







An Analysis of Phenotypic Diversity in Multi-solution Optimization

Alexander Hagg^{1,2}(✉) , Mike Preuss² , Alexander Asteroth¹ ,
and Thomas Bäck² 

¹ Bonn-Rhein-Sieg University of Applied Sciences, Sankt Augustin, Germany
{alexander.hagg,alexander.asteroth}@h-brs.de

² Leiden Institute of Advanced Computer Science, Leiden University,
Leiden, The Netherlands
{m.preuss,t.h.w.baeck}@liacs.leidenuniv.nl

Abstract. More and more, optimization methods are used to find diverse solution sets. We compare solution diversity in multi-objective optimization, multimodal optimization, and quality diversity in a simple domain. We show that multiobjective optimization does not always produce much diversity, multimodal optimization produces higher fitness solutions, and quality diversity is not sensitive to genetic neutrality and creates the most diverse set of solutions. An autoencoder is used to discover phenotypic features automatically, producing an even more diverse solution set with quality diversity. Finally, we make recommendations about when to use which approach.

Keywords: Evolutionary computation · Multimodal optimization · Multi-objective optimization · Quality diversity · Autoencoder

1 Introduction

With the advent of 3D printing and generative design, a new goal in optimization is emerging. Having the option of choosing from different solutions that are good enough to fulfill a task can be more effective than being guided by single-solution algorithms. The optimization field should aim to understand how to solve a problem in different ways.

Three major paradigms for multi-solution optimization exist. The major difference between multi-objective optimization (MOO), multimodal optimization (MMO) and quality diversity (QD) is the context in which solution diversity is maintained. In MOO the goal is to find the Pareto set, which represents the trade-offs between multiple criteria. MMO finds solutions that cover the search space as well as possible. QD finds combinations of phenotypic features to maximize the variation in solutions' expressed shape or behavior - a new focus in evolutionary optimization [17].

This work received funding from the German Federal Ministry of Education and Research (BMBF) (grant agreement no. 03FH012PX5).

We analyze the diversity of solution sets in the three paradigms and introduce a new niching method that allows comparing genetic and phenotypic diversity (Sect. 2). State of the art diversity metrics (Sect. 3) are used in a new problem domain (Sect. 4) to evaluate all paradigms (Sect. 5) after which we make recommendations when to use which approach (Sect. 6).

2 Diversity in Optimization

The intuitive understanding of diversity assumes that there are more ways to “do” or to “be” something and involves the concepts of *dissimilarity* and *distance*. Evidence can be found in the large number of approaches and metrics, and the lack of agreement in when to use which one. This section gives an overview over three paradigms that have arisen in the last decades.

Finding solutions that are diverse with respect to objective space has been a paradigm since the 1970s. Multi-objective optimization tries to discover the Pareto set of trade-off solutions with respect to two or more objectives. The method has no control over the diversity of genomes or their expression other than the expectation that trade-offs require different solutions. The most successful method is the Non-dominated Sorting Genetic Algorithm (NSGA-II) [5].

The first ideas to use genetic diversity in optimization were not used to find different solutions, but to deal with premature convergence to local optima. The concept of *niching* was integrated into evolutionary optimization by introducing sharing and crowding [6, 8]. In the 1990s, multi-local or multimodal optimization came into focus. This paradigm has the explicit goal to find a diverse set of high quality locations in the search space, based on a single criterion. Various algorithms have been introduced, like basin hopping [26], topographical selection [23], nearest-better clustering [16] and restarted local search (RLS) [15].

The introduction of novelty search [11] led to studying the search for novel, non-optimal solutions. QD, reintroducing objectives [3, 12], finds a diverse set of high quality optimizers by performing niching in phenotypic space. In applications for developing artificial creatures and robot controller morphologies [3, 12], QD only allows solutions that belong the same phenotypic niche to compete. To this end it keeps track of an archive of niches. Solutions are added to the archive if their phenotype is novel enough or better than that of a similar solution.

This work does not aim at giving an exhaustive overview over all methods, for which we refer to some of the many survey papers [1, 4, 15, 21, 22, 27]. We consciously choose not to talk about methods that combine ideas from the three paradigms, but rather compare the three paradigms in their “purest” form.

