works mentioned above, like most other works in the field, have been evaluated with people. This fact raises concerns regarding the applicability of the suggested and well thought out theoretical modelings when accounting for human argumentative behavior. Thus far, very little investigation has been done regarding how well the proposed Argumentation Theory modelings apply to humans. As far as we know, only a handful of papers address the topic [3, 36, 18, 12, 41]. These papers do not account for persuasive argumentative interactions.

The Natural Language Processing (NLP) community has also addressed the issue of automatic persuasion in various settings. For example, by generating personalized smoking cessation letters [37], ranking textual arguments by their persuasiveness [21] and the analysis of the persuasiveness of arguments in online forum discussions [46]. The proposed methods focus on linguistic features rather than strategic human-agent interaction, and therefore complement the proposed notions of this paper. Note that the development of automated argumentation-based agents, such as the one presented in this study, necessitates the assumption that natural language statements can be automatically mapped into arguments. Despite recent advancements in NLP and Information Retrieval (IR) and their studied connections to argumentation [1, 31, 9, 32, 45, 8], this assumption is not completely satisfied by existing automated tools. Hence, throughout this work we use a human expert annotator whom we hired as a research assistant. We hope that this work will inspire other

researchers in NLP and IR to tackle the problem of automatically mapping natural language statements into arguments as well as other open problems of great importance in argumentation-based systems.

3 Theoretical Modeling

In order to perform reasoning in a persuasive context, an argumentation framework needs to be defined (see [6, 10] for recent reviews). In its most basic form, an argumentation framework consists of a set of arguments A and an attack relation R over $A \times A$ (see [14]). In our previous investigations of human argumentative behavior [40, 39, 41] we noticed that people often use supportive arguments rather than attacking ones, which necessitates the addition of the support relation as suggested in [11]. Furthermore, we noticed that people associate different belief levels in arguments, as suggested in [47, 4], and different strength levels with interactions between arguments, as suggested in [15].

Therefore, throughout this work we use the newly proposed *Weighted Bipolar Argumentation Framework* (WBAF) which integrates the basic notions from the Bipolar Argumentation Framework [11], the Weighted Argumentation Framework [15], the Quantitative Argumentation Debate (QuAD) Framework [4] and the Trust Argumentation Framework [47].

Definition 1. *Let V be a completely ordered set with a minimum element (V_{min}) and a maximum element (V_{max}). A Weighted Bipolar Argumentation Framework (WBAF) $\langle A, R, S, W, B, \omega \rangle$ consists of a finite set A called arguments⁴, two binary relations over A called attack (R) and support (S), an interaction weighing function $W : R \cup S \rightarrow V$ and an argument belief function $B : A \rightarrow V$. $\omega \in A$ is a designated argument which represents the discussed issue.*

We will refer to the WBAF as the “argumentation framework” from this point forward.

In Definition 1, we assume 2 types of possible interactions between arguments: attack and support. That is, if argument $a \in A$ relates to an argument $b \in A$, then aRb or aSb holds, respective of the relation type. It is argued that the use of both support and attack relations in argumentation frameworks is essential to representing realistic knowledge (see [2] for a survey). We also allow relations to carry different weights. The weighing function $W : R \cup S \rightarrow V$ returns a value for each pair of arguments belonging to the attack or support relations representing the degree to which one argument attacks or supports the other. Based on preliminary experiments, we assume that while the attack and support relations are not disputable in our modeling, each agent may have a different W function. We also incorporate a belief function $B : A \rightarrow V$ in our model. The belief function represents the belief that a reasoner has in each argument on its own, regardless of other arguments. Again, beliefs are personal and different agents may have different beliefs. ω denotes the argument of interest. Specifically, a reasoner seeks to evaluate ω in the context of the argumentation framework.

Example 1. *The following is a part of an argumentation framework on the topic “You should have a Computer Science Master’s Degree” as collected in Section 6.1.*

A consists of the following arguments: ω = “You should have a Computer Science Master’s Degree”, a = “A Master’s Degree helps in getting well-paying jobs.”, b = “Experience is more important than education. Therefore, a master’s degree will not help in getting better jobs.” and c = “Conducting academic research is challenging

⁴ In this work, represented as short textual statements.

and interesting". R is defined to be $\{ \langle b, a \rangle \}$ as argument b attacks argument a . S is defined to be $\{ \langle a, \omega \rangle, \langle c, \omega \rangle \}$ as both a and c directly support ω . W and B could be defined differently by each reasoner. For example, W can be defined as $W(\langle b, a \rangle) = 0.5, W(\langle a, \omega \rangle) = W(\langle c, \omega \rangle) = 0.2$, indicating that the reasoner who uses this argumentation framework believes that b 's attack on a is stronger than a 's support of ω and c 's support of ω . B can be defined as $B(a) = 0.1, B(\omega) = 0.5, B(b) = B(c) = 0.7$, indicating that the belief of the reasoner in a (not taking into account any other arguments) is lower than she has in b . See Figure 1 for an illustration.

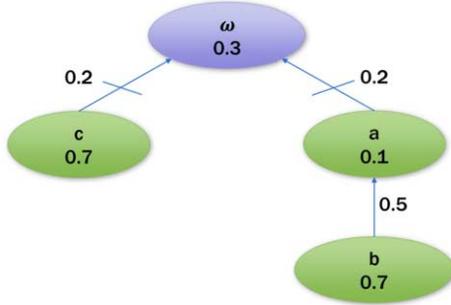


Figure 1. An example of a WBAF, as specified in Example 1. Nodes represent the arguments, arrows indicate attacks and arrows with diagonal lines indicate support. The numbers within the nodes represent the belief function and the numbers next to the edges represent the weighing function.

In our framework we assume that a reasoner uses an evaluation function $v : A \rightarrow V$ which assigns a real value to each argument while contemplating the argumentation framework. Note that v is different from the belief function B as the belief function captures the belief a reasoner has in an argument on its own, regardless of other arguments in the argumentation framework. The evaluation function v can be defined in various ways capturing different underpinning principles. In any suggested evaluation function one needs to address 3 issues:

1. **Propagation** - how does the valuation of argument a influence argument b given the weight of the relation between a and b ?
2. **Summation** - how do attacking (supporting) arguments accrue?
3. **Consolidation** - how are the attacking arguments and supporting arguments incorporated?

These three issues are addressed in Definition 2, which is an extension of the gradual valuation in Bipolar Argumentation Frameworks [11] to gradual belief valuation in WBAFs.

Definition 2. Let $WBAF = \langle A, R, S, W, B, \omega \rangle$ where W and B are defined over V , and let V^* be the set of all finite length tuples of values in V . Let $h : V \times V \rightarrow V$ be a **propagation function**, evaluating the quality of attack/support which one argument has on another; let $f_{att} : V^* \rightarrow F_{att}$ (resp. $f_{sup} : V^* \rightarrow F_{sup}$) be the **summation function**, evaluating the quality of a set of attacking (resp. supporting) arguments; and let $g : F_{att} \times F_{sup} \times V \rightarrow V$ be the **consolidation function** which combines the impact of the attacking arguments with the quality of the supporting arguments and the initial belief in the argument.

Consider $a \in A$ with arguments b_1, \dots, b_n as attacking arguments ($b_i R a$) and c_1, \dots, c_m as supporting arguments ($c_i S a$). A

gradual belief valuation function on AF is $v : A \rightarrow V$ such that $v(a) = g(f_{sup}(h(v(b_1), w(b_1, a)), \dots, h(v(b_n), w(b_n, a))), f_{att}(h(v(c_1), w(c_1, a)), \dots, h(v(c_m), w(c_m, a))), B(a))$.

An instantiation of a gradual belief valuation function is considered *legitimate* if it satisfies the following axioms (for convenience, in the following f indicates both f_{att} and f_{sup}):

1. $h(x, y)$ must be non-decreasing in both x and y .
2. $x_i > x'_i \rightarrow f(x_1, \dots, x_i, \dots, x_n) > f(x_1, \dots, x'_i, \dots, x_n)$
3. $f(x_1, \dots, x_n) > f(x_1, \dots, x_n, x_{n+1})$
4. $f() = \alpha \leq f(x_1, \dots, x_n) \leq \beta^2$.
5. $g(x, y, z)$ must be non-decreasing in x and z and non-increasing in y .

The above axioms capture basic principles that should be followed by any legitimate gradual belief valuation function. Axiom 1 assures that the propagating value from one argument to another depends on the source argument's justification level and the interaction weight in a non-negative manner. Axioms 2, 3 and 4 assure that the summation function increases in the number and quality of the relevant arguments, yet the value is bounded. Axiom 5 assures that the consolidation function does not decrease in the summed strength of the supporting arguments and does not increase in the summed strength of the attacking arguments. Furthermore, it assures that the function does not decrease in the belief level of the argument.

Definition 2 gives rise to a family of valuation functions. Given an argument of interest, the value returned by the valuation function represents the reasoner's ability to support that argument and defend it against potential attacks. The higher the strength level, the easier it is to support and defend the argument, and the harder it is to attack it. In this study we use the following instantiation:

$$h = \min, V = [-1, 1], F_{sup} = F_{att} = [0, \infty],$$

$$f_{sup}(x_1, \dots, x_n) = f_{att}(x_1, \dots, x_n) = \sum_{i=0}^n \frac{x_i + 1}{2}$$

$$\text{and } g(x, y, z) = \max\left\{\frac{1}{1+y} - \frac{1}{1+x}, z\right\}.$$

The above instantiation is inspired by the ArgTrust application [47], which uses a propagation function of \min , and extends the gradual valuation function definition in [11] to incorporate belief and propagation. The motivation for this choice is twofold: first, the selection of \min as a propagation function induces an upper bound on the affect one argument has on the other. The selection of the summation and consolidation functions is a natural extension of [11] and they provide desirable properties such as the ones described above as axioms. An example for the use of the above gradual belief valuation function is presented in Example 2.

Proposition. The suggested instantiation is a gradual belief valuation function.

Example 2. Using Example 1's argumentation framework, our proposed gradual belief valuation function will provide the following: $v(\omega) = 0.53, v(a) = -0.43, v(b) = 0.7$ and $v(c) = 0.7$. If we were to remove a and b from the argumentation framework, $v(\omega)$ would decrease to 0.37. Similarly, if we remove c from the argumentation framework, $v(\omega)$ would decrease to 0.3 (its belief level).

⁵ α (β) is the minimal (maximal) value of F_{sup} (resp. F_{att})

We assume that the higher $v(\omega)$ is within a reasoner’s argumentation framework, the more positive the reasoner’s attitude will be towards the topic of interest (ω). Therefore, in a persuasive setting, it is the persuader’s task to try and maximize the persuadee’s valuation $v(\omega)$. We discuss this task next in Section 4.

4 Persuasive Dialog Optimization

A persuasive dialog is a finite sequence of arguments $\langle a_1, a_2, \dots, a_n \rangle$ where arguments at odd indices are presented by the persuader and arguments at even indices are presented by the persuadee. A dialog is terminated when the persuader uses the “terminate” argument, which is only available to him.

We denoted D as the set of all finite length dialogs. At every even index of the dialog, the persuader observes the current state of the dialog $d \in D$ and posits an argument a according to a persuasive policy π . That is, π maps each possible even length dialog to an argument that the persuader should posit.

A persuasive agent seeks to execute an *optimal* persuasive dialog policy, π^* . Namely, π^* maximizes the expected value of $v(\omega)$ in the persuadee’s argumentation framework by following it until the dialog terminates.

We consider an environment in which the persuader is *Omniscient*, namely, it is aware of all arguments affecting ω , the designated argument which represents the discussed issue. On the other hand, we assume that the persuadee may only be aware of a subset of the arguments of which the persuader is aware. This asymmetrical situation is common when the persuader is an expert in the discussed issue and the persuadee is not. Namely, an extensive WBAF which contains all possible arguments affecting ω is maintained by the persuader. However, each persuadee holds a different WBAF that may differ from the one held by the persuader. Consequently, the persuader seeks to estimate the persuadee’s WBAF and strives to influence it. The persuader can influence the persuadee’s evaluation of ω (denoted as $v(\omega)$) under the persuadee’s argumentation framework by introducing new arguments of which the persuadee was unaware. Once presented with an argument of which the persuadee was unaware, we assume that the argument is added to the persuadee’s argumentation framework and $v(\omega)$ is updated accordingly. In our environment, the persuadee’s argumentation framework is not assumed to be known to the persuader prior to or during the dialog. However, we do assume that the persuader can obtain a probability distribution χ of the possible persuadee’s argumentation frameworks, possibly from past interactions. However, due to the infinite number of possible argumentation frameworks (recall that the WBAF’s B and W functions may return continuous numbers), constructing and using χ is not straightforward. Note that the persuader can only be certain that arguments that have actually been presented in the dialog are in the persuadee’s argumentation framework.

We assume the persuadee’s choice of arguments depends heavily on her argumentation framework. Namely, after an argument is presented by the persuadee, the persuader may change its estimations of the persuadee’s argumentation framework as the persuadee’s arguments act as “signals” to her argumentation framework. Namely, when the persuadee posits argument a , the persuader learns that a is part of the persuadee’s argumentation framework. Then the persuader can speculate *why* the persuadee chose to posit that argument. For example, a reasonable explanation may be that the persuadee thinks that a is well supported in her argumentation framework. A more practical way to look at this phenomenon, which we use later in this paper, is that the persuader speculates which argumentation

frameworks are likely to result in the persuadee presenting argument a given the current state of the dialog.

Given any non-terminated dialog d , an optimal persuasive dialog policy π^* satisfies the following equation:

$$\pi^*(d) = \operatorname{argmax}_a \mathbb{E}_{\pi^*} [v(\omega) | da] \quad (1)$$

where \mathbb{E}_{π^*} denotes the expected value given that the agent consistently follows policy π^* until the dialog terminates.

Note that calculating π^* is infeasible due to the infinite number of possible argumentation frameworks and the exponential number of possible dialogs. Therefore, a persuader can only approximate the optimal persuasive dialog policy. We address both issues next in the design of our agent, SPA, in Section 5.

5 Strategic POMDP Agent (SPA)

In order to approximate the optimal solution for the dialog optimization problem (Section 4), we show a reduction of this problem to a Partially Observable Markov Decision Process (POMDP) [27]. As discussed in Section 4, we do not assume that the persuadee’s argumentation framework is known to the persuader prior to or during the dialog. In other words, the persuadee’s **state**, i.e., her argumentation framework, is only partially observable to the persuader. Nevertheless, the persuader does see the dialog that takes places, which we will refer to as the **observation**, and can use it to derive insights regarding the persuadee’s state. As the dialog progresses more arguments are presented, which change the persuader’s observation and possibly the persuadee’s state if new arguments are added to her argumentation framework. The persuader posits arguments, i.e., takes **actions**, in order to influence the persuadee’s argumentation framework. That is, following an action by the persuader a change in the system occurs according to some **transition function** which we will soon discuss. The persuadee also presents arguments in the dialog. However, the persuadee cannot posit arguments of which she is unaware (i.e., they are not in her argumentation framework), therefore the persuadee’s arguments only change the observation and thereby, as discussed in Section 4, play an important role in estimating the persuadee’s state. Naturally, the persuader seeks to maximize the expected value of $v(\omega)$ in the persuadee’s argumentation framework by following the optimal persuasion policy. At the same time, the persuader wishes to avoid prolonging the dialog, as long dialogs may bother or annoy the persuadee. To that end, the persuader may use a **discounting factor** to favor short dialogs.

We model the persuasive dialog optimization problem as a Partially Observable Markov Decision Process (POMDP).

Definition 3. A Partially Observable Markov Decision Process is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, D, \mathcal{R}, \Phi, \gamma \rangle$ where:

- \mathcal{S} is the set of all possible argumentation frameworks. $s \in \mathcal{S}$ is a persuadee’s argumentation framework.
- \mathcal{A} is the set of all arguments that may affect the evaluation of ω and the argument “terminate”. $a \in \mathcal{A}$ is an argument that the persuader can posit, i.e., a is in the persuader’s argumentation framework. Recall that we assume that the persuader is Omniscient, and thus aware of all arguments affecting ω (see Section 4).
- \mathcal{T} represents the state transition dynamics, where $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \{0, 1\}$. $\mathcal{T}(s, a, s')$ is an indicator function specifying whether a transition from s to s' using a is valid. Formally,

$$\mathcal{T}(s, a, s') = \begin{cases} 1 & \text{if } s' = s \oplus a \\ 0 & \text{otherwise} \end{cases}$$

where $s \oplus a$ is the resulting framework from adding argument a to s .

- D is the set of all possible finite length dialogs. In our setting, D is also the set of all possible observations.
- $\mathcal{R} : \mathcal{S} \times D \mapsto V$ is the reward function for arriving at state s with dialog d . We define

$$\mathcal{R}(s, d) = \begin{cases} 0 & \text{if } d \text{ is non-terminated} \\ v(\omega) & \text{otherwise} \end{cases}$$
- Φ is the conditional probability $\Phi(d \mid s', a)$. We will discuss Φ later in this section.
- γ is the discounting factor, representing the likelihood for the persuadee to be bothered or annoyed by a prolonged dialog.

SPA approximates the optimal solution for the above POMDP using Monte-Carlo Planning, an algorithm known as POMCP [43]. POMCP is a general purpose algorithm for approximating an optimal policy in large POMDPs. The POMCP algorithm uses a Monte-Carlo search tree to evaluate each argument that the persuader can posit (at odd levels of the tree) given the state of the dialog (a node in the tree). The search tree is rooted in the empty dialog.

The deployment of the POMCP algorithm does not necessitate the explicit definition of Φ . Instead, POMCP requires 3 components:

1. \mathcal{I} , a black-box simulator for sampling $s \in \mathcal{S}$ according to each state's initial probability χ (discussed in Section 4).
2. $\mathcal{G}(s, d, a)$, a generative model of the POMDP. This simulator returns a sample of a successor state (s'), dialog (d') and reward (r) given (s, d, a) , denoted $(s', d', r) \sim \mathcal{G}(s, d, a)$.
3. π_{rollout} , a policy that is deployed once leaving the scope of the search tree.

SPA approximates \mathcal{I} using Algorithm 1. In words, SPA is given an annotated corpus C of dialogs between humans (without any agent intervention) on a given topic ω . SPA assumes that the use of an argument a in C is an indicator of its likelihood to appear in the persuadees' argumentation frameworks. Therefore, Algorithm 1 samples an argument subset A' out of all arguments available in C according to each argument's Maximum Likelihood Estimation (MLE) [42]. R and S are defined according to a manual annotation of the relation between each pair of arguments in C . More details regarding the annotation process are provided in [40]. For the definition of B and W SPA is given two answer sets of questionnaires answered by human participants, denoted Q_1 and Q_2 . In Q_1 , human participants rate the persuasiveness of each argument in C on its own, namely, while disregarding all other arguments they may be aware of. We model each subject's answers in Q_1 as the subject's B function in the corresponding argumentation framework. In Q_2 , the same participants whose answers were recorded in Q_1 rate the degree to which arguments affect others. That is, participants are presented with pairs of arguments from C for which a relation was annotated in the first place. Participants rate the degree to which the first argument affects the second one. We model each subject's answers in Q_2 as the subject's W function in the corresponding argumentation framework. In order to sample B and W , and thus complete the definition of the sampled argumentation framework, SPA uses the well-established Kernel Density Estimation (KDE) sampling method [44]: First, SPA samples a participant at random from the participant list and retrieves her answers in both Q_1 and Q_2 . Then, SPA uses a Gaussian KDE method to smooth out the contribution of each of the subject's answers (in Q_1 and Q_2) over a local neighborhood of possible answers, resulting in a probability distribution centered around the subject's

actual answers. Then, SPA samples the probability distribution and uses the sample as B and W . This process makes it possible to sample an infinite variety of B and W , while using a finite set of points as "anchors" for the sampling process.

Algorithm 1 Simulating \mathcal{I}

Require: Dialog corpus C , answers sets Q_1, Q_2 .

