A Discretization-free Metric for Assessing Quality Diversity Algorithms

Paul Kent
Warwick University
paul.kent@warwick.ac.uk

Juergen Branke
Warwick Business School
Juergen.branke@wbs.ac.uk

Adam Gaier
Autodesk Research
adam.gaier@autodesk.com

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

ABSTRACT

While Quality-Diversity algorithms attempt to produce a set of high quality solutions that are diverse throughout descriptor space, in reality decision makers are often interested in solutions with specific descriptor values. In this paper we suggest that current methods of evaluating Quality Diversity algorithm performance do not properly account for a decision maker's preference in a continuous descriptor space and suggest three approaches that attempt to capture the real-world trade-off between a solution's objective performance and distance from a desired set of target descriptors.

In this paper we propose a randomised metric, a process of Monte-Carlo sampling of *n* target points in descriptor space and a small number of random weights that represent different tolerances for mis-specification in a solution's descriptor values. This sampling allows us to simulate the requirements of all possible combinations of target-tolerance pairs and, by taking sufficient samples, estimate average performance.

We go on to formulate three simple methods for comparing average performance of algorithms; Continuous Quality Diversity score (CQD) and Hypervolume of the objective/distance Pareto front. We show that these measures are simple to implement and robust measures of performance without introducing artificial discretisation of the descriptor space.

KEYWORDS

Quality-diversity, metrics, optimisation

ACM Reference Format:
Paul Kent, Juergen Branke, Adam Gaier, and Jean-Baptiste Mouret. 2018. A Discretization-free Metric for Assessing Quality Diversity Algorithms. In *Proceedings of Gecco 2022 - Quality Diversity Stream (Gecco 2022).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Quality Diversity (QD) algorithms [2, 6, 7] aim at finding a large set of solutions to an optimization problem that perform well *and* behave differently. Since there is currently no single metric that captures both dimensions of this goal, there is no straightforward way to compare different algorithms or the same algorithm at different iterations. Mouret and Clune [7] introduced four quality measures to analyze the performance of the MAP-Elites algorithm that all rely on a discretization of the behavior space into a grid (or "map"): coverage (how many cells of the grid are filled), precision (when a cell is filled, how close is the solution to the best known solution? that is, ignore the coverage), global reliability (same as precision, but use 0 when a cell is not filled, that is, combine performance and coverage), and global performance (the best solution found in the map). Pugh et al. [9] compared MAP-Elites, which relies on a grid, to Novelty Search with Local Competition, which does not use a grid. Nevertheless, they use the "MAP-Elites grid" to compare the performance algorithm, that is, they discretized the behavior space, looked at the best performance in each cell, then used the reliability introduced in [7] (the sum of the fitness of all the filled cells), which they renamed "QD-score". Recent papers mostly (e.g., [1, 3, 8]) focus on the QD-score and the coverage (a notable exception is [2], which uses a novelty-score to estimate the density in behavior space without a grid).

QD-score and coverage can be used to compare two grid-based algorithms, but comparing continuous algorithms or grids at different resolutions is not possible. For instance, a decision-maker might be more interested in matching a precise value for a behavior descriptor than having a slightly better performance. While there are only a few continuous algorithms for now [4, 5], we can expect more algorithms in the future as the field develops. In addition, high-dimensional behavior spaces may require using a Centroidal Voronoi Tesselation [10] instead of a standard grid, which makes the QD-score a bit harder to compute. Last, the QD-score in its current form implicitly assumes that the minimum fitness is 0 (otherwise, filling a cell would result in a reduction of the QD-score).

Our key idea is that decision makers (DMs) often have a set of specific preferences over descriptor space and would ideally like a solution that matches their requirements within an acceptable tolerance. In this work, we attempt to formalise the language of coverage in a continuous descriptor space and recommend three ways to measure the balance between quality and diversity from the perspective of an end user who wishes to use a Quality-Diversity algorithm. In fact, we define the goal of a QD algorithm to identify

Gecco 2022,
,

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/1122445.1122456

a mapping $S : G \longrightarrow X$ which returns, for every possible descriptor in feature space, the best (in terms of objective function) solution $x \in X$ with the specified descriptors.

Let us assume we have a well formulated QD problem and have implemented a suitable QD algorithm to obtain a solution. We consider an end user querying the solution produced by the QD algorithm for a single solution that matches a set of target feature values G. In the case of archive based QD algorithms such as MAP-Elites, we would return the elite point in the same niche as G but without loss of generality we introduce $S(.)$ as a way of generating a solution for any point in feature space. Now $S(G) = x^r_G$ is the point recommended by a QD solution for a point with descriptor values G.