2.1 Niching with Voronoi Tessellation

To remove variations in the search dynamics when comparing different algorithms, we introduce a niching variant using ideas from Novelty Search with Local Competition (NSLC) [12] and CVT-Elites [25]. Voronoi-Elites (VE)

accepts all new solutions until the maximum number of archive bins is surpassed (Algorithm 1). Then the pair of elites that are phenotypically closest to each other are compared, rejecting the worst-performing. An example archive is shown in Fig. 6 at step five). By locating selection pressure on the closest solutions, VE tries to equalize the distances between individuals. The generators of the Voronoi cells do not have to coincide with the centroids, like in CVT-Elites, and the boundaries of the archive are not fixed. VE can be used to compare archive spaces of different dimensionality. When the genetic parameters are used as archive dimensions, VE behaves like an MMO algorithm by performing niching in genetic space. When we use phenotypic descriptors, VE behaves like a QD algorithm.

Algorithm 1. Voronoi-Elites

```

Initialize population
for iter 1 to n do
  Select parents  $\mathcal{P}$  randomly
  Mutate  $\mathcal{P}$  using normal distribution to create offspring  $\mathcal{O}$ 
  Evaluate performance and descriptors of  $\mathcal{O}$ 
  Add  $\mathcal{O}$  to archive  $\mathcal{A}$ 
  while  $|\mathcal{A}| > maxSize$  do
    Find pair in  $\mathcal{A}$  with smallest distance
    Remove individual (in pair) with lowest fitness
  end while
end for

```

2.2 Related Work

A number of survey and analysis articles have appeared in the last decade. In [1] a taxonomy for diversity in optimization was introduced. [28] investigates how genetically diverse solution sets in MOO are found and shows that quality indicators used in MOO can be applied to MMO. [24] compares two algorithms from MMO to two QD algorithms in a robotics task, showing that clearing’s performance can be comparable to that of QD. Finally, [13] discusses 100 solution set quality indicators in MOO and [22] discusses diversity indicators for MOO.

3 Metrics

From the large number of diversity metrics available we only consider metrics that do not depend on precise domain knowledge, because no knowledge about actual local optima is available in real world applications. Three commonly used distance-based metrics are selected to evaluate the experiments in this work. The *Sum of Distances to Nearest Neighbor (SDNN)* measures the size of a solution

set as well as the dispersion between members of that set. *Solow-Polasky Diversity (SPD)* measures the effective number of species by using pairwise distances between the species in the set [20]. If the solutions are similar with respect to each other, SPD tends to 1, otherwise to N . The sensitive parameter θ , which determines how fast a population tends to N with increasing distance, needs to be parameterized for every domain. It is set to 1 for genetic distances and to 100 for phenotypic distances in this work. *Pure Diversity (PD)* is used in high-dimensional many-objective optimization [21, 27]. It does not have parameters, which makes it more robust, and depends on a dissimilarity measure ($L_{0,1}$ -norm).

Publications in the field of QD have focus on a small number of metrics. The total fitness is used directly or through the *QD-score* [18], which calculates the total fitness of all filled niches in a phenotypic archive. To achieve this, the solutions from a non-QD algorithm are projected into a fixed phenotypic niching space. This score is domain-dependent and does not allow comparing QD algorithms that have different archiving methods. A comparison between archives created from different features introduces a bias towards one of the archives. The *collection size* indicates the proportion of the niching space that is covered by the collection, but again can only be used on a reference archive [4]. Archive-dependent metrics do not generalize well and introduce biases. We therefore only use distance-based diversity metrics. The high dimensionality of phenotypic spaces is taken into account by using appropriate distance norms.

4 Polygon Domain

We construct a domain of free form deformed, eight-sided polygons. The genome (Fig. 1a) consists of 16 parameters controlling the polar coordinate deviation of the polygon control points. The first eight genes determine the deviation of the radius of the polygon’s control points, the second eight genes determine their angular deviation. Since the phenotypes can be expressed as binary bitmap images (Fig. 1b and c, resolution of 64×64 pixels) we use the Hamming distance in the diversity metrics to circumvent the problem of high dimensionality [7].