- 1: $A \leftarrow \text{getArguments}(C)$
 - 2: $A' \leftarrow \{\omega\}$ ▷ Create a set with the designated argument
 - 3: **for all** $a \in A$ **do**
 - 4: $MLE(a) \leftarrow (\#a \text{ appearances in } C)/|C|$
 - 5: Add a to A' with probability $MLE(a)$
 - 6: $S, R \leftarrow$ manually annotated relations among argument in A'
 - 7: $id \leftarrow$ uniformly sample a participant id.
 - 8: $B \leftarrow KDE(Q_1(id))$
 - 9: $W \leftarrow KDE(Q_2(id))$
 - 10: **return** $\langle A', R, S, W, B, \omega \rangle$
-

SPA approximates $(s', d', r) \sim \mathcal{G}(s, d, a)$ using Algorithm 2. In words, similar to the input of Algorithm 1, SPA is given (the same) annotated corpus C of dialogs between humans (without any agent intervention) on a given topic ω . If $a = \text{"terminate"}$ then $s' = s$, $d' = da$ (denoting the concatenation of a to the end of dialog d), and $r = v_s(\omega)$ (the evaluation of ω in the argumentation framework s). Recall that once the persuader posits "terminate", the dialog ends. Otherwise, the dialog continues with an argument by the persuadee. To simulate the persuadee's answer, a Machine Learning algorithm, $P(a|d)$, is trained *offline* using C to predict the likelihood of each argument being presented next, given dialog d . Algorithm 2 returns $s' = s \oplus a$, denoting the addition of argument a to s and $d' = dab$ where b is an argument sampled according to $P(b|da)$. The reward can be defined as $r = -c_a$ where c_a is the cost of positing argument a . We define $r = 0$ for all arguments as we assume there is no direct cost for positing arguments. In our modeling, the cost of prolonging the dialog is captured by γ (the discounting factor).

Algorithm 2 Simulating $\mathcal{G}(s, d, a)$

Require: Dialog corpus C .

- 1: $P \leftarrow \text{predModel}(C)$ ▷ Constructed once.
 - 2: **if** $a = \text{"terminate"}$ **then**
 - 3: **return** $\langle s, da, v_s(\omega) \rangle$
 - 4: Add a to s .
 - 5: $b \sim P(da)$ s.t $b \in s$.
 - 6: **return** $\langle s, dab, 0 \rangle$
-

As for the rollout policy π_{rollout} , SPA uses a simple policy where an argument a is selected at random at odd indices of the dialog and the predication model $P(a|d)$ (see Algorithm 2) is used at even indices to simulate the persuadee's responses.

Training SPA

In order to train SPA, we need to construct a prediction model P for estimating the likelihood that an argument b will be presented next, given dialog d . To that end, SPA uses the ML methodology suggested in [40]. The method uses a standard decision tree learning algorithm that returns a probability model estimating the probability of each possible argument being presented next. P is used in the definition of $\mathcal{G}(s, d, a)$ – the generative process of the POMDP (see Definition 3).

During its training, the POMCP algorithm maintains a search tree which keeps changing and expanding as long as the algorithm is running. Many POMDP-based applications that implement the POMCP Algorithm, especially in game playing [24], train the POMCP algorithm offline against itself. Namely, two instances of the POMCP algorithm are implemented and are trained simultaneously. The first POMCP learns to play against the second POMCP, which in turn learns to play against the first. This methodology cannot be implemented in the scope of this work as we assume that the persuadee is not strategic and hence cannot be represented as a POMCP instance. However, the prediction model P does capture the non-strategic behavior of the persuadee, hence it is used in the definition of $\mathcal{G}(\cdot)$. Note that during actual deployment SPA uses the persuadee’s actual arguments instead of simulated arguments provided by sampling P .

6 Evaluation in Attitude Change

First we evaluate SPA in an attitude change task. In an attitude change task the agent’s goal is to increase positive attitude and decrease negative attitude towards a given topic. The topic we chose to focus on was the “*You should have a Computer Science Master’s Degree*”, where the persuader’s goal is to change senior computer science students’ attitude towards the enrollment to a master’s program. The topic is of great interest to senior students and hence was selected.

6.1 Data Collection

Phase 1 - We recruited 56 senior bachelors students studying Computer Science – 37 males and 19 females with an average age of 28. First, each student was asked to rate a series of five statements using an online questionnaire. The statements were regarding the students’ personal academic experience, such as “I would volunteer during my studies if I would get credit for it”. The statement of interest to us was “I plan to enroll in a Master’s degree program”. For each statement, students provided a rating on the following Likert scale; 1-Strongly Agree, 2-Agree, 3-Neutral, 4-Disagree and 5-Strongly Disagree.

Students were represented in the system using a special identification number that was given to them prior to the experiment by our research assistant. We made sure that the students were aware that the identifier could not be traced back to their identity in order to avoid possible influences on the students’ behavior. Students were divided into 3 groups according to their answers to the question of interest: Positive, Neutral and Negative.

A week afterwards, we matched the students such that each student was coupled with a student from outside her group. The coupling was carried out manually by our research assistant who asked the subjects for their preferred time slots and matched every couple accordingly. The students were asked to converse about the topic “*You should have a Computer Science Master’s Degree*” for a minimum of 5 minutes, and to try and convince their interlocutor to adopt their point of view. Dialogs ranged in length from 5 arguments to 11 (mean 7). Each dialog ended when one of the deliberants chose to exit the chat environment. All dialogs were manually annotated for arguments and the relation between those arguments by a human expert using the annotation methodology used in [40], resulting in an annotated dialog corpus C .⁶ Immediately after the chat, students

⁶ A second expert human annotator was also asked to annotate the dialogs in order to ensure the quality of the annotation. In 10% of the cases, a disagreement between the two annotators was recorded, making the annotation relatively reliable.

were again asked for their rating of the statement “I plan to enroll in a Master’s degree program” using the same scale.

In our previous study [40], we showed that people do not adhere to the reasoning rules proposed by the argumentation theory in real-world deliberations. It turns out that this result extends to persuasive interactions as well. For example, only 67% of the students participating in this phase of the data collection used a *conflict free* argument set in their dialog. Namely, 33% of the students used at least two arguments a and b such that a attacks b or vice versa during their dialog.

Phase 2 - We recruited an additional 107 senior bachelors students studying Computer Science – 68 males and 39 females with an average age of 27. Students were asked to answer two online questionnaires, a week apart. In the first one, students were asked to rate the persuasiveness of each of the 16 arguments in C on its own on a scale of 0 to 1, where 0 is “The argument is not persuasive at all” and 1 is “The argument is very persuasive”. In the second one, students were asked to rate the degree to which one argument effects another over pairs of arguments. The scale that was used was again of 0 to 1, where 0 stands for “No effect” and 1 is “Very strong effect”.

In C , 16 distinct arguments were detected (8 pro and 8 con). First, a prediction model P was trained using the methodology discussed in Section 5. For comparison, we also considered using a *Bigram model* [26]. In Bigram, the model calculates the probability $P(a_2|a_1)$ for every pair of arguments a_1, a_2 . That is, the probability that a_2 follows a_1 . These probabilities are estimated using a Maximum Likelihood Estimator with smoothing on the dialogs from C . Both models were evaluated in a one-left-out fashion where each dialog was taken out of C one at a time, both models were trained over the remaining dialogs and were evaluated in relation to the left-out dialog. The perplexity measurement of P was significantly lower than that of Bigram ($p < 0.05$), which makes it preferable.

6.2 Experimental Setting

For the evaluation of SPA we recruited 30 senior bachelors students studying Computer Science, 20 males and 10 females with an average age of 28. Students were first asked to rate two statements using an online questionnaire. The statements were: 1) “I plan to enroll in a Master’s degree program”, and 2) “A Master’s degree will help me in the future”. For each statement, students provided a rating on the same Likert scale as used in Section 6.1, namely 1-Strongly Agree, 2-Agree, 3-Neutral, 4-Disagree and 5-Strongly Disagree.

The agent’s goal is to persuade students to change their opinion and rate the 1st statement higher. That is, to encourage them to enroll in a master’s degree program. If a student has already planned to enroll in a master’s degree program prior to the experiment (i.e, she ranked the 1st statement as “Strongly Agree”, which was the case for 2 students), then the agent seeks to persuade the student to rate the 2nd statement higher. Note that none of the students provided the highest rating for the 2nd statement prior to the dialog.

We use a between-subjects experimental design with 3 conditions:

1. **SPA**. SPA was trained for 72 hours in which more than 22,700 sessions were simulated. For the evaluation of SPA, we replaced the use of the prediction model P with the persuadee’s actual arguments. Recall that P was used to simulate the persuadee’s response in the offline training of the POMCP (Section 5). For the evaluation we use the student’s actual arguments as presented in the dialog.
2. **Baseline**. The Baseline agent uses the relevance heuristic suggested in [40] and presents a random argument that has not yet

been presented in the dialog and directly relates to the last argument presented in the dialog. Of course, the agent only suggests arguments that positively relate to ω , that is, support it indirectly. If no such argument exists, the agent suggests an argument which directly supports argument ω and does not relate to the last argument presented in the dialog. If all directly supporting arguments of ω were already presented, the agent finishes the dialog.⁷

3. **Human.** Recall that during the data collection of human dialogs (with no agent intervention, see Section 6.1) the students' rating changes were recorded. We use the 56 subjects' answers as an additional benchmark in the analysis.

Subjects were pseudo-randomly assigned to each of the first 2 conditions (the 3rd condition is described as part of the data collection in Section 6.1), such that each of the two subjects who rated the 1st statement in the questionnaire as "Strongly Agree" was assigned to a different agent (SPA or Baseline).

A week after answering the questionnaire, each student was asked to engage in a chat with her agent. Note that students were not told that they would interact with an automated agent. On the other hand, they were not told that they would interact with another human either. This was done to avoid biasing the results.

As discussed earlier in this paper, the automatic extraction of arguments from texts is not in the scope of this work. Therefore, the identification of the arguments used by the students was done using a *Wizard of Oz* methodology, where during the chat a human expert⁸ mapped each of the persuadee's sentences into an argument extracted from C (see Section 6.1). In case no argument in C fits the presented statement, a designated "NULL" argument was selected. This was rarely used. The possibility of adapting the agent's framework online will be addressed in future work. In order to bolster the natural flow of the dialog, the Wizard of Oz was also in charge of framing the agent's argument using discourse markers. Namely, the wizard was not allowed to alter the content of the argument but could add discourse markers such as "However", "Moreover", etc.

At the end of the dialog, subjects were asked to answer the online questionnaire once again.

6.3 Results

Out of the 15 students who were equipped with SPA, 4 students (26.6%) changed their rating by one category. Three subjects changed from Positive to Very Positive and one from Neutral to Very Positive. Only a single student (6.6%) from the 15 students who were equipped with the Baseline agent changed her rating (from Negative to Neutral). Out of the 56 students who were asked to persuade each other in Section 6.1, 15 (26.7%) changed their opinion by at least 1 category. Out of these 15 students, 4 (7.4%) changed their opinion by 2 categories. This result is slightly better than the results obtained by SPA, yet the difference is not statistically significant. Nevertheless, the Baseline agent was significantly outperformed by the other examined conditions using Fisher's exact test ($p < 0.05$).

⁷ We chose to compare our method with another method that has already been tested with human subjects. Unfortunately, existing proposals in persuasive argumentation were not tested with people thus far. We hope to inspire other researchers in the field to test their proposed methods with human subjects.

⁸ In order to prevent the expert from being biased toward one of the agents, the expert was not involved in any other part of the research, in particular in building the agents.

7 Evaluation in Behavior Change

We also evaluate SPA in a behavior change task. In a behavior change task the agent's goal is to persuade its interlocutor to choose a desired action that does not fit with the interlocutor's initial choice. A prominent example of such a behavior change task is persuading people to make healthier life styles choices [17]. The practical decision we chose to focus on was "Would you rather receive a piece of chocolate cake or an energy bar as a free snack?", where the persuader's goal is to change its interlocutor's choice given her initial one. Therefore, two persuasive policies were learned, one that is aimed at persuading people to choose the piece of chocolate cake, and one to persuade people to choose the energy bar. Unlike attitude change (Section 6), in behavior change evaluation we wish to make the decision-making concrete and practical in order to assert that the change had taken place. Therefore, subjects were awarded with their chosen snack at the end of the experiment (see Section 7.2).

SPA assumes that a higher $v(\omega)$ value suggests a higher probability that an alternative will be chosen by the persuadee. Therefore, the agent seeks to maximize the probability that the persuadee will take the desired action by maximizing its $v(\omega)$ value.

7.1 Data Collection

Phase 1 - We recruited 28 subjects – 18 males and 20 females, with an average age of 33. Instead of rating a series of questions on a Likert scale, as done in Section 6.2, in this experiment we asked subjects to answer only a single question with a binary answer - "Would you rather receive a piece of chocolate cake or an energy bar as a free snack?". Students were divided into 2 groups according to their answers.

A week afterwards, we again matched the subjects such that each subject was coupled with a subject from outside her group. The coupling was carried out manually by our research assistant who asked the subjects for their preferred time slots and matched every couple accordingly. The subjects were asked to discuss the topic "Would you rather receive a piece of chocolate cake or an energy bar as a free snack?" for a minimum of 5 minutes, and try and convince their interlocutor to adopt their point of view. Dialogs ranged in length from 4 arguments to 11 (mean 7). Each dialog ended when one of the participants chose to exit the chat environment. Immediately after the chat, subjects were again asked to answer the binary question "Would you rather receive a piece of chocolate cake or an energy bar as a free snack?". All dialogs were manually annotated for arguments and the relation between those arguments by a human expert using the annotation methodology used in [40], resulting in an annotated dialog corpus C .

Similar to the analysis presented in Section 6.1, only 79% of the subjects participating in this phase of the data collection used a *conflict free* argument set in their dialog. Namely, 21% of the students used at least two arguments a and b such that a attacks b or vice versa during their dialog.

Phase 2 - We recruited 40 additional subjects – 24 males and 16 females, with an average age of 30. Subjects were asked to answer two online questionnaires. In the first one, subjects were asked to rate the persuasiveness of each of the 26 arguments extracted from the dialogs collected in Phase 1 (denoted C) on its own on a scale of 0 to 1, where 0 is "The argument is not persuasive at all" and 1 is "The argument is very persuasive". In the second questionnaire, subjects were asked to rate the degree to which one argument effects another over pairs of arguments. The scale that was used was 0 to 1,

where 0 stands for “No effect” and 1 is “Very strong effect”.

In C , 26 distinct arguments were detected (13 in favor of a piece of chocolate cake and 13 against). A prediction model P was trained to estimate the likelihood that an argument b will be presented next given dialog d . Similar to the analysis in Section 6.1, the perplexity measurement of P was again significantly lower than that of a Bigram prediction method ($p < 0.05$), which makes it preferable.

7.2 Experimental Setting

For the evaluation of SPA we recruited 30 subjects - 15 males and 15 females, with an average age of 29. Subjects were first asked to answer the question “Would you rather receive a piece of chocolate cake or an energy bar as a free snack?”. Out of the 30 subjects, 16 preferred to have a piece of chocolate cake and 14 preferred to have an energy bar.

We used a between-subjects experimental design with 3 conditions, the same conditions used in Section 6. Namely: SPA, Baselines and Human.

Subjects were pseudo-randomly assigned to each of the 2 agents (SPA and Baseline), such that each agent was assigned 15 subjects, 8 of which prefer to have a piece of chocolate cake and 7 who prefer to have an energy bar. At the end of the dialog, subjects were again asked to choose between a piece of chocolate cake and an energy bar. Subjects were awarded with their snack of choice in return for their participation in the experiment.

7.3 Results

Out of the 15 students who were equipped with SPA, 3 students (20%) changed their decision – 2 from a piece of cake to an energy bar and 1 from an energy bar to a piece of chocolate cake. Only a single subject (6.6%) from the 15 subjects who were equipped with the Baseline agent changed her decision (from a piece of cake to an energy bar). Out of the 28 subjects who were asked to persuade each other in Section 7.1, only 3 subjects (10.7%) changed their decisions following the chat. This result is worse than the results obtained by SPA, yet the difference is not statistically significant.

8 Conclusions and Future Work

In this paper we presented and evaluated a novel methodology for human persuasion through argumentative dialogs. To that end, we proposed a new argumentation framework called Weighted Bipolar Argumentation Framework (WBAF) and suggested a gradual belief valuation method for allowing reasoning within that framework. Our methodology, combining the WBAF argumentative modeling, machine learning on human generated dialogs and argumentative data, and Markovian optimization techniques enabled our automated agent, SPA, to persuade people in 2 distinct environments. In both an attitude change environment and a behavior change environment, SPA was able to perform on a human-like level and significantly better than a baseline agent.

This study is part of our ongoing effort to investigate the connections and challenges between Argumentation Theory and people [40, 39, 41]. We hope that the encouraging results shown in this work (and in previous ones) will inspire other researchers in the field to investigate other argumentation-based methods in human experiments. We believe that bridging the gap between formal argumentation and human argumentation is essential for making argumentation practical for a wider range of applications.

We plan to continue this line of work by investigating other human argumentative interactions such as negotiations [16]. In negotiations, both parties try to maximize some personal utility in the face of partially conflicting interests, while striving to reach an agreement.

We will be pleased to share the constructed corpora for future research.

ACKNOWLEDGEMENTS

We would like to thank Intel Collaboration Research Institute for Computational Intelligence, the ERC (grant #267523), and Israel Science Foundation (grant #1488/14) for their support in this research.

REFERENCES

- [1] Ehud Aharoni, Carlos Alzate, Roy Bar-Haim, Yonatan Bilu, Lena Dankin, Iris Eiron, Daniel Hershcovich, and Shay Hummel, ‘Claims on demand—an initial demonstration of a system for automatic detection and polarity identification of context dependent claims in massive corpora’, *COLING 2014*, 6, (2014).
- [2] Leila Amgoud, Claudette Cayrol, Marie-Christine Lagasquie-Schiex, and Pierre Livet, ‘On bipolarity in argumentation frameworks’, *International Journal of Intelligent Systems*, **23**(10), 1062–1093, (2008).
- [3] Edmond Awad, Jean-François Bonnefon, Martin Caminada, Thomas Malone, and Iyad Rahwan, ‘Experimental assessment of aggregation rules in argumentation-enabled collective intelligence’, *arXiv preprint arXiv:1604.00681*, (2016).
- [4] Pietro Baroni, Marco Romano, Francesca Toni, Marco Aurisicchio, and Giorgio Bertanza, ‘Automatic evaluation of design alternatives with quantitative argumentation’, *Argument & Computation*, **6**(1), 24–49, (2015).
- [5] Trevor JM Bench-Capon, ‘Persuasion in practical argument using value-based argumentation frameworks’, *Journal of Logic and Computation*, **13**(3), 429–448, (2003).
- [6] Gerhard Brewka, Sylwia Polberg, and Stefan Woltran, ‘Generalizations of dung frameworks and their role in formal argumentation’, *Intelligent Systems, IEEE*, **29**(1), 30–38, (2014).
- [7] Katarzyna Budzyńska, Magdalena Kacprzak, and Paweł Rembelski, ‘Perseus. software for analyzing persuasion process’, *Fundamenta Informaticae*, **93**(1), 65–79, (2009).
- [8] Elena Cabrio, Graeme Hirst, Serena Villata, and Adam Wyner, eds. *Report of Dagstuhl Seminar on Natural Language Argumentation: Mining, Processing, and Reasoning over Textual Arguments (16161)*, April 17–22, 2016, 2016.
- [9] Elena Cabrio, Serena Villata, and Adam Wyner, eds. *Proceedings of the Workshop on Frontiers and Connections between Argumentation Theory and Natural Language Processing, Forlì-Cesena, Italy, July 21–25, 2014*, volume 1341 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2014.
- [10] Lucas Carstens, Xiuyi Fan, Yang Gao, and Francesca Toni, ‘An overview of argumentation frameworks for decision support’, in *Graph Structures for Knowledge Representation and Reasoning*, 32–49, Springer, (2015).
- [11] Claudette Cayrol and Marie-Christine Lagasquie-Schiex, ‘On the acceptability of arguments in bipolar argumentation frameworks’, in *Symbolic and quantitative approaches to reasoning with uncertainty*, 378–389, Springer, (2005).
- [12] Federico Cerutti, Nava Tintarev, and Nir Oren, ‘Formal arguments, preferences, and natural language interfaces to humans: an empirical evaluation’, in *ECAI*, (2014).
- [13] Joseph Devereux and Chris Reed, ‘Strategic argumentation in rigorous persuasion dialogue’, in *Argumentation in Multi-Agent Systems*, 94–113, Springer, (2010).
- [14] 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).
- [15] Paul E Dunne, Anthony Hunter, Peter McBurney, Simon Parsons, and Michael Wooldridge, ‘Weighted argument systems: Basic definitions, algorithms, and complexity results’, *Artificial Intelligence*, **175**(2), 457–486, (2011).

