An Analysis of Phenotypic Diversity in Multi-Solution Optimization

Alexander Hagg, Mike Preuss, Alexander Asteroth and
Thomas Bäck

An Analysis of Phenotypic Diversity in Multi-Solution Optimization

Alexander Hagg[1,2]([✉])[0000−0002−8668−1796], Mike Preuss[2][0000−0003−4681−1346], Alexander Asteroth[1][0000−0003−1133−9424], and Thomas Bäck[2][0000−0001−6768−1478]

[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. In optimization methods that return diverse solution sets, three interpretations of diversity can be distinguished: multi-objective optimization which searches diversity in objective space, multimodal optimization which tries spreading out the solutions in genetic space, and quality diversity which performs diversity maintenance in phenotypic space. We introduce niching methods that provide more flexibility to the analysis of diversity and a simple domain to compare and provide insights about the paradigms. We show that multiobjective optimization does not always produce much diversity, quality diversity is not sensitive to genetic neutrality and creates the most diverse set of solutions, and multimodal optimization produces higher fitness solutions. An autoencoder is used to discover phenotypic features automatically, producing an even more diverse solution set. Finally, we make recommendations about when to use which approach.

Keywords: evolutionary computation · diversity · multimodal optimization · multi-objective optimization · quality diversity · autoencoder · feature discovery.

1 Introduction

The number of solutions to engineering problems that are of high quality can be very large - it often can be intractable to discover them all. But in real world optimization, having the option of choosing from different solutions can be more effective than being guided by single-solution algorithms. Fields like robust design optimization, designing with multiple criteria, real world design and interactive optimization benefit from *multi-solution optimization*, understanding how to solve a problem in different ways and considering different trade-offs.

Through different views on solution diversity, computer scientists have tried to identify a representative solution set. Three major paradigms have been introduced in the last decades. 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 [19].

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

2   Diversity in Optimization

Diversity is a concept that is hard to define. The intuitive understanding of diversity assumes that there are more ways to "do" or to "be" something and thus 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 and the metrics that are used to measure diversity.

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

Genetic Space. Arguably, the first ideas to use genetic diversity in optimization were not used to find different solutions, but to deal with the problem of premature convergence to local optima. The concept of *niches* was integrated into evolutionary optimization by introducing sharing and crowding [9, 6]: fitness values are penalized due to proximity of other solutions, or child solutions only replace the most similar parent in the population. By being able to discover different, locally optimal solutions makes it easier to reach the global optimum.

In the 1990s, multi-solution, multi-local or multimodal optimization was put 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. The first metric that was intended to be used on benchmark problems was peak ratio [26]. Various algorithms have been introduced, like basin hopping [29], topographical selection [25], nearest-better clustering [18] and restarted local search (RLS) [17].

Phenotypic Space. The introduction of novelty search [12] led to the effort of studying the search for novel, non-optimal solutions. In QD, objectives were reintroduced with novelty search with local competition (NSLC) and MAP-Elites [3,

13]. QD finds a diverse set of high quality optimizers by performing niching in phenotypic space. In applications for developing new artificial creatures and robot controller morphologies [3, 13], QD only allows solutions that belong to a known phenotypic niche to compete locally. To this end it keeps track of a growing archive of niches. Solutions are only added to the archive if their phenotype is novel enough or their quality is higher 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 [4, 1, 15, 17, 19, 23, 24, 30, 31]. We also consciously choose not to talk about specific methods that combine ideas from the three paradigms, but rather compare the three paradigms in their "purest" fashion.

2.1   Niching with Voronoi Tessellation

To remove variations in the search dynamics when comparing different algorithms, we introduce a niching variant which uses ideas from NSLC [13] and CVT-Elites [28]. Voronoi-Elites (VE) accepts all new solutions until the maximum number of bins is surpassed (Alg. 1, an example VE archive is shown in Fig. 6 at step five). If this happens, the pair of elites that are closest to each other in terms of their phenotypic niches are compared, rejecting the worst-performing one. By putting 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 which we will summarize here. In [1] a taxonomy for diversity in optimization was introduced. [32] investigates how genetically diverse solution sets in MOO are found and shows that quality indicators used in MOO can be applied to MMO. [27] 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, [14] discusses 100 solution set quality indicators in MOO and [24] discusses diversity indicators for MOO.

3   Metrics

There is a large number of diversity metrics used in multi-solution optimization. To preserve generality w.r.t. real world applications, we only consider metrics that do not depend on precise domain knowledge, because in real world applications, no knowledge about actual local optima or basins is available. Diversity metrics are either based upon maximizing the volume in the search space (*spread*), the uniformity of their distribution (*discrepancy*), a combination of the two (*coverage*) and *maximin* diversity, which tries to maximize the minimum distance between points [31]. Three commonly used distance-based metrics are selected to evaluate the experiments in this work.

