Gaining Insight into Quality Diversity

Joshua E. Auerbach, Giovanni Iacca[*] and Dario Floreano
Laboratory of Intelligent Systems
Ecole Polytéchnique Fédérale de Lausanne
Lausanne, Switzerland
{first name}.{last name}@epfl.ch

ABSTRACT

Recently there has been a growing movement of researchers that believes innovation and novelty creation, rather than pure optimization, are the true strengths of evolutionary algorithms relative to other forms of machine learning. This idea also provides one possible explanation for why evolutionary processes may exist in nervous systems on top of other forms of learning. One particularly exciting corollary of this, is that evolutionary algorithms may be used to produce what Pugh et al have dubbed *Quality Diversity* (QD): as many as possible different solutions (according to some characterization), which are all as fit as possible. While the notion of QD implies choosing the dimensions on which to measure diversity and performance, we propose that it may be possible (and desirable) to free the evolutionary process from requiring defining these dimensions. Toward that aim, we seek to understand more about QD in general by investigating how algorithms informed by different measures of diversity (or none at all) create QD, when that QD is measured in a diversity of ways.

Keywords

Evolutionary Computation; Non-objective search; Behavior Characterization; Robotics; Quality Diversity; Neuroevolution

1. INTRODUCTION

Traditionally, the primary focus of evolutionary computation has been on developing algorithms for black box optimization problems. That is, the performance of different algorithms are compared on their relative ability to quickly approach optimal solutions to problems where the only feedback is in the form of a fitness value corresponding to the performance of a given candidate solution. Accordingly, better algorithms are considered to be those that can achieve the most fit solutions in the fewest number of iterations. While the domains to which these algorithms are applied may be quite diverse, this notion of finding a single global optimum[1] holds sway over the vast majority of work in the field.

However, there has recently been a growing movement of researchers that believes innovation and novelty creation, rather than pure optimization, are the true strengths of evolutionary algorithms (EAs) relative to other forms of machine learning. In particular, a number of recent works in the field, have chosen to reframe the question of "can we develop an algorithm to find solutions better and/or faster than other techniques?", and have instead asked whether the innovative aspects of evolution can be leveraged to produce a diverse assortment of high quality solutions. Pugh et al [9] have dubbed this to be a quest for *Quality Diversity* (QD), and have presented an initial comparison of the existing algorithms created for this purpose: MAP-Elites, Novelty Search, and variants thereof.

At the same time, there is a growing body of evidence that evolutionary principles may be operating within the nervous systems of humans and other animals, and may be a crucial ingredient to insightful problem solving [5]. If this is the case, then it seems likely that ability of evolution to produce a diversity of solutions (e.g. self models [2], features [1, 10], motor actions [3]) may be what justifies their existence.

While these QD algorithms have been shown to be quite successful in several domains, they all contain one strongly limiting constraint: the *a priori* definition of a behavioral characterization (BC): a set of features that defines the dimensions along which diversity is sought. As Pugh et al [9] demonstrated the choice of BC can significantly impact the performance of various algorithms. However, while they compare different BCs, their comparisons are all made between variants of different algorithms that are each informed by the same BC on which QD is measured.

While computing a QD metric implies choosing the dimensions on which to measure diversity and performance, we propose that it may be possible (and desirable) to free the evolutionary process from requiring defining these dimensions. Ultimately, we seek to create Darwinian neurodynamic methods that allow robots to invent their own notion of what defines "useful" diversity, but here, as a first step, we seek to understand more about QD in general by

[*] G. Iacca is also affiliated with the Département d'Ecologie et d'Evolution, Université de Lausanne, Keller Group, and with INCAS^3, The Netherlands.

[1] Or a Pareto front of global optima in the case of multi-objective problems.

GECCO'16 Companion, July 20 - 24, 2016, Denver, CO, USA

ISBN 978-1-4503-4323-7/16/07…$15.00

DOI: http://dx.doi.org/10.1145/2908961.2931675

investigating how algorithms informed by different BCs (or none at all) create QD when QD is measured on BCs that may be entirely different from (and unrelated to) the ones used to drive the evolutionary process.

