Improving the Data Efficiency of Multi-Objective Quality-Diversity through Gradient Assistance and Crowding Exploration

Hannah Janmohamed
Imperial College London, InstaDeep
London, United Kingdom
hnj21@imperial.ac.uk

Thomas Pierrot
InstaDeep
Boston, USA
t.pierrot@instadeep.com

Antoine Cully
Imperial College London
London, United Kingdom
a.cully@imperial.ac.uk

ABSTRACT

Quality-Diversity (QD) algorithms have recently gained traction as optimisation methods due to their effectiveness at escaping local optima and capability of generating wide-ranging and high-performing solutions. Recently, Multi-Objective MAP-Elites (MOME) extended the QD paradigm to the multi-objective setting by maintaining a Pareto front in each cell of a MAP-ELITES grid. MOME achieved a global performance that competed with NSGA-II and SPEA2, two well-established multi-objective evolutionary algorithms, while also acquiring a diverse repertoire of solutions. However, MOME is limited by non-directed genetic search mechanisms which struggle in high-dimensional search spaces. In this work, we present Multi-Objective MAP-Elites with Policy-Gradient Assistance and Crowding-based Exploration (MOME-PGX): a new QD algorithm that extends MOME to improve its data efficiency and performance. MOME-PGX uses gradient-based optimisation to efficiently drive solutions towards higher performance. It also introduces crowding-based mechanisms to create an improved exploration strategy and to encourage greater uniformity across Pareto fronts. We evaluate MOME-PGX in four simulated robot locomotion tasks and demonstrate that it converges faster and to a higher performance than all other baselines. We show that MOME-PGX is between 4.3 and 42 times more data-efficient than MOME and doubles the performance of MOME, NSGA-II and SPEA2 in challenging environments.

CCS CONCEPTS

• **Theory of computation** → **Evolutionary algorithms**; • **Applied computing** → **Multi-criterion optimization and decision-making**; • **Computer systems organization** → *Evolutionary robotics.*

KEYWORDS

Quality-Diversity, Multi-Objective Optimisation, MAP-Elites, Neuroevolution



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike International 4.0 License.

GECCO '23, July 15–19, 2023, Lisbon, Portugal
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0119-1/23/07…$15.00
https://doi.org/10.1145/3583131.3590470

ACM Reference Format:
Hannah Janmohamed, Thomas Pierrot, and Antoine Cully. 2023. Improving the Data Efficiency of Multi-Objective Quality-Diversity through Gradient Assistance and Crowding Exploration. In *Proceedings of Genetic and Evolutionary Computation Conference (GECCO '23).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3583131.3590470

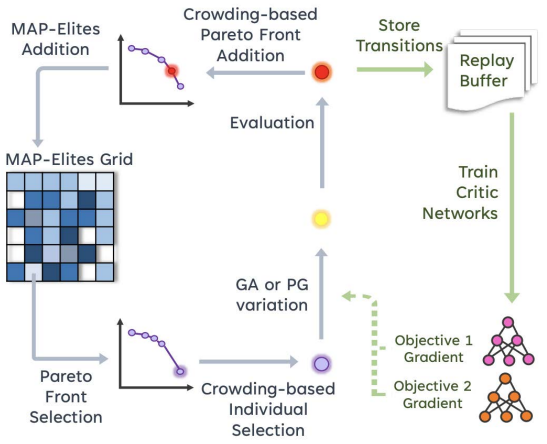


Figure 1: MOME-PGX algorithm. At each iteration, solutions are sampled, mutated and added back to the archive based on their Pareto-dominance and crowding distance. Separate to this main QD loop, a critic network per objective function is trained periodically, using transitions from a buffer.

1 INTRODUCTION

Quality-Diversity (QD) algorithms form a new optimisation paradigm designed to acquire a set of high-performing and diverse solutions, rather than the single optimum obtained by many traditional optimisation methods. This approach has proven useful in many domains where having more than one solution is a desirable outcome. For example, in robotics, learning a repertoire of controllers allows the robot to adapt in the event of damage [1, 4]. Similarly, QD algorithms have also been used to generate portfolios of architectural designs [15], create video game content [20] and manage scheduling tasks [30]. Even in applications where only a single solution is sought, QD has been observed to be a highly effective optimisation approach [14]. Specifically, maintaining a divergent population of solutions creates an implicit exploration

strategy, helping the optimisation process to avoid getting stuck in local minima and resulting in a higher global performance [14, 21].

As QD optimisation becomes increasingly capable of tackling complex problems, attention turns to how we can leverage them in the multi-objective case. Indeed, many real-life applications don't involve just a single objective, but rather a multitude of sophisticated and potentially contrasting ones. Canonical examples, such as wind turbine control [16] or computer hardware design [36], often seek the maximisation of some performance objective whilst minimising a cost function and obeying other safety constraints. But the scope and range of multi-objective problems is vast, including domains that range from drug design [18] to satellite communication [7].

The recently introduced Multi-Objective MAP-Elites (MOME) [28] unifies methods from Multi-Objective (MO) optimisation with the QD framework. Traditional MO algorithms don't aim to find a single optimal solution, but rather a set of solutions, each of which represents a different optimal trade-off of the given objectives. This set, referred to as the Pareto front, provides the end user with a variety of solutions to pick from. To incorporate this goal into QD, MOME aims to find a large collection of Pareto fronts, each of which occupies a different behavioural niche. While MOME shows promising results, it relies on non-directed genetic mutation operators, making it limited to simple tasks with low-dimensional search spaces.

In this paper, we present Multi-Objective MAP-Elites with Policy-Gradient Assistance and Crowding-based Exploration (MOME-PGX): a data-efficient and high-performing Multi-objective Quality-Diversity (MOQD) algorithm for tasks that can be formulated as Markov Decision Processes (MDP). The algorithm is visualised in Figure 1 and builds on top of MOME: Pareto fronts of solutions are stored in cells of a MAP-ELITES grid and, at each iteration, solutions are selected, mutated and potentially re-added to the grid based on their fitness. However, our proposed method makes two key contributions to improve the performance and data efficiency of MOME. Firstly, MOME-PGX uses gradient-based mutations to find high performing solutions for each objective. Secondly, it uses crowding-based criteria to alter the selection and addition mechanisms of MOME in order to increase exploration in sparse regions of the objective space and encourage a uniform distribution of points on the Pareto fronts.