- [16] Shaheen Fatima, Sarit Kraus, and Michael Wooldridge, *Principles of automated negotiation*, Cambridge University Press, 2014.
- [17] BJ Fogg, *Persuasive Technology: Using Computers to Change What We Think and Do (Interactive Technologies)*, Morgan Kaufmann, 2002.
- [18] Yaakov Gal, Sohan Dsouza, Philippe Pasquier, Iyad Rahwan, and Sherief Abdallah, 'The effects of goal revelation on computer-mediated negotiation', in *Proceedings of the Annual meeting of the Cognitive Science Society (CogSci), Amsterdam, The Netherlands*, (2009).
- [19] Jacob Glazer and Ariel Rubinstein, 'A study in the pragmatics of persuasion: a game theoretical approach', *New perspectives on games and interaction*, 121–140, (2008).
- [20] Guido Governatori, Francesco Olivieri, Simone Scannapieco, Antonino Rotolo, and Matteo Cristani, 'Strategic argumentation is np-complete', *arXiv preprint arXiv:1312.4287*, (2013).
- [21] Ivan Habernal and Iryna Gurevych, 'Which argument is more convincing? analyzing and predicting convincings of web arguments using bidirectional lstm', in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, (2016).
- [22] Christos Hadjilokis, Yiannis Siantos, Sanjay Modgil, Elizabeth Black, and Peter McBurney, 'Opponent modelling in persuasion dialogues.', in *IJCAI*, (2013).
- [23] Emmanuel Hadoux, Aurélie Beynier, Nicolas Maudet, Paul Weng, and Anthony Hunter, 'Optimization of probabilistic argumentation with markov decision models', in *IJCAI*, (2015).
- [24] Johannes Heinrich and David Silver, 'Smooth uct search in computer poker', in *IJCAI*, (2015).
- [25] Anthony Hunter, 'Modelling the persuadee in asymmetric argumentation dialogues for persuasion', in *IJCAI*, pp. 3055–3061, (2015).
- [26] Fred Jelinek, 'Self-organized language modeling for speech recognition', *Readings in speech recognition*, 450–506, (1990).
- [27] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra, 'Planning and acting in partially observable stochastic domains', *Artificial intelligence*, **101**(1), 99–134, (1998).
- [28] Yilin Kang, Ah-Hwee Tan, and Chunyan Miao, 'An adaptive computational model for personalized persuasion', in *Proceedings of the 24th International Conference on Artificial Intelligence*, pp. 61–67. AAAI Press, (2015).
- [29] Sarit Kraus, Katia Sycara, and Amir Evenchik, 'Reaching agreements through argumentation: a logical model and implementation', *Artificial Intelligence*, **104**(1), 1–69, (1998).
- [30] Peter McBurney and Simon Parsons, 'Dialogue games for agent argumentation', in *Argumentation in artificial intelligence*, 261–280, Springer, (2009).
- [31] Sanjay Modgil, Francesca Toni, Floris Bex, Ivan Bratko, Carlos I Chesñevar, Wolfgang Dvořák, Marcelo A Falappa, Xiuyi Fan, Sarah Alice Gaggl, Alejandro J García, et al., 'The added value of argumentation', in *Agreement Technologies*, 357–403, Springer, (2013).
- [32] Marie-Francine Moens, 'Argumentation mining: Where are we now, where do we want to be and how do we get there?', in *Post-proceedings of the forum for information retrieval evaluation (FIRE 2013)*, (2014).
- [33] Nir Oren, Timothy J Norman, and Alun Preece, 'Information based argumentation heuristics', in *Argumentation in Multi-Agent Systems*, 161–174, Springer, (2007).
- [34] Henry Prakken, 'Formal systems for persuasion dialogue', *The Knowledge Engineering Review*, **21**(02), 163–188, (2006).
- [35] Henry Prakken, 'Models of persuasion dialogue', in *Argumentation in artificial intelligence*, 281–300, Springer, (2009).
- [36] Iyad Rahwan, Mohammed I Madakkatel, Jean-François Bonnefon, Ruqiyabi N Awan, and Sherief Abdallah, 'Behavioral experiments for assessing the abstract argumentation semantics of reinstatement', *Cognitive Science*, **34**(8), 1483–1502, (2010).
- [37] Ehud Reiter, Roma Robertson, and Liesl M Osman, 'Lessons from a failure: Generating tailored smoking cessation letters', *Artificial Intelligence*, **144**(1), 41–58, (2003).
- [38] Tjitze Rienstra, Matthias Thimm, and Nir Oren, 'Opponent models with uncertainty for strategic argumentation', in *IJCAI*, pp. 332–338, (2013).
- [39] Ariel Rosenfeld and Sarit Kraus, 'Argumentation theory in the field: An empirical study of fundamental notions', in *Proceedings of the Workshop on Frontiers and Connections between Argumentation Theory and Natural Language Processing, Forlì-Cesena, Italy, July 21-25, 2014.*, (2014).
- [40] Ariel Rosenfeld and Sarit Kraus, 'Providing arguments in discussions based on the prediction of human argumentative behavior', in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pp. 1320–1327, (2015).
- [41] Ariel Rosenfeld and Sarit Kraus, 'Providing arguments in discussions on the basis of the prediction human argumentative behavior', *ACM Transactions on Interactive Intelligent Systems (TiiS)*, (2016). (in press).
- [42] FW Scholz, 'Maximum likelihood estimation', *Encyclopedia of Statistical Sciences*, (1985).
- [43] David Silver and Joel Veness, 'Monte-carlo planning in large pomdps', in *Advances in neural information processing systems*, pp. 2164–2172, (2010).
- [44] Bernard W Silverman, *Density Estimation for Statistics and Data Analysis*, volume 26, CRC press, 1986.
- [45] Christian Stab and Ivan Habernal, 'Detecting argument components and structures', *Report of Dagstuhl Seminar on Debating Technologies (15512)*, **5**(12), 32–32, (2016).
- [46] Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, and Lillian Lee, 'Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions', in *Proceedings of the 25th International Conference on World Wide Web*, pp. 613–624, (2016).
- [47] Yuqing Tang, Elizabeth Sklar, and Simon Parsons, 'An argumentation engine: Argtrust', in *Ninth International Workshop on Argumentation in Multiagent Systems*, (2012).
- [48] Matthias Thimm, 'Strategic argumentation in multi-agent systems', *KI-Künstliche Intelligenz*, **28**(3), 159–168, (2014).
- [49] Tangming Yuan, Viar Svansson, David Moore, and Alec Grierson, 'A computer game for abstract argumentation', in *Proceedings of the 7th Workshop on Computational Models of Natural Argument (CMNA07)*, (2007).

Formalizing Commitment-Based Deals in Boolean Games¹

Sofie De Clercq² and Steven Schockaert³ and Ann Nowé⁴ and Martine De Cock^{2,5}

Abstract. Boolean games (BGs) are a strategic framework in which agents' goals are described using propositional logic. Despite the popularity of BGs, the problem of how agents can coordinate with others to (at least partially) achieve their goals has hardly received any attention. However, negotiation protocols that have been developed outside the setting of BGs can be adopted for this purpose, provided that we can formalize (i) how agents can make commitments and (ii) how deals between coalitions of agents can be identified given a set of active commitments. In this paper, we focus on these two aims. First, we show how agents can formulate commitments that are in accordance with their goals, and what it means for the commitments of an agent to be consistent. Second, we formalize deals in terms of coalitions who can achieve their goals without help from others. We show that verifying the consistency of a set of commitments of one agent is Π_2^P -complete while checking the existence of a deal in a set of mutual commitments is Σ_2^P -complete. Finally, we illustrate how the introduced concepts of commitments and deals can be used to achieve game-theoretical properties of the deals and to configure negotiation protocols.

1 Introduction

Boolean games (BGs) are a game-theoretic framework which uses propositional logic to represent the goals of agents in a compact way [24]. A key feature of BGs is that each agent controls the truth value of a subset of the atoms from which these goals are built; these atoms are referred to as the action variables of the agent. In standard BGs, goals are of a binary nature [24]. In the context of negotiation, however, it is usually more natural to consider prioritized goal bases [6, 14], as these allow agents to partially concede. The basic intuition underlying BGs is illustrated in the next example.

Example 1

Suppose there are four agents, denoted by 1, 2, 3 and 4, representing four nations. Each agent i controls an action variable d_i . If agent i sets its action variable d_i to true, this means that i will disarm its nation. Nation 1 considers nation 2 a threat, nation 3 an ally and nation 4 irrelevant for its military strategy. It believes to be safe if either nation 2 disarms or nations 1 and 3 both keep their arms, i.e. $d_2 \vee (\neg d_1 \wedge \neg d_3)$. Nation 2 considers nation 1 as the only real threat, but prefers to disarm itself due to the associated costs of maintaining

its arms. Therefore, its highest priority goal is $d_1 \wedge d_2$. Nation 3 strongly believes in the possibility of an alien invasion and prefers all nations to be armed, i.e. its highest priority goal is $\neg d_1 \wedge \neg d_2 \wedge \neg d_3 \wedge \neg d_4$. The pacifistic nation 4's first priority is the disarmament of all nations, i.e. $d_1 \wedge d_2 \wedge d_3 \wedge d_4$.

Although Boolean games have been widely studied in recent years, leading among others to the characterization of numerous solution concepts, the literature on BGs provides surprisingly few tools for agents to actually coordinate towards mutually beneficial agreements (see Section 5). The broader literature on multi-agent systems, however, has provided numerous negotiation protocols [34, 26, 3, 18, 33]. From a high-level point of view, many of these protocols are based on agents formulating commitments, intuitively encoding what they are prepared to offer in return for their goals being (partially) fulfilled. After a number of rounds, in which agents may progressively weaken their stance, the agents may end up with a set of mutual commitments which are such that a deal can be made. There are many technical details that need to be specified as part of a negotiation protocol (related e.g. to how agents communicate), but most of these are not dependent on how the agent's goals are encoded. In particular, to adapt existing negotiation protocols to the BG setting, it suffices to specify how agents can formulate commitments (i.e. proposal submission) and how deals based on these commitments can be made (i.e. agreement formation). The incorporation of the introduced notions into existing protocols is illustrated in Section 4.3.

Central to the discussion in this paper is the notion of a commitment. In the literature, a commitment is commonly stated as $(i; j; ante; con)$, with the interpretation that agent i commits to agent j to bring about con when $ante$ is made true by the other agents [37]. For instance, in the context of Example 1, a sensible commitment for agent 2 would be $(2; 1; d_1; d_2)$: if agent 1 disarms, agent 2 is prepared to do this as well. Commitments provide an intuitive way to formulate a propositional proposal and at the same time capture the fact that particular action variables are controlled by particular agents. This makes commitments a natural fit for the framework of BGs. Moreover, by identifying creditors, a commitment-based protocol allows the formation of deals between a subset of agents, i.e. the formation of coalitions. For instance, suppose in Example 1 that agent 1 and 2 respectively formulate the commitments $(1; 2; d_2; \top)$ and $(2; 1; d_1; d_2)$, where the former communicates agent 1's willingness to play any strategy if agent 2 disarms. Note such a commitment merely informs agent 2 of possibilities, but yields no guarantees, since \top is brought about by default. The commitments of agents 1 and 2 lead to a possible deal: they can form a coalition $\{1, 2\}$ and play $\{d_1, d_2\}$, i.e. both nations disarm. To confirm the deal, however, agent 1 has to make a stronger commitment, i.e. it must specifically commit to bring about d_1 if d_2 is brought about. If this deal between agents 1 and 2 is closed, the BG can be reduced, allowing the remaining agents to update their goals: agent 3's

¹ This research was funded by a Research Foundation-Flanders project.

² Dept. of Applied Mathematics, CS and Statistics, Ghent University, Ghent, Belgium, {SofieR.DeClercq, Martine.DeCock}@ugent.be

³ School of Computer Science and Informatics, Cardiff University, Cardiff, UK, SchockaertS1@cardiff.ac.uk

⁴ Computational Modeling Lab, Vrije Universiteit Brussel, Brussels, Belgium, Ann.Nowe@vub.ac.be

⁵ Center for Data Science, University of Washington, Tacoma, US, MDeCock@uw.edu

highest priority goal is no longer achievable and it should now turn to its lower priority goals. Agent 4's highest priority goal is reduced to $d_3 \wedge d_4$, i.e. in order to disarm all nations it remains to disarm itself and convince agent 3 to disarm as well. Note that by identifying deals between coalitions of agents, a global consensus is not required to obtain local deals. Moreover, the reduction of the BG induced by the previous deals facilitates matters for the agents that were not yet able to close a deal. These advantages are especially important in large-scale games, which many real-life applications are.

An important requirement for commitments is that agents should be able to guarantee that they can fulfill them. For instance, suppose that agent 1 in Example 1 communicates the commitments $(1; 2; d_2; d_1)$ and $(1; 3; \neg d_3; \neg d_1)$. Clearly, in case agent 2 and 3 play $\{d_2, \neg d_3\}$, the agent cannot play a strategy such that all its commitments are fulfilled. In Section 3 we formalize a notion of consistency, which captures the intuition that agents should not make commitments that cannot be jointly fulfilled. As we will show, checking whether a given set of commitments is consistent is a Π_2^P -complete problem.

Given a set of commitments, the main inference task we consider is verifying whether any corresponding deals can be made. Some of the issues underlying this process are illustrated in the next example.

Example 2

Suppose the agents in Example 1 communicate the following commitments:

$$\begin{array}{ll} (1; 2; d_2; \top) & (3; \{1, 2, 4\}; \neg d_1 \wedge \neg d_2 \wedge \neg d_4; \neg d_3) \\ (1; 3; \neg d_3; \neg d_1) & (4; \{1, 2, 3\}; d_1 \wedge d_2 \wedge d_3; d_4) \\ (2; 1; d_1; d_2) & \end{array}$$

Intuitively, the commitments agents 1 and 2 make to each other allow a deal between them to both disarm: $\{d_1, d_2\}$. However, as stated earlier, agent 1 has not specifically committed to bring about d_1 when d_2 is brought about. Therefore, we cannot consider this a confirmed deal. Moreover, agent 1's commitment to agent 3 blocks the possible deal with 2 when agent 3 decides to play $\neg d_3$. As a result, agents 1 and 2 cannot form a coalition by themselves based on the current set of commitments, as they would be reliant on what agent 3 subsequently decides.

Note that to identify deals, the control assignment of the BG should be taken into account, i.e. it does not suffice to check the satisfiability of conjunctions of formulas corresponding to goals of coalitions. Suppose, for instance, that agents 1 and 2 want agent 3 to disarm, i.e. the conjunction of their highest priority goal is satisfied if d_3 holds. Then no deal can be reached unless agent 3 can be convinced to set d_3 to true. As we will show, as a result, the problem of checking whether a deal can be made given a set of commitments is Σ_2^P -complete. To the best of our knowledge, this paper is the first to study commitment-based deals in BGs.

The paper is structured as follows. First, we give some background on BGs in Section 2. In Section 3 we formalize commitments in BGs, defining important concepts, including the consistency of a set of commitments, which guarantees that agents can honour their commitments, irrespective of the strategies of the other agents. After investigating the computational complexity of verifying consistency in Section 3.1, we explain how an agent can formulate commitments that accord with a single goal or with a prioritized goal base in respectively Sections 3.2 and 3.3. Then we formalize how agents can identify deals based on a set of commitments and investigate the computational complexity in Section 4.1. Next, we illustrate how our concepts can be used to guarantee game-theoretical properties of the

deals in Section 4.2 and how they can be implemented in existing protocols in Section 4.3. We discuss related work in Section 5 and present our conclusion and interesting questions for further research in Section 6. The proofs of all results are available in an online appendix⁶.

2 Preliminaries

The propositional language L_Φ is built from a finite set of atoms Φ in the usual way. We write $Lit(\Phi) = \Phi \cup \{\neg p \mid p \in \Phi\}$. An interpretation of Φ is defined as a subset ν of $Lit(\Phi)$ such that for every atom $p \in \Phi$ either $p \in \nu$ or $\neg p \in \nu$. We denote the set of all interpretations of Φ as $Int(\Phi)$. An interpretation can be extended to L_Φ in the usual way. We write $\nu \models \varphi$ to denote that formula φ is true in interpretation ν . Whenever we write an interpretation ν where a formula is expected, this should be interpreted as the conjunction of ν 's literals. For two formulas φ and $\psi \in L_\Phi$, it holds that φ entails ψ , denoted $\varphi \models \psi$, iff for every interpretation ν it holds that $\nu \models \varphi$ whenever $\nu \models \psi$. We say that a variable p is irrelevant in a formula if there exists an equivalent formula in which p does not occur [30]. The relevant variables of a formula γ are denoted as $DepVar(\gamma)$. We say that φ and ψ are *equivalent modulo δ* , denoted $\varphi \equiv \psi \pmod{\delta}$, iff $\varphi \wedge \delta$ is equivalent with $\psi \wedge \delta$ [30]. We use the variant of Boolean games from [14].

Definition 1 (Boolean Game with Priorities)

A Boolean game (BG) with priorities is a tuple $G = (N, (\Phi_i)_{i \in N}, (\delta_i)_{i \in N}, (\Gamma_i)_{i \in N})$, where N is a finite set of agents, Φ_i is a finite set of atoms such that $\Phi_i \cap \Phi_j = \emptyset$ for $j \neq i$, δ_i is a consistent formula in L_{Φ_i} , and $\Gamma_i = \{\gamma_i^1; \dots; \gamma_i^p\}$ is i 's prioritized goal base. We write $\Phi = \bigcup_{i \in N} \Phi_i$ and $\delta = \bigwedge_{i \in N} \delta_i$. The formula $\gamma_i^m \in L_\Phi$ is agent i 's goal of priority m . We assume w.l.o.g. that every agent has p priority levels.

Definition 1 is a particular case of generalized BGs [6] in which the preference relations are total, but with the addition of constraints δ_i [8]. The set Φ contains all action variables (or atoms). Agent i controls the truth value of the atoms in Φ_i , with the restriction that δ_i must be satisfied. We write $S_i = \{\nu_i \in Int(\Phi_i) \mid \nu_i \models \delta_i\}$ for the set of permissible strategies of agent i . A non-empty subset of N is called a coalition. We straightforwardly extend definitions w.r.t. agents to coalitions, e.g. Φ_J is the set of action variables controlled by a coalition J , S_J denotes the permissible joint strategies. We denote singleton coalitions $\{i\}$ as i when there can be no confusion. By convention, goals are ordered from high (level 1) to low priority (level p).

Example 1 (Continued)

We have $N = \{1, 2, 3, 4\}$, $\Phi_i = \{d_i\}$ and $\delta_i = \top$ for each $i \in N$. Consequently, $S_i = \{d_i, \neg d_i\}$, i.e. each agent can either disarm or not. The prioritized goal bases in the BG are:

$$\begin{array}{l} \Gamma_1 = \{d_2 \vee (\neg d_1 \wedge \neg d_3); \neg d_1\}, \Gamma_2 = \{d_1 \wedge d_2; \neg d_2\} \\ \Gamma_3 = \{\neg d_1 \wedge \neg d_2 \wedge \neg d_3 \wedge \neg d_4; (\neg d_1 \vee \neg d_2 \vee \neg d_4) \wedge \neg d_3\} \\ \Gamma_4 = \{d_1 \wedge d_2 \wedge d_3 \wedge d_4; d_4\} \end{array}$$

This means, e.g. for agent 2, that its first priority is for agents 1 and 2 to disarm and its second priority is to arm.

We now define the concept of relevant agents for a formula, which is related to the concept of relevant agents for another agent described in [7].

⁶ <http://www.cwi.ugent.be/sofie/ECAI16appendix.pdf>

Definition 2 (Relevant Agents for a Formula)

The set of relevant agents $RelAg(\varphi)$ for $\varphi \in L_\Phi$ is defined as

$$\bigcup_{i \in N} \left\{ \left(\bigcap_{\psi \equiv \varphi \pmod{\delta}} DepVar(\psi) \right) \cap \Phi_i \neq \emptyset \right\}$$

In other words, the relevant agents for a formula are the agents controlling its relevant variables, taking into account the constraint δ . For instance, in Example 1, the relevant agents for $(d_1 \wedge d_2) \wedge (\neg d_3 \vee d_3)$ are $\{1, 2\}$.

Definition 3 (Outcome)

An interpretation of Φ satisfying δ is called an outcome of G . We denote the set of all outcomes as S_N .

An outcome ν corresponds to a tuple (ν_1, \dots, ν_n) with $\nu_i \in S_i$ for every $i \in N$. We write $\nu_J = \nu \cap Lit(\Phi_J)$ for the restriction of ν to the strategies of agents in coalition J . The restriction of ν to the agents outside J is denoted as ν_{-J} . For disjoint coalitions J and J' , $\nu \in S_J$ and $\nu' \in S_{J'}$, we write $(\nu, \nu') \in S_{J \cup J'}$ to denote their joint strategy. We also use the notation (ν, ν') if J and J' are not disjoint, but $\nu_i = \nu'_i$ for every $i \in J \cap J'$. For the ease of presentation, we define a numerical utility in $[0, 1]$, which is determined by the highest priority for which the corresponding goal is satisfied by ν .