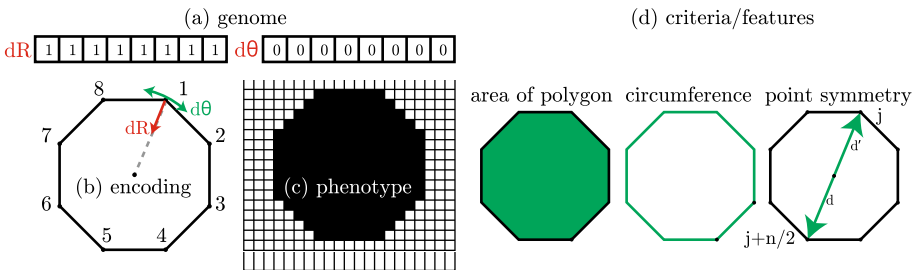


Fig. 1. Free form encoding of polygons. The genome (a) consists of 16 parameters that define axial and radial deformations (b). The phenotype is considered to be the pixel representation of the polygon (c). Shown is a 20×20 phenotype, although we use 64×64 pixels. Features/criteria are shown in (d).

Three aspects describing the polygons are defined that can be used either as criteria or as features (Fig. 1d): the area of the polygon A , its circumference l and point symmetry P through the center. The polygon is sampled at $n = 1000$ equidistant locations on the polygon circumference. The symmetry error E_s is calculated as the sum of distances of all $n/2$ opposing sampling locations. The symmetry metric is calculated as shown in Eq. 1.

$$f_P(x_i) = \frac{1}{1 + E_s(x_i)}, E_s(x_i) = \sum_{j=1}^{n/2} \|x_i^j, x_i^{j+n/2}\| \quad (1)$$

5 Evaluation

We ask which paradigm (objective space, search space or phenotype space) provides the highest phenotypic diversity of shapes. We compare VE, RLS and NSGA-II in multiple experiments. Throughout these experiments we fix the number of function evaluations and solutions and use five replicates per configuration. In NSGA-II the features are used as optimization criteria, maximizing A and minimizing l . The true Pareto set consists of circles with varying sizes. The number of generations is set to 1024 and mutation strength to 10% of the parameter range. The probability of crossover for NSGA-II is 90% and probability of mutation $\frac{1}{dof} = 0.0625\%$, with $dof = 16$ degrees of freedom. VE’s archive size is varied throughout the experiments. The number of children and population size is set to the same value. RLS uses as many restarts as the size of the VE archive, the step size is set to $\rho = 0.065$ (after a small parameter sweep) and L-BFGS-B is used as a local search method (within the bounds of the domain). The initial solution set for VE and NSGA-II is created with a Sobol sequence - the initial RLS solution is in the center of the parameter range but RLS’ space filling character assures a good search space coverage.

5.1 Genetic or Phenotypic Diversity

Biology has inspired evolutionary optimization to compose a solution of a genome, its encoding, and a phenotype, its expression. The phenotype often is a very high-dimensional object, for example a high-resolution 2D image, and can involve the interaction with the environment. Since the phenotypic space is usually too large, a low-dimensional representation, the genome, is used as search space. An expression function is constructed that turns a genome into its phenotype. Although the expression function should ideally be a bijective mapping, it often does not prevent multiple genomes to be mapped to the same phenotype. The phenomenon of such a surjective mapping is called genetic neutrality, which is not the same but akin to genetic neutrality in biology. In biology, a neutral mutation is understood to be a mutation that has no effect on the survivability of a life form. In evolutionary computation, genetic neutrality is referred to as genetic variants that have the same phenotype [9].

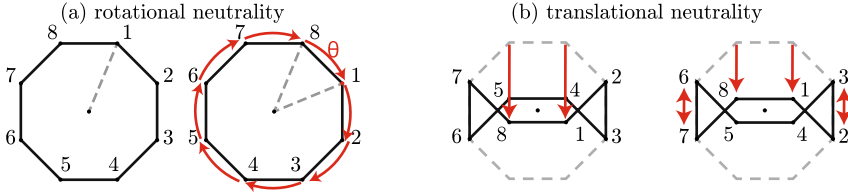


Fig. 2. Genetic neutrality. The same phenotype is expressed when rotating the control points by a $\frac{\pi}{8}$ angle (left) or by translating the control points as shown (right).