We evaluate MOME-PGX on four simulated robotic control tasks using large neural-networks and compare its performance to a variety of standard QD and MO baselines. We find that MOME-PGX outperforms all standard baselines across all MOQD metrics in each of the tasks and doubles the performance of MOME, NSGA-II and SPEA2 in challenging tasks. Furthermore, MOME-PGX is shown to be between 4.3 and 42 times more data-efficient than MOME. Finally, we provide an ablation study to demonstrate that encompassing gradients and using crowding-mechanisms are both essential for improving the performance of MOME. Our code implementation was containerised and implemented using the QDax framework [22] for all methods and baselines, and is available at: https://github.com/adaptive-intelligent-robotics/MOME_PGX.

2 BACKGROUND

2.1 Quality-Diversity Optimisation

Traditional optimisation problems aim to find the single solution x from a search space $\mathcal{X}$ which maximises an objective function $f : \mathcal{X} \to \mathbb{R}$. By contrast, Quality-Diversity (QD) algorithms extend this objective by additionally encompassing a descriptor function $c : \mathcal{X} \to \mathbb{R}^d$ which maps solutions from the search space to d-dimensional descriptor vectors. Rather than seek a single solution x, QD algorithms aim to find a population of solutions $x_i \in X$ which both maximise the objective f and have diverse descriptor vectors.

MAP-ELITES algorithms [25, 28, 31] are a simple, yet powerful family of QD algorithms which achieve the QD goal by discretising the set of possible behaviour descriptors $S = c(\mathcal{X})$, referred to as the descriptor space, into a grid structure with k cells S_i which is used to store solutions. Specifically, the solution x_i is stored in the cell S_i that corresponds to its descriptor $c(x_i)$. In each iteration of MAP-ELITES, a batch of solutions is selected from the grid and mutated to form new solutions. Then, the new solutions are evaluated and potentially added back to the grid structure. Specifically, if the cell corresponding to the new solution's descriptor is empty, it is added. Otherwise, the new solution only replaces the existing one if it is higher-performing. This loop is repeated for a set budget and, as the algorithm progresses, the grid structure becomes incrementally populated with more diverse and higher-quality solutions.

2.2 Multi-objective Optimisation

A multi-objective optimisation problem considers the simultaneous maximisation of many objectives, $f_1(x), f_2(x), ...f_m(x)$, rather than a single one. Usually, these objectives come as a trade-off so there is not a single optimal solution, but rather a set of solutions that achieve different compromises across these objectives. Accordingly, the notion of Pareto-dominance is often used to induce a preference over solutions. Given two solutions x_1 and x_2 we say that x_1 dominates x_2 [36], or $x_1 \succ x_2$, if:

$$\forall i \in \{1, 2, ..., m\} : f_i(x_1) \geq f_i(x_2) \quad \wedge \quad \exists j \in \{1, 2, ..., m\} : f_j(x_1) > f_j(x_2)\,. \tag{1}$$

That is, one solution dominates another if it scores at least as high as the other across all objectives and strictly higher in at least one. Given a set of candidate solutions $\mathcal{S}$, the Pareto front is the set of solutions $\mathcal{P}(\mathcal{S}) \in \mathcal{S}$ that are not dominated by any other solutions in $\mathcal{S}$. In other words, the Pareto front represents all of the best possible trade-offs of the objective functions. The goal of MO optimisation is to approximate the Pareto front over the entire search space of solutions, $\mathcal{P}(\mathcal{X})$, known as the optimal Pareto front.

The performance of MO algorithms is commonly quantified via the hypervolume metric [2, 34] which reflects the coverage of the objective space achieved by solutions on the Pareto front. As visualised in Figure 2, for two objectives, this corresponds to the union of rectangles covered by solutions on the Pareto front [37]. More generally, the hypervolume Ξ of a Pareto front $\mathcal{P}$ is defined as:

$$\Xi(\mathcal{P}) = \lambda(x \in \mathcal{X} \mid \exists\, s \in \mathcal{P}, s \succ x \succ r)\,, \tag{2}$$

where λ denotes the Lebesgue measure and r is a pre-determined and fixed reference point [2, 28]. Limitations to this metric include its reliance on defining a suitable reference point r which can impact on its magnitude. However, it is regarded as a reliable MO performance metric due to its monotonic and scale-invariant properties [2].

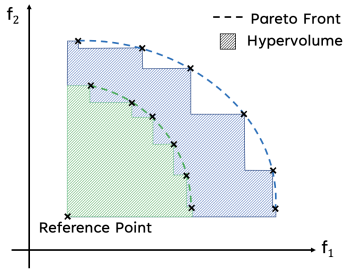


Figure 2: Two Pareto fronts and their hypervolumes with respect to a given reference point for a bi-objective problem. All of the points on the blue (outer) Pareto front strictly dominate the points on the green (inner) Pareto front and, accordingly, the blue Pareto front has a larger hypervolume.

3 RELATED WORKS

3.1 Multi-Objective Quality-Diversity

Multi-Objective Quality-Diversity (MOQD) combines the goals of QD and MO optimisation, with the aim of returning the Pareto front with maximum hypervolume in each cell of the descriptor space. To the best of our knowledge, Multi-Objective MAP-Elites (MOME) [28] is the only proposed method that has been specifically designed to tackle the MOQD problem. Its algorithmic approach is a natural extension to MAP-ELITES where, in each iteration of the algorithm, solutions are randomly selected from the population, mutated and potentially re-added to the population based on their fitness. However, unlike MAP-ELITES which stores at most one solution in each cell, MOME stores a Pareto front in each descriptor cell. If a new solution is on the Pareto front of the cell that corresponds to its descriptor, it is added to the grid. Otherwise, if the new solution dominates an existing solution on this front, it replaces it. Since the new solution could dominate several solutions in the cell's existing Pareto front, it is possible for it to replace more than one solution. MOME was shown to achieve a competitive MO performance compared to popular MO baselines, while also acquiring a diverse repertoire of behaviours.

Another method of note is Multi-Criteria Exploration (MCX) [15] which also follows a usual MAP-ELITES loop but uses a tournament ranking strategy to assess a solution's performance across multiple objectives. Specifically, MCX uses a "T-DominO score" which reflects how balanced solutions are across all objectives. If new a solution achieves a higher T-DominO score than the existing solution in the cell, it replaces it. While this approach is desirable in some contexts, it makes the assumption a priori that all objectives are of equal preference. Furthermore, MCX does not aim to obtain a Pareto front of solutions and so, while we note its relevance, we consider it to be closer related to traditional mono-objective QD methods.