The quality of any recommended point to the final user will depend on both the objective quality of the recommended point and the user's tolerance for a mis-match with the requested descriptors. This preference could be encapsulated by a linearly weighted expression.

$$\omega(x, G, \theta) = \frac{f(x)}{|f_{max} - f_{min}|} - \theta \frac{\delta(g(x), G)}{\delta_{max}} \tag{1}$$

for an objective function $f(x)$, descriptor function $g(x)$, distance measure δ and some weighting on the penalty θ. The values are normalised by the values $f_{max} - f_{min}$ for the objective and δ_{max} for the distance. f_{max} represents the maximum value of the objective function. For benchmark problems these max/min values can be provided but in practice, when comparing performance between multiple algorithms, f_{max} and f_{min} can simply be the largest and smallest observation from the unison of the observation sets. As the descriptor limits are generally known a-priori δ_{max} can be easily obtained. In the extreme cases, if the DM were to have infinite tolerance for deviating from the specified descriptor values ($\theta = 0$) we would be searching for the global optimum. Alternatively, as $\theta \longrightarrow 1$ the DM would value any point that does not have our specific descriptors as being quite poor. Theoretically θ need not be limited to 1 and $\theta > 1$ would represent an extremely strong preference for obtaining specific descriptor values but we do not explore that in the current work.

We abuse notation slightly to extend S with an extra parameter

$$S(G, \theta) = x^r_{G,\theta}$$

$x^r_{G,\theta}$ is now the solution recommended by a completed QD algorithm for some desired descriptor values G and a mismatch tolerance θ.

2 PROPOSAL 1. CONTINUOUS QUALTY DIVERSITY SCORE (CQD)

Assuming some distribution over the target descriptor G and the distance penalty weight θ, we can then compute the expected quality of a QD solution mapping S as

$$\int_G \int_\theta \omega(S(G, \theta), G, \theta). \tag{2}$$

As (2) does not have an analytical form, in order to measure an algorithm's performance we must generate targets by performing Monte Carlo sampling over the target descriptor and penalty weight space. We then query our QD solution with $S(G, \theta)$ which will select the best performing point from our archive or predict the best point in the case of model based QD approaches.

$$CQD = \sum_G \sum_\theta \omega(x^r, G, \theta)$$

For n Monte Carlo sampled points and a selection of weights, taking x^r to be $x^r_{G,\theta}$, the point recommended for descriptor G.

As the QD algorithms do not know which points in feature space will be required a-priori, algorithms cannot simply overfit to the requirements. In order for an algorithm to perform well on this benchmark, they must therefore achieve good coverage and identify good points throughout the search domain, aligning with the original goals of QD.

2.1 CQD pseudocode

```
Require: $S(G, \theta) \longrightarrow x^r$ : A function that selects the best point
    from the QD algorithm for some G and $\theta$
$G \longleftarrow$ n sampled target points in descriptor space
$\theta \longleftarrow$ m uniformly distributed weights $\in [0, 1]$
score = 0
for $G_i \in G$ do
    for $\theta_j \in \theta$ do
        $x^r = S(G_i, \theta_j)$
        score += $\omega(x^r, G_i, \theta_j)$
    end for
end for
```

3 VISUALISATION OF CQD SCORE

To visualise the metric, we consider a 1 dimensional objective function with a 1 dimensional linear descriptor function. When an algorithm suggests a point for a target descriptor value (the vertical line in Figure 1), we take the objective performance of the point and discount it by its distance, in this case euclidean distance in descriptor space, from the target. In Figure 1, we have a weight of 0, so the global optimum will be the best recommendation for all targets.

In Figure 2, we see how the value of θ affects the value of points depending on the distance away from the target value. The 'x' indicates the respective optimal solution.

4 VISUALISING ALGORITHM COMPARISONS

We can compare algorithms over different observation budgets, here is an example comparing MAP-Elites to Sobol sampling on a 6 dimensional Rosenbrock function. For both algorithms we use the MAP-Elites style niched archive and search the archive for the best point. The problem is defined over a 2-dimensional descriptor space formed by 2 descriptor functions D with x_i referring to the ith input dimension of point x:

$$D_1 = \frac{x_0 + x_1}{2}, \quad D_2 = (x_2 - 1)^2 \tag{3}$$

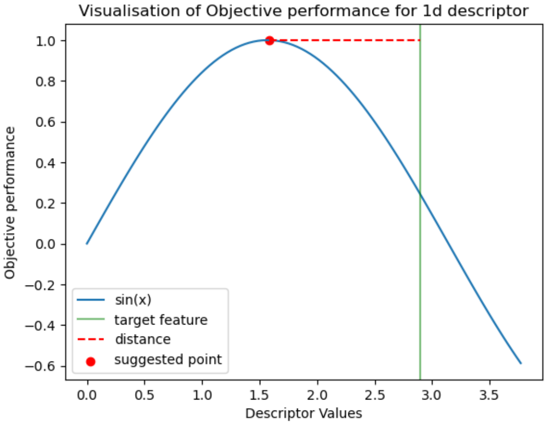

Figure 1: a visualisation of objective and distance from a target descriptor

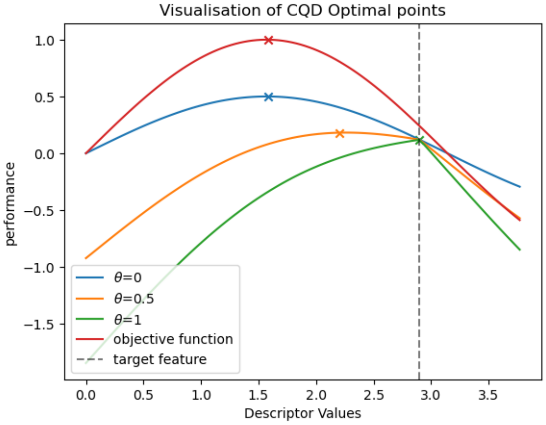

Figure 2: A visualisation of the changing optimal solution as the weighting changes

We show the performance over 10 uniform distance weights (mean and standard error of average performance over 100 target points, calculated over 10 runs) and compare at two search budgets.

Note that the standard error displayed in these results is the standard error around the mean performance of the algorithms on 100 random target points per run. There are 3 sources of variance for this measure, from randomly chosen targets (target variance), variance from the stochasticity of the algorithm itself (performance variance), and variance in the distribution of the objective measure (objective variance). The latter source of variability is a characteristic of the problem and should not enter the error bar calculation, so we generate targets independently for each run and record as a result the average over all the target points. In the following plots we average performance over 10 runs for Fig 3. and Fig 4. and 100 runs for Fig. 5.

Clearly MAP-Elites is leveraging its higher performing solution set due to its evolutionary strategy. It found much better solutions after 10,000 evaluations than Sobol sampling, even much better than what Sobol sampling achieved after 90,000 observations.

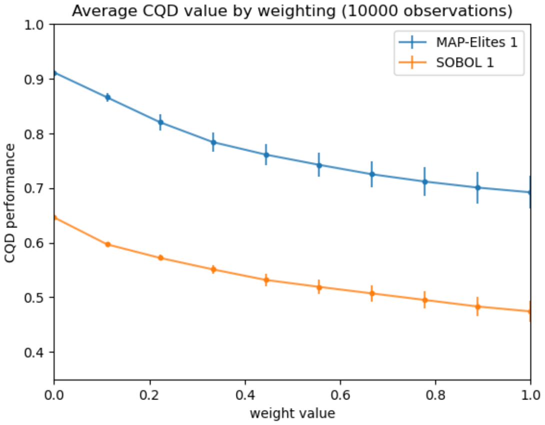

Figure 3: MAP-Elites performance vs Sobol sampling on the Rosenbrock6d problem with 2 black-box features and a budget of 10,000 observations, average of 10 runs

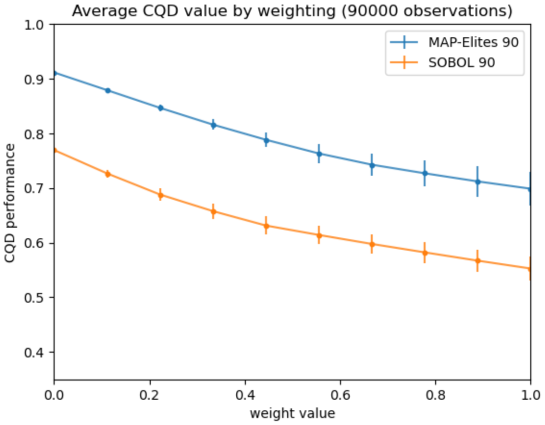

Figure 4: MAP-Elites performance vs Sobol sampling on the Rosenbrock6d problem with 2 black-box features and a budget of 90,000 observations, average of 10 runs