Fig. 2(a) shows an example polygon. If the angle θ equals 0° or 45° , phenotypically speaking, these shapes are the same. In this case, eight genomes all point to the same phenotype. Similarly, Fig. 2(b) shows how, through translations of the keypoints, a similar shape can appear based on different genomes. We postulate the first hypothesis: diversity maintenance in a neutral, surjective genetic space leads to lower phenotypic diversity than when using phenotypic niching.

While diversity is often thought about in terms of the distribution of points in the search space, we make a case to measure diversity in phenotypic space, which is independent of the encoding and does not suffer from the effects of genetic neutrality. Phenotypes may also include other factors that are not embodied within the solution’s shape itself, but emerge through interaction with the environment. This is taken advantage of in several publications on neuroevolution [11, 12]. In this work we only analyse the narrow interpretation of phenotypes, which does not include behavior.

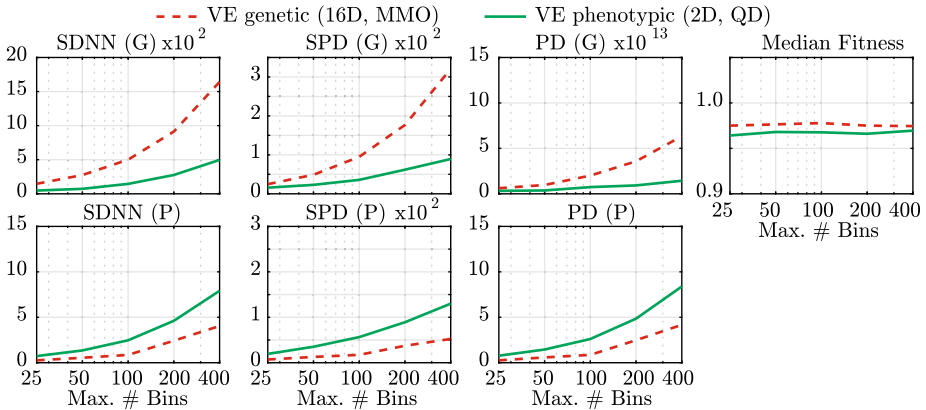


Fig. 3. Voronoi-Elites (VE) performed in 16D genetic and 2D phenotypic space. Top: genetic diversity (SDNN = Sum of Distances to Nearest Neighbor, SPD = Solow-Polasky Diversity, and PD = Pure Diversity) and median fitness, bottom: phenotypic diversity. The number of bins/solutions is increased (x-axis).

The Voronoi tessellation used in VE makes it easy to compare archives of different dimensionality by fixing the number of niches. We apply VE as an MMO algorithm, performing niching in 16-dimensional genetic space, and as a QD algorithm with a two-dimensional phenotypic space. The number of bins is increased to evaluate when differences between genetic and phenotypic VE appear (Fig. 3). At 25 solutions, the approaches produce about the same diversity, but genetic VE finds higher quality solutions. As the number of bins is increased, based on where niching is performed (genetic or phenotypic space), the diversity in that space becomes higher. Phenotypic VE beats genetic VE in terms of phenotypic diversity, which gives us evidence that the first hypothesis is valid. At the same time, the average fitness values of genetic VE are higher than that of phenotypic VE, although the difference gets lower towards 400 solutions.

Table 1. Parameter settings in order of increasing genetic neutrality.

Case	Axial min	Axial max	Radial min	Radial max	Neutrality
A	0	1	-0.05	0.05	-
B	0	1	-0.125	0.125	+
C	-0.25	1	-0.25	0.25	++
D	-0.5	1	-0.5	0.5	+++
E	-1	1	-1	1	++++

We compare phenotypic VE to NSGA-II and RLS. When we bound dr between 0 and 1 and $d\theta$ between $\pm 0.125 \times \pi$, we can minimize genetic neutrality. Neutrality is increased by expanding those bounds (Table 1). In contrast to VE, the phenotypic diversity of RLS' solutions is expected to decrease as genetic neutrality increases. Since there is no mechanism to distinguish between similar shapes with different genomes, there is an increasing probability that RLS finds similar solutions. We expect that the solution set produced by RLS due to its space filling character is more diverse than using NSGA-II.