3.2 Data-efficient Quality-Diversity

Recent works have proposed modifications to QD algorithms in order to improve their computational cost. For example, some methods use surrogate models to estimate the performance of solutions without directly evaluating them [13, 19, 23, 24]. As such, solutions that are predicted to be low-performing can be filtered out and unnecessary computation can be avoided. While this can reduce the number of required evaluations by more than an order of magnitude [23], it relies on having a accurate surrogate model which may be difficult to obtain.

Other works focus on creating a better exploration strategy. For example, Covariance Matrix Adaptation MAP-Elites (CMA-ME) [10] uses several emitter distributions to guide the exploration process toward the direction of, for example, highest diversity or performance. Or, more recently, Multi-Emitter MAP-Elites (ME-MAP-ELITES) includes a heterogeneous set of emitters and then uses a bandit algorithm to determine which combination of them to use to maximise the proportion of solutions that are added to the archive.

Some QD methods only consider the subset of problems that can be formalised as MDPs. In these tasks, solutions $x_i \in \mathcal{X}$ actually encode parameters for policies π_{x_i}, for an agent that interacts in a sequential environment at discrete time-steps t. At each time step, the agent observes its current state s_t and chooses an action a_t which leads to a subsequent state s_{t+1} and reward r_t. The policy of the agent determines which action is taken in each time-step, $\mathbb{P}(a_t|s_t)$, and the aim of the task is to find the policy that maximises the expected return of the agent over T time-steps:

$$J(x) = \sum_{j=0}^{T} \gamma^j r(s_j, a_j)\,. \tag{3}$$

Here, γ is known as a discount factor: a weighting coefficient that expresses the preferences between short-term and future rewards.

Policy Gradient Assisted MAP-Elites (PGA-ME) takes tasks from this MDP setting and employs techniques from the TD3 algorithm [12] in order to improve the optimisation process. Specifically, as solutions are evaluated in the main QD loop, transition tuples of (s_t, a_t, r_t, s_{t+1}) are collected and stored in a replay buffer $\mathcal{B}$. These samples are used to train a critic network Q_θ that approximates the action-value function,

$$Q(s_t, a_t) = \mathbb{E}\left[\sum_{j=t}^{T} \gamma^j r(s_j, a_j)\right], \tag{4}$$

which captures the value of taking action a_t from the state s_t. These critic estimates can consequently be used to form a policy gradient estimate:

$$\nabla_{x_i} J(x_i) = \mathbb{E}\left[\nabla_{x_i}\, \pi_{x_i}(s)\, \nabla_a Q_\theta(s,a)\,|_{\,a=\pi_{x_i}(s)}\right]. \tag{5}$$

Intuitively, the policy gradient provides the direction in which a solution x_i should be updated in order to maximise the reward of the agent. In each iteration of PGA-ME some solutions are mutated via gradient-based updates and, to maintain exploration, others are mutated via genetic mutations. This accelerates the optimisation process since gradient-information quickly highlights promising regions of the solution space, while genetic search enforces thorough exploration in these regions [8].

More recently, Pierrot and Macé et al. [27] built upon PGA-ME by using gradients to improve both the fitness and the diversity of solutions. Specifically, rather than rely on genetic variations for exploration, they introduced a Diversity Policy Gradient operator to improve exploration of states at each time step. Alternatively, Differentiable Quality-Diversity (DQD) [9] is a gradient-based approach that does not rely on an MDP formulation but assumes

differentiable objective and descriptor functions, and incorporates their gradients into a variation operator which greatly improves search efficiency. Subsequently, Tjanaka et al. adapted this algorithm for MDP tasks where the environment is not differentiable by instead using evolutionary strategies and techniques from TD3 to form gradient approximations [29].

3.3 Multi-objective Evolutionary Algorithms

Evolutionary algorithms are often applied to MO problems since their population-based approach lends itself naturally to finding Pareto fronts. In general, they work by initialising a population of solutions and then employing carefully designed selection, mutation and replacement operators in order to encourage exploration in promising regions of the solution space and hence drive performance of solutions toward the optimal Pareto front.

NSGA-II [6] is one such method that is driven by biased selection mechanisms. At each iteration, the population is first organised into sets of Pareto fronts: the first front is the Pareto front over all solutions; the second front is the Pareto front over all solutions except those that belong to the first front, and so on. Then solutions in each front are assigned a crowding distance score which reflects the density of the region of objective space that they occupy. Finally, an ordering over all solutions is calculated based on the previous two steps: solutions belonging to higher fronts and lying in less dense areas of the objective space are preferred. This ordering is used to bias the selection probability of solutions. As a result, NSGA-II has been demonstrated to achieve Pareto fronts that are both high performing and well spread.

SPEA2 [35] is an alternative, long-established MO algorithm that has a similar approach. Specifically, each individual is assigned a strength score that reflects the number of solutions that it dominates. Then, each solution is assigned a raw strength equal to the total strength scores of the solutions that dominate it. A lower raw strength is desirable as it means that it is dominated by fewer solutions. The selection probability of a solution is proportional to its raw strength and the corresponding density of the objective space that it lies in. Similar to NSGA-II, this encourages more exploration around individuals that are high-performing and that lie in less populated regions of the objective space.

A multitude of other MO evolutionary algorithms exist and we refer the reader to a comprehensive survey [34] for a summary of other approaches. However, to the best of our knowledge, MOME is the only MO method that actively maintains diversity over the descriptor space.

4 MOME-PGX

In this section, we present MOME-PGX, a new MOQD algorithm. Our proposed method uses policy gradient (PG) mutation operators in order to efficiently drive solutions towards high performance in each objective, while maintaining a divergent search through traditional genetic variation operators. MOME-PGX also improves on the uniform selection and replacement mechanisms from MOME by instead encouraging exploration around solutions that lie in less crowded regions of the objective space. Details for each of these contributions are found in the next sections and the pseudo-code for the MOME-PGX can be found in Algorithm 1.

