MAP-Elites for Noisy Domains by Adaptive Sampling

Niels Justesen
IT University of Copenhagen
noju@itu.dk

Sebastian Risi
IT University of Copenhagen
sebr@itu.dk

Jean-Baptiste Mouret
Inria, CNRS, Université de Lorraine
jean-baptiste.mouret@inria.fr

ABSTRACT

Quality Diversity algorithms (QD) evolve a set of high-performing phenotypes that each behaves as differently as possible. However, current algorithms are all elitist, which make them unable to cope with stochastic fitness functions and behavior evaluations. In fact, many of the promising applications of QD algorithms, for instance, games and robotics, are stochastic. Here we propose two new extensions to the QD-algorithm MAP-Elites — adaptive sampling and drifting-elites — and demonstrate empirically that these extensions increase the quality of solutions in a noisy artificial test function and the behavioral diversity in a 2D bipedal walker environment.

ACM Reference Format:
Niels Justesen, Sebastian Risi, and Jean-Baptiste Mouret. 2019. MAP-Elites for Noisy Domains by Adaptive Sampling. In *Genetic and Evolutionary Computation Conference Companion (GECCO '19 Companion), July 13–17, 2019, Prague, Czech Republic.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3319619.3321904

1 INTRODUCTION

Traditional optimization algorithms attempt to find a single configuration that minimizes some error function. In contrast, Quality-Diversity (QD) algorithms, inspired by natural evolution, attempt to find a large set of high-performing and diverse configurations. Two popular QD-algorithms are Novelty Search with Local Competition [3] and Multi-dimensional Archive of Phenotypic Elites (MAP-Elites) [4]. In this paper, we investigate the effect of noise on diversity, performance, and robustness of the solutions found by MAP-Elites, which has not been studied in depth despite most real-world problems being noisy. To deal with the issues introduced by noise we propose an adaptive sampling technique that gradually increases the number of samples used in each solution evaluation.

We formulate the QD-optimization problem as the maximization of $M(x_1, \cdots, x_N) = \sum_{i=1}^{N} \mathbb{E}(F(x_i))$, where x_i are the parameters of the solution in cell i within a user-defined feature space divided into N cells (or niches), $F(x) = f(x) + \delta_f(x)$, where $f(x)$ is the true performance/quality of x and δ_f is noise from an unknown distribution. The features determining which cell x belongs to is sampled from $B(x) = b(x) + \delta_b(x)$ and $M(\cdot)$ refers to the total expected quality and is maximal when the archive is filled with high-quality solutions.

A common approach to deal with noise in evolutionary algorithms is to evaluate each solution multiple times and take the average of all the performance measurements, also known as explicit averaging [5]. For QD-algorithms, the same can be done to determine the behavioral features. Explicit averaging introduces several issues for an elitist algorithm like MAP-Elites: (1) The number of evaluation trials is usually found experimentally by balancing between inaccurate estimations, leading to unstable solutions, and longer running times for the algorithm, (2) domains with a high level of noise require many trials to reach accurate results, which require more computation, and (3) the algorithm will over time over-estimate the fitness and thus prioritize unstable solutions.

2 METHODS

2.1 Adaptive Sampling & Drifting Elites

Intuitively, we need to re-evaluate more at the end of the evolutionary process (to get precise results) than at the beginning (when a rough approximate is acceptable). We thus propose an adaptive sampling approach along with an early-stopping rule. Previous adaptive sampling methods for EAs [1] are not directly applicable to MAP-Elites. The approach for evaluating a solution is shown in Algorithm 1, where lines in gray are unnecessary when $B(x) = b(x)$.