Definition 4 (Utility Function)

Let G be a BG. For each $i \in N$ and $\nu \in S_N$, the utility of i in ν is defined as:

$$u_i(G, \nu) = \frac{p + 1 - \min\{k \mid 1 \leq k \leq p, \nu \models \gamma_i^k\}}{p}$$

with $\min \emptyset = p + 1$.

Note that if ν does not satisfy any goal, the utility is 0, while it is 1 iff the first priority goal is satisfied. In Example 1, for instance, the utility of agent 1 is 1 for every outcome ν in which the coalition $\{1, 3\}$ plays the joint strategy $\{\neg d_1, \neg d_3\}$, i.e. if both nations 1 and 3 decide against disarmament. Note that there exist alternative ways to extract utilities from prioritized goal bases [6], for which similar results as the ones presented in this paper can be obtained.

A well-known solution concept in BGs is the pure Nash equilibrium. This notion is based on best responses: we say that $\nu \in S_i$ is a *best response* to $\nu^* \in S_{-i}$, written $\mathcal{BR}(G, \nu, \nu^*)$, iff for every $\nu' \in S_i$ it holds that $u_i(G, (\nu, \nu^*)) \geq u_i(G, (\nu', \nu^*))$. Intuitively, this means that given the strategies of the other agents, an agent is not better off by deviating from its current strategy. In Example 1, for instance, agent 2's first priority is the disarmament of agents 1 and 2. Therefore, if agent 1 disarms, agent 2's best response is to disarm as well.

Definition 5 (Pure Nash Equilibrium)

An outcome ν is a *pure Nash equilibrium (PNE)* of the BG G iff $\mathcal{BR}(G, \nu_i, \nu_{-i})$ holds for every $i \in N$.

In Example 1, for instance, the outcomes $\{d_1, d_2, \neg d_3, d_4\}$ and $\{\neg d_1, \neg d_2, \neg d_3, d_4\}$ are PNEs.

3 Formalizing Commitments in BGs

In this section, we adapt the notion of commitment from [37] to the context of BGs. Among others, we analyze how agents can formalize consistent commitments based on the goal they want to achieve. Throughout this section, we will assume that $G = (N, (\Phi_i)_{i \in N}, (\delta_i)_{i \in N}, (\Gamma_i)_{i \in N})$ is a BG.

Definition 6 (Commitment in BG)

The tuple $c = (deb(c); cred(c); ante(c); con(c))$ is called a *commitment* in G if $deb(c) \in N$, $cred(c) \subseteq N \setminus \{deb(c)\}$, $ante(c)$ is a formula such that $RelAg(ante(c)) = cred(c)$, $ante(c) \wedge \delta \not\models \perp$, $con(c)$ is a formula containing only variables from $\Phi_{deb(c)}$, and $con(c) \wedge \delta \not\models \perp$.

Intuitively, a commitment c describes a state of affairs in which the debtor $deb(c)$ commits to the creditors⁷ $cred(c)$ to bring about the consequent $con(c)$ if the antecedent $ante(c)$ is satisfied. To exclude meaningless commitments, we assume that $con(c)$ contains only variables from $\Phi_{deb(c)}$, i.e. we do not allow an agent to commit to anything outside its own control. Moreover, we assume that $con(c)$ is consistent with $\delta_{deb(c)}$, meaning that there exists at least one strategy $\nu \in S_{deb(c)}$ such that $\nu \models con(c)$. This restriction, which excludes impossible promises, corresponds to the *consistency postulate* for active commitments in [37]. We also exclude the debtor as one of its own creditors, as the commitment is meaningless to the other creditors if the debtor itself can control whether or not the antecedent is fulfilled. The condition $RelAg(ante(c)) = cred(c)$ makes the formulation of other definitions more convenient, but is not an explicit requirement for commitments [37]. We demand that $ante(c)$ is satisfiable w.r.t. δ , in line with the *nonvacuity postulate* for active commitments [37].

If $ante(c) \equiv \top$ (and thus $cred(c) = \emptyset$), the commitment is called unconditional. The case where $con(c) \equiv \top$ contradicts the first postulate in [37], which says that a commitment is *discharged* (no longer active) when the consequent holds. This makes sense, since such a commitment makes no guarantees to the creditor. However, as illustrated in Section 1, we will use such commitments to allow an agent to express that it is prepared to bring about any of its strategies when the antecedent is satisfied. This will be useful in our formalization of a possible deal, as will become clear in Definition 12. To obtain a confirmed deal though, the agent will have to make a stronger commitment, as will become clear in Definition 14.

In the following example and throughout this paper, subscripts are used to indicate the controlling agent of each action variable.

Example 3

In the context of Example 1, consider the pair of commitments $(1; 2; d_2; d_1)$ and $(1; 3; \neg d_3; \neg d_1)$. In the first commitment, agent 1 commits to disarm, i.e. bring about d_1 , when agent 2 disarms, i.e. brings about d_2 . In the second commitment, agent 1 commits to arm, i.e. bring about $\neg d_1$, when agent 3 arms, i.e. brings about $\neg d_3$.

Note that the commitments in Example 3 cannot be fulfilled simultaneously. We will call such commitments inconsistent and formalize the concept of consistent commitments in the following section.

3.1 Consistency of Commitments

Given that commitments are meant to be binding, it is important that an agent can jointly fulfil the set of all commitments it has made. To formalize this notion of consistency, we introduce the following abbreviations:

$$\begin{aligned} deb(\mathcal{C}) &= \bigcup_{c \in \mathcal{C}} deb(c), & cred(\mathcal{C}) &= \bigcup_{c \in \mathcal{C}} cred(c), \\ ante(\mathcal{C}) &= \bigwedge_{c \in \mathcal{C}} ante(c), & con(\mathcal{C}) &= \bigwedge_{c \in \mathcal{C}} con(c). \end{aligned}$$

⁷ In contrast to [37], we allow multiple creditors.

Definition 7 (Consistent Commitments)

Suppose \mathcal{C} is a set of commitments of agent i , i.e. $\text{deb}(\mathcal{C}) = i$. Then \mathcal{C} is consistent iff for every non-empty subset \mathcal{C}' of \mathcal{C} :

$$(\text{ante}(\mathcal{C}') \wedge \delta) \text{ is consistent} \Rightarrow (\text{con}(\mathcal{C}') \wedge \delta) \text{ is consistent}$$

Intuitively, whenever there exists a strategy of the creditors of a subset \mathcal{C}' of \mathcal{C} that satisfies the antecedents in \mathcal{C}' , there should exist a strategy of the debtor of \mathcal{C}' that satisfies all consequents in \mathcal{C}' . Note that it is not sufficient for consistency that every pair of commitments in the set \mathcal{C} is consistent, as illustrated in the following example.

Example 4

Every pair in the following set of commitments is consistent, but the three commitments together are not: $(1; 2; d_2; d_1 \leftrightarrow \neg b_1)$, $(1; 3; d_3; b_1 \leftrightarrow \neg c_1)$, $(1; 4; d_4; c_1 \leftrightarrow \neg d_1)$.

Proposition 1

Deciding whether a given set of commitments with fixed debtor is consistent, is Π_2^P -complete.

3.2 Relating Goals and Commitments

In this section, we investigate how an agent with a goal γ can formulate a corresponding commitment.

Example 5

Reconsider the context of Example 1. Let $\gamma = d_1 \wedge d_2 \wedge d_3 \wedge d_4$ be a goal of agent 4. To achieve this goal, agent 4 can make the commitment $c = (4; \{1, 2, 3\}; d_1 \wedge d_2 \wedge d_3; d_1)$, which intuitively expresses that agent 4 commits to play a strategy that involves its disarmament if agents 1, 2 and 3 do the same.

We now formalize how a commitment can be created to match a propositional goal.

Definition 8 (Commitments Coinciding with Goal)

The set \mathcal{C} of commitments with debtor i coincides with the goal γ of agent i iff for every $\nu \in \mathcal{S}_N$:

$$(\nu \models \gamma) \Leftrightarrow ((\exists c \in \mathcal{C} : (\nu_{-i} \models \text{ante}(c)) \wedge (\nu_i \models \text{con}(c))) \wedge (\nu_i \models \text{con}(\{c \in \mathcal{C} \mid \nu_{-i} \models \text{ante}(c)\}))) \quad (1)$$

The ‘ \Rightarrow ’ direction expresses that the set of commitments *covers* the goal γ , i.e. for every outcome ν that satisfies γ , i has an active commitment c corresponding to ν , which does not result in the violation of any other commitment of i . The ‘ \Leftarrow ’ direction expresses that the set of commitments *is covered* by the goal γ , i.e. any outcome made possible by a commitment of i results in the satisfaction of γ .

Proposition 2

For any formula $\gamma \in L_\Phi$ and agent $i \in N$, there exists a consistent set \mathcal{C} of commitments with debtor i that coincides with γ . Moreover, if γ can be rewritten as a conjunction of literals, then \mathcal{C} can be chosen as a singleton.

It is easy to verify that some goals cannot coincide with a singleton, e.g. the goal $(d_1 \leftrightarrow d_2)$ requires two separate commitments. We illustrate the specification of commitments coinciding with goals in the following example.

Example 6

Suppose that $\gamma = (d_1 \wedge d_2) \vee (\neg d_1 \wedge \neg d_3)$ is a goal of agent 1. The set $\mathcal{C} = \{(1; \{2, 3\}; d_2 \wedge d_3; d_1), (1; \{2, 3\}; \neg d_2 \wedge \neg d_3; \neg d_1), (1; \{2, 3\}; d_2 \wedge \neg d_3; \top)\}$ coincides with γ .

3.3 Commitments for Prioritized Goal Bases

We now relate prioritized goal bases and commitments.

Definition 9 (Commitments Guaranteeing Utility)

Let $\Gamma = \{\gamma^1; \dots; \gamma^p\}$ be a prioritized goal base of i . The set \mathcal{C} of commitments with debtor i guarantees utility k iff for each $\nu \in \mathcal{S}_N$:

$$u_i(G, \nu) \geq k \Leftrightarrow ((\nu_i \models \text{con}(\{c \in \mathcal{C} \mid \nu_{-i} \models \text{ante}(c)\})) \wedge (\exists c \in \mathcal{C} : (\nu_{-i} \models \text{ante}(c)) \wedge (\nu_i \models \text{con}(c)))) \quad (2)$$

A straightforward way to construct a set of commitments that guarantees utility k is by constructing the set of commitments that coincides with formula $\bigvee_{m=1}^{p-kp+1} \gamma^m$, using the construction from Proposition 2. Note that for this particular choice, the ‘ \Rightarrow ’ direction of (2) also holds. However, we do not require this direction in Definition 9, due to incompatibility with Definition 10, which formalizes when an agent’s commitments are in line with its best responses.

Definition 10 (Commitment Respecting Best Responses)

The commitment c with debtor i respects i ’s best responses in G iff for each $\nu \in \mathcal{S}_N$:

$$(\nu_{-i} \models \text{ante}(c)) \wedge (\nu_i \models \text{con}(c)) \Rightarrow \text{BR}(G, \nu_i, \nu_{-i}) \quad (3)$$

A set of commitments with debtor i respects i ’s best responses iff every commitment in the set does.

Example 7

Reconsider the context of Example 1. Suppose the pacifistic nation 4 has the goal base $\Gamma_4 = \{d_1 \wedge d_2 \wedge d_3 \wedge d_4; \neg d_4\}$, i.e. the first priority of nation 4 is global disarmament, but in case this turns out to be unachievable, it distrusts the other nations and prefers to maintain its arms. The commitment $(4; \emptyset; \top; \neg d_4)$ guarantees utility 0.5, but does not respect 4’s best responses: if the other nations decide to disarm, 4 would have been better off by disarming as well. The commitment $(4; \{1, 2, 3\}; d_1 \wedge d_2 \wedge d_3; d_4)$ guarantees utility 1 and respects agent 4’s best responses. The pair of commitments $(4; \{1, 2, 3\}; d_1 \wedge d_2 \wedge d_3; d_4)$ and $(4; \{1, 2, 3\}; \neg d_1 \vee \neg d_2 \vee \neg d_3; \neg d_4)$ guarantees utility 0.5 and respects agent 4’s best responses.

Note that a set \mathcal{C} of commitments with debtor i which respect i ’s best responses is not necessarily consistent. Consider for instance a BG with $\Gamma_1 = \{(a_1 \vee b_1) \wedge a_2\}$ and $\delta_1 = \neg(a_1 \wedge b_1)$. Agent 1’s commitments $(1; 2; a_2; a_1)$ and $(1; 2; a_2; b_1)$ both respect its best responses, but are not consistent. However, we can show the following result.

Proposition 3

Let $\Gamma = \{\gamma^1; \dots; \gamma^p\}$ be the goal base of some agent i . For any $k \in \{\frac{1}{p}, \dots, \frac{p}{p}\}$ such that $\bigvee_{m=1}^{p-kp+1} \gamma^m \wedge \delta \not\models \perp$ there exists a non-empty consistent set \mathcal{C} of commitments with debtor i that guarantees utility k and respects i ’s best responses.

4 Commitment-based Deals in BGs

From a high-level point of view, many negotiation protocols are based on agents formulating proposals or commitments, intuitively encoding what they are prepared to offer in return for their goals being (partially) fulfilled. After a number of rounds, in which agents may progressively weaken their stance, the agents may end up with a set of mutual commitments which are such that a deal can be made. Note that a deal does not require the involvement of all agents. For instance, suppose two neighbouring nations 1 and 2’s first priority it to both disarm. In order to play the coalition strategy $\{d_1, d_2\}$, they

do not need the approval of all other nations, as the action variables involved in the deal are controlled by the agents that closed the deal.

As deals can be closed between coalitions of agents, the BG can iteratively be reduced based on the chosen strategies of the agents who have already closed a deal. To formalize this, we use the notion of a formula $\varphi \in L_{\Phi}$ being conditioned by an interpretation ν of $\Phi' \subseteq \Phi$ [13], written as $\text{cond}(\varphi, \nu)$. The formula $\text{cond}(\varphi, \nu) \in L_{\Phi \setminus \Phi'}$ is obtained from φ in the following way: for every atom $p \in \Phi'$ such that $\neg p \in \nu$ we replace every occurrence of p in φ by \perp , and for every atom $p \in \Phi'$ such that $p \in \nu$ we replace every occurrence of p in φ by \top . Next, the tautologies $\neg \top \equiv \perp$, $\neg \perp \equiv \top$, $(\top \wedge \varphi) \equiv \varphi$, $(\top \vee \varphi) \equiv \top$, $(\perp \wedge \varphi) \equiv \perp$ and $(\perp \vee \varphi) \equiv \varphi$ are iteratively used to simplify the formula.

Definition 11 (Reduced BG)

Let $G = (N, (\Phi_i)_{i \in N}, (\delta_i)_{i \in N}, (\Gamma_i)_{i \in N})$, J a coalition of N and $\nu \in S_J$. The reduced BG is defined as $\text{red}(G, J, \nu) = (N \setminus J, (\Phi_i)_{i \in N \setminus J}, (\delta_i)_{i \in N \setminus J}, (\text{cond}(\Gamma_i, \nu))_{i \in N \setminus J})$, with $\text{cond}(\Gamma_i, \nu) = \{\text{cond}(\gamma_i^1, \nu), \dots, \text{cond}(\gamma_i^p, \nu)\}$ for $i \in N \setminus J$.

The next example illustrates this concept, which generalizes the notion of a *projection* from [7].

Example 8

Reconsider the context of Example 1, i.e. the BG G with four agents. Each agent i controls one variable, i.e. $\Phi_i = \{d_i\}$. Consider the coalition $\{1, 2\}$ with the joint strategy to disarm, i.e. $\{d_1, d_2\}$. Then the reduced game $\text{red}(G, \{1, 2\}, \{d_1, d_2\})$ consists of two agents (namely agents 3 and 4), with Φ_3 and Φ_4 as in G . The goal bases of agents 3 and 4 are reduced from

$$\begin{aligned} \Gamma_3 &= \{-d_1 \wedge \neg d_2 \wedge \neg d_3 \wedge \neg d_4; (\neg d_1 \vee \neg d_2 \vee \neg d_4) \wedge \neg d_3\} \\ \Gamma_4 &= \{d_1 \wedge d_2 \wedge d_3 \wedge d_4; d_4\} \end{aligned}$$

to

$$\Gamma_3 = \{\perp; \neg d_4 \wedge \neg d_3\} \quad \Gamma_4 = \{d_3 \wedge d_4; d_4\}$$

Note that in particular, agent 3 can no longer achieve utility 1.

The following results are straightforward to prove.

Proposition 4

Let G be a BG with J and J' two disjoint coalitions, $\nu \in S_J$ and $\nu' \in S_{J'}$. It holds that $\text{red}(\text{red}(G, J, \nu), J', \nu') = \text{red}(G, J \cup J', (\nu, \nu'))$.

Proposition 5

Let G be a BG, J a coalition and $\nu \in S_J$. For $i \in N \setminus J$ and $\nu' \in S_{N \setminus J}$ it holds that $u_i(G, (\nu, \nu')) = u_i(\text{red}(G, J, \nu), \nu')$.

To illustrate Proposition 5, note that in Example 8 agent 4's utility in $\{d_1, d_2, d_3, d_4\}$ in G is 1, which is the same as its utility in $\{d_3, d_4\}$ in the reduced game $\text{red}(G, \{1, 2\}, \{d_1, d_2\})$. The following corollary follows immediately from Proposition 5.

Corollary 6

Let G be a BG, J a coalition and $\nu \in S_J$. For $i \in N \setminus J$, $\nu' \in S_i$ and $\nu^* \in S_{N \setminus (J \cup \{i\})}$ it holds that $\mathcal{BR}(G, \nu', (\nu, \nu^*)) \Leftrightarrow \mathcal{BR}(\text{red}(G, J, \nu), \nu', \nu^*)$.

Intuitively, Corollary 6 expresses that the reduced game preserves the best responses of the original game. For instance, in Example 8 the strategy $\{d_4\}$ is agent 4's unique best response to $\{d_3\}$ in $\text{red}(G, \{1, 2\}, \{d_1, d_2\})$, as well as its unique best response to $\{d_1, d_2, d_3\}$ in the original game G . As a consequence, PNEs are also preserved.

Corollary 7

Let G be a BG with $J \subset N$, ν a strategy of J and ν' a strategy of $N \setminus J$. It holds that (ν, ν') is a PNE of G iff ν' is a PNE of $\text{red}(G, J, \nu)$ and for every $i \in J$ it holds that $\mathcal{BR}(G, \nu_i, (\nu_{-i}, \nu'))$.

For instance, in Example 8 the outcome $\{d_1, d_2, d_3, d_4\}$ is a PNE of G . It also holds that $\{d_3, d_4\}$ is a PNE of the reduced game $\text{red}(G, \{1, 2\}, \{d_1, d_2\})$ and d_1 and d_2 are best responses to $\{d_2, d_3, d_4\}$ respectively $\{d_1, d_3, d_4\}$ in G . Similarly, $\{d_1, d_2\}$ is a PNE of $\text{red}(G, \{3, 4\}, \{d_3, d_4\})$ and d_3 and d_4 are best response to $\{d_1, d_2, d_4\}$ respectively $\{d_1, d_2, d_3\}$ in G .

Now that we have explained how deals between coalitions can be used to reduce the BG, it remains to formalize how agents can identify these deals based on a given set of commitments.

4.1 Identifying Deals

Given a set of commitments, intuitively, a possible deal corresponds to a coalition with a joint strategy such that all agents in the coalition actively support the coalition strategy through their commitments. Throughout this paper we assume that commitments are either common knowledge or known by one central entity.

Definition 12 (Possible Deal)

Let $\mathcal{C} = (C_i)_{i \in N}$ be a tuple with C_i a set of commitments for every agent $i \in N$. The coalition J and strategy $\nu \in S_J$ correspond to a possible deal based on \mathcal{C} iff:

$$\forall i \in J, \exists c \in C_i : ((\nu_{-i} \wedge \delta) \models \text{ante}(c)) \wedge (\nu_i \models \text{con}(c)) \quad (4)$$

Intuitively, Definition 12 expresses that a deal between a coalition of agents must be backed up by an active commitment of every participating agent, i.e. no agent agrees to a deal without benefiting from it. The computational complexity of the associated decision problem is situated at the second level of the polynomial hierarchy.