Finally, it can make more sense to treat objectives as features and, instead of searching for the Pareto set, allowing all combinations of features and increasing the diversity of the solution set. We expect NSGA-II to easily find the Pareto set, which consists of circles of various scales, maximizing the area while minimizing the length of the circumference, while QD should find a variety of shapes that can be any combination of large and small A and l . We postulate the second hypothesis: allowing all criteria combinations, instead of using a Pareto approach, leads to higher diversity, while still approximating the Pareto set.

The number of solutions is set to 400. A result similar to Fig. 3 appears for the standard algorithms in Fig. 4. Phenotypic diversity is highest for VE, especially after the genetic neutrality threshold is crossed (at B). Diversity of NSGA-II is lowest, as is expected for this setup. Although diversity of VE is higher than that of RLS, the latter's solutions are all maximally symmetric (see

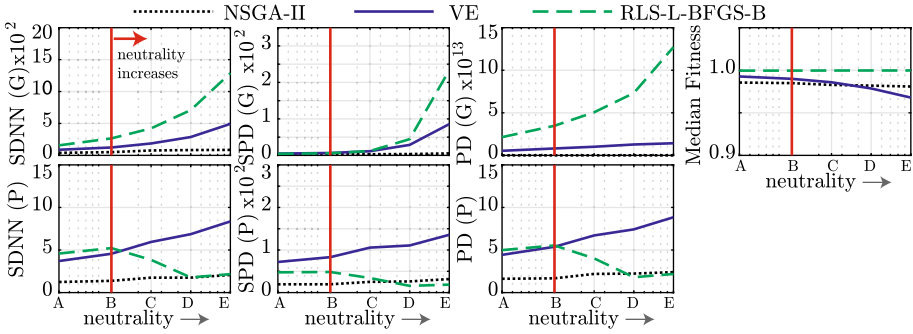


Fig. 4. Genetic (top) and phenotypic (bottom) diversity, and median fitness. Right of red marker: neutrality increases, using parameter bounds shown in Table 1.

fitness plots), making RLS much more appropriate when quality is more important than diversity. These results confirm the first part of the second hypothesis. The Pareto set can be calculated a priori, as we know that circular shapes maximize area while minimizing circumference. The members of the Pareto set adhere to the following genome: $(r_1, \dots, r_8, \theta_1, \dots, \theta_8)$, where r_i and θ_i have the same respective value. To create 100 shapes from the Pareto set we take ten equidistant values for r and θ and combine them.

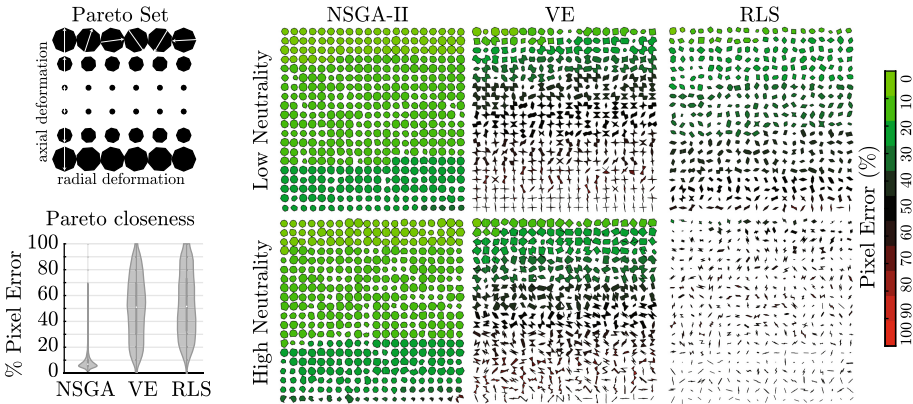


Fig. 5. The ground truth Pareto set is shown over the entire parameter range, with negative as well as positive values for the radial deformation. Bottom left: closeness to Pareto set, measured as pixel errors. The six figures on the right show example solution sets for low and high neutrality.