Algorithm 1 MAP-Elites with **adaptive sampling** and **drifting elites** for noisy performance measures and feature descriptors.

```
procedure EVALUATE(x)
    e ← ∅                                      ▷ The elite to challenge
    V ← ∅                                      ▷ Visited cells
    while e = ∅ or (|x_P| < |e_P|) do
        b, p ← SIMULATE(x)                     ▷ Measure behavior and perf.
        x_B ← x_B ∪ {b}, x_P ← x_P ∪ {p}
        e ← A(x̄_B)                             ▷ Get current elite from archive A
        c ← CELLINDEX(A, ē_B)                  ▷ The cell occupied by e
        if e = ∅ then
            return
        else if x̄_P < ē_P then                 ▷ Mean perf. of x and e is compared
            V(c) ← V(c) + 1                    ▷ Increment visit count
            b, p ← SIMULATE(e)                 ▷ Re-evaluate e once
            e_B ← e_B ∪ {b}, e_P ← e_P ∪ {p}
            c′ ← CELLINDEX(A, ē_B)
            if c ≠ c′ then
                Remove e from cell c           ▷ Elite is drifting
                EVALUATE(e)                    ▷ Resume evaluation of e
            e ← A(x̄_B)
            if e ≠ ∅ and x̄_P < ē_P then
                return
        if V(c) > |V|/2 and |x_P| < |e_P| then
            return
```

In this scheme, solutions are evaluated until one of the following occurs: 1) The solution is estimated to be in an empty cell, 2) the solution has been evaluated the same number of times as the corresponding elite and the solution has a higher mean performance, or 3) the mean performance is lower than the corresponding elite's.

Niels Justesen was financially supported by the Elite Research travel grant from The Danish Ministry for Higher Education and Science.

GECCO '19 Companion, July 13–17, 2019, Prague, Czech Republic

ACM ISBN 978-1-4503-6748-6/19/07…$15.00
https://doi.org/10.1145/3319619.3321904

If feature measures are noisy, case number 3 is only activated if the solution has visited the current cell more than 50% of the time; we believe it has *settled*. In case 1 and 2, the solution is added to the archive in the corresponding cell and in case 3 the solution is discarded. When discarding a solution, the elite will be evaluated one additional time to improve our estimations.

As solutions in the archive are re-evaluated over time and thus their behavioral descriptors change, solutions will have to be moved to new cells; i.e. they are *drifting*. A naive implementation of this idea will leave behind an empty cell whenever a solution *drifts*. Our solution to this is to store the k best solutions in each cell while only treating the fittest one as the elite. When an elite is moved to a new cell it will continue its evaluation procedure and the second most-fit solution in the cell becomes the new elite. We observed improvement by using $k = 10$ instead of $k = 1$ but saw no difference when using $k = 100$.

3 EXPERIMENTAL SETUP

We test three domains: (1) 6-D Rastrigin with noisy performance measures and deterministic feature measures: $f(\mathbf{x}) = 10n + \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i)\right]$ where x in $[-5, 10]$, $n = 6$, $F(x) = f(x) + \mathcal{N}(0, 625)$, and $B(x) = b(x)$; where $b(x)$ is equal to the two first values in x. 2) 6-D Rastrigin with noisy performance measures and noisy feature measures, such that $F(x) = f(x) + N(0, 625)$ and $B(x) = b(x) + \mathcal{N}(0, 0.01)$, and 3) the OpenAI Gym *BipedalWalker* environment [2] which is stochastic and thus no artificial noise is added. The variances of the added noise were determined by randomly sampling the search space.

We use the CVT variant of MAP-Elites [6], a batch size of 100, random initialization of 1,000 solutions, and for the mutation operator, sigma iso was set to 0.01 and sigma line to 0.2, with 25,000 samples to generate the CVT archive. We use 5,000 niches in the two Rastrigin experiments; for the BipedalWalker the number of niches is set to 1,000 and the neural network has 24 inputs, two fully-connected layers of 256 tanh units, and 4 outputs. Initial parameters are sampled uniformly random within $[-0.5, 0.5]$ ($[0, 1]$ for Rastrigin). We compare our approach to three baselines with explicit averaging of $n = 1$, $n = 10$, and $n = 100$. Rastrigin experiments were repeated three times each, while the BipedalWalker experiment was executed once. To analyze the solutions found by each algorithm we *correct* the archives by re-evaluating every elite (either by using the true value for Rastrigin or re-evaluating each elite 100 times for the BipedalWalker) and move them to their correct cells. This sometimes leaves cells empty (see Figure 1).