Algorithm 1 MOME-PGX pseudo-code

Input:
- A MAP-ELITES archive $\mathcal{A}$, with k cells, and an empty Pareto front of size P in each cell
- the batch size B and total number of iterations N
- m objective functions $f_1, ..., f_m$ and a descriptor function c
- an empty replay buffer $\mathcal{B}$
- pairs of actor, critic and target networks for $f_1, ..., f_m$

// Initialisation
Generate initial candidate solutions x_k
Find descriptors $c(x_k)$ and fitnesses $f_1(x_k), ..., f_m(x_k)$ of x_k
Add initial solutions to $\mathcal{A}$ and update Pareto fronts
Add transitions collected by x_k to buffer $\mathcal{B}$
Train actor and critic networks on samples from $\mathcal{B}$

// Main loop
for iter = $1 \rightarrow N$ **do**
  // Sample solutions
  Sample B non-empty cells from $\mathcal{A}$ with uniform probability
  Calculate crowding distances in each sampled cell
  In each sampled cell's Pareto front, sample a solution x with probability proportional to the crowding distances of solutions

  // Generate offspring
  Apply GA variation to $B/2$ sampled solutions
  Apply PG variation to $(B/2) \div m$ sampled solutions for m objective functions

  // Evaluate offspring
  Find descriptors and fitness values for offspring
  Store transitions from offspring in buffer $\mathcal{B}$
  Train PG networks

  // Add to archive
  Find cell corresponding to descriptor of each offspring
  Add offspring to cells and recompute Pareto fronts
  if Pareto front length > P **then**
    Calculate crowding distances of solutions in cell
    Remove solutions with minimum crowding distance

  // Update iterations
  iter ← iter +1

return $\mathcal{A}$

4.1 Gradient-Assisted Mutations

In MOME-PGX, we extend the benefits of gradient-based optimisation that have been observed in PGA-ME to the multi-objective case by maintaining a critic network per objective function. In each iteration, half of the solutions are mutated via a Genetic Algorithm (GA) variation operator and the remaining solutions are divided into equal batch-sizes to be mutated with policy-gradients per objective function. So, for example, given a total batch-size of 256 and a bi-objective task, 128 of the solutions are mutated via the GA variation operator, 64 are mutated according to the gradient for the

first objective and the remaining 64 solutions are mutated according to the gradient of the second objective. MOME-PGX also employs the same network training procedures as the TD3 and PGA-ME algorithms, such as using an actor network π_ϕ to calculate a target value for critic training and introducing pairs of critic networks $Q_{\theta_1}, Q_{\theta_2}$ to minimise bootstrapping errors. We refer the interested reader to the original PGA-ME [26] and TD3 [12] papers for further implementation details.

4.2 Crowding-based selection and replacement

In MOME-PGX, we also introduce crowding-based selection and replacement mechanisms in order to induce a more targeted exploration strategy and improve the quality of the Pareto fronts obtained in each cell of the MAP-ELITES grid.

To select offspring solutions at each iteration, MOME first uniformly samples a cell from the MAP-ELITES grid and then uniformly samples a solution from the corresponding Pareto front. To illustrate the limitations of this, we first note that since the objective functions are continuous, it is possible to fill gaps in a Pareto front with solutions that represent marginally different trade-offs, making the existing Pareto front more dense rather than actively extending its coverage. Therefore, if solutions are not well spread on the Pareto front, uniform selection mechanisms can create an unintentional cycle: exploration is more likely to take place in more crowded areas of the Pareto front since there are more solutions there, which in turn may make that region even more dense, and so on. As this cycle continues, solutions that lie in sparser regions of the objective space will become increasingly less likely to be picked, and exploration in these regions will be neglected.

To address this, MOME-PGX uses a biased selection mechanism based on a solution's crowding distance, similar to NSGA-II [6]. As illustrated in Figure 3, the crowding distance is calculated as the average Manhattan distance between solutions and their neighbours in the objective space [6]. For most solutions, we take the average distance to their two neighbouring solutions. However, for solutions at the boundaries of the Pareto front, we use the single nearest-neighbour. When sampling solutions in each iteration of MOME-PGX: first, a cell is chosen with uniform probability and then a solution from the corresponding Pareto front is selected with probability proportional to its crowding distance. By consequence, solutions in sparser regions of the front are more likely to be selected.

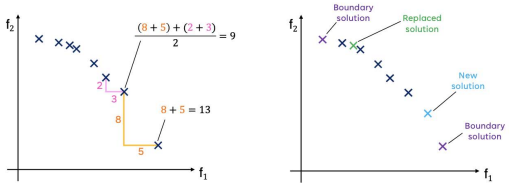


Figure 3: Left: crowding distances calculations for creating selection probabilities. Right: illustration of MOME-PGX replacement mechanism. When the Pareto Front is at maximum capacity, the new solution replaces the solution which has the smallest crowding distance.

Another point of difference between MOME and MOME-PGX is their Pareto front addition mechanisms. In both algorithms, the Pareto front in each cell is chosen to be a fixed maximum size (e.g., 50), which allows the algorithm to be parallelised and enables us to exploit the recent advances in massive hardware acceleration [22]. However, in MOME, if a new solution is added to a cell that is at full capacity according to this maximum, it randomly replaces an existing solution. This approach pays no attention to the value of different solutions in terms of the trade-off of objectives they provide, and may result in loss of objective space coverage if boundary solutions are replaced. In particular, we note that two solutions that lie close together in the objective space provide the end user with a more similar trade-off of objectives than two solutions that lie farther apart. Therefore, we should prefer to retain solutions that are dissimilar in the objective space, to provide the end-user with the greatest variety and smoothness of contiguous trade-offs.

MOME-PGX addresses this limitation by retaining solutions according to their crowding distances. As illustrated in Figure 3, if the Pareto front exceeds the maximum length, the solutions with the maximum crowding distances are retained. In order to prevent erosion of coverage of the front, boundary points are always assigned infinite crowding distances, as in SPEA2 [35].

5 EXPERIMENTS

5.1 Evaluation Tasks

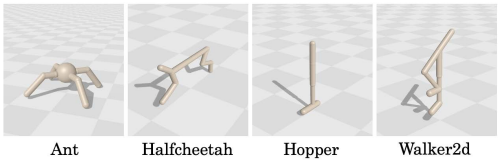


Figure 4: Brax tasks: agents aim to move forward as quickly as possible, while minimising their energy consumption.

We evaluate our approach on continuous control robotic tasks using the Brax suite [11]. We choose four robot morphologies from the suite: the Ant, HalfCheetah, Hopper and Walker2d, visualised in Figure 4. The solutions of the task correspond to the parameters of closed-loop neural-network controllers that control the torque commands $a_t \in [-1, 1]$ of the robot at each time step t, based on their current state s_t characterised by the robot's proprioceptive sensors. For each task, we used a two layer neural network each with 64 neurons, corresponding to solution space sizes of 6472, 5766, 5123 and 5702 for the Ant, HalfCheetah, Hopper and Walker2d respectively.

