Open-Ended Evolution with Multi-Containers QD

Stephane Doncieux
Sorbonne Université, CNRS, Institut des Systèmes Intelligents et de Robotique, ISIR, F-75005 Paris, France
stephane.doncieux@sorbonne-universite.fr

Alexandre Coninx
Sorbonne Université, CNRS, Institut des Systèmes Intelligents et de Robotique, ISIR, F-75005 Paris, France
alexandre.coninx@sorbonne-universite.fr

CCS CONCEPTS

• **Computing methodologies** → **Evolutionary robotics**; **Evolutionary robotics**; **Evolutionary robotics**; *Cognitive robotics*; *Batch learning*; • **Computer systems organization** → **Robotics**; **Evolutionary robotics**; **Robotic autonomy**;

KEYWORDS

Quality Diversity algorithms; novelty search; evolutionary robotics; open-ended evolution

ACM Reference Format:
Stephane Doncieux and Alexandre Coninx. 2018. Open-Ended Evolution with Multi-Containers QD. In *GECCO '18 Companion: Genetic and Evolutionary Computation Conference Companion, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3205651.3205705

Evolution in nature has allowed biological systems to develop and survive in many different environments. It can be attributed to one of the major features of natural evolution: its open-endedness, that can be considered as the ability to continuously produce novelty and/or complexity [1]. This feature is critical to allow an agent to continuously adapt to its environment. We propose here an extension of Quality Diversity algorithms to make it more open-ended. Quality Diversity algorithms aim at generating a large set of diverse solutions [3, 10]. They can be used in a two-steps process in which (1) a diverse set of solutions is generated offline before (2) the agent can exploit it online to find the behavior that fits to the situation [2, 4–6]. QD algorithms main goal is to fill a behavior space whose dimensions are defined by behavior descriptors. Current QD algorithms can generate novelty and complexity, but within the frame of these behavior descriptors. To make the evolutionary process more open-ended, we extend Cully and Demiris framework [3] and introduce Multi-Container Quality Diversity algorithms (MCQD). MCQD are QD algorithms relying on multiple behavior spaces that can be added on the fly. MCQD thus aims at implementing an open-ended evolutionary process that would explore any new behavior space that could be found of interest during the search process.

Cully and Demiris framework define the container as the core component of a QD algorithm [3]. The container is the repertoire of behaviors that the QD algorithm will try to fill. It is an archive in novelty-based approaches [7], and a grid in MAP-Elites based approaches [8]. The QD algorithm relies on this container to select the individuals that will feed the next generation. In the proposed extension, we use multiple containers, each having its own behavior descriptor. Each container is also associated to a condition of eligibility. An individual will be eligible to a container if it fulfills this condition. For instance, in a robot locomotion problem, an individual would be eligible in a container focused on flight only if the individual can fly. An individual has a behavior descriptor in each of the containers it is eligible for. This approach allows to add new containers during the QD search, for instance, when new environments are discovered, associating them with new descriptor spaces and eligibility conditions suited to this environment.

The MCQD approach proposed here is an extension of novelty search with local competition [7] in which the novelty and local competition scores have been adapted to deal with multiple containers. Because of the triangle inequality, the sum of novelties in each container is greater than the novelty computed on the concatenation of the descriptor spaces. We have thus chosen to compute the novelty of an individual i as the sum of novelties in $\mathcal{E}_i$, the behavior spaces the individual is eligible for:

$$n_i(t) = \sum_{k \in \mathcal{E}_i} n_i^{(k)}(t) \quad (1)$$

where $n_i^{(k)}(t)$ is the novelty score of individual i in the behavior space k.