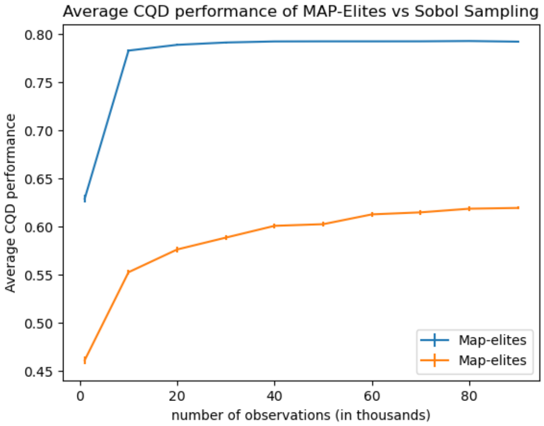

Figure 5: Average CQD performance for 100 runs of MAP-Elites and Sobol sampling on the Rosenbrock6d problem with 2 black-box features. Error bars are standard error around the mean performance for 100 independent Monte Carlo sampled targets per run

5 COMPARING CQD CONVERGENCE

When averaged over weight values, CQD provides a single performance value for the comparison of algorithms, see Table 1 for an example. This also allows to compare algorithm convergence, as depicted in Fig. 5.

Table 1: Performance of three algorithms on the Rosenbrock6d problem. With a budget of 90,000 evaluations, MAP-Elites obtains a much better CQD score than Sobol Sampling. On the other hand, BOP-Elites[5] obtains a similar CQD score already after 700 function evaluations

Algorithm	budget	Mean CQD	Std Error
BOP-Elites	700	0.7922515	0.0010045
MAP-Elites	90000	0.7923912	0.00057338
Sobol Sampling*	90000	0.60649299	0.01103941

6 CQD WITH A THRESHOLD

Above we proposed the CQD measure assuming a linear penalty on the distance from the target descriptor. However, we could also use a variant that assumes the DM has a tolerance δ and values each solution with a distance less than or equal to δ from the target descriptor at its objective performance, and as zero otherwise, i.e.,

$$CQD_{\beta} = \sum_{G} \sum_{\theta} \Omega(x^r, G, \theta) \tag{4}$$

where

$$\Omega = \begin{cases} f(x^r) & \delta(x^r, G) \leq \beta \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

This could be seen as a continuous version of the of the QD-score.

7 PROPOSAL 2: CQD HYPERVOLUME

In a real-world setting, a DM looking for a good solution near or at a particular target descriptor may be interested to learn about the trade-off between objective performance and distance from the target. If a significant increase in objective performance can be gained by accepting a point slightly further away from the target, a DM may be willing to accept such a compromise. In this case it is informative to present the DM with the Pareto front of the points in a solution archive.

In other words, rather than the DM providing a target descriptor and a weight factor θ and the algorithm returning a recommended solution $x^r_{G,\theta}$, the DM would only provide a target descriptor G, and the algorithm would return a Pareto front of solutions from the solution archive, maximising objective performance and minimising distance from the target.

In this setting, the proposed performance measure would make use of the average Hypervolume [11] of the Pareto fronts returned for each target, i.e.,

$$CQD_{HV} = \sum_{G} HV(S_{HV}(G)) \tag{6}$$

where S_{HV} returns a Pareto front of solutions given a target description, and HV returns the Hypervolume of this set. Note that the computation of the Hypervolume requires to set a reference point. We follow the same normalisation methods presented in Eqn(1) but specify δ'_{norm} so that we can maximise both axes for the hypervolume calculation. We can now use the common reference point $(0, 0)$.

$$f_{norm} = \frac{f(x)}{|f_{max} - f_{min}|} \quad , \quad \delta'_{norm} = 1 - \frac{\delta(g(x), G)}{\delta_{max}}$$

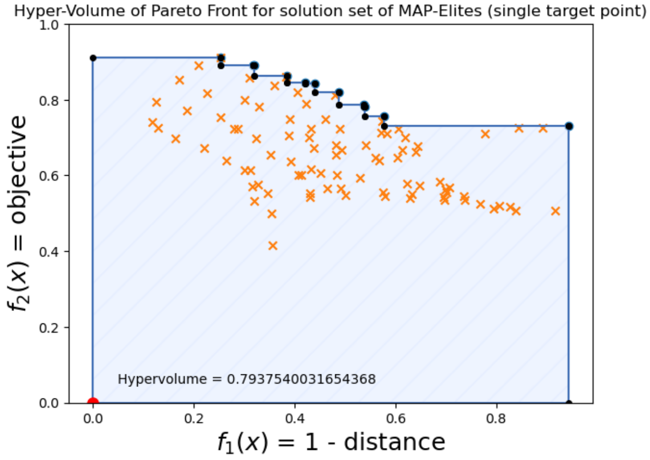

Figure 6: A visualisation showing the Pareto Front of a solution set from MAP-Elites on the Rosenbrock6d problem with a 10x10 archive and distances measured from a single target point. Orange crosses indicate dominated solutions, blue area shows the hypervolume