The aim of the task is to acquire a set of controllers that enable the robot to walk forward in different gaits, as fast as possible while minimising its energy consumption over $T = 1000$ timesteps. To characterise a solution's gait, its descriptor corresponds to the proportion of time that the robot spends on each of its legs [4, 26, 28]:

$$c_i(x) = \frac{1}{T}\sum_{t=1}^{T} c_{i,t} \quad \text{for } i = 1, ..., L, \tag{6}$$

where L is the number of legs of the robot and $c_{i,t} = 1$ when the leg is in contact with the floor at time step t and 0 otherwise. The energy consumption f_1 and forward motion f_2, are taken as:

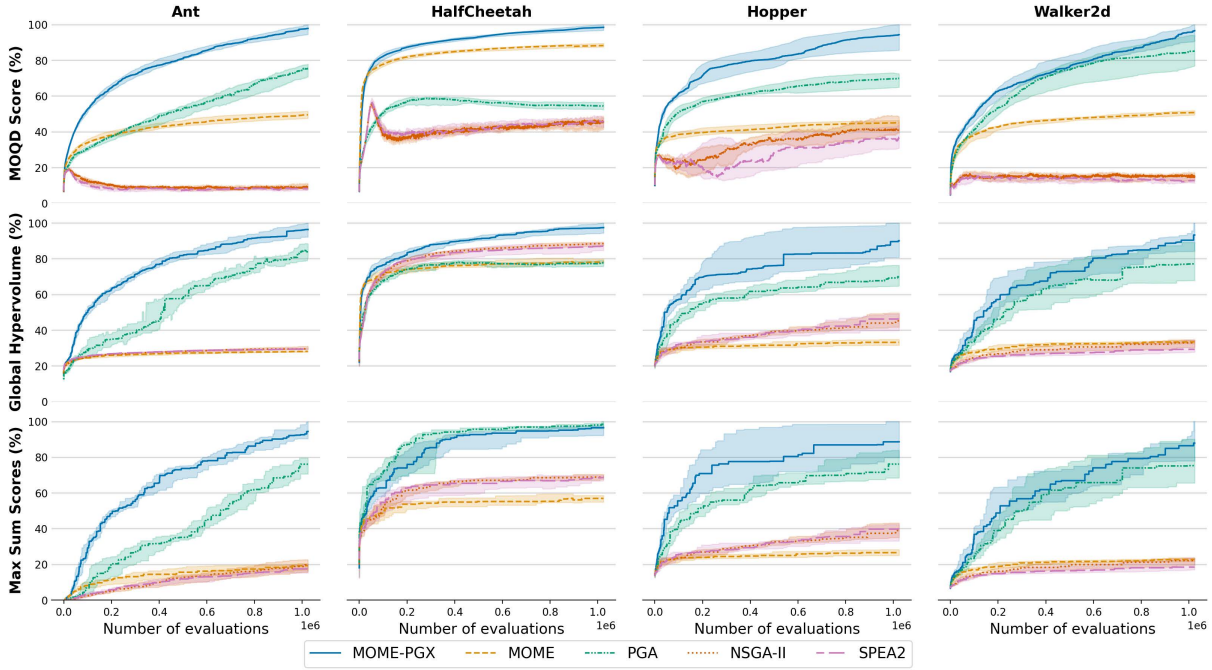


Figure 5: Performance of MOME-PGX and all other baseline algorithms for each of the tasks. The curves show the median performance across 15 replications and the shaded regions show the inter-quartile range.

$$\begin{cases} f_1 = -\sum_{t=1}^{T} ||a_t||_2 \,, \\ f_2 = \sum_{t=1}^{T} \frac{x_t - x_{t-1}}{\delta t} \,, \end{cases} \tag{7}$$

where $||\cdot||_2$ denotes the Euclidean norm, x_t denotes the robot's center of gravity position at time step t and δt denotes the length of one time-step. Following the implementation of the Brax authors, we also add a weighting coefficient to f_1, which is different for each task, to satisfy constraints on the robot's centre of gravity.

5.2 Evaluation Metrics

We evaluate the performance of algorithms based on four metrics:

1) **MOQD-SCORE**. Given the set of all solutions in the current archive $\mathcal{A}$ and a descriptor space $\mathcal{S}$ that has been tessellated into k cells $\mathcal{S}_i$, the MOQD-SCORE is the sum of hypervolumes of Pareto fronts from each cell in the MAP-ELITES grid [28]:

$$\sum_{i=1}^{k} \Xi(\mathcal{P}_i), \text{ where } \forall i, \mathcal{P}_i = \mathcal{P}(x \in \mathcal{A} | c(x) \in S_i) \,. \tag{8}$$

Analogous to the QD-SCORE in the mono-objective case [5], this metric aims to reflect both the coverage of the descriptor space and the performance of the solutions.

2) **GLOBAL-HYPERVOLUME**. We also evaluate performance via the global hypervolume. This is the hypervolume of the Pareto front formed from all solutions in the archive, equivalent to the standard Pareto front of canonical MO algorithms [28]. This metric allows us to compare the best set of possible trade-offs of the objectives that we can achieve regardless of a solution's descriptor.

3) **MAXIMUM SUM OF SCORES**. We also compare the performance of solutions using the maximum sum of scores of the objective functions. This allows direct comparison with traditional QD algorithms, which use the sum of rewards as a single objective.

4) **COVERAGE**. The COVERAGE is the proportion of cells of the archive that are non-empty, reflecting the diversity of solutions in the descriptor space. This metric is reported in Appendix A as all QD methods achieve a similar coverage.

5.3 Baselines

We compare MOME-PGX to the following four baselines: 1) MOME, 2) PGA-ME, 3) NSGA-II, 4) SPEA2. Since PGA-ME is not explicitly a multi-objective algorithm, we sum the two objectives in Equation 7 to form a single objective. Additionally, we note that using the same number of cells for PGA-ME would not be a fair comparison. For example, when we use a MAP-ELITES grid with k cells each with a maximum Pareto front length of P, this would lead to a maximum population size of $k \times P$ in MOME-PGX, but only a maximum population size of k in PGA-ME. Therefore, we use a grid tessellated into $k \times P$ cells for PGA-ME. By the same logic, we use a population size of $k \times P$ for NSGA-II and SPEA2.

On the other hand, having a different number of cells for PGA-ME and MOME-PGX also makes comparison of the MOQD-SCORE and COVERAGE unfair. Moreover, neither NSGA-II or SPEA2 even use a tessellated grid to maintain their populations. Therefore, for these baselines, we also maintain a passive archive with the same number of cells and maximum Pareto front length as MOME and MOME-PGX.

At each iteration we take all of the solutions from the PGA-ME, NSGA-II and SPEA2 archives and use them to fill the passive archives via the normal Pareto front addition rules. All metrics that we report for comparison are then calculated from these passive archives. Since the passive archives do not interact within the main algorithmic loop, the behaviour of the baselines remains unchanged.