Proposition 8

Deciding whether there exists a possible deal based on a given set \mathcal{C} of commitments is Σ_2^P -complete.

Condition (4) is obviously a necessary condition to reach an agreement. However, participating in a possible deal may require the agent to play a strategy which is incompatible with some of its other commitments. Moreover, a commitment might make a deal possible, yet not be strong enough to guarantee it. We illustrate these issues with an example.

Example 9

Reconsider the context of Example 1 and Example 2: a BG with $N = \{1, 2, 3, 4\}$, $\Phi_i = \{d_i\}$, $\delta = \top$, $\Gamma_1 = \{(\neg d_1 \wedge \neg d_3) \vee d_2; \dots\}$, $\Gamma_2 = \{d_1 \wedge d_2; \dots\}$, $\Gamma_3 = \{-d_1 \wedge \neg d_2 \wedge \neg d_3 \wedge \neg d_4; \dots\}$ and $\Gamma_4 = \{d_1 \wedge d_2 \wedge d_3 \wedge d_4; \dots\}$. Suppose the agents announce the following commitments:

$$\begin{aligned} C_1 &= \{c_1 = (1; 3; \neg d_3; \neg d_1), c'_1 = (1; 2; d_2; \top)\} \\ C_2 &= \{c_2 = (2; 1; d_1; d_2)\} \\ C_3 &= \{c_3 = (3; \{1, 2, 4\}; \neg d_1 \wedge \neg d_2 \wedge \neg d_4; \neg d_3)\} \\ C_4 &= \{c_4 = (4; \{1, 2, 3\}; d_1 \wedge d_2 \wedge d_3; d_4)\} \end{aligned}$$

These sets are consistent and respect each agent's best responses. The unique possible deal is $(\{1, 2\}, \{d_1, d_2\})$: c'_1 and c_2 back up this deal. However, if agent 3 decides to bring about $\neg d_3$, agent 1 cannot play $\{d_1\}$ without violating its commitment c_1 . Moreover, agent 1 has not specifically committed to bring about d_1 when d_2 is brought about.

To address these issues, we introduce the notion of *confirmed deals*. To this end, we first define the concept of a stable set of commitments, which captures the intuition of a coalition whose commitments entail the willingness of all members to participate in a corresponding deal, i.e. to jointly satisfy the antecedents of the commitments that are part of the deal.

Definition 13 (Stable Set of Commitments)

A set of commitments \mathcal{C} with $J = \text{deb}(\mathcal{C})$ is called *stable* iff $\text{con}(\mathcal{C}) \wedge \delta \models \text{ante}(\mathcal{C})$ and \mathcal{C} has at least one playable coalition strategy, i.e. $\exists \nu \in \mathcal{S}_J$ such that $\nu \models \text{con}(\mathcal{C})$.

In Example 9 there are no stable sets of commitments. If agent 1 replaces the commitment c_1 with $(1; \{2, 3\}; \neg d_3 \wedge \neg d_2; \neg d_1)$ and c'_1 with $c''_1 = (1; 2; d_2; d_1)$, then $\{c'_1, c_2\}$ is a stable set of commitments with $\{d_1, d_2\}$ its unique playable coalition strategy.

As the consequent of any commitment c contains only variables of $\Phi_{\text{deb}(c)}$ — see Definition 6 — it follows that $\nu \models \text{con}(\mathcal{C})$ is equivalent to $\nu_i \models \text{con}(\{c \in \mathcal{C} \mid \text{deb}(c) = i\})$ for every $i \in \text{deb}(\mathcal{C})$. Therefore, one can unambiguously speak of a playable strategy of an agent in $\text{deb}(\mathcal{C})$.

Definition 14 (Confirmed Deal)

Let $\mathcal{C} = (\mathcal{C}_i)_{i \in N}$ be such that each \mathcal{C}_i only contains commitments with debtor i and let $\mathcal{C}' \subseteq \mathcal{C}$ be a stable set of commitments with $J = \text{deb}(\mathcal{C}')$. Then J and \mathcal{C}' form a *confirmed deal* based on \mathcal{C} iff \mathcal{C}' has at least one safely playable coalition strategy, i.e. there is some $\nu' \in \mathcal{S}_J$ such that $\nu' \models \text{con}(\mathcal{C}')$, and for every $i \in J$ and $c \in \mathcal{C}_i$ we have

$$\forall \nu \in \mathcal{S}_{N \setminus J} : ((\nu, \nu'_{-i}) \models \text{ante}(c)) \Rightarrow (\nu'_i \models \text{con}(c)) \quad (5)$$

For $J = N$, we drop the universal quantifier and ν in (5).

Intuitively, a confirmed deal guarantees for all coalition partners that the antecedent of their commitments in \mathcal{C}' can be satisfied and that they can at the same time honour all their commitments in \mathcal{C} , regardless of what the agents outside the coalition do. This is achieved by playing a safely playable coalition strategy. The computational complexity of the associated decision problem is also situated at the second level of the polynomial hierarchy.

Proposition 9

Deciding whether there exists a confirmed deal based on a given set \mathcal{C} of commitments is Σ_2^P -complete.

It turns out that if the coalition partners have expressed consistent commitments, having a stable set is sufficient to obtain a confirmed deal.

Proposition 10

Let $\mathcal{C} = (\mathcal{C}_i)_{i \in N}$ be such that each \mathcal{C}_i is a consistent set of commitments with debtor i . If $\mathcal{C}' \subseteq \mathcal{C}$ is a stable set of commitments, then $(\text{deb}(\mathcal{C}'), \mathcal{C}')$ is a confirmed deal based on \mathcal{C} .

Moreover, a connection can be made between stable sets and possible deals.

Proposition 11

Let \mathcal{C} be a stable set of commitments with $J = \text{cred}(\mathcal{C})$, then for every playable coalition strategy $\nu \in \mathcal{S}_J$ of \mathcal{C} it holds that (J, ν) is a possible deal.

Since every safely playable strategy of a confirmed deal (J, \mathcal{C}') is in particular a playable strategy of the stable set \mathcal{C}' , we immediately get the following result.

Corollary 12

Let (J, \mathcal{C}') be a confirmed deal based on a set of commitments \mathcal{C} , then for every safely playable coalition strategy $\nu \in \mathcal{S}_J$ of (J, \mathcal{C}') it holds that (J, ν) is a possible deal.

4.2 Nash Equilibria

To illustrate how the introduced notions can be applied, we now show that agents can choose their commitments such that the union of the deals is a PNE (if one exists). The following results link the best response of a debtor to the best responses of the other agents involved in a deal.

Proposition 13

Let \mathcal{C} be a set of commitments in the BG G such that every $c \in \mathcal{C}$ respects the debtor's best responses and let (J, ν) be a possible deal based on \mathcal{C} . For every $i \in J$ and every $\nu' \in \mathcal{S}_{N \setminus J}$ it holds that $\mathcal{BR}(G, \nu_i, (\nu_{-i}, \nu'))$.

Intuitively, Proposition 13 expresses that every possible deal based on commitments which respect the debtor's best responses has the property that, regardless of what the agents who are not part of the deal decide to do, the agents who are part of the deal play a best response by playing the strategy specified in the deal.

Corollary 14

Let \mathcal{C} be a set of commitments in the BG G such that every $c \in \mathcal{C}$ respects the debtor's best responses and let (J, \mathcal{C}') be a confirmed deal based on \mathcal{C} . For every $i \in J$, every safely playable coalition strategy $\nu \in \mathcal{S}_J$ and every $\nu' \in \mathcal{S}_{N \setminus J}$ it holds that $\mathcal{BR}(G, \nu_i, (\nu_{-i}, \nu'))$.

We can now straightforwardly derive the following proposition from Proposition 4 and Corollaries 6 and 14.

Proposition 15

Suppose that agents only announce consistent sets of commitments respecting their best responses. If all agents are part of a confirmed deal, in either the original BG or one of its reductions based on previously closed deals, then the union of the safely playable coalition strategies is a PNE.

Note that, by definition, no agent has an incentive to individually deviate from an obtained deal which is a PNE. We moreover have the following result.

Proposition 16

If the BG G has a PNE, then there exists a sequence of commitments such that every agent is guaranteed to become part of a confirmed deal and the union of these deals is a PNE, without requiring prior knowledge of the other agents' goals.

Note, however, that the existence of a PNE is not required to obtain a confirmed deal. Consider for instance a 2 agent BG with $\Phi_i = \{d_i\}$, $\delta_i = \top$, $\Gamma_1 = \{d_1 \wedge d_2; \neg d_1 \wedge \neg d_2\}$ and $\Gamma_2 = \{\neg d_1 \wedge \neg d_2; d_1 \wedge d_2\}$. In other words, agent 1 prefers disarmament of both nations over arming both nations, and vice versa for agent 2. If the agents announce the commitments $c_1 = (1; 2; d_2; d_1)$ and $c_2 = (2; 1; \neg d_1; \neg d_2)$ in the first round, there are no possible deals. If they additionally announce the commitments $c'_1 = (1; 2; \neg d_2; \neg d_1)$ and $c'_2 = (2; 1; d_1; d_2)$ in the second round, 2 confirmed deals involving both agents are obtained, namely $(\{1, 2\}, \{c'_1, c_2\})$ and $(\{1, 2\}, \{c_1, c'_2\})$. The unique safely playable strategies are respectively $\{\neg d_1, \neg d_2\}$, i.e. both nations arm, and $\{d_1, d_2\}$, i.e. both nations disarm. These two outcomes are Pareto optimal, i.e. no other outcome exists such that both agents would be better off.

4.3 Commitment-based Deals in Negotiation Protocols

The formalized concepts of commitments and commitment-based deals can easily be plugged into existing negotiation protocols. This allows to select protocols that best suit the needs of the application. As an illustration, we configure two kinds of protocols.

Firstly, we explain how our notions can be used in a multilateral monotonic concession protocol [34, 18], in which agents incrementally make concessions to reach an agreement. Given our framework of BGs with prioritized goal bases, this is a very intuitive approach for the agents: if a commitment corresponding to the first priority goal of an agent does not lead to any deal, the agent can concede by considering its second priority goal as well.

Example 10

Reconsider the context of disarmament of nations and suppose a pacifistic agent 1 is in a BG with 3 agents and has the goal base $\Gamma_1 = \{d_1 \wedge d_2 \wedge d_3; d_1 \wedge (d_2 \vee d_3); d_1\}$. Its first commitment would be $c_1 = (1; \{2, 3\}; d_2 \wedge d_3; d_1)$, coinciding with its first priority goal of global disarmament. Now assume that agents 2 and 3 have communicated commitments such that not a single possible deal arose, e.g. $c_2 = (2; \{1, 3\}; \neg d_1 \vee \neg d_3; \neg d_2)$ and $c_3 = (3; 1; d_1; d_3)$. Then agent 1 can concede by communicating the commitment $c'_1 = (1; \{2, 3\}; d_2 \vee d_3; d_1)$, coinciding with its second priority goal.

Concessions can thus easily be captured by opening with commitments corresponding to the first priority goal, then conceding to the disjunction of the first and second priority goal, next to the disjunction of the first, second and third priority goal etc. Note that in Example 10, the disjunction of agent 1's first and second priority goal is equivalent with its second priority goal.

The notion of a confirmed deal can fulfill the concept of agreement used in the monotonic concession protocol [34, 18]. In Example 10, the concession of agent 1 would lead to the confirmed deal $(\{1, 3\}, \{c'_1, c_3\})$ based on $\{c'_1, c_2, c_3\}$, even without concession of agents 2 and 3. The corresponding unique safely playable coalition strategy is $\{d_1, d_3\}$.

Recall that e.g. proposals corresponding to propositional formulas would not be sufficient to obtain deals between coalitions, as they do not take the control assignment of the action variables into account. Finally note that previous work on monotonic commitment concession does not address goal-related concession [42].

As another example, consider the negotiation protocol described in [33], in which a broker agent matches proposals, and then notifies the agents — which submitted the proposals — about the possibility of agreement. It is easy to see that, by using commitments as proposals, we can straightforwardly use our definitions of a deal to capture the notion of matching proposals. In [33], the agents are supposed to negotiate about which of the possible agreements is to be chosen after they received the notification. However, the presence of a broker agent — a central entity — can also bypass the need of this extra negotiation by selecting at most one deal per agent. It could even use a social criterion, e.g. by selecting the deals which involve the largest number of agents. Whether such interventions are desirable or even justifiable strongly depends on the context of the application.

5 Related Work

To the best of our knowledge, this paper is the first to study commitments in BGs. Therefore, we structure this section by addressing the related work w.r.t. BGs and commitments separately.

5.1 Related Work w.r.t. Boolean Games

Our study on opportunities for agreement and the formation of coalitions is reminiscent of cooperative game theory [2]. The study of Boolean games from a cooperative point of view has led to a variety of concepts, such as e.g. the core, stable sets of outcomes [17], efficient coalitions, weak and strong core [8] and stable coalitions [35]. In this existing work only BGs with a single goal are considered, yet costs associated with strategies are used to obtain non-binary utilities. In this paper, however, the coalition concepts, i.e. the deals, are based on a set of commitments instead of on the BG. Nonetheless, since it is sensible that agents' commitments are related to their goals (see Section 3), it is likely that, under certain assumptions, links can be found between the different concepts. For instance, in [8] a coalition is called efficient iff it can satisfy the goal of every coalition partner, regardless of the strategies of the agents outside the coalition. It is clear that our concepts offer a way to obtain the efficient coalitions in the case of BGs with a single goal: if every agent communicates commitments coinciding with their goal, the efficient coalitions are exactly the possible deals. Analogously, negotiation strategies might be characterized such that agents obtain an agreement corresponding to alternative solution concepts, such as the weak and strong core [8], as these can be linked to the concept of efficient coalitions. Further investigation of these links, however, lies beyond the scope of this paper.

Although negotiation [29, 1] and BGs [9, 17, 14, 7, 8, 41, 5] have been widely studied, only few works have considered protocols in BGs which allow agents to actually coordinate towards agreeable outcomes. A multilateral negotiation protocol for BGs has been investigated in [17], where it is shown that, when the logical structure of the goals is restricted to positive goals (i.e. no connectives other than \wedge and \vee are used), the protocol is guaranteed to end in a Pareto optimal outcome, meaning that no agent can improve its position without another agent being worse off. BGs have also been extended to endogenous variants [39], involving a pre-play negotiation phase, in which agents can try to influence the decisions of other agents by means of side payments, i.e. transferring parts of their utility to other agents. In [19], an extension of BGs is used to model voting strategies in binary aggregation. A multilateral negotiation protocol for BGs with prioritized goals has been developed in [15]. The protocol is driven by an agreement rule which guarantees a fair and efficient outcome under complete knowledge about the other agents' goals. It is extended to BG settings with incomplete knowledge, in which case a weaker notion of fairness and efficiency is guaranteed. In this agreement protocol, the order of the agents strongly influences the outcome. In contrast, in this paper we introduce building blocks for multilateral negotiation protocols in which arbitrary propositional formulas can be used to specify goals. Moreover, the building blocks can be used in protocols in which the ordering of the agents does not influence the agreements, such as the multilateral protocol with the broker agent. Furthermore, it is easy to see that the assumption of transferable utility or knowledge about the other agents' goals is not required to obtain deals.

5.2 Related Work w.r.t. Commitments

Outside the setting of BGs, the characterization of commitments and commitment-based protocols has been extensively studied [36, 25, 40, 16, 31, 32, 37, 12, 28, 22, 38]. Commitments, which can be considered a form of pre-play moves [25], are studied from a game-theoretical perspective in Schelling's seminal work [36]. Schelling

explains how commitments can be threats, when one commits to deviate from its own best response to damage the opponent, or promises, when one commits to deviate from its own best response to cooperate with the opponent [36]. However, since we assume no knowledge about the other agents' goals in this paper, the agents do not have the required information to determine whether their commitments are either one of these. Instead, commitments here should be understood primarily as a way to communicate its own incentives, a point of view which is also noted by Schelling [36].

In this paper, we have introduced the concept of a consistent set of commitments. A series of postulates for commitments has been described in [37], in which the debtor and creditor are fixed. Using the *monotonicity* postulate, it is clear that an inconsistent set of commitments with a fixed debtor and creditor would violate the *strong consistency* postulate. To obtain the notion of consistency as defined in our paper though, a meaningful generalization of the postulates to variable creditors is required. In [22], conflicts between commitments are discussed, where it is assumed that the domain dependent conflict knowledge is already present. For instance, the same car cannot be rented simultaneously by two different individuals. The framework in [22] is more involved than ours, as it relies on event calculus to formalize conflicts between commitments. The debtor and creditor are irrelevant in the detection of conflicts [22]. Three different notions of conflict are defined, but even if we fix the debtor, none these notions coincide with our concept of consistency, as none of them seem to take into account whether the antecedents can simultaneously be satisfied or not. In [23], the notion of feasibility of commitments is introduced. The intuitive idea behind this concept is the same as ours w.r.t consistency, i.e. checking whether it is possible for an agent to fulfill a set of commitments all together [23]. Their elaboration, however, is slightly different from ours: the feasibility of the commitments of an agent i does not only take the commitments with debtor i into account, but also those with creditor i . For instance, if you have committed to make two payments to two different agents but only have sufficient money for one of the payments, we consider this pair of commitments inconsistent. In [23], this set might still be considered feasible in case there is e.g. a commitment of a reliable agent to make a payment to you. Moreover, for a set of commitments to be feasible there should exist an execution that discharges them all, i.e. that brings about all the consequences [23]. Translated to our framework, that would imply that a pair of commitments of the form $(1; 2; d_2; d_1)$ and $(1; 2; \neg d_2; \neg d_1)$ is not feasible. In this paper however, this pair is consistent, due to the inconsistency of the antecedents.

Several studies on the relationship between goals and commitments can be found in the multi-agent literature [11, 38, 20]. The variety among this work can be partially explained by the usage of different representations for the goals. The goal models in [11] are specified in Tropos, an agent-oriented software engineering framework [10]. Goals can either be decomposed as conjunctions or as disjunctions of subgoals. Moreover, the link between the possible achievement of pairs of goals is formalized. In this framework, goals are not prioritized. The authors exploit commitments for goal support and vice versa and provide elaborated semantics. In [38], goals consists of a precondition, an in-condition, a post-condition, a success condition and a failure condition. An agent can have multiple goals, but it is assumed that they are mutually consistent. No priority between goals is used. A formalization from goal to commitment and vice versa is provided, based on guarded rules. These rules operate on the goals (e.g. consider, activate, suspend, reactivate, drop) and commitments (e.g. release, cancel). In [20], goals are represented in

a similar way as in [38], but the in-condition and post-condition are dropped. The framework also involves beliefs about the other agents' goals and capabilities. It is likely that BGs could be embedded into this framework in case it were to be generalized to allow priorities in the set of goals. As in [11], a notion of goal support is introduced and the authors provide algorithms to generate commitments to support their goals [20, 27, 21]. To this end, agents also use their beliefs about the goals of other agents to formulate commitments which are more likely to be accepted [20]. The notion of goal support takes all commitments into account and considers a goal to be supported if there is a chain of commitments leading to the satisfaction of the goal. For example, if agent 1 and 2 have communicated the commitments $(1; 2; d_2; d_1)$ and $(2; 1; d_1; d_2)$, then none of these agents is considered to support the goal $\gamma = d_1 \wedge d_2$. In contrast, we consider these as commitments coinciding with γ , which moreover form a stable set to bring about γ . Investigating the possible formal links between the aforementioned work and ours is an interesting topic for further research.

A large amount of the prior work on commitment-based protocols mainly focuses on practical aspects, e.g. investigating the life-cycle of commitments [40], solving misalignment [12], detecting exceptions in commitments [28], applications in a business context [16], etc., and pays less attention to the underlying multi-agent system. In contrast, we analyze commitment-based deals specifically in the context of BGs, allowing us to exploit game-theoretical concepts such as utility and best response to define sensible commitments and to introduce suitable notions of deals between coalitions of agents.

6 Conclusion

The aim of this paper was to study, at a general level, the notion of commitments and commitment-based deals in Boolean games. First we have formalized the notion of commitments and explained how goals can be related to sets of commitments. Then, we investigated commitment-based deals, which rely on the idea of identifying stable sets among the commitments made by a group of agents. Finally, we have illustrated how the notions of commitments and deals can be used to guarantee e.g. the Nash property in the obtained deals, or to configure existing negotiation protocols. The latter allows a flexible use of the introduced concepts: depending on the application context one can, for instance, either plug our building blocks into a distributed or a centralized protocol.