2. RELATED WORK

In this section we describe some of the most relevant work. However, due to the brevity of this contribution, this overview is necessarily incomplete. We direct the reader to elsewhere in the literature, especially [9], for a more thorough background.

Novelty Search (NS) [6] introduced the concept of "non-objective" search, or searching without objectives. Effectively it replaces the fitness in a traditional EA with a novelty score, which is computed as the mean distance of an individual's BC to that of its k-nearest neighbors amongst the current population and an archive of previously evaluated individuals. Counter-intuitively, NS is often able to outperform standard objective search when compared on the fitness of evolved solutions–even though NS is not selecting for fitness it can actually find more fit individuals than algorithms that explicitly select for fitness. Furthermore, more recent work has demonstrated that in addition to finding ultimately more fit solutions, selecting for novelty is useful in itself for building up a repertoire of different behaviors [4].

It was this latter idea that inspired MAP-Elites [8], which makes this concept more explicit by building up a map of *diverse and fit* individuals: a discretization of the behavioral space into bins, where each bin maintains the most fit individual whose BC vector falls into that bin.

An alternative algorithm, which has not been previously studied in the context of Quality Diversity is Viability Evolution (ViE) [7]. ViE represents an alternative abstraction of artificial evolution, which, similar to non-objective search, does not require the formulation of an explicit fitness function. Taking inspiration from viability theory in dynamical systems, natural evolution and ethology, ViE instead works by eliminating individuals that do not meet a set of changing criteria, but shows no preference within the set of viable individuals.

3. EXPERIMENTS

Our experimental setup is structured as follows. We consider four different variants of a 2D robot maze navigation task inspired by the task explored in [9]. These four variants are referred to as *base maze* (BM), *hard maze* (HM), *base maze freeze on contact* (BMFC), and *hard maze freeze on contact* (HMFC). All tasks involve a 2D circular robot, controlled by an evolved neural network attempting to reach a goal location. The base and hard maze are shown in Fig. 1, where the blue and yellow circles indicate the starting position and the goal, respectively. The topology of the two mazes is loosely modeled after the "QD-Maze" used in [9], the main difference being only the shape and position of some of the obstacles. As their "QD-Maze", our two mazes contain numerous deceptive traps (to navigate to the goal a robot must first navigate *away* from it, possibly multiple times), but also allow many possible paths to the goal.

The two mazes differ in that the central cul-de-sac is closer to the goal in the hard maze, and the hard maze additionally contains two walls that narrow the passages towards the goal, thus making the problem more deceptive.

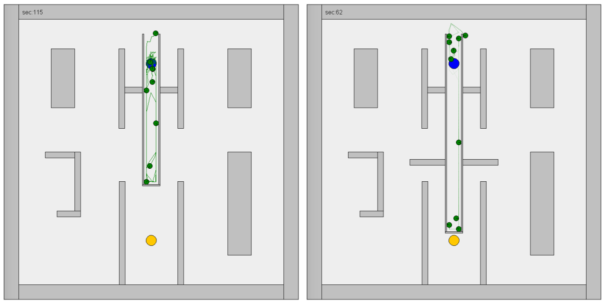


Figure 1: The base maze (left) and hard maze (right)

In the "freeze on contact" variants, we use the same mazes but we make the task even harder since we stop the robots as soon as they touch a wall. This forces the robots to additionally learn to avoid walls and should penalize "wall sliding" behaviors or any other behavior that would bounce the robots between obstacles until they reach the goal.

In the simulations[2], we model the kinematics of an e-puck[3] like wheeled robots, controlled by a Feed Forward Neural Network with 9 input nodes, 5 hidden nodes, and 2 output nodes. The first 8 inputs are binary signals coming from 8 infrared sensors evenly spaced around the robot. The last input measures the relative bearing towards the goal. All inputs are scaled in $[-1, 1]$. The two outputs are the commands to the two wheels. Hidden and output layers use the sigmoid activation function. Each network is encoded as a set of 62 real-valued weights in the range $[-3, 3]$.