Table 2: Average HyperVolume for 100 random targets each evaluated on 100 converged solution archives for 3 algorithms, BOP-Elites uses it's evaluated solution archive and does not suggest points it has not seen

Algorithm	budget	Mean HV	Std Error
BOP-Elites	700	0.841845	2E-10
MAP-Elites	90000	0.842580	1E-10
Sobol Sampling*	90000	0.647926	0.010546

8 CONCLUSIONS

In this paper we have set out a frame work to measure the performance of QD algorithms over continuous descriptor space, suggested a simple Monte-Carlo sampling method and 2 new metrics for evaluating performance.

Both new metrics offer an opportunity to quantify the value of increased resolution in a QD algorithm as we can directly compare, for instance, the CQD performance of MAP-Elites with 100 niches to the same setup with 1000 niches. This has the potential to provide new insights into the performance and tuning of QD algorithms.

We briefly highlight the pros and cons of each method below:

CQD - Pros

- Measures performance on continuous descriptors without introducing synthetic discretization
- Represents real-world trade off of descriptor preference and performance
- Simple to implement
- Allows for comparison of algorithms with very different properties, with different budgets and different resolution etc.

CQD - Considerations

- Stochastic performance indicator due to Monte Carlo sampling, several runs are required to remove variance in the measure.
- Requires knowledge of f_{max} and f_{min} though these can be replaced with sampled values for comparisons between algorithms.
- Has to assume a distribution not only over the target descriptor space, but also over the weight penalty θ.

CQD$_{\beta}$ - Pros

- Measures performance on continuous descriptors without introducing synthetic discretization
- Represents real-world trade off of descriptor preference and performance
- Simple to implement
- Allows for comparison of algorithms with very different properties, with different budgets and different resolution etc.

CQD$_{\beta}$ - Considerations

- Stochastic performance indicator due to Monte Carlo sampling, several runs are required to remove variance in the measure.
- Requires a tolerance threshold β.
- Zero has to be a sensible value for a solution outside the requested tolerance or if the algorithm cannot return a solution at all.

CDQ$_{HV}$ - Pros

- Known metric that is easy to compute
- Allows for a visualisation that provides insights on the algorithm performance.
- Does not introduce synthetic discretisation

CDQ$_{HV}$ - Considerations

- Not suitable for algorithms that predict descriptor values, as the Hypervolume of estimated performance and distance values is not very meaningful.
- Requires reference point, though this can be (0,0) as long as $\theta < 1$

When considering which metric should be used we believe that CQD$_{HV}$ will appeal to the current QD community as it makes the trade-off of distance from target vs. objective performance explicit, without having to make an assumption about the distribution of DM utility functions.

However, CQD is a more robust measure as it is capable of comparing against future methods that predict descriptor values.

REFERENCES

[1] 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.

[2] Antoine Cully and Yiannis Demiris. 2017. Quality and diversity optimization: A unifying modular framework. *IEEE Transactions on Evolutionary Computation* 22, 2 (2017), 245–259.

[3] 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.

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

[5] Paul Kent and Juergen Branke. 2020. Bop-elites, a bayesian optimisation algorithm for quality-diversity search. *arXiv preprint arXiv:2005.04320* (2020).

[6] Jean-Baptiste Mouret. 2020. Evolving the behavior of machines: from micro to macroevolution. *iScience* 23, 11 (2020), 101731.

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

[8] Jean-Baptiste Mouret and Glenn Maguire. 2020. Quality diversity for multi-task optimization. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. 121–129.

[9] 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), 40.

[10] Vassilis Vassiliades, Konstantinos Chatzilygeroudis, and Jean-Baptiste Mouret. 2017. Using centroidal voronoi tessellations to scale up the multidimensional archive of phenotypic elites algorithm. *IEEE Transactions on Evolutionary Computation* 22, 4 (2017), 623–630.

[11] Eckart Zitzler and Lothar Thiele. 1998. Multiobjective optimization using evolutionary algorithms—a comparative case study. In *International conference on parallel problem solving from nature*. Springer, 292–301.