Note that in a practical implementation, additional aspects might need to be addressed. However, to a large extent we can rely on the available results w.r.t. (algebraic) formalization of commitments [32, 37, 38] and commitment-based protocols [40, 31, 12]. For instance, to escape deadlock agreements (characterized by cyclic dependencies) or avoid the possibility of cheating caused by imperfect synchrony, one could respectively rely on the 2PC protocol [40] and the lockstep-protocol [4]. Deciding which confirmed deal is closed when more than one arises should also be addressed (e.g. choosing the largest coalition).

Finally, an important issue is how agents can act strategically in how they formulate commitments. This should be based on their beliefs about the other agents' goals, their risk aversion and/or their readiness to concede, and could be implemented using techniques such as Monte Carlo tree search. Alternatively, an approach similar to the one in [20] can be investigated, where the beliefs about the agents' goals are used to generate commitments that are more likely to be accepted by other agents.

REFERENCES

- [1] L. Amgoud and H. Prade, 'A possibilistic logic modeling of autonomous agents negotiation', in *Progress in Artificial Intelligence*, 360–365, Springer, (2003).
- [2] R.J. Aumann, 'Game theory', in *The New Palgrave*, eds., J. Eatwell, M. Milgate, and P. Newman, 1–53, Macmillan, London and Basingstoke, (1989).
- [3] C. Bartolini, C. Preist, and N.R. Jennings, 'A software framework for automated negotiation', in *Software Engineering for Multi-Agent Systems III*, 213–235, Springer, (2004).
- [4] N.E. Baughman and B.N. Levine, 'Cheat-proof payout for centralized and distributed online games', in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pp. 104–113, (2001).
- [5] J. Ben-Naim and E. Lorini, 'Evaluating power of agents from dependence relations in Boolean games', in *Proc. AAMAS*, pp. 853–860, (2014).
- [6] E. Bonzon, M.-C. Lagasquie-Schiex, and J. Lang, 'Compact preference representation for Boolean games', in *Proc. PRICAI*, pp. 41–50, Springer-Verlag, (2006).
- [7] E. Bonzon, M.-C. Lagasquie-Schiex, and J. Lang, 'Dependencies between players in Boolean games', in *Proc. ECSQARU*, volume 4724 of *LNCIS*, pp. 743–754, Springer, (2007).
- [8] E. Bonzon, M.-C. Lagasquie-Schiex, and J. Lang, 'Effectivity functions and efficient coalitions in Boolean games', *Synthese*, **187**, 73–103, (2012).
- [9] E. Bonzon, M.-C. Lagasquie-Schiex, J. Lang, and B. Zanuttini, 'Boolean games revisited', in *Proc. ECAI*, pp. 265–269, ACM, (2006).
- [10] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, and J. Mylopoulos, 'Tropos: An agent-oriented software development methodology', *Autonomous Agents and Multi-Agent Systems*, **8**(3), 203–236, (2004).
- [11] A.K. Chopra, F. Dalpiaz, P. Giorgini, and J. Mylopoulos, 'Reasoning about agents and protocols via goals and commitments', in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, volume 1, pp. 457–464, International Foundation for Autonomous Agents and Multiagent Systems, (2010).
- [12] A.K. Chopra and M.P. Singh, 'Multiagent commitment alignment', in *Proceedings of AAMAS*, pp. 937–944, International Foundation for Autonomous Agents and Multiagent Systems, (2009).
- [13] A. Darwiche, 'Decomposable negation normal form', *Journal of the ACM (JACM)*, **48**(4), 608–647, (2001).
- [14] S. De Clercq, S. Schockaert, M. De Cock, and A. Nowé, 'Possibilistic Boolean games: Strategic reasoning under incomplete information', in *Proc. JELIA*, pp. 196–209, (2014).
- [15] S. De Clercq, S. Schockaert, A. Nowé, and M. De Cock, 'Multilateral negotiation in Boolean games with incomplete information using generalized possibilistic logic', in *Proceedings of IJCAI*, pp. 2890–2896, (2015).
- [16] N. Desai, A.U. Mallya, A.K. Chopra, and M.P. Singh, 'Interaction protocols as design abstractions for business processes', *Software Engineering, IEEE Transactions on*, **31**(12), 1015–1027, (2005).
- [17] P.E. Dunne, W. van der Hoek, S. Kraus, and M. Wooldridge, 'Cooperative Boolean games', in *Proc. AAMAS*, volume 2, pp. 1015–1022, IFAAMAS, (2008).
- [18] U. Endriss, 'Monotonic concession protocols for multilateral negotiation', in *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pp. 392–399, ACM, (2006).
- [19] U. Grandi, D. Grossi, and P. Turrini, 'Equilibrium refinement through negotiation in binary voting', in *Proceedings of IJCAI*, pp. 540–546, (2015).
- [20] A. Günay, M. Winikoff, and P. Yolum, 'Commitment protocol generation', in *Declarative agent languages and technologies X*, 136–152, Springer, (2012).
- [21] A. Günay, M. Winikoff, and P. Yolum, 'Dynamically generated commitment protocols in open systems', *Autonomous Agents and Multi-Agent Systems*, **29**(2), 192–229, (2015).
- [22] A. Günay and P. Yolum, 'Detecting conflicts in commitments', in *Declarative Agent Languages and Technologies IX*, 51–66, Springer, (2011).
- [23] A. Günay and P. Yolum, 'Constraint satisfaction as a tool for modeling and checking feasibility of multiagent commitments', *Applied intelligence*, **39**(3), 489–509, (2013).
- [24] P. Harrenstein, W. van der Hoek, J.-J. Meyer, and C. Witteveen, 'Boolean games', in *Proc. TARK*, pp. 287–298, MKP Inc., (2001).
- [25] J. Hirshleifer, 'Game-theoretic interpretations of commitment', *Evolution and the capacity for commitment*, 77–93, (2001).
- [26] N.R. Jennings, P. Faratin, A.R. Lomuscio, S. Parsons, M.J. Wooldridge, and c. Sierra, 'Automated negotiation: prospects, methods and challenges', *Group Decision and Negotiation*, **10**(2), 199–215, (2001).
- [27] O. Kafali, A. Günay, and P. Yolum, 'Gosu: Computing goal support with commitments in multiagent systems', in *Proceedings of 21st European Conference on Artificial Intelligence*, pp. 477–482, (2014).
- [28] Ö. Kafali and P. Yolum, 'Detecting exceptions in commitment protocols: Discovering hidden states', in *Languages, Methodologies, and Development Tools for Multi-Agent Systems*, 112–127, Springer, (2009).
- [29] S. Kraus, 'Negotiation and cooperation in multi-agent environments', *Artificial intelligence*, **94**(1), 79–97, (1997).
- [30] J. Lang, P. Liberatore, and P. Marquis, 'Propositional independence-formula-variable independence and forgetting', *Journal of Artificial Intelligence Research*, 391–443, (2003).
- [31] A.U. Mallya and M.P. Singh, 'Introducing preferences into commitment protocols', in *Agent Communication II*, 136–149, Springer, (2006).
- [32] A.U. Mallya and M.P. Singh, 'An algebra for commitment protocols', *Autonomous Agents and Multi-Agent Systems*, **14**(2), 143–163, (2007).
- [33] Piotr Palka, 'Multilateral negotiations in distributed, multi-agent environment', in *Computational Collective Intelligence. Technologies and Applications*, 80–89, Springer, (2011).
- [34] Jeffrey S Rosenschein and Gilad Zlotkin, *Rules of encounter: designing conventions for automated negotiation among computers*, MIT press, 1994.
- [35] L. Sauro and S. Villata, 'Dependency in cooperative Boolean games', *Journal of Logic and Computation*, **23**(2), 425–444, (2013).
- [36] T.C. Schelling, *The strategy of conflict*, Harvard university press, 1980.
- [37] M.P. Singh, 'Semantical considerations on dialectical and practical commitments', in *AAAI*, volume 8, pp. 176–181, (2008).
- [38] P.R. Telang, M.P. Singh, and N. Yorke-Smith, 'Relating goal and commitment semantics', in *Programming Multi-Agent Systems*, 22–37, Springer, (2012).
- [39] P. Turrini, 'Endogenous games with goals: side-payments among goal-directed agents', *Autonomous Agents and Multi-Agent Systems*, 1–28, (2015).
- [40] F. Wan and M.P. Singh, 'Formalizing and achieving multiparty agreements via commitments', in *Proceedings of AAMAS*, pp. 770–777, ACM, (2005).
- [41] M. Wooldridge, U. Endriss, S. Kraus, and J. Lang, 'Incentive engineering for Boolean games', *Artif. Intell.*, **195**, 418–439, (2013).
- [42] P. Yolum and M.P. Singh, 'Enacting protocols by commitment concession', in *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, pp. 116–123, ACM, (2007).

A Joint Model for Sentiment-Aware Topic Detection on Social Media

Kang Xu and Guilin Qi and Junheng Huang and Tianxing Wu¹

Abstract. Joint sentiment/topic models are widely applied in detecting sentiment-aware topics on the lengthy review data and they are achieved with Latent Dirichlet Allocation (LDA) based model. Nowadays plenty of user-generated posts, e.g., tweets and E-commerce short reviews, are published on the social media and the posts imply the public's sentiments (i.e., positive and negative) towards various topics. However, the existing sentiment/topic models are not applicable to detect sentiment-aware topics on the posts, i.e., short texts, because applying the models to the short texts directly will suffer from the context sparsity problem. In this paper, we propose a Time-User Sentiment/Topic Latent Dirichlet Allocation (TUS-LDA) which aggregates posts in the same timeslice or user as a pseudo-document to alleviate the context sparsity problem. Moreover, we design approaches for parameter inference and incorporating prior knowledge into TUS-LDA. Experiments on the Sentiment140 and tweets of electronic products from Twitter7 show that TUS-LDA outperforms previous models in the tasks of sentiment classification and sentiment-aware topic extraction. Finally, we visualize the sentiment-aware topics discovered by TUS-LDA.

1 Introduction

With the rapid growth of Web 2.0, a mass of user-generated posts, e.g., tweets and E-commerce short reviews, which capture people's interests, thoughts, sentiments and actions. The posts have been accumulating on the social media with each passing day. Sentiment analysis attempts to find user preference, likes and dislikes from the posts on social media, such as reviews, blogs and microblogs [21] and topic modeling attempts to discover the topics or aspects from from reviews, blogs and microblogs etc [3]. Topic modeling and sentiment analysis on the posts are two significant tasks which can benefit many people. For example, we can discover a topic about "Apple Inc." and the overall sentiment of the topic. The sentiment of the topic about "Apple Inc." is implicitly associated with the stock trading of "Apple Inc.", because negative sentiments towards the company on social media can fall sales and financial gains but positive sentiments can improve sales [2]. Topic modeling [1] focuses on extracting word-level or document-level topics, while sentiment analysis [23] is to analyze the sentiments of words or documents.

Topic modeling and sentiment analysis on the social media are complementary where sentiments on the social media often change over different topics and topics on the social media are always related to public sentiments. So jointly modeling topics and sentiments on the social media is a feasible and significative task and it can reflect people's sentiment on different topics. However, unlike the nor-

mal documents (e.g., news and long reviews), the short and informal characteristic of the posts, e.g., tweets and short reviews, on the social media makes the tasks of topic modeling and sentiment analysis more challenging.

By jointly modeling topics and sentiments on social media, we want to obtain sentiment-aware topics from the posts, e.g., a topic about "Apple Inc." ('ipad', 'iphone', 'itouch', 'imac', 'beautiful' and 'popular') with the overall sentiment polarity "positive". Topic models, e.g., LDA [1] and pLSA [10], originally focus on mining topics from texts, but the models can also be extended to extract an extra aspect of texts, i.e., sentiment. Conventional sentiment-aware topic models, like Joint Sentiment/Topic Model (JST) [15] and Aspect/Sentiment Unification Model (ASUM) [11], are utilized for uncovering the hidden topics and sentiments from text corpus where each document is a mixture of sentiment/topics and each sentiment/topic is a mixture of words. Thereinto, each sentiment label in the models is viewed as a special kind of topic where topics are unknown and data-driven but sentiments are known and specified. However, for the short and informal characteristic of the posts, applying the models to the short posts on the social media directly always suffers from the context sparsity problem. So the models fail to recognize the accurate sentiments and senses of words in the posts.

One simple and effective way to alleviate the sparsity problem is to aggregate short posts into lengthy pseudo-documents [5, 31]. Here we assume that the posts on the social media are a mixture of two kinds of topics: temporal topics which are related to current events (e.g., tweets about a topic "Announcement of iphone SE" in Fig 1(a) which are produced in a timeslice) and stable topics which are related to personal interests (e.g., tweets about a topic "Apple products" in Fig 1(b) which are produced by a user). Thereinto, temporal topics are sensitive to time. If posts belong to temporal topics, we aggregate the posts in the same timeslice as a single document. We assume each timeslice is a mixture of sentiment-aware topics, i.e., each sentiment in the timeslice corresponds to several topics. Similar to temporal topics, stable topics are related to specific users and each user is a mixture of sentiment-aware topics. If a post belongs to a temporal topic, the post is assigned to a sentiment-aware topic in its publishing timeslice; otherwise, it is assigned to a sentiment-aware topic in its publishing user.

Moreover, based on the analysis of the characteristics of topics and sentiments, we exploit the important observation of topics: A single post always talks about a single topic [31]. Although a post usually talks about a single topic, a post may talk about multiple aspects of the topic with different sentiment polarities [12, 18].

For example, while the following short review of cannon camera from Amazon.com expresses the overall sentiment polarity of *Camera*, which corresponds to the part in italics, as positive, it addi-

¹ Southeast University, Nanjing, China
Email: {kxu, gqi, jhhuang, wutianxing}@seu.edu.cn



Figure 1. (a) A temporal topic (b) A stable topic

tionally expresses a negative opinions towards the camera's lenses which corresponds to the part in bold.

Camera is great, but lenses are **crap** and cheap and don't work on auto focus. Buy body and lenses separately.

For a tweet, it can express a positive, a negative or neutral sentiment, and it can also express both positive and negative sentiments[24].

So, for sentiment polarities, we exploit the observation that words in a single post may correspond to multiple sentiment polarities [12, 18]. A post can talk about the same topic with different sentiments. For better modeling topics and sentiments respectively, we follow the assumption that words in the same post should belong to the same topic, but they can have different sentiments.

Moreover, we add a sentiment label for each post. The sentiment label represents the overall sentiment polarities of the post and is determined by the sentiment polarities of words in the post. If words of a post express both positive and negative sentiments, the overall sentiment polarities of the post should be judged as the stronger one [24]. The sentiment label is utilized to model the association between sentiments and topics.

In this paper, we propose a novel Time-User Sentiment/Topic Latent Dirichlet Allocation (TUS-LDA) to mine sentiment-aware topics from the user-generated posts on social media.

There exist four main contributions of TUS-LDA:

1) TUS-LDA aggregates posts in the same timeslice or user as a single document to alleviate the context sparsity problem.

2) We design different ways to model topics and sentiments based on the characteristics of topics and sentiments. Thereinto, the sentiments of a post and the words in the post are all drawn from document-level sentiment distribution. Within the chosen sentiment of the post, the topic of the post is drawn from a user-level or timeslice-level sentiment/topic distribution.

3) We design approaches of parameter inference and incorporating prior sentiment knowledge for TUS-LDA.

4) We implement experiments on two datasets to evaluate the effectiveness of sentiment classification and topic extraction in TUS-LDA and visualize sentiment-aware topics discovered by TUS-LDA.

The rest of the paper is organized as follows: in Section 2, we introduce the related work about topic models on short texts and joint sentiment/topic models; in Section 3, we give the definitions of the basic terminologies we will use in our paper; in Section 4 we present our proposed model Time-User Sentiment/Topic Latent Dirichlet Allocation (TUS-LDA); Experimental settings and results are shown in Section 5. Finally, in Section 6, we conclude this paper and lists the future work.

2 Related Work

2.1 Topic Models on Short Texts

LDA [1] and PLSA [10] originally focus on mining topics from lengthy documents. Recently topic modeling in the posts on social media is popular, however, it also suffers from the context sparsity problem of the posts. To overcome the sparsity problem of posts on the social media, there exist some work of aggregating posts into pseudo-documents. In [31], Twitter-LDA aggregated posts published by a user into one lengthy pseudo-document and made words in the same post belong to the same topic. In [5], posts in TimeUserLDA were aggregated by timeslices or users for finding bursty topics where posts belong to two kinds of topics: personal topics and temporal topics. Similar to TimeUserLDA, posts in TUK-TTM [29] were also aggregated by timeslices or users and TUK-TTM was utilized for time-aware personalized hashtag recommendation. Although these models can alleviate the problem of the context sparsity of posts on social media, they did not model an extra aspect of posts, i.e., sentiment.

2.2 Joint Sentiment/Topic Models

Recently, some topic models have been extended to model topics and sentiments jointly. The first work of topic and sentiment modeling is Topic-Sentiment Mixture model TSM [19]. In TSM, a sentiment is a special kind of topic and each word is generated from either a sentiment or a topic. The relation between sentiments and topics cannot be mined by TSM. At the same time, TSM is based on PLSA and suffers from the problems of inferencing on new documents and overfitting the data. To overcome these shortcomings, **Joint Sentiment-Topic model (JST)** [15] which is a two-level sentiment-topic model based on Latent Dirichlet Allocation (LDA) was proposed. In JST, sentiment labels are associated with documents, under which topics are associated with sentiment labels and words are associated with both sentiment labels and topics. Reverse-JST (RJST) [16] is a variant of JST where the position of sentiment and topic layer is swapped. In JST, topics were generated conditioned on a sentiment polarity, while in RJST sentiments were generated conditioned on a topic. **Aspect/Sentiment Unification Model (ASUM)** [11] is similar to JST. In ASUM, words in the same sentence belong to the same sentiment and topic. Sentiment Topic Model with Decomposed Prior (STDP) [32] is another variant of JST. STDP first determined whether the word is used as a sentiment word or ordinary topic words and then chose the accurate sentiments for sentiment words. Time-aware

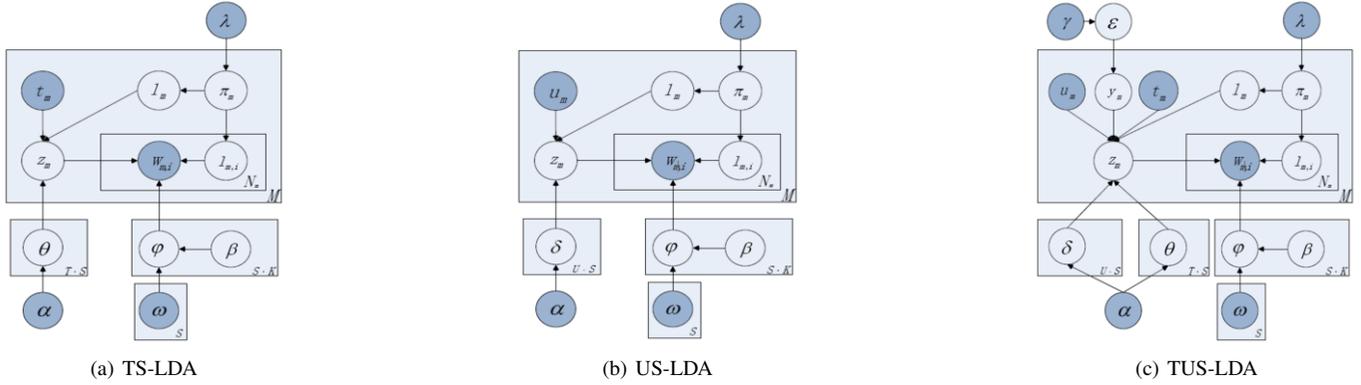


Figure 2. The graphical representation of the proposed model (TS-LDA (a), US-LDA (b), TUS-LDA (c)). Shaded circles are observations or constants. Unshaded ones are hidden variables.

Topic-Sentiment Model (TTS) [4] extracted the hidden topics from texts and modeled the association between topics and sentiments and tracked the strength of topic-sentiment association over time. In TTS, time is viewed as a special word to bias the topic-sentiment distributions. But in our model, we use time to aggregate short texts and generate pseudo documents for modeling topics and sentiments. JST, RJST, ASUM, STDP and TTS are designed for normal texts where each piece of text has rich context to infer topics and sentiments, but our work models posts (i.e., short and informal texts) on social media and all of these models lose efficacy in the short and informal texts. MaxEnt-LDA [30] jointly discovers both aspects and aspect-specific opinion words by integrating a supervised maximum entropy algorithm to separate opinion words from objective ones. However, it does not further discover aspect-aware sentiment polarities of opinion words, which are very useful for sentiment analysis.