4 RESULTS & CONCLUSION

Figure 1 shows the *corrected* collection size, total normalized quality (the sum of all solution qualities normalized from the range $[-250, 0]$ for Rastrigin and $[-50, 300]$ for the BipedalWalker) and the number of elite evaluation trials. For Rastrigin with noisy performance measures our approach results in the best total normalized quality compared to the three baselines. For the baselines on Rastrigin with noisy performance and feature measures, the collection size and total quality degrade over time as unstable solutions populate the archive. While it seems that $n = 100$ is best here, if we were able to stop early, it requires many re-evaluations to monitor the degradation. For the BipedalWalker the same trend is not as apparent, only for $n = 1$. The best results here were obtained by $n = 10$ followed by our approach. These results suggest that the adaptive sampling approach needs to be controlled better as it may be domain-specific and depend on other hyper-parameters. How to control the growth of evaluation trials remain a challenge for future work while our results demonstrate the potential of adaptive sampling in elitist QD-algorithms.

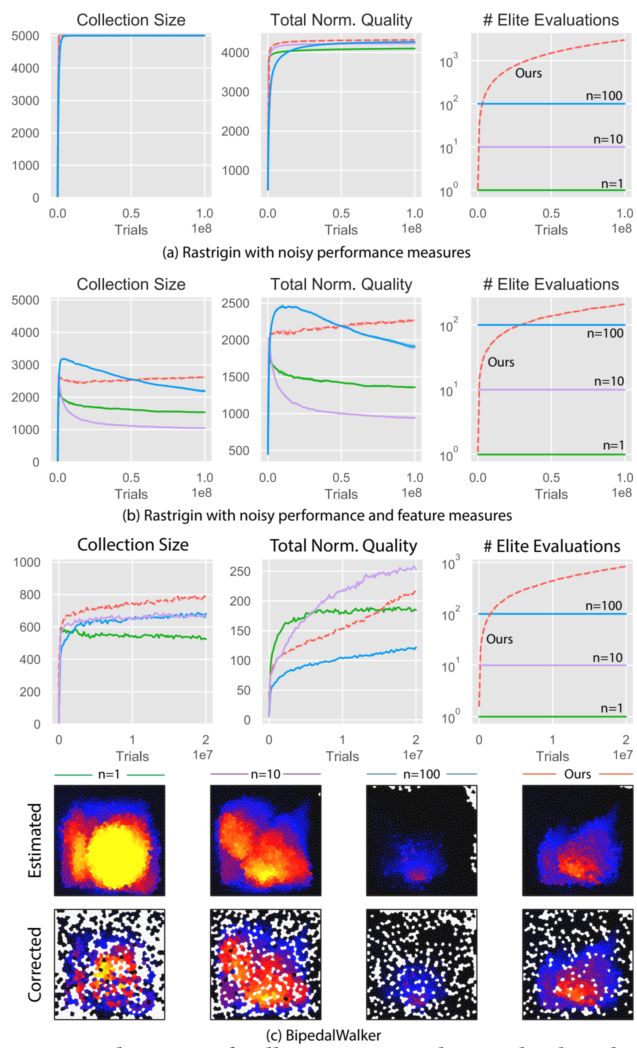

Figure 1: The *corrected* collection size, total normalized quality, and the mean number of elite evaluations for (a) Rastrigin with noisy performance measures and (b) feature measures, and (c) for the BipedalWalker which also shows the estimated and corrected archives.

REFERENCES

[1] Erick Cantú-Paz. 2004. Adaptive sampling for noisy problems. In *Proc. of GECCO*.
[2] O Klimov. 2016. BipedalWalkerHardcore-v2. http://gym.openai.com/. (2016).
[3] J Lehman and K O Stanley. 2011. Evolving a diversity of virtual creatures through novelty search and local competition. In *Proc. of GECCO*. ACM, 211–218.
[4] J-B Mouret and J Clune. 2015. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909* (2015).
[5] I Paenke, J Branke, and Y. Jin. 2006. Efficient search for robust solutions by means of evolutionary algorithms and fitness approximation. *IEEE Trans. on Evolutionary Computation* 10, 4 (2006), 405–420.
[6] V Vassiliades, K Chatzilygeroudis, and J-B Mouret. 2018. Using centroidal voronoi tessellations to scale up the multidimensional archive of phenotypic elites algorithm. *IEEE Trans. on Evolutionary Computation* 22, 4 (2018), 623–630.