In order to select individuals whose fitness is locally high in all descriptor spaces, the local competition score of individual i is defined as the min of the local competition scores in each container:

$$lc_i = min_{k \in \mathcal{E}_i}(lc_i^{(k)}) \quad (2)$$

To test the proposed approach, we introduce "the cube" an experimental setup design to test open-ended learning ability[1]. In this setup, the agent has to go through a sequence of rooms. Each room is associated with a task that the agent has to solve to enter the next room. The goal for the agent is to solve as many rooms as possible. The proposed experiment is an extension of the collectball experiment [9]. There are 5 different kinds of tasks: (1) finding the exit of a room (exit), (2) activating a switch (switch), (3) exploring a given percentage of a room (surface), (4) finding a key and bringing it to a door (key) and (5) collecting balls (collectball), similar to key, but the agent needs to release the ball when in front of the basket whereas the agent just needs to be in front of the door while holding the key to exit a key-room. The agent is a 2-wheeled robot with 3 effectors, 2 for the wheel motors and a third one to trigger actions like collecting a ball. It has 3 laser range sensors and 2 bumpers to detect walls, 2 ball detectors and likewise 2 switch detectors and 2 lock/basket detectors. There is also a room-specific input that is

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07.
https://doi.org/10.1145/3205651.3205705

[1] The name comes from a science fiction movie called "Cube" that has inspired this setup.

constant and different for each room. It is controlled by a neural network using a direct NEAT-like encoding [9].

Although MCQD does not require to predefine the number of rooms and new ones could be defined procedurally, we have limited the number of rooms to 10 to make comparisons with mono-container approaches, i.e. novelty+fitness and novelty+local competition (Figure 1). In these setups, the behavior descriptors for a room is a set of 10 points in the trajectory of the robot. These descriptors have been concatenated into a single behavior descriptor for the mono-container approaches. We have also compared the results to a setup in which the fitness only is used. In all setups, the fitness is the number of solved rooms.

Figure 1 shows the behavior of an individual that has solved all 10 rooms. Figure 2 shows the fitness after 1000 generations and a population size of 200. The proposed approach is not statistically different from the novelty+fitness and is slightly superior to novelty+local competition. Unsurprisingly, these three setups are clearly above the fitness alone setup.

The proposed approach shows that decomposing the computation of the novelty score and local competition and computing it separately in each container does not have any negative impact on the proposed task. Contrary to mono-container approaches, in the proposed MCQD, containers can be added on-the-fly, thus making one step towards open-ended QD algorithms.

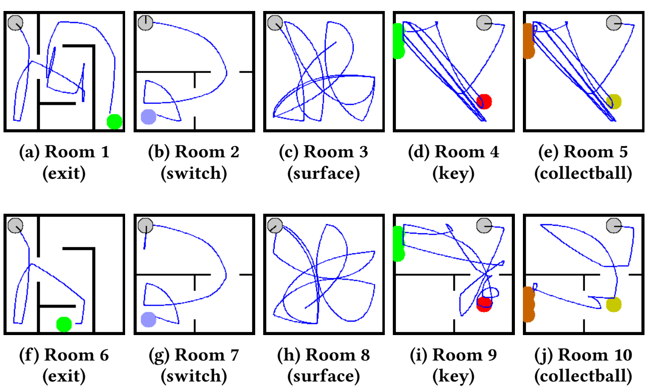

Figure 1: Sequence of rooms used for the experiment. The initial position and orientation of the robot are shown in grey, and an exemple trajectory of an individual with maximum fitness is drawn in blue. Exits are figured in green, switches in light blue, baskets in orange, initial position of balls in yellow, and initial position of keys in red. In surface rooms the robot needs to explore 50% of the room to succeed.