Sum of Distances to Nearest Neighbor (SDNN). The sum of the nearest neighbor distances, or N-dispersion-min-sum measure [16], measures the size of a solution set as well as the dispersion between members of that set.

Solow-Polasky Diversity (SPD). This metric measures the effective number of species by using pairwise distances between the species in the set [22]. Just like in clustering algorithms, SPD tends towards N if the distance between all species is very large. If the species 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). This metric is used in high-dimensional many-objective optimization [23, 30]. It does not use parameters, which makes it more robust, and depends on a dissimilarity measure, which is fixed to the $L_{0.1}$-norm. The metric is used for MOO but we apply it to genetic and phenotypic spaces.

Diversity of Phenotypes. Publications in the field of QD have mainly focused on a small number of metrics. The total fitness is used directly or through the *QD-score* [20], which calculates the total fitness of all filled niches in a (fixed-size) phenotypic archive. To achieve this, the solutions from a non-QD algorithm are projected into a fixed phenotypic niching space. The QD-score therefore is
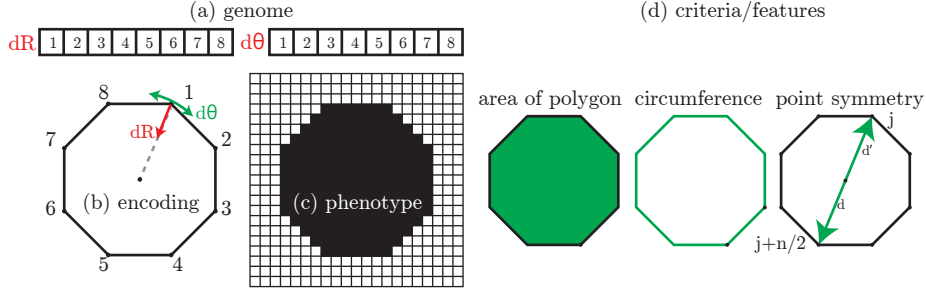
Fig. 1. Free form encoding of polygons. The genome (a) consists of 16 parameters that define axial and radial deformations (b). The phenotype is considered to be the pixel representation of the polygon (c). Shown is a 20x20 phenotype, although we use 64x64 pixels. Features/criteria are shown in (d).

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 can introduce biases towards certain niching spaces. We therefore only use distance-based diversity metrics. The high dimensionality of phenotypic spaces has to be taken into account by using appropriate distance norms.

4   Polygon Domain

We construct a domain that consists 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 (Fig. 1c) can be expressed (Fig. 1b) as binary bitmap images (with a resolution of 64x64 pixels) we can use the Hamming [8] distance in the diversity metrics to circumvent the problem of high dimensionality.

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.

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

(1)

5   Evaluation

We select those algorithms that provide simple interpretations of diversity in objective space, search space or phenotype space. We ask which one of these interpretations of diversity produces a larger diversity of shapes.

5.1   Experimental Setup

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 runs per configuration. In NSGA-II the features are used as optimization criteria, maximizing A and minimizing l. The Pareto set consists of circles with varying sizes. The number of generations is set to 1024, mutation strength is set to 10% of parameter range, the initial solution set for VE and NSGA-II is created with a Sobol sequence - the initial RLS solution is always in the center of the parameter range but the space filling character will assure a good coverage of the search space. The archive size is varied throughout the experiments, whereby we set the number of children and population size to the same value. The probability of crossover for NSGA-II is 90% and probability of mutation ($\frac{1}{dof} = 0.0625\%$, with $dof = 16$ degrees of freedom). 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 as a local search method we use L-BFGS-B within the bounds of the domain.

5.2   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, 3D mesh, or structure of a large neural network, and can also entail the interaction of the object with its environment. Since the phenotypic space is usually too large to search directly (for example, 3D printed shapes can have millions of filled volume units), a low-dimensional representation of a solution, the genome, is used as search space. A predetermined function then needs to be constructed that expresses 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. However, in evolutionary computation, genetic neutrality is referred to as genetic variants that have the same phenotype [10].