On each maze variant, we run five search algorithms: MAP-Elites [9], Novelty Search [6], Viability Evolution [7], random search, and a standard fitness-based Genetic Algorithm (GA). The latter is tested in two different configurations of selection pressure, i.e. with a parent population (μ) of 100 and 2000 individuals. The robot's fitness is measured as its Euclidean distance to the goal at the end of the simulation. Each algorithm is repeated 30 times, each continued for $250,000$ evaluations. To compare the algorithms, we consider the four BCs used in [9] (EndpointBC, FullTrajectoryBC, HalfTrajectoryBC, DirectionBC) and the QD metric proposed therein.

To assess the importance of the feature choices on quality diversity, we run both MAP-Elites and Novelty Search on each of the four BCs, while also collecting the information on the three other BCs (although these are not used in the search). This is accomplished by maintaining a "QD map" for each run that holds, for each bin of a "MAP-Elites style" map, defined for each BC, the best fitness seen in that bin over the entire run. Similarly, for the other algorithms that are not explicitly informed by one of the BCs we also record information about all four BCs so that we can ultimately compare each algorithm using any of the BCs, even those different from what informed their evolution (if any). In total, considering 30 replicates of 12 different algorithmic variants (4 MAP-Elites, 4 Novelty, 4 other), and 4 task variants (BM, HM, BMFC, HMFC), we run 1440 experiments.

From the saved QD maps for each run we are then able to compute QD scores based on the different BCs to score

[2] The source code of the simulator and algorithms is available at https://github.com/lis-epfl/qd_experiments.

[3] http://www.e-puck.org

how well the different algorithms compared on producing a diverse assortment of well-performing solutions. We do this in two ways. The first, *Fitness QD*, is computed as QD is computed in [9]: we sum up the best fitness achieved in all bins of the relevant QD map for each run to arrive at a final score. In this way QD is increased by filling more bins and finding more fit solutions in each bin. However, since fitness may be a deceptive measure, QD computed in this way may also be deceptive (e.g. robots navigating to the bottom of the cul-de-sac receive high fitness even though stuck in a dead-end), so here we introduce a second QD metric, dubbed *Success QD*. Success QD does not consider fitness directly, but only considers whether a solution is successful in finding the goal (defined in terms of reaching a distance to the goal location under a defined threshold that excludes solutions stuck in the cul-de-sac). Specifically, Success QD is defined as the number of bins where a successful solution has been found. Since this is not based on fitness, it will be free from the the aforementioned deceptiveness.

4. RESULTS

A sampling of results from these experiments is presented in Fig. 2, and explained in the caption. The first observation is that our experiments allow us to recover the results obtained in [9]. In particular, we observe that the standard GA shows comparatively lower Fitness QD in all experiments, w.r.t. the remaining algorithms. We also confirm that among the algorithms that make explicit use of a behavior space during the search (MAP-Elites and Novelty Search), Novelty Search performs best when using BCs more "aligned" with fitness, while MAP-Elites tend to perform increasingly better as the behavior features get less aligned with the fitness (i.e., moving from the strongly-aligned EndpointBC to the least-aligned DirectionBC).

Additionally, our extensive experimental configuration also allow us to collect evidence for several new findings, which can be summarized as follows: **(1)** The relative performance of different algorithms varies depending on the difficulty of the task environment. **(2)** The relative performance of different algorithms on a given task also depends on whether the comparison is based on Fitness QD or Success QD. Fitness QD, like fitness itself, may be deceptive and lead to the false conclusion that one algorithm is outperforming another, when in fact it discovers fewer different, successful solutions. **(3)** Viability Search [7], which like the fitness and random variants is not informed by any BC, can often achieve better performance than other methods which are not informed by the BC that is used to calculate QD. **(4)** Different BCs drive the search in different ways, and counter-intuitively algorithms driven by different BCs from which QD is calculated on can outperform variants that were actually informed by the BC used to calculate QD (compare how MAP-Elites run with either FullTrajectoryBC or HalfTrajectoryBC rate on Success QD in the top-right plot).