Part of the resulting Pareto set is shown in Fig. 5. The distance to the Pareto set is measured in phenotypic space, by measuring the smallest pixel error, the sum of pixel-wise differences, between a solution and the Pareto set. We see that the a number of solutions in VE and RLS are close to the Pareto set (Fig. 5 bottom left). Example results with low and high neutrality are shown on the right. Solutions that are close to the Pareto set are shown in the brightest green color. This is evidence for the second half of the second hypothesis. VE again seems to be more robust w.r.t. genetic neutrality, as it finds more solutions close to the Pareto set in high-neutrality domains (bottom row) than RLS.

5.2 Phenotypic Diversity Without Domain Knowledge

Up to this point we have used domain knowledge to construct a phenotypic niching space with VE. Intuitively, the area and circumference seem like good indicators for phenotypic differences. But this comparison between QD and MMO is not completely fair, as the latter does not get any domain information. On the other hand, the features used in QD might not be the most diversifying.

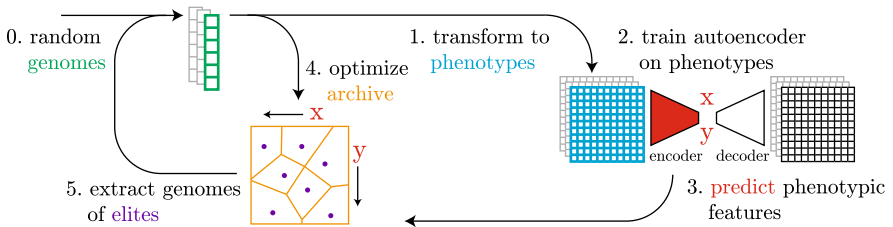


Fig. 6. AutoVE. Generating phenotypic features with an autoencoder. A random set of genomes is created (0), their phenotypes are calculated (1) and used as a training set for an autoencoder (2). The autoencoder can now be used to predict phenotypic features of new solutions (3), which is used to fill the archive (4), after which the elite solutions are extracted from the archive (5) and used to retrain the autoencoder.

We remove the domain knowledge from QD and construct a phenotypic niching space by using a well known dimensionality reduction technique to map the phenotypes to a latent space, as was done in [2, 14]. To our best knowledge, this data driven phenotypic niching approach, which we name Auto-Voronoi-Elites (AutoVE), has never been applied to shape optimization. An initial set of genomes, drawn from a quasi-random, space-filling Sobol sequence [19] and expressed into their phenotypes, is used to train a convolutional autoencoder (cAE) (see Fig. 6). The bottleneck in the cAE is a compressed, latent space that assigns every phenotype to a coordinate tuple. The encoder predicts these coordinates of new shapes in the latent space, which are used as phenotypic features. QD searches phenotypes that expand and improve the cAE archive. The cAE is retrained with the new samples. The cAE consists of two convolutional

layers in the encoder and four transposed convolutional layers in the decoder. We set the filter size to three pixels, the stride to two pixels, and the number of filters to eight. The cAE is trained using ADAM [10] with a learning rate of 0.001 and 350 training epochs and a mean square error loss function. Latent coordinates are normalized between 0 and 1. The number of generations (1024) is divided over two iterations for AutoVE and the number of latent dimensions is set to two (to compare with manual VE), five or ten.

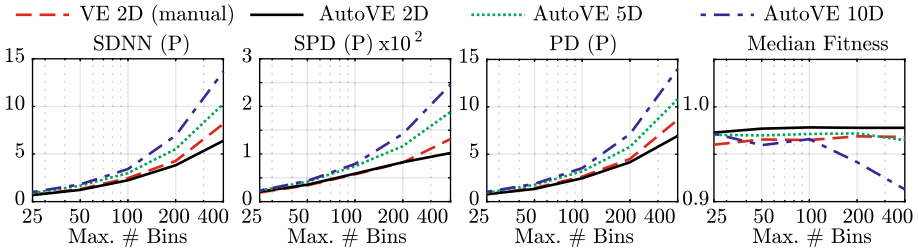


Fig. 7. Phenotypic diversity and fitness of manually crafted features (VE) compared to using an autoencoder (AutoVE) with 2, 5 or 10 latent dimensions.