In our model, we focus on short and informal texts on social media. There exists some work about LDA-based sentiment analysis on social media. Twitter Opinion Topic Model (TOTM) [14] aggregated or summarized opinions of a product from tweets, which can discover target specific opinion words and improve opinion prediction. Topic Sentiment Latent Dirichlet Allocation (TSLDA) [22] utilized sentiments on social media for predicting stock price movement. TSLDA distinguished topic words and opinion words where topic words were drawn from the topic-word distribution and opinion words were drawn from the sentiment-topic-word distribution. Although these two work focuses on posts on social media, they do not consider and solve the context sparsity problem of posts.

3 Problem Definition

In this section, we define the basic terminologies we will use in this paper.

- **Post:** A post contains a sequence of words which express the opinions and thoughts of people towards different things (e.g., a tweet or a review).
- **User:** Each user-generated post has a user identification that specifies who publishes the post.
- **Timeslice:** Each user-generated post has a timeslice that specifies when the user publishes the post, in this paper, the length of timeslice is a day.
- **Topic:** A topic is a discrete piece of content that is about a specific subject, has an identifiable purpose (e.g., an event, a current hot problem and a product). Here, a topic is represented as a list of words.

- **Aspect:** An aspect refers to a distinct ratable facet of an entity. For a product, an aspect is an attribute or a component of the product that has been commented on in a review, e.g., “screen” for a digital camera. For an event or other kinds of topics, an aspect can be participants of the topic [25], e.g., “Obama” in the event of “Obama’s visit to cuba”.
- **Sentiment:** Sentiment is a label which refers to the polarity in which a concept or opinion is interpreted [17], i.e., “positive” and “negative”. For example, “positive” is a sentiment for the post “Tom was glad to visit his friends.”
- **Sentiment-aware topic:** A sentiment-aware topic is a topic labeled with a sentiment polarity. For example, the overall sentiment of the topic “Obama’s visit to cuba” is positive, so the topic “Obama’s visit to cuba” is a positive topic.

4 The Proposed Models

In this section, we firstly introduce the notation and formally formulate our problem. Then, we describe the method utilized for learning parameters. Finally, we present the method of incorporating prior knowledge into our model.

4.1 The Generation Process

It is assumed that there exists a stream of M posts, denoted as d_1, d_2, \dots, d_M . Each post d_m is generated by a user u_m within a timeslice t_m and the post d_m contains a bag of words, denoted as $\{w_{m,1}, w_{m,2}, \dots, w_{m,N_m}\}$.

In LDA, a document is viewed as a multinomial distribution over topics and a topic is a multinomial distribution over words. In JST, each document is associated with the sentiment/topic distribution, i.e., each sentiment in the document has a topic distribution; the document also has a sentiment distribution for document-level sentiment-classification and a sentiment/topic is a multinomial distribution over words. LDA and JST only work well for lengthy documents, because the lengthy document have rich contexts. Based on the analysis of posts on the social media, words in the same post tend to be about a single topic [31]. However, the sentiment polarities of words in the same post can be different [12]. At the same time, to model the association between sentiments and topics, we also add a sentiment label for each post which is determined by the overall sentiment of all the words in the post.

On social media, a part of posts talks about stable topics which are related to users’ personal interests with certain sentiments, so we

introduce a global sentiment/topic distribution δ for each user to capture personal long-term topical interests and sentiment preferences. Another part of posts is about temporal topics which are related to current events with the corresponding sentiments, so we add a time-dependent sentiment/topic distribution θ for each timeslice to capture temporal topics and the sentiments towards the topics.

Here, we construct the generative process of all the posts in the stream. When a user u_m publishes a post d_m within a timeslice t_m , the user first utilizes the variable y_m , which is drawn from the global user-timeslice switch distribution ε , to decide whether the post talks about a stable topic or a temporal topic. Then the user chooses a sentiment label l_m for the post from the document-sentiment π_m . If the user chooses a stable topic u_m and a sentiment label l_m , the user then selects a topic z_m from δ_{u_m, l_m} ; otherwise, the user selects a topic z_m from θ_{t_m, l_m} . For each word $w_{m,i}$ in the post d_m , the user first chooses a sentiment label $l_{m,i}$; with the chosen topic z_m and sentiment label $l_{m,i}$, the word is drawn from the sentiment-topic word distribution $\varphi_{l_{m,i}, z_m}$.

The notations in this paper are summarized in Table 1. Fig 2(c) shows the graphical representation of the generation process. Formally, the generative story for each post is as follows:

1. Draw $\varepsilon \sim \text{Beta}(\gamma)$
2. For each timeslice $t = 1, \dots, T$
 - i. For each sentiment label $s = 0, 1, 2$
 - a. Draw $\theta_{t,s} \sim \text{Dir}(\alpha)$
3. For each user $u = 1, \dots, U$
 - i. For each sentiment label $s = 0, 1, 2$
 - a. Draw $\delta_{u,s} \sim \text{Dir}(\alpha)$
4. For each sentiment label $s = 0, 1, 2$
 - i. For each topic $k = 1, \dots, K$
 - a. Draw $\varphi_{s,k} \sim \text{Dir}(\beta)$
5. For each post $d_m, m = 1, \dots, M$
 - i. Draw $\pi_m \sim \text{Dir}(\lambda)$
 - ii. Draw $l_m \sim \text{Multi}(\pi_m)$
 - iii. Draw $y_m \sim \text{Bernoulli}(\varepsilon)$
 - iv. if $y_m=0$, Draw $z_m \sim \text{Multi}(\theta_{u_m, l_m})$ or if $y_m=1$, Draw $z_m \sim \text{Multi}(\delta_{u_m, l_m})$
 - v. For each word $w_i = 1, \dots, N_m$
 - a. Draw $l_{m,i} \sim \text{Multi}(\pi_m)$
 - b. Draw $w_{m,i} \sim \text{Multi}(\varphi_{z_m, l_{m,i}})$

There are two degenerate variations of our model which are shown in the experiments. The first one is depicted in Fig 2(a), which considers the temporal topic-sentiment distribution. The second one is depicted in Fig 2(b), which only considers the stable topic-sentiment distribution. We refer to our complete model as TUS-LDA, the model in Fig 2(a) as TS-LDA and the model in Fig 2(b) as US-LDA.

4.2 Parameters Inference

Like LDA, exact inference is intractable in our models. Hence approximate estimation approaches, such as Gibbs Sampling [9], are utilized to solve the problem. Gibbs Sampling, a special case of Markov Chain Monte Carlo (MCMC) [6], is a relatively simple algorithm of approximate inference for our models. Due to space limitation, only the final formulas are given here.

Table 1. Notation used in the TUS-LDA model

Symbol	Description
M, K	number of documents, topics
V, U, T	number of vocabulary, users, timeslices
Z, W, Y	all the topics, words, user-timeslice switches
T, U	all the timeslices and users
L, L	all the sentiments of posts and words
N_m	number of word tokens in post d_m
u_m, t_m	user, timeslice, user-timeslice switch
y_m, l_m	and sentiment of post d_m
$l_{m,i}$	sentiment of word $w_{m,i}$
ε	beta distribution of stable topics and temporal topics
π_m	document-sentiment distribution, $\Omega = \{\pi_m\}_{m=1}^M$
$\theta_{t,s}$	timeslice-sentiment topic distribution, $\Theta = \{\theta_{t,s}\}_{t=1, s=1}^{T \times S}$
$\delta_{u,s}$	user-sentiment topic distribution, $\Phi = \{\delta_{u,s}\}_{u=1, s=1}^{U \times S}$
$\varphi_{s,k}$	sentiment-topic word distribution, $\Psi = \{\varphi_{s,k}\}_{s=1, k=1}^{S \times K}$
α	hyperparameters of $\theta_{t,s}$ and $\delta_{u,s}$
β, λ	hyperparameters of $\varphi_{s,k}, \pi_m$
γ	hyperparameters of ε
ω_s	prior knowledge of $\varphi_{s,k}$

4.2.1 Joint Distribution

The joint probability of words, users, timeslices, timeslices-user switches, topics and sentiments can be factored in Eq 1, where $\varepsilon, \pi, \varphi, \delta$ and θ are integrated and \vec{n}_m counts the number of three sentiment labels of a post and the words in the post (All the notations are illustrated in Table 1.).

$$\begin{aligned}
& P_{TUS-LDA}(\mathbf{Z}, \mathbf{W}, \mathbf{T}, \mathbf{U}, \mathbf{Y}, \mathbf{L}, \bar{\mathbf{L}} | \alpha, \gamma, \lambda, \beta, \omega) = \\
& P(\mathbf{Y} | \gamma) P(\mathbf{L} | \lambda) P(\mathbf{Z} | \mathbf{Y}, \mathbf{L}, \alpha) P(\bar{\mathbf{L}} | \lambda) P(\mathbf{W} | \mathbf{Z}, \bar{\mathbf{L}}, \beta, \omega) = \\
& \frac{\Delta(\vec{n}_y + \vec{\gamma})}{\Delta(\vec{\gamma})} \times \prod_{m=1}^M \frac{\Delta(\vec{n}_m + \vec{\lambda})}{\Delta(\vec{\lambda})} \times \prod_{u=1}^U \prod_{s=1}^S \frac{\Delta(\vec{n}_{u,s} + \vec{\alpha})}{\Delta(\vec{\alpha})} \\
& \times \prod_{t=1}^T \prod_{s=1}^S \frac{\Delta(\vec{n}_{t,s} + \vec{\alpha})}{\Delta(\vec{\alpha})} \times \prod_{s=1}^S \prod_{k=1}^K \frac{\Delta(\vec{n}_{s,k} + \vec{\beta})}{\Delta(\vec{\beta})}; \\
& \Delta = \frac{\prod_{k=1}^{dim \vec{x}} \Gamma(x_k)}{\Gamma(\prod_{k=1}^{dim \vec{x}} x_k)}, \vec{n}_y = \{n_y^0, n_y^1\}, \vec{n}_m = \{n_m^{pos}, n_m^{neg}\} \\
& \vec{n}_{u,s} = \{n_{u,s}^k\}_{k=1}^K, \vec{n}_{t,s} = \{n_{t,s}^k\}_{k=1}^K, \vec{n}_{s,k} = \{n_{s,k}^v\}_{v=1}^V
\end{aligned} \tag{1}$$

4.2.2 Posterior Distribution

Posterior distribution is estimated as follows: for the i -th post, the user u_i and timeslice t_i are known. y_i, z_i and l_i can be jointly sampled given all other variables. Here, we use \mathbf{y} to denote all the hidden variables y and \mathbf{y}_{-i} to denote all the other y except y_i . All the hyperparameters are omitted.

$$\begin{aligned}
& P(y_i = 0, z_i = k, l_i = s | \mathbf{y}_{-i}, \mathbf{z}_{-i}, \mathbf{l}_{-i}, \bar{\mathbf{l}}, \mathbf{w}) \propto \frac{\gamma_0 + n_{y,-i}^0}{\sum_{p=1}^2 \gamma_p + n_{y,-i}^p} \\
& \times \frac{\lambda_s + n_{m,-i}^s}{\sum_{s'=1}^S \lambda_{s'} + n_{m,-i}^{s'}} \times \frac{\alpha_k + n_{u,s,-i}^k}{\sum_{k=1}^K \alpha_{k'} + n_{u,s,-i}^{k'}} \\
& \times \frac{\prod_{v=1}^V \prod_{n_v=0}^{N(v)-1} (\beta_{s,k} + n_{s,k,-i}^v + n_v)}{\prod_{n=0}^{N-1} (\sum_{v=1}^V (\beta_{s,k} + n_{s,k,-i}^v) + n)}
\end{aligned} \tag{2}$$

If $y_i=0$, the i -th post talks about a stable topic, the sampling formula is shown in Eq 2; otherwise, the i -th post talks about a temporal topic, the sampling formula is shown in Eq 3.

$$\begin{aligned}
P(y_i = 1, z_i = k, l_i = s | \mathbf{y}_{-i}, \mathbf{z}_{-i}, \mathbf{l}_{-i}, \bar{\mathbf{l}}, \mathbf{w}) &\propto \frac{\gamma_1 + n_{y,-i}^1}{\sum_{p=1}^2 \gamma_p + n_{y,-i}^p} \\
&\times \frac{\lambda_s + n_{m,-i}^s}{\sum_{s'=1}^S \lambda_{s'} + n_{m,-i}^s} \times \frac{\alpha_k + n_{t,s,-i}^k}{\sum_{k'=1}^K \alpha_{k'} + n_{t,s,-i}^{k'}} \\
&\times \frac{\prod_{v=1}^V \prod_{n_v=0}^{N^{(v)}-1} (\beta_{s,k} + n_{s,k,-i}^v + n_v)}{\prod_{n=0}^{N-1} (\sum_{v=1}^V (\beta_{s,k} + n_{s,k,-i}^v) + n)}
\end{aligned} \quad (3)$$

For the j -th word in the i -th post, the sample formula of is shown in Eq 4.

$$\begin{aligned}
P(\bar{l}_{ij} = s | \mathbf{z}, \bar{\mathbf{l}}_{-ij}, \mathbf{w}, \mathbf{y}, \mathbf{l}) &\propto \frac{\lambda_s + n_{m,-ij}^s}{\sum_{s'=1}^S (\lambda_{s'} + n_{m,-ij}^{s'})} \\
&\times \frac{\beta_{s,k}^v + n_{s,k,-ij}^v}{\sum_{v'=1}^V (\beta_{s,k}^{v'} + n_{s,k,-ij}^{v'})}
\end{aligned} \quad (4)$$

Samples obtained from MCMC are then utilized for estimating the distributions π (Eq 5), δ (Eq 6) and θ (Eq 7), ϕ (Eq 8).

$$\pi_m^s = \frac{\lambda_s + n_m^s}{\sum_{s'=1}^S (\lambda_{s'} + n_m^{s'})} \quad (5)$$

$$\delta_{u,s}^k = \frac{\alpha_k + n_{u,s}^k}{\sum_{k'=1}^K (\alpha_{k'} + n_{u,s}^{k'})} \quad (6)$$

$$\theta_{t,s}^k = \frac{\alpha_k + n_{t,s}^k}{\sum_{k'=1}^K (\alpha_{k'} + n_{t,s}^{k'})} \quad (7)$$

$$\varphi_{s,k}^v = \frac{\beta_{s,k}^v + n_{s,k}^v}{\sum_{v'=1}^V (\beta_{s,k}^{v'} + n_{s,k}^{v'})} \quad (8)$$

4.2.3 Gibbs Sampling Algorithm

A complete overview of Gibbs sampling procedure is given in Algorithm 1 (All the notations are listed in Table 1).

4.3 Incorporating Prior Knowledge

Drawing on the experience of JST and RJST [16], we also add an additional dependency link of φ on the matrix ω of size $S * V$, which is utilized for encoding word prior sentiment information into the TUS-LDA and its variants. To incorporate prior knowledge into TUS-LDA and its variants, we first set all the values of ω as 1. Then the matrix ω is updated with a sentiment lexicon which contains words with the corresponding sentiment labels, i.e., positive and negative. For each term $w \in \{1, \dots, V\}$ in the corpus, if w is found in the sentiment lexicon with the sentiment label $l \in \{1, \dots, S\}$, the element ω_{lw} is set as 1 and other elements of the word w are set as 0. The element lw is updated as follows:

$$\omega_{lw} = \begin{cases} 1 & \text{if } S(w)=l \\ 0 & \text{otherwise} \end{cases}$$

The Dirichlet prior β of the size $S * K * V$ are multiplied by the matrix ω (a transformation matrix) to capture the word prior sentiment polarity.

Algorithm 1: Inference on TUS-LDA

Input: $\alpha, \gamma, \lambda, \beta, \omega$

- 1 Initialize matrices $\Omega, \Theta, \Phi, \Psi$ and ε .
- 2 **for** iteration $c=1$ to $numIterations$ **do**
- 3 **for** post $m=1$ to M **do**
- 4 Exclude post m and update count variables.
- 5 Sample a timeslice-user switch, topic and sentiment label for post m .
- 6 **if** $y=0$ **then**
- 7 Use Eq 2
- 8 **if** $y=1$ **then**
- 9 Use Eq 3
- 10 Update count variables with new timeslice-user switch, topic and sentiment label.
- 11 **for** $n=1$ to n_m **do**
- 12 Exclude word w_n and update count variables.
- 13 Sample the sentiment label for word w_n using Eq 4.
- 14 Update count variables with new sentiment label.
- 15 Update matrices $\Omega, \Phi, \Theta, \Psi$ using Eq 5, 6, 7, 8

5 Experiment Analysis

5.1 Dataset Description and Preprocessing

For experiments, we performed sentiment-aware topic discovery and sentiment classification on tweets, which are characterized by their limited 140 characters text. We selected tweets, which are related to electronic products such as camera and mobile phones, from Tweet7² and all the queried words are listed in Table 2). These tweets contain the description and reviews of various electronic products and correspond to multiple sentiment-aware topics. Besides, each tweet contains the content, the release timeslice, the user information.

Table 2. Selected Words for Extracting Tweets Related to Electronics Products

iphone, blackberry, nokia, palmpre, sony, motorola, canon, nikon, dell, lenovo, toshiba, acer, asus, macbook, hp,alienware, camera, laptop, tablet, netbook, ipad, ipod, xbox,playstation, wii, phone, nintendo, printer, panasonic, epson,samsung, kyocera, ibm, sony, microsoft, lg, hitachi, scanner,computer, fujitsu, kodak, gameboy, sega, squareenix, android,ios, windows, operatingsystem, apple

Due to the lack of sentiment labels on the Tweet7, we utilized the Sentiment140³ [8], which contains 1.6 million tweets, for sentiment classification evaluation. Each tweet in Sentiment140 has the content, a release timeslice, a user and the overall polarity label (positive or negative). The number of positive and negative tweets are nearly identical.

We followed the preprocessing steps in BTM [28]. To improve the quality of our model, we added two extra steps: (1) Part-of-speech tagging of tweet contents using the Part-of-speech tagger⁴ specially trained on tweets [7], retaining the words tagged as nouns, verbs or adjectives; (2) Lemmatizing words tagged as noun, verb, which was used to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. After preprocessing,

² <https://snap.stanford.edu/data/twitter7.html>

³ <http://help.sentiment140.com/for-students/>

⁴ <http://www.ark.cs.cmu.edu/TweetNLP/>

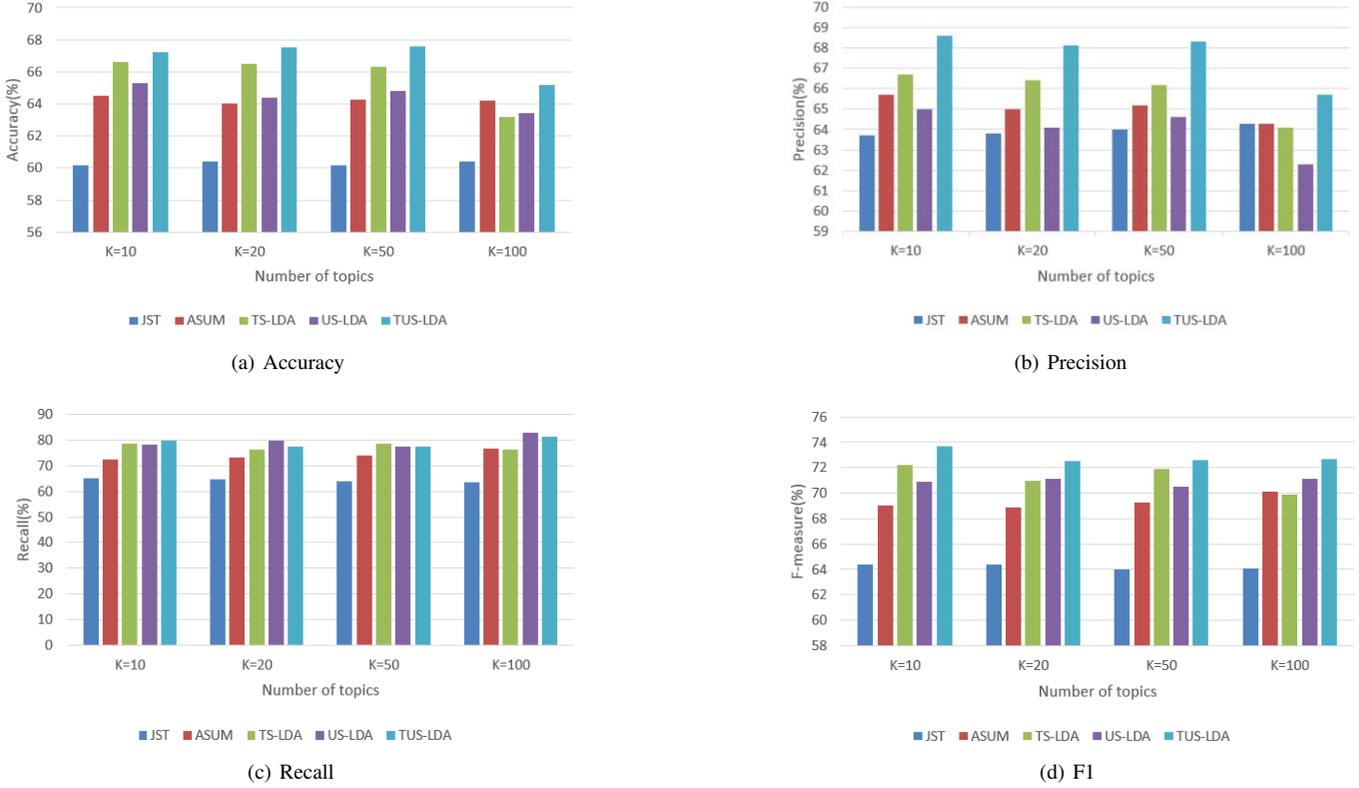


Figure 3. (a) Accuracy (b) Precision (c) Recall (d) F1 of sentiment classification

as is shown in Table 3, we left 2,766,325 valid tweets, 80,083 distinct words, 174 timeslices (days) and 572,238 users in Tweet7, and we left 258,268 valid tweets, 29,486 distinct words, 48 timeslices (days) and 21,815 users in Sentiment140.