5. CONCLUSION

Here, we have presented a snapshot of ongoing work attempting to understand the influence of behavioral characterizations on algorithms that explicitly or implicitly attempt to replicate the Quality Diversity seen in the natural world. While this exploration is still a work in progress, we have presented a number of interesting observations that can be made when different characterizations of QD are considered, including those calculated on Behavior Characterizations different from what had been used to inform the evolutionary processes, as well as those calculated by only considering task success rather than (a possibly deceptive) fitness value.

The general notion of QD – using evolutionary algorithms to find a diverse assortment of high performing solutions – is clearly an important advance in the field of Evolutionary Computation (and in machine learning more generally): it has enabled impressive results, especially as a mechanism for a robot to learn a useful behavioral repertoire (as done in [3, 4]), which may be indicative of similar processes occurring in the brains of natural organisms. Still, the true goal is to find algorithms that do not require defining a BC *a priori*, and when comparing algorithms caution is required in terms of how QD is calculated, as different formulations of QD may themselves be deceptive. Ongoing and future work is examining alternative algorithms that do not rely on an explicitly defined BC and exploring the notion of QD in more complex task domains.

6. ACKNOWLEDGMENTS

This work was supported by the Swiss National Science Foundation under grant no 141063, the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 308943 (FET-OPEN Insight project), and the European Union's Horizon 2020 research and innovation programme under grant agreement no 665347 (FET-OPEN Phoenix project).

7. REFERENCES

[1] J. E. Auerbach, C. Fernando, and D. Floreano. Online Extreme Evolutionary Learning Machines. In *Artificial Life 14: Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems*, pages 465–472. The MIT Press, 2014.

[2] J. Bongard, V. Zykov, and H. Lipson. Resilient machines through continuous self-modeling. *Science*, 314:1118–1121, 2006.

[3] A. Cully, J. Clune, D. Tarapore, and J.-B. Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, 2015.

[4] A. Cully and J.-B. Mouret. Behavioral repertoire learning in robotics. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 175–182. ACM, 2013.

[5] C. Fernando, R. Goldstein, and E. Szathmáry. The neuronal replicator hypothesis. *Neural computation*, 22(11):2809–2857, 2010.