Fig. 7 shows that the two-dimensional manual and autoencoded phenotypic space (AutoVE 2D) produce similar diversity, whereby the quality of solutions from AutoVE 2D is higher. The higher-dimensional latent spaces increase the solution set diversity at the cost of fitness. This is to be expected, as lower-fitness optima are protected in their own niches. Finally, the diversity of higher-dimensional AutoVE is around 50% higher than any of the other tested methods.

6 Conclusion

The main contributions of this work are as follows: a domain was introduced that allows comparing three different diversity paradigms; a case was made to measure diversity in phenotypic rather than genetic space; the hypothesis that QD is less sensitive to genetic neutrality than MMO was confirmed; the hypothesis that while the diversity of solutions sets of QD and RLS is higher than that of MOO, they also find some solutions close to the ground truth Pareto set, was confirmed; we showed that phenotypic diversity in QD is higher than MMO and MOO. Furthermore, we introduced VE, a simpler and self-expanding version of QD. We also used an autoencoder to discover phenotypic features in a shape optimization problem, showing that we do not need to manually predefine features to get a highly diverse solution set, allowing us to fairly compare QD to MOO and MMO. Using an autoencoder produces higher diversity than manually defined features, making AutoVE a strong choice for high diversity multi-solution optimization.

Since all paradigms have their strengths and weaknesses, we propose a guide for when to use which approach. MOO should be used when you want to optimize

all the criteria and want to know the trade-off solutions between those criteria. MMO is appropriate when you have a non-neutral bijective encoding, when you have a single criterion you want to optimize for or if you want to perform a gradient-based, Quasi-Newton or (direct) evolutionary local search to refine local optima. We cannot easily do this in QD due to the effect of neutrality that allows a search to “jump out of” a phenotypic niche. QD should be used if you have some criteria where you are less determined about whether to optimize for them, for example during the first phase of a design process. Some representatives from the Pareto set will still be discovered. When you are interested in the largest diversity of solutions and are more willing to get some solutions with lower fitness than when using MMO, QD is the better alternative. One of the biggest strengths of QD is the possibility to understand relationships between features or even to discover features automatically.

Some research effort should be focused on hybridization. MOO and QD are connected, as the boundary of valid solutions in the phenotypic archive is close to the Pareto front, yet there is room for improvement. Connecting MMO and QD means to use a local search method in QD, which needs to overcome the genetic neutrality problem. We cannot search close to a solution in genetic space and expect newly created solutions to be close in phenotypic space.

We gave insights about different variations of diversity and when and where to apply them, depending on whether one is most interested in trade-offs between criteria, increasing diversity while maximizing fitness, or maximizing diversity while finding high-performing solutions in a manually defined or automatically extracted phenotypic space. It is often easy to manually define two or three phenotypic descriptors, but human imagination can run out of options quickly. Automatic discovery of phenotypic features is a more attractive option for increasing solution diversity. Real world multi-solution optimization and understanding solution diversity are important steps towards increasing the efficacy and efficiency at which engineers solve problems.

References

1. Basto-Fernandes, V., Yevseyeva, I., Emmerich, M.: A survey of diversity-oriented optimization. In: *EVOLVE 2013-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computing*, pp. 101–109 (2013)
2. Cully, A.: Autonomous skill discovery with quality-diversity and unsupervised descriptors. In: *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 81–89 (2019)
3. Cully, A., Clune, J., Tarapore, D., Mouret, J.B.: Robots that can adapt like animals. *Nature* **521**(7553), 503–507 (2015)
4. Cully, A., Demiris, Y.: Quality and diversity optimization: a unifying modular framework. *IEEE Trans. Evol. Comput.* **22**(2), 245–259 (2017)
5. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)
6. DeJong, K.: *Analysis of the Behavior of a Class of Genetic Adaptive Systems*. Department of Computer and Communication Sciences, University of Michigan, Ann Arbor (1975)