Table 3. Corpus Statistics

	Electronic	Senti140
Number of tweets	2,766,325	258,268
Users	572,238	21,815
Timeslices	174	48

5.2 Sentiment Lexicon

In JST [15] and our models, each sentiment label is viewed as a special kind of topic that we have known in advance. To improve the accuracy of sentiment detection, we need to incorporate prior knowledge or subjectivity lexicon (i.e., words with positive or negative polarity). Here, we chose PARADIGM [26], which consists of a set of positive and negative words, e.g., happy and sad. It defines the positive and negative semantic orientation of words. Moreover, emoticons are also strong emotion indicators on social media. The entire list of emoticons is taken from Wikipedia⁵. To adjust to our scenario on social media, we just chose a subset of the emoticons in Table 4.

5.3 Parameter Settings

To optimize the number of topics K , we empirically ran the models with four values of K : 10, 20, 50 and 100 in Sentiment140 and ran

⁵ https://en.wikipedia.org/wiki/List_of_emoticons

Table 4. Emoticons

Positive	Negative
:-) :o) :) :3 :c)	>:-(>:[:-(:c
:> =] 8) : } :-D	@ >:(:(-(:'-(-
;-D :D 8-D \o/^	:'(D; (T-T) (:;)
:} (o) ()/	(;:) T.T !:!

the models with three values of K : 10, 20, 50 in Twitter7 (In Twitter7, these tweets only contain a small number of electronic product-related topics). In our model, we simply selected symmetric Dirichlet prior vectors as is empirically done in JST and ASUM. For JST and ASUM, $\alpha = \frac{50}{K}, \beta = 0.01$ and $\gamma = 0.01$. For TUS-LDA, we set $\alpha = 0.5, \gamma = 0.01, \lambda = 0.01$ and $\beta = 0.01$. These LDA-based models are not sensitive to the hyperparameters [27]. In all the methods, Gibbs sampling was run for 1,000 iterations with 200 burn-in periods.

5.4 Quantitative Evaluations

5.4.1 Sentiment Classification

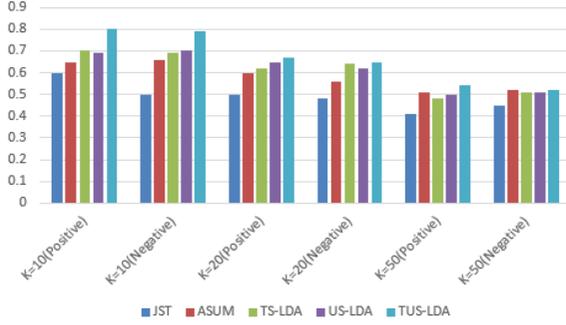
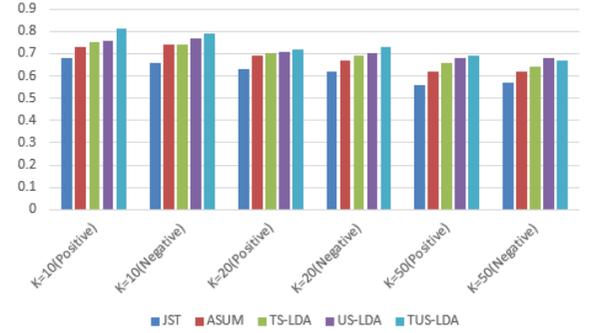
In this section, we performed a sentiment classification task to predict the sentiment labels of the test data in Sentiment140. Note that the Sentiment140 tweets do not contain neutral tweets. We determined the polarity of a tweet m by selecting the polarity s that has a higher probability in π_m^s (π_m is the sentiment distribution of the m -th post), the function is shown in Eq 9.

$$polarity(m) = \operatorname{argmax}_{s \in \{neg, pos\}} \pi_m^s \tag{9}$$

We present the results of sentiment classification with *Accuracy*, *Precision*, *Recall* and *F1* which are defined in the following.

Table 5. Average coherence score on the top T words in the K topics discovered on tweets of electronic products

T	Top 5			Top 10			Top 20		
	10	20	50	10	20	50	10	20	50
JST	-39.88	-42.08	-41.68	-242.74	-246.79	-251.97	-1139.43	-1145.36	-1142.01
ASUM	-38.02	-39.86	-39.58	-240.47	-243.97	-246.47	-1135.44	-1131.96	-1135.27
TS-LDA	-35.53	-37.66	-38.91	-238.29	-240.87	-244.71	-1136.04	-1133.02	-1130.81
US-LDA	-36.42	-36.84	-36.59	-237.01	-238.67	-245.29	-1134.07	-1130.45	-1132.23
TUS-LDA	-33.91	-35.7	-35.61	-233.08	-234.72	-241.78	-1030.83	-1127.64	-1130.02

(a) Proportion of *coherent* topics generated by each model in $K = 10, 20, 50$ (b) Average Precision @20 (p @20) of words in *coherent* topics generated by each model in $K = 10, 20, 50$ **Figure 4.** (a) Proportion of *coherent* topics (b) Average Precision @20 (p @20) of words in *coherent* topics

Accuracy is the proportion of true results (both true positives and true negatives) among the total number of cases examined in the binary classification.

Precision is the proportion of the true positives against all the predicated positive results (both true positives and false positives) in the binary classification.

Recall is the proportion of the true positives against all the actual positive results (both true positives and false negatives) in the binary classification.

F1 is the harmonic mean of **Precision** and **Recall**.

Based on the results of sentiment classification, we can see that TUS-LDA outperformed JST, ASUM, TS-LDA and US-LDA in $F1$ (Fig 3(d)). For *Recall* (3(c)), ASUM, TS-LDA, US-LDA and TUS-LDA performed equally well, JST performed worst. For *Accuracy* (Fig 3(a)) and *Precision* (Fig 3(b)), TUS-LDA performed best and TS-LDA performed better than US-LDA. There exist 48 timeslices and 21,815 users, the number of users is far more than that of timeslices which causes that modeling tweets aggregated in timeslices performed better than tweets aggregated in users. Aggregating tweets in timeslices or users (i.e., TUS-LDA) with $K = 10$ performed best in Sentiment140.

5.4.2 Topic Coherence

Another goal of TUS-LDA is to extract coherent sentiment-aware topics from user-generated post collection and evaluate the effectiveness of topic and sentiment captured by our models. In order to conduct quantitative evaluation of topic coherence, we used an automated metric proposed in [20], which is shown in Eq 10, where topic coherence, denoted as $D(v)$, is the document frequency of word v , $D(v, v')$ is the co-document frequency of word v and v' and $V^{(k)} = (v_1^{(k)}, \dots, v_T^{(k)})$ is a list of the T most probable words in topic k . The key idea of the coherence score is that if a word pair is related

to the same topic, they will co-occur frequently in the corpus. In order to quantify the overall coherence of the discovered topics, the average coherence score, $\frac{1}{K} \sum^k C(z_k; V^{(z_k)})$, was utilized. We conducted and evaluated the topic extraction experiments on the tweets of electronic products. Here we also compared TUS-LDA with four sentiment-topic models: JST, ASUM, TS-LDA and US-LDA. In this collection, we set the number of topics $K = 10, 20, 50$ for all the methods. The result is listed in Table 5. From the topic coherent results, it is clear that aggregating tweets in timeslices or users (TUS-LDA) directly leads to significant improvement of topic coherent. Note that TUS-LDA also performed best in the topic coherent and the performance of TS-LDA (aggregating tweets in timeslices) was similar to US-LDA (aggregating tweets in users).

$$C(t; V^{(t)}) = \sum_{m=2}^M \sum_{l=1}^{m-1} \log \frac{D(v_m^{(t)}, v_l^{(t)}) + 1}{D(v_l^{(t)})} \quad (10)$$

5.4.3 Human Evaluation

As our objective is to discover more coherent sentiment-aware topics, so we chose to evaluate the topics manually which is based on human judgement. Without enough knowledge, the annotation will not be credible. Following [20], we asked two human judges, who are familiar with common knowledge and skilled in looking up the test tweet dataset, to annotate the discovered sentiment-aware topics manually. To ensure the annotation reliable, we labeled the generated topics by all the baseline models and our proposed model at learning iteration 10.

Topic Labeling: Following [20], we asked the judges to label each sentiment-aware topic as *coherent* or *incoherent*. Each sentiment-aware topic is represented as a list of 20 most probable words in word distribution φ of the topic. Here they annotated a sentiment-aware topic as *coherent* when at least half of top 20 words were

Table 6. Example of topics extracted by TUS-LDA

Positive sentiment label			Negative sentiment label		
Topic 1	Topic 2	Topic 3	Topic 1	Topic 2	Topic 3
camera	ipod	xbox	printer	window	phone
digit	song	game	ink	vista	problem
canon	phone	live	print	us	information
nikon	listen	sale	cartridge	microsoft	security
new	music	console	toner	install	strange
len	love	plai	laser	download	risk
photograph	new	playstat	color	software	finiance
review	play	ps3	laserjet	file	mobile
panason	shuffle	microsoft	paper	free	digit
slr	good	new	scanner	server	on-line

related to the same semantic-coherent concept (e.g., an event, a hot topic) and the sentiment polarities of the words are accurate, others were *incoherent*.

Word Labeling: Then we chose *coherent* sentiment-aware topics which were judged before and asked judges to label each word of the top 20 words among these *coherent* sentiment-aware topics. When a word was in accordance with the main semantic-coherent concept that represents the topic, the word was annotated as *correct* and others were *incorrect*. After topic labeling, the judges had known the concept of each sentiment-aware topic and the overall sentiment of the topic, it is easy to label words of each sentiment-aware topic. As is shown in Table 7, the annotation of both judges in *Precision@20* (or $p@20$) also have good agreements (Cohen’s Kappa score is greater than 0.8 [13]).

Table 7. Cohen’s Kappa for pairwise inter-rater agreements

	Topic Labeling	Word Labeling		
		p@5	p@10	p@20
Kappa	0.820	0.911	0.821	0.816

Figure 4(a) shows that TUS-LDA can discover more *coherent* topics than JST, ASUM, TS-LDA and US-LDA. Thereinto, TUS-LDA can discover the nearly equal number of positive and negative topics. Figure 4(b) gives the average *Precision@20* of all coherent topics. TUS-LDA performed better than other four models and performed best in $K = 10$.

From the above, we can observe that aggregating posts in the same timeslice or user as a single document can indeed improve the performance in sentiment classification and sentiment-aware topic extraction in user-generated posts as TUS-LDA consistently outperformed the baseline models except in $K = 50$ (*Negative*). Also the empirical results reveal that the most likely number of topic for tweets of electronic products in Twitter7 is 10.

5.5 Qualitative Analysis

To investigate the quality of topics discovered by TUS-LDA, we randomly choose some topics for visualization. We randomly selected six topics, i.e., three positive topics and three negative topics. For each topic, we choose the top 10 words which can most represent the topic.

Table 6 presents the top words of the selected topics. The three topics with a positive sentiment label respectively talk about “Camera”, “apple music product” and “game” and these topics are listed in the left columns of Table 6; the three negative topics are related to “printer”, “window product” and “phone” are listed in right columns of Table 6. As we can see clearly from Table 6, the six topics are

quite explicit and coherent, where each of them tried to capture the topic of a kind of electronic product. In terms of topic sentiment, by checking each of the topics in Table 6, it is clear that Topic 2 under the positive sentiment label and Topic 3 under the negative sentiment label indeed bear positive and negative sentiment labels respectively. However, other topics under positive and negative sentiment label carry fewer sentiment words than the above two topics. By manually examining the tweet data, we observe that the sentiment labels of these topics are accurate. The analysis of these topics shows that TUS-LDA can indeed discover coherent sentiment-aware topics.

6 Conclusion and Future Work

In this paper, we studied the problem of sentiment-aware topic detection from the user-generated posts on the social media. The existing work is not suitable for the short and informal posts, we proposed a new sentiment/topic model that considers the time, user information of posts to jointly model topics and sentiments. Based on the different characteristics of sentiments and topics, we limited that words in the same post belong to the same topic, but they can belong to different sentiments. We compared our model with JST, ASUM as well as two degenerate variations of our model on two Twitter datasets. Our quantitative evaluation showed that our model outperformed other models both in sentiment classification and topic coherence. At the same time, we asked two judges to evaluate our models and baseline methods and the result also showed that our model TUS-LDA performed best in sentiment-aware topic extraction. Moreover, we used six examples to visualize some sentiment-aware topics. In the future work, we want to further mine sentiment-aware events in the posts which can monitor the sentiment variation over time of each event. Moreover, we can also utilize the user’s topic and sentiment information to cluster similar users. We will also consider to expand our model for aspect-based opinion mining.

ACKNOWLEDGEMENTS

We would like to thank the reviewers for their comments, which helped improve this paper considerably. This work is supported in part by the National Natural Science Foundation of China (NSFC) under Grant No.61272378 and the 863 Program under Grant No.2015AA015406.

REFERENCES

- [1] David M Blei, Andrew Y Ng, and Michael I Jordan, ‘Latent dirichlet allocation’, *Journal of Machine Learning Research*, **3**, 993–1022, (2003).

- [2] Johan Bollen, Huina Mao, and Xiaojun Zeng, 'Twitter mood predicts the stock market', *Journal of Computational Science*, **2**(1), 1–8, (2011).
- [3] Zhiyuan Chen, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh, 'Leveraging multi-domain prior knowledge in topic models', in *Proc. of IJCAI*, pp. 2071–2077. AAAI, (2013).
- [4] Mohamed Dermouche, Julien Velcin, Leila Khouas, and Sabine Loudcher, 'A joint model for topic-sentiment evolution over time', in *Proc. of ICDM*, pp. 773–778. IEEE, (2014).
- [5] Qiming Diao, Jing Jiang, Feida Zhu, and Ee-Peng Lim, 'Finding bursty topics from microblogs', in *Proc. of ACL*, pp. 536–544. ACL, (2012).
- [6] Charles J Geyer, 'Practical markov chain monte carlo', *Statistical Science*, 473–483, (1992).
- [7] Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith, 'Part-of-speech tagging for twitter: Annotation, features, and experiments', in *Proc. of ACL*, pp. 42–47. ACL, (2011).
- [8] Alec Go, Richa Bhayani, and Lei Huang, 'Twitter sentiment classification using distant supervision', *CS224N Project Report, Stanford*, **1**, 12, (2009).
- [9] Gregor Heinrich, 'Parameter estimation for text analysis', Technical report, Technical report, (2005).
- [10] Thomas Hofmann, 'Probabilistic latent semantic indexing', in *Proc. of SIGIR*, pp. 50–57. ACM, (1999).
- [11] Yohan Jo and Alice H Oh, 'Aspect and sentiment unification model for online review analysis', in *Proc. of WSDM*, pp. 815–824. ACM, (2011).
- [12] Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad, 'Nrc-canada-2014: Detecting aspects and sentiment in customer reviews', in *SemEval*, pp. 437–442. ACL, (2014).
- [13] JR Landis and GG Koch, 'The measurement of observer agreement for categorical data.', *Biometrics*, **33**, 159–174, (1977).
- [14] Kar Wai Lim and Wray Buntine, 'Twitter opinion topic model: Extracting product opinions from tweets by leveraging hashtags and sentiment lexicon', in *Proc. of CIKM*, pp. 1319–1328. ACM, (2014).
- [15] Chenghua Lin and Yulan He, 'Joint sentiment/topic model for sentiment analysis', in *Proc. of CIKM*, pp. 375–384. ACM, (2009).
- [16] Chenghua Lin, Yulan He, Richard Everson, and Stefan Ruger, 'Weakly supervised joint sentiment-topic detection from text', *IEEE Transactions on Knowledge and Data Engineering*, **24**(6), 1134–1145, (2012).
- [17] Bing Liu, *Web data mining: exploring hyperlinks, contents, and usage data*, Springer Science & Business Media, 2007.
- [18] Bin Lu, Myle Ott, Claire Cardie, and Benjamin K Tsou, 'Multi-aspect sentiment analysis with topic models', in *Proc. of ICDMW*, pp. 81–88. IEEE, (2011).
- [19] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai, 'Topic sentiment mixture: modeling facets and opinions in weblogs', in *Proc. of WWW*, pp. 171–180. ACM, (2007).
- [20] David Mimno, Hanna M Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum, 'Optimizing semantic coherence in topic models', in *Proc. of EMNLP*, pp. 262–272. ACL, (2011).
- [21] Subhabrata Mukherjee, Gaurab Basu, and Sachindra Joshi, 'Joint author sentiment topic model', in *SDM*, pp. 370–378. SIAM, (2014).
- [22] Thien Hai Nguyen and Kiyooki Shirai, 'Topic modeling based sentiment analysis on social media for stock market prediction', in *Proc. of ACL*, pp. 1354–1364. ACL, (2015).
- [23] Alexander Pak and Patrick Paroubek, 'Twitter as a corpus for sentiment analysis and opinion mining.', in *Proc. of LREC*, volume 10, pp. 1320–1326, (2010).
- [24] Sara Rosenthal, Preslav Nakov, Svetlana Kiritchenko, Saif M Mohammad, Alan Ritter, and Veselin Stoyanov, 'Semeval-2015 task 10: Sentiment analysis in twitter', *SemEval*, (2015).
- [25] Kim Schouten and Flavius Frasincar, 'Survey on aspect-level sentiment analysis', *IEEE Transactions on Knowledge & Data Engineering*, **28**(3), 813–830, (2016).
- [26] Peter D Turney and Michael L Littman, 'Measuring praise and criticism: Inference of semantic orientation from association', *ACM Transactions on Information Systems*, **21**(4), 315–346, (2003).
- [27] Hanna M Wallach, David M Mimno, and Andrew McCallum, 'Rethinking lda: Why priors matter', in *NIPS*, pp. 1973–1981, (2009).
- [28] Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng, 'A bitern topic model for short texts', in *Proc. of WWW*, pp. 1445–1456. Springer, (2013).
- [29] Qi Zhang, Yeyun Gong, Xuyang Sun, and Xuanjing Huang, 'Time-aware personalized hashtag recommendation on social media', in *Proc. of COLING*, pp. 203–212. ACL, (2014).
- [30] Wayne Xin Zhao, Jing Jiang, Hongfei Yan, and Xiaoming Li, 'Jointly modeling aspects and opinions with a maxent-lda hybrid', in *Proc. of EMNLP*, pp. 56–65. ACL, (2010).
- [31] Wayne Xin Zhao, Jiang Jing, Weng Jianshu, He Jing, Lim Ee-Peng, Yan Hongfei, and Li Xiaoming, 'Comparing twitter and traditional media using topic models', in *Proc. of ECIR*, pp. 338–349. Springer, (2011).
- [32] Chen Zheng, Li Chengtao, Sun Jian-Tao, and Jianwen Zhang, 'Sentiment topic model with decomposed prior', in *Proc. of SDM*, pp. 767–775. SIAM, (2013).