  Figure 2(a) shows a polygon that is defined by its 16 genes. If the angle θ equals 45˚, the same shape appears, but with the keypoints at the location of the next neighbor. Phenotypically speaking, these shapes are the same. In this case, eight different genomes all point to the same phenotype. Similarly, Figure 2(b)
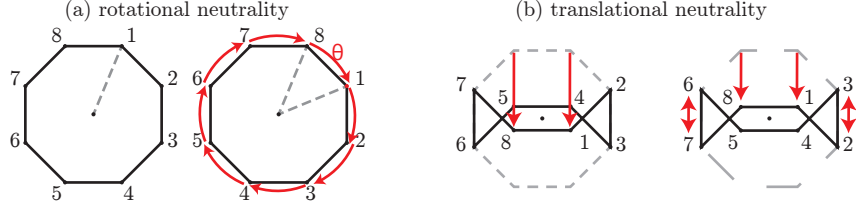
Fig. 2. Genetic neutrality. The same phenotype is expressed when rotating the control points by a $\frac{\pi}{8}$ angle (left) or by translating the control points as shown (right).

shows how, through various translations of the keypoints, a similar shape can appear based on different genomes. We therefore postulate:

Hypothesis I: Diversity maintenance in a neutral, surjective genetic space leads to a lower phenotypic diversity than when using a phenotypic niching method.

While researchers often think of diversity 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 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, for example [13, 12]. In this work, however, we only analyse the narrow interpretation of phenotypes, which does not include behavior.

The Voronoi tessellation used in VE makes it easy to compare archives that have 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.

Result. The number of bins or solutions 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 Hypothesis I 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.

Comparison to Standard Algorithms. We now compare VE to NSGA-II and RLS. When we bound dr between 0 and 1 and $d\theta$ between $+/-0.125 \times \pi$, we can minimize genetic neutrality. Neutrality is increased by expanding those bounds, which are shown in Table 1. In contrast to (phenotypic) VE, the phenotypic diversity of RLS' solutions is expected to decrease as genetic neutrality
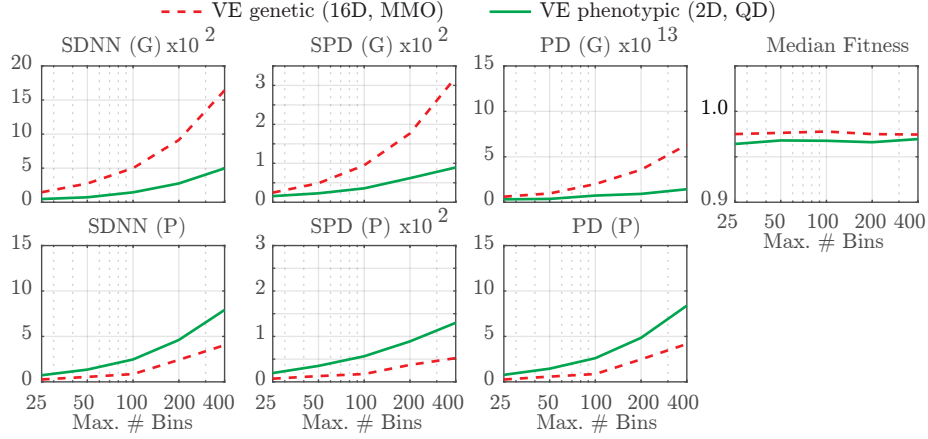
Fig. 3. Voronoi-Elites (VE) performed in 16D genetic and 2D phenotypic space. Top: genetic diversity (SDNN = Sum of Distances to Nearest Neighbor, SPD = Solow-Polasky Diversity, and PD = Pure Diversity) and median fitness, bottom: phenotypic diversity. The number of bins/solutions is increased (x-axis).

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	++++

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 still is more diverse than using NSGA-II due to the space filling character of RLS.

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:

Hypothesis II: Allowing all criteria combinations, instead of using a Pareto approach, leads to higher diversity, while still approximating the Pareto set.

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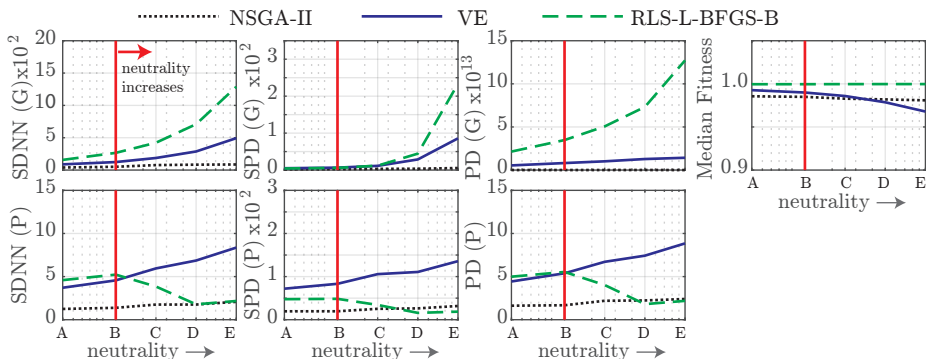
Fig. 4. Genetic (top) and phenotypic (bottom) diversity, and median fitness. Right of red marker: neutrality increases, using parameter bounds shown in Table 1.