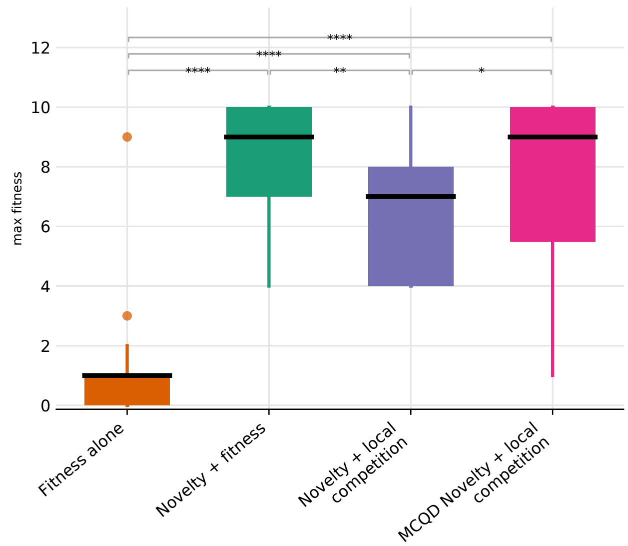

Figure 2: Fitness obtained at generation 1000. Global statistical significance was first validated using a Kruskal-Wallis test ($p < 0.001$) and pairwise differences were then evaluated using Mann-Whitney tests with p-values adjusted for multiple comparisons using the Holm-Bonferroni method. Statistical significance on the corrected Mann-Whitney tests are shown, with thresholds of 5% (*), 1% (**), 0.1% (***) and 0.01% (****).

ACKNOWLEDGEMENT

This work is supported by the DREAM project [2] through the European Unions Horizon 2020 research and innovation program under grant agreement No 640891.

[2] http://www.robotsthatdream.eu

REFERENCES

[1] Wolfgang Banzhaf, Bert Baumgaertner, Guillaume Beslon, René Doursat, James A. Foster, Barry McMullin, Vinicius Veloso De Melo, Thomas Miconi, Lee Spector, and Susan Stepney. 2016. Defining and simulating open-ended novelty: requirements, guidelines, and challenges. *Theory in Biosciences* 135, 3 (2016), 131–161.

[2] Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. 2015. Robots that can adapt like animals. *Nature* 521, 7553 (May 2015), 503–507. https://doi.org/10.1038/nature14422

[3] A. Cully and Y. Demiris. 2017. Quality and Diversity Optimization: A Unifying Modular Framework. *IEEE Transactions on Evolutionary Computation* PP, 99 (2017), 1–1. https://doi.org/10.1109/TEVC.2017.2704781

[4] A. Cully and J.-B. Mouret. 2015. Evolving a Behavioral Repertoire for a Walking Robot. *Evolutionary Computation* 24, 1 (Jan. 2015), 59–88. https://doi.org/10.1162/EVCO_a_00143

[5] M. Duarte, J. Gomes, S. M. Oliveira, and A. L. Christensen. 2017. Evolution of Repertoire-based Control for Robots with Complex Locomotor Systems. *IEEE Transactions on Evolutionary Computation* PP, 99 (2017). https://doi.org/10.1109/TEVC.2017.2722101

[6] Seungsu Kim and Stephane Doncieux. 2017. Learning Highly Diverse Robot Throwing Movements through Quality Diversity Search. In *2nd Evolution in Cognition Workshop, Proceedings of the Genetic and Evolutionary Computation Conference Companion.* ACM, 1177–1178.

[7] Joel Lehman and Kenneth O. Stanley. 2011. Evolving a Diversity of Virtual Creatures Through Novelty Search and Local Competition. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation (GECCO '11)*. ACM, New York, NY, USA, 211–218. https://doi.org/10.1145/2001576.2001606

[8] Jean-Baptiste Mouret and Jeff Clune. 2015. Illuminating search spaces by mapping elites. *arXiv:1504.04909 [cs, q-bio]* (April 2015). arXiv: 1504.04909.

[9] Jean-Baptiste Mouret and Stephane Doncieux. 2012. Encouraging Behavioral Diversity in Evolutionary Robotics: An Empirical Study. *Evolutionary computation* 20, 1 (2012), 91–133. https://doi.org/10.1162/EVCO_a_00048

[10] Justin K. Pugh, Lisa B. Soros, and Kenneth O. Stanley. 2016. Quality Diversity: A New Frontier for Evolutionary Computation. *Frontiers in Robotics and AI* 3 (2016). https://doi.org/10.3389/frobt.2016.00040