7. Hamming, R.W.: Error detecting and error correcting codes. *Bell Syst. Tech. J.* **29**(2), 147–160 (1950)
8. Holland, J.H.: *Adaptation in Natural and Artificial Systems*. MIT Press, Cambridge (1975)
9. Hu, T., Payne, J.L., Banzhaf, W., Moore, J.H.: Robustness, evolvability, and accessibility in linear genetic programming. In: Silva, S., Foster, J.A., Nicolau, M., Machado, P., Giacobini, M. (eds.) *Genetic Programming. EuroGP 2011. Lecture Notes in Computer Science*, vol. 6621, pp. 13–24. Springer, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20407-4_2
10. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
11. Lehman, J., Stanley, K.O.: Abandoning objectives: evolution through the search for novelty alone. *Evol. Comput.* **19**(2), 189–223 (2011)
12. Lehman, J., Stanley, K.O.: Evolving a diversity of virtual creatures through novelty search and local competition. In: *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, pp. 211–218 (2011)
13. Li, M., Yao, X.: Quality evaluation of solution sets in multiobjective optimisation: a survey. *ACM Comput. Surv. (CSUR)* **52**(2), 1–38 (2019)
14. Meyerson, E., Lehman, J., Miikkulainen, R.: Learning behavior characterizations for novelty search. In: *GECCO 2016 - Proceedings of the 2016 Genetic and Evolutionary Computation Conference*, pp. 149–156 (2016)
15. Pošík, P., Huyer, W.: Restarted local search algorithms for continuous black box optimization. *Evol. Comput.* **20**(4), 575–607 (2012)
16. Preuss, M.: Improved topological niching for real-valued global optimization. In: Di Chio, C., et al. (eds.) *Applications of Evolutionary Computation. EvoApplications 2012. Lecture Notes in Computer Science*, vol. 7248, pp. 386–395. Springer, Berlin, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29178-4_39
17. Pugh, J.K., Soros, L.B., Stanley, K.O.: Quality diversity: a new frontier for evolutionary computation. *Front. Robot. AI* **3**, 40 (2016)
18. Pugh, J.K., Soros, L.B., Szerlip, P.A., Stanley, K.O.: Confronting the challenge of quality diversity. In: *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pp. 967–974 (2015)
19. Sobol', I.M.: On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki* **7**(4), 784–802 (1967)
20. Solow, A.R., Polasky, S.: Measuring biological diversity. *Environ. Ecol. Stat.* **1**(2), 95–103 (1994)
21. Tian, Y., Cheng, R., Zhang, X., Jin, Y.: Platemo: a matlab platform for evolutionary multi-objective optimization. *IEEE Comput. Intell. Mag.* **12**(4), 73–87 (2017)
22. Tian, Y., Cheng, R., Zhang, X., Li, M., Jin, Y.: Diversity assessment of multi-objective evolutionary algorithms: performance metric and benchmark problems [research frontier]. *IEEE Comput. Intell. Mag.* **14**(3), 61–74 (2019)
23. Törn, A., Viitanen, S.: *Topographical global optimization*. In: Floudas, C.A., Pardalos, P.M. (eds.) *Recent Advances in Global Optimization*, pp. 384–398. Princeton University Press, New Jersey (1992)
24. Vassiliades, V., Chatzilygeroudis, K., Mouret, J.B.: Comparing multimodal optimization and illumination. In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 97–98 (2017)
25. Vassiliades, V., Chatzilygeroudis, K., Mouret, J.B.: Using centroidal voronoi tessellations to scale up the multidimensional archive of phenotypic elites algorithm. *IEEE Trans. Evol. Comput.* **22**(4), 623–630 (2017)

26. Wales, D.J., Doye, J.P.: Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms. *J. Phys. Chem. A* **101**(28), 5111–5116 (1997)
27. Wang, H., Jin, Y., Yao, X.: Diversity assessment in many-objective optimization. *IEEE Trans. Cybern.* **47**(6), 1510–1522 (2016)
28. Wessing, S., Preuss, M.: On multiobjective selection for multimodal optimization. *Comput. Optim. Appl.* **63**(3), 875–902 (2015). <https://doi.org/10.1007/s10589-015-9785-x>