5.4 Experiment designs

We run each experiment for the same total budget of 4000 iterations with a batch-size of 256, corresponding to a total of 1,024,000 evaluations. For MOME and MOME-PGX, we use a CVT tessellation [31] with 128 centroids and a maximum Pareto front length of 50 in each centroid. Accordingly, PGA-ME uses a CVT tessellation of $128 \times 50 = 6400$ centroids, while NSGA-II and SPEA2 have population sizes of 6400. For all experiments, we chose the GA variation operator to be Iso+LineDD operator [32] with $\sigma_1 = 0.005$ and $\sigma_2 = 0.05$. The reference points were chosen to be the empirically observed minima of the objective functions for of the each environments, as given in Appendix B, and were kept the same for each of the experiments. The hyperparameters used for neural networks in PGA-ME, MOME-PGX and the ablations were the same for each experiment and are provided in Appendix C. Each experiment was repeated for 15 different seeds.

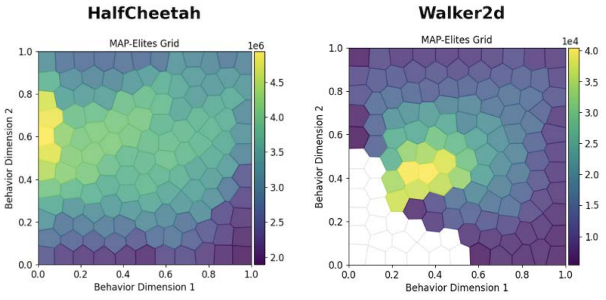


Figure 6: The final repertoires obtained from one optimisation run of MOME-PGX, colour-coded by hypervolume. Only tasks with two-dimensional descriptors are visualised.

5.5 Experimental Results

The results of our experiments are displayed in Figure 5. We also report the p-values obtained from a Wilcoxon signed-rank test [33] with a Holm-Bonferroni correction [17], under the null hypothesis that the MOME-PGX results come from the same distribution as the other baselines. The key conclusion is that MOME-PGX outperforms or matches the performance of all other baselines, across all metrics in each of the environments.

The results demonstrate that MOME-PGX achieves a higher MOQD-SCORE in all tasks ($p < 2.2 \times 10^{-2}$ in every experiment), indicating that it is better at finding high-performing Pareto fronts that span the descriptor space compared to other baselines. This performance is highlighted by the final archives visualised in Figure 6. Notably, Figure 5 shows that in the challenging Ant and Hopper tasks, the MOQD-SCORE of MOME-PGX is double that of MOME, NSGA-II and SPEA2. Across all tasks, MOME-PGX requires between 4.3 and 42 times fewer evaluations than MOME to reach the same MOQD-SCORE.

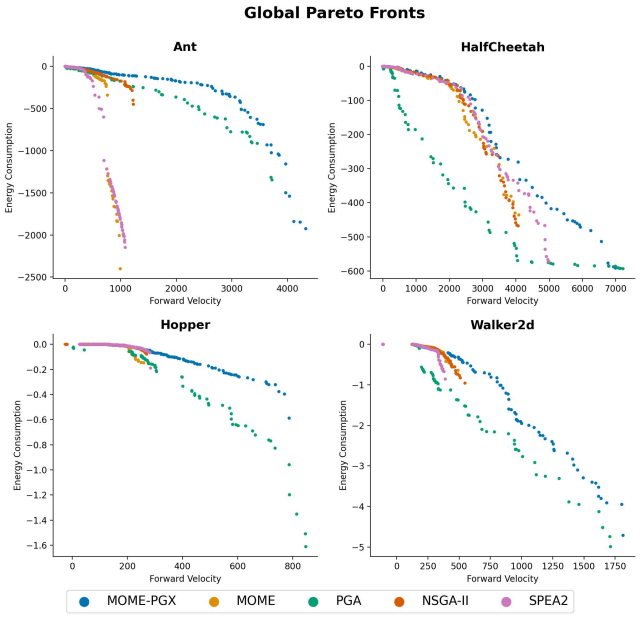


Figure 7: The global Pareto fronts obtained from one optimisation run for MOME-PGX and all other baseline algorithms.

MOME-PGX also outperforms all other baselines on the GLOBAL-HYPERVOLUME metric ($p < 4.3 \times 10^{-3}$ in every experiment) in all of the tasks, meaning that it is able to produce the best set of possible trade-offs across each of the objectives. Interestingly, PGA-ME also outperformed the other MO baselines, despite the fact that it does not actively seek a Pareto front of solutions. The strong performance of MOME-PGX and PGA-ME highlights the value of gradient-based approaches for tasks with large search spaces. Indeed, examining the global Pareto fronts attained by each of the algorithms in Figure 7, we see that MOME-PGX and PGA-ME were able to find solutions that were high-performing across both of the objectives but MOME, SPEA2 and NSGA-II often failed to find solutions with high forward velocity. We also note that, although PGA-ME does explicitly include energy consumption within its reward, MOME-PGX finds many solutions that achieve the same forward velocity as solutions from PGA-ME, but at a lower energy cost. This is most likely due to the difference in scales of each of the objectives, meaning that the mono-objective fitness used in PGA-ME is predominantly composed of the forward velocity reward and does not sufficiently prioritise energy consumption. However, this highlights one problem with using such a scalarised reward, which is that finding the coefficients with which to weight each objective can prove difficult.

Finally, we note that MOME-PGX also demonstrates excellent mono-objective performance via the MAXIMUM SUM OF SCORES metric. MOME-PGX outperforms all other MO baselines ($p < 4.8 \times 10^{-4}$ in every experiment). Moreover, MOME-PGX outperformed PGA-ME in the Ant task ($p = 5 \times 10^{-4}$) and there was no statistically significant difference between MOME-PGX and PGA-ME on the other three tasks. This result is particularly interesting because while PGA-ME explicitly optimises for the sum of rewards, MOME-PGX does not. This suggests that optimising over each of the objectives separately