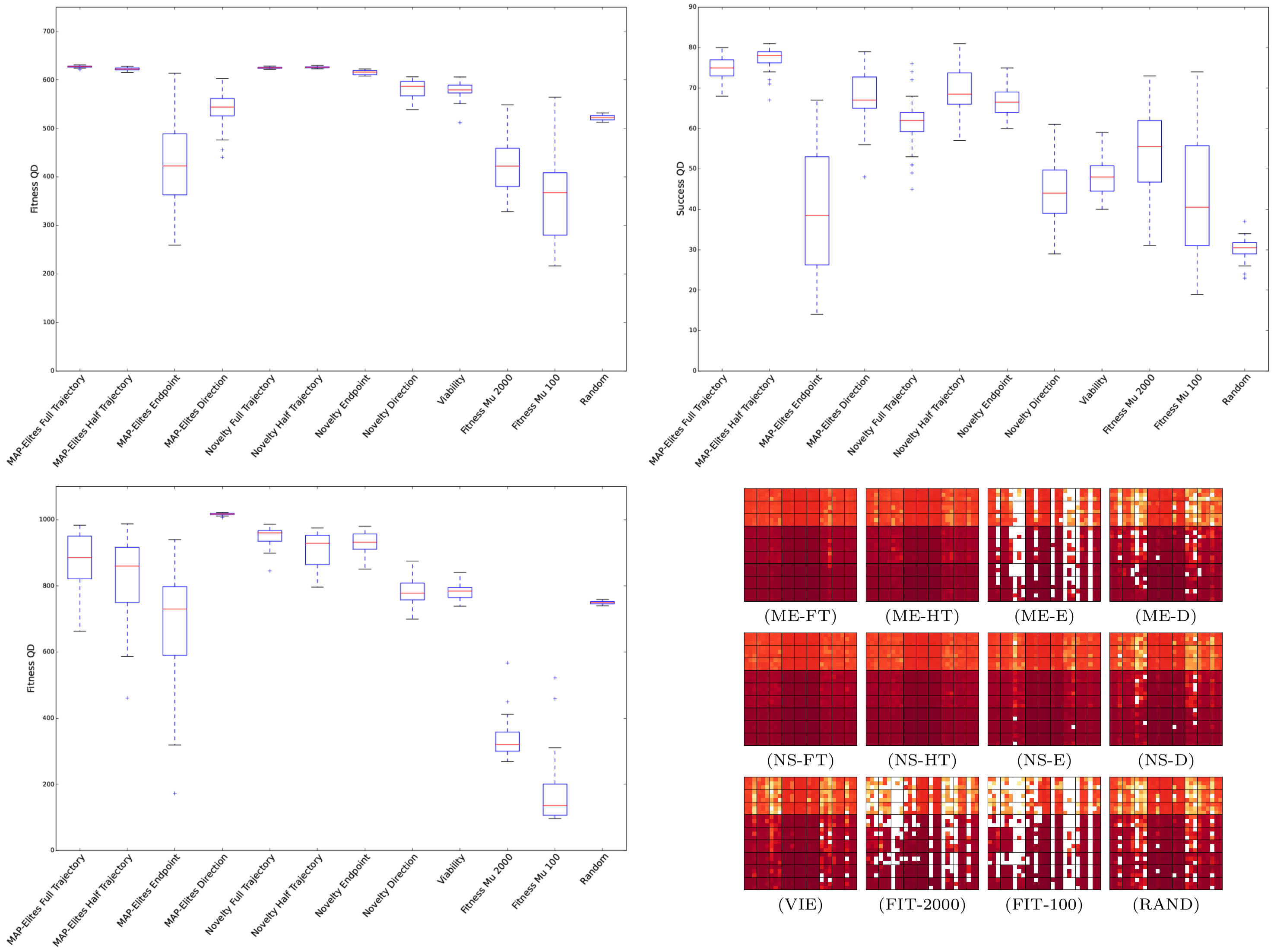
[6] J. Lehman and K. O. Stanley. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary Computation*, 19(2):189–223, 2011.

[7] A. Maesani, P. R. Fernando, and D. Floreano. Artificial evolution by viability rather than competition. *PLoS ONE*, 9(1):1–12, 01 2014.

[8] J. Mouret and J. Clune. Illuminating search spaces by mapping elites. *CoRR*, abs/1504.04909, 2015.

[9] J. K. Pugh, L. B. Soros, P. A. Szerlip, and K. O. Stanley. Confronting the challenge of quality diversity. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, GECCO '15, pages 967–974, New York, NY, USA, 2015. ACM.

[10] P. A. Szerlip, G. Morse, J. K. Pugh, and K. O. Stanley. Unsupervised feature learning through divergent discriminative feature accumulation. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.



(ME-FT) (ME-HT) (ME-E) (ME-D)

(NS-FT) (NS-HT) (NS-E) (NS-D)

(VIE) (FIT-2000) (FIT-100) (RAND)

Figure 2: **Comparison of QDs**. Box plots of the final Fitness QDs calculated on FullTrajectoryBC on the base maze (top-left); Success QD calculated on FullTrajectoryBC on the base maze (top-right); Fitness QD calculated on DirectionBC on the hard maze freeze on contact (bottom-left). **Fitness QD heatmaps**. Final Fitness QD map of the median-performing run of each algorithm on FullTrajectoryBC on the base maze (bottom-right). Legend: MAP-Elites FullTrajectory (ME-FT); MAP-Elites HalfTrajectory (ME-HT); MAP-Elites Endpoint (ME-E); MAP-Elites Direction (ME-D); Novelty Search FullTrajectory (NS-FT); Novelty Search HalfTrajectory (NS-HT); Novelty Search Endpoint (NS-E); Novelty Search Direction (NS-D); Viability (VIE); Fitness-based GA with $\mu = 2000$ (FIT-2000); Fitness-based GA with $\mu = 100$ (FIT-100); Random search (RAND).