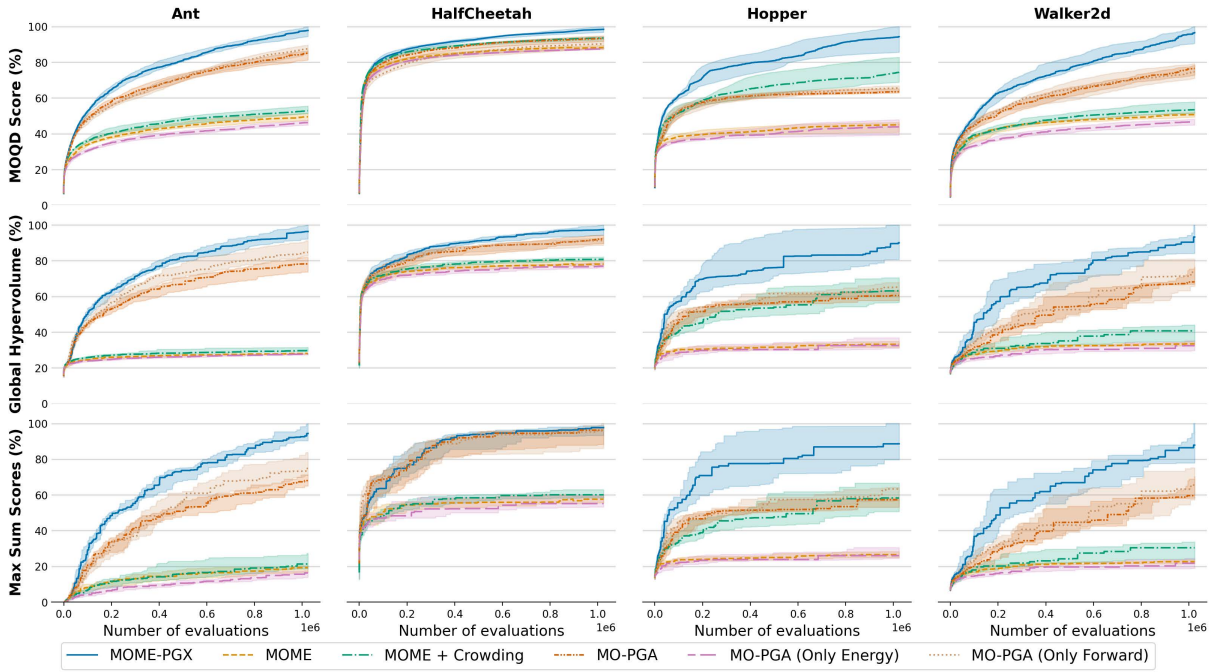


Figure 8: Performance of MOME-PGX and all ablation algorithms for each of the tasks. The curves show the median performance across 15 replications and the shaded regions show the inter-quartile range.

could help to provide better stepping stone solutions than those found in PGA-ME.

5.6 Ablation Study

We also compare MOME-PGX to a series of ablations. The first, MO-PGA, is the same as MOME but with PG variations for each objective. The second, MO-PGA (ONLY FORWARD), is the same as MO-PGA but only uses PG variations for the forward velocity. Similarly, MO-PGA (ONLY ENERGY), is also the same as MO-PGA but only uses PG variations for the energy consumption. Finally, MOME-CROWDING uses no policy gradients and simply extends MOME to include crowding-based sampling and addition.

The first key observation from the ablation study is that MOME-PGX outperforms all ablation algorithms in the Ant, Hopper and Walker2d tasks ($p < 1 \times 10^{-2}$ in all experiments). This highlights that neither the crowding-based mechanism nor the policy gradient variation operator is solely responsible for the strong performance of MOME-PGX but rather a combination of the two. Interestingly, the value each of the MOME-PGX components varies according to the task. For example, in the Hopper task, MOME-CROWDING has a higher MOQD-SCORE than MO-PGA ($p = 4.8 \times 10^{-4}$) but the reverse is true in the Ant and Walker2d tasks ($p = 6.1 \times 10^{-5}$ on both tasks).

The second key observation is that MO-PGA (ONLY FORWARD) outperforms MO-PGA (ONLY ENERGY) in the Ant, Hopper and Walker2d tasks ($p < 1 \times 10^{-2}$ in all experiments), demonstrating that policy gradient mutations across different objectives are not equally valuable. In fact, MO-PGA (ONLY FORWARD) led to a higher a GLOBAL-HYPERVOLUME and MAXIMUM SUM OF SCORES than MO-PGA in the Ant task, and there was no statistical difference between MO-PGA and MO-PGA (ONLY FORWARD) on the MOQD-SCORE in the Ant, Hopper and Walker2d tasks. This can be explained by the fact that the objectives are not equally straightforward to optimise. Specifically, we note that low-energy solutions can be found relatively straightforwardly through GA search mechanisms alone, as demonstrated by MOME, SPEA2 and NSGA-II performances. This lessens the need to use PG mutations to directly minimise the energy cost and weakens the benefit of these mutations for the overall performance of the algorithm. Despite this, since this result is likely to be highly task-specific, we still propose that MOME-PGX uses mutations from all objective functions. However, we note that an interesting future line of work would be to dynamically adjust the proportion of solutions mutated according to each objective [3].

6 CONCLUSION

In this work, we introduced MOME-PGX, a new MOQD algorithm that demonstrably succeeds at generating large collections of diverse and high-performing solutions, across multiple objective functions. Our experimental analysis shows that MOME-PGX improves the sample-efficiency of MOME in tasks that can be formalised as MDPs. It does this by using PG variation operators to accelerate the search process and crowding-based mechanisms to maintain exploration over the objective space. In future lines of work, we would like to explore MOQD approaches in tasks with many more objectives.

ACKNOWLEDGEMENTS

This work was supported by PhD scholarship funding for Hannah from InstaDeep.

REFERENCES

[1] Maxime Allard, Simón C Smith, Konstantinos Chatzilygeroudis, Bryan Lim, and Antoine Cully. 2022. Online Damage Recovery for Physical Robots with Hierarchical Quality-Diversity. *arXiv preprint arXiv:2210.09918* (2022).

[2] Yongtao Cao, Byran J Smucker, and Timothy J Robinson. 2015. On using the hypervolume indicator to compare Pareto fronts: Applications to multi-criteria optimal experimental design. *Journal of Statistical Planning and Inference* 160 (2015), 60–74.

[3] Antoine Cully. 2021. Multi-emitter MAP-elites: improving quality, diversity and data efficiency with heterogeneous sets of emitters. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 84–92.

[4] Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. 2015. Robots that can adapt like animals. *Nature* 521, 7553 (2015), 503–507.

[5] Antoine Cully and Yiannis Demiris. 2018. Quality and Diversity Optimization: A Unifying Modular Framework. *IEEE Transactions on Evolutionary Computation* 22, 2 (April 2018), 245–259. https://doi.org/10.1109/TEVC.2017.2704781

[6] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and Tanaka Meyarivan. 2000. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *International conference on parallel problem solving from nature*. Springer, 849–858.

[7] Paulo Victor R. Ferreira, Randy Paffenroth, Alexander M. Wyglinski, Timothy M. Hackett, Sven G. Bilen, Richard C. Reinhart, and Dale J. Mortensen. 2019. Reinforcement Learning for Satellite Communications: From LEO to Deep Space Operations. *IEEE Communications Magazine* 57, 5 (2019), 70–75. https://doi.org/10.1109/MCOM.2019.1800796

[8] Manon Flageat, Felix Chalumeau, and Antoine Cully. 2022. Empirical analysis of PGA-MAP-Elites for Neuroevolution in Uncertain Domains. *ACM Transactions on Evolutionary Learning* (2022).

[9] Matthew Fontaine and Stefanos Nikolaidis. 2021. Differentiable quality diversity. *Advances in Neural Information Processing Systems* 34 (2021), 10040–10052.

[10] Matthew C Fontaine, Julian Togelius, Stefanos Nikolaidis, and Amy K Hoover. 2020. Covariance matrix adaptation for the rapid illumination of behavior space. In *Proceedings of the 2020 genetic and evolutionary computation conference*. 94–102.

[11] C Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. 2021. Brax–A Differentiable Physics Engine for Large Scale Rigid Body Simulation. *arXiv preprint arXiv:2106.13281* (2021).

[12] Scott Fujimoto, Herke Hoof, and David Meger. 2018. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*. PMLR, 1587–1596.

[13] Adam Gaier, Alexander Asteroth, and Jean-Baptiste Mouret. 2018. Data-efficient design exploration through surrogate-assisted illumination. *Evolutionary computation* 26, 3 (2018), 381–410.

[14] Adam Gaier, Alexander Asteroth, and Jean-Baptiste Mouret. 2019. Are Quality Diversity Algorithms Better at Generating Stepping Stones than Objective-Based Search?. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (Prague, Czech Republic) *(GECCO '19)*. Association for Computing Machinery, New York, NY, USA, 115–116. https://doi.org/10.1145/3319619.3321897

[15] Adam Gaier, James Stoddart, Lorenzo Villaggi, and Peter J Bentley. 2022. Exploring Multiple Criteria with Quality-Diversity and the Tournament Dominance Objective. *arXiv preprint arXiv:2207.01439* (2022).

[16] Conor F Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M Zintgraf, Richard Dazeley, Fredrik Heintz, et al. 2022. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems* 36, 1 (2022), 26.

[17] Sture Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics* (1979), 65–70.

[18] Julien Horwood and Emmanuel Noutahi. 2020. Molecular design in synthetically accessible chemical space via deep reinforcement learning. *ACS omega* 5, 51 (2020), 32984–32994.

[19] Leon Keller, Daniel Tanneberg, Svenja Stark, and Jan Peters. 2020. Model-based quality-diversity search for efficient robot learning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 9675–9680.

[20] Ahmed Khalifa, Scott Lee, Andy Nealen, and Julian Togelius. 2018. Talakat: Bullet hell generation through constrained map-elites. In *Proceedings of The Genetic and Evolutionary Computation Conference*. 1047–1054.

[21] Joel Lehman and Kenneth O Stanley. 2011. Abandoning objectives: Evolution through the search for novelty alone. *Evolutionary computation* 19, 2 (2011), 189–223.

[22] Bryan Lim, Maxime Allard, Luca Grillotti, and Antoine Cully. 2022. Accelerated Quality-Diversity for Robotics through Massive Parallelism. *arXiv preprint arXiv:2202.01258* (2022).

[23] Bryan Lim, Luca Grillotti, Lorenzo Bernasconi, and Antoine Cully. 2022. Dynamics-aware quality-diversity for efficient learning of skill repertoires. In *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 5360–5366.

[24] Bryan Lim, Alexander Reichenbach, and Antoine Cully. 2022. Learning to walk autonomously via reset-free quality-diversity. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 86–94.

[25] Jean-Baptiste Mouret and Jeff Clune. 2015. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909* (2015).

[26] Olle Nilsson and Antoine Cully. 2021. Policy gradient assisted map-elites. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 866–875.

[27] Thomas Pierrot, Valentin Macé, Felix Chalumeau, Arthur Flajolet, Geoffrey Cideron, Karim Beguir, Antoine Cully, Olivier Sigaud, and Nicolas Perrin-Gilbert. 2022. Diversity Policy Gradient for Sample Efficient Quality-Diversity Optimization. In *ICLR Workshop on Agent Learning in Open-Endedness*.

[28] Thomas Pierrot, Guillaume Richard, Karim Beguir, and Antoine Cully. 2022. Multi-Objective Quality Diversity Optimization. *arXiv preprint arXiv:2202.03057* (2022).

[29] Bryon Tjanaka, Matthew C. Fontaine, Julian Togelius, and Stefanos Nikolaidis. 2022. Approximating Gradients for Differentiable Quality Diversity in Reinforcement Learning. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Boston, Massachusetts) *(GECCO '22)*. Association for Computing Machinery, New York, NY, USA, 1102–1111. https://doi.org/10.1145/3512290.3528705

[30] Neil Urquhart, Emma Hart, and William Hutcheson. 2019. Quantifying the effects of increasing user choice in map-elites applied to a workforce scheduling and routing problem. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*. Springer, 49–63.

[31] Vassilis Vassiliades, Konstantinos Chatzilygeroudis, and Jean-Baptiste Mouret. 2018. Using Centroidal Voronoi Tessellations to Scale Up the Multidimensional Archive of Phenotypic Elites Algorithm. *IEEE Transactions on Evolutionary Computation* 22, 4 (2018), 623–630. https://doi.org/10.1109/TEVC.2017.2735550

[32] Vassiiis Vassiliades and Jean-Baptiste Mouret. 2018. Discovering the elite hypervolume by leveraging interspecies correlation. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 149–156.

[33] Frank Wilcoxon. 1992. *Individual comparisons by ranking methods*. Springer.

[34] Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Qingfu Zhang. 2011. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation* 1, 1 (2011), 32–49. https://doi.org/10.1016/j.swevo.2011.03.001

[35] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. 2001. SPEA2: Improving the strength Pareto evolutionary algorithm. *TIK-report* 103 (2001).

[36] Eckart Zitzler and Lothar Thiele. 1998. An evolutionary algorithm for multiobjective optimization: The strength pareto approach. *TIK-report* 43 (1998).

[37] Eckart Zitzler and Lothar Thiele. 1998. Multiobjective optimization using evolutionary algorithms—a comparative case study. In *Parallel Problem Solving from Nature—PPSN V: 5th International Conference Amsterdam, The Netherlands September 27–30, 1998 Proceedings 5*. Springer, 292–301.