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 fitness plots), making RLS much more appropriate when quality is more important than diversity. These results confirm the first part of Hypothesis II.

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: $(r_1, \ldots, r_8, \theta_1, \ldots, \theta_8)$, 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. 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 can see that the a number of solutions in VE and RLS are close to the Pareto set (Fig. 5 bottom left). Example result sets with low and high neutrality are shown on the right for all three algorithms. Solutions that are close to the Pareto set are shown in the brightest green color. We can confirm the second half of Hypothesis II based upon this evidence. 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.

5.3   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 the other methods is not completely fair. MMO does not get any information other than what it finds itself, neither does MOO. On the other hand, the features used in QD might not be the most diversifying we could find.

We remove the domain knowledge from QD and construct a phenotypic niching space by using a well known dimensionality reduction technique to map the
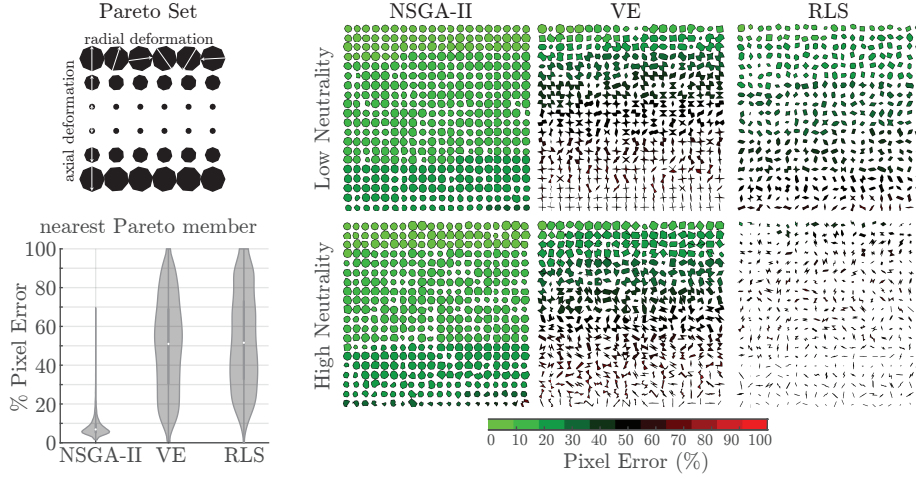
Fig. 5. The ground truth Pareto set is shown over the entire parameter range, with negative as well as positive values for the radial deformation. Bottom left: closeness to Pareto set, measured as pixel errors. The six figures on the right show example solution sets for low and high neutrality.

phenotypes to a latent space, as was done in [2], which automatically constructs a behaviorial niching space for control tasks and in [7] to measure novelty or interestingness of evolved images. 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 random genomes, drawn from a pseudo-random, space-filling Sobol sequence [21], is used to train a convolutional autoencoder (cAE). The bottleneck in the cAE is a compressed, latent space that assigns every training example a coordinate tupel. The encoder predicts the coordinates of new shapes in the latent space, which are used as phenotypic features. QD searches genomes that expand the cAE archive and improve its quality. The cAE is retrained with the new elites. The algorithm is further explained in Fig. 6. 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 [11] 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. The number of latent dimensions is set to two (to compare with manual VE), five or ten.

Result. Fig. 7 shows that the two-dimensional manual and autoencoded phenotypic space (AutoVE 2D) produce a similar diversity, whereby the quality of solutions from AutoVE 2D is higher. The higher-dimensional latent spaces increase the diversity of the solution set, but at the cost of fitness. This is to
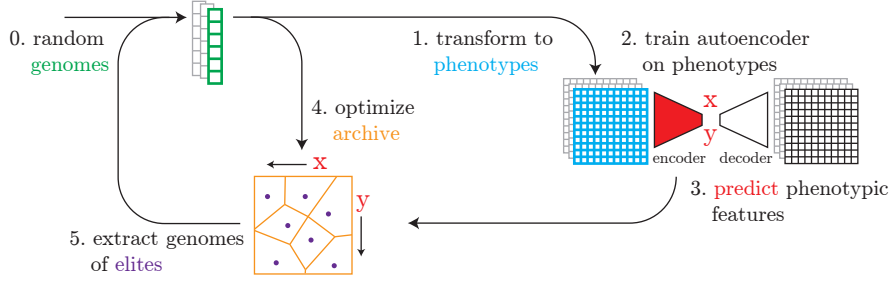
Fig. 6. AutoVE. Generating phenotypic features with an autoencoder. A random set of genomes is created (0), their phenotypes are calculated (1) and used as a training set for an autoencoder (2). The autoencoder can now be used to predict phenotypic features of new solutions (3), which is used to fill the archive (4), after which the elite solutions are extracted from the archive (5) and used to retrain the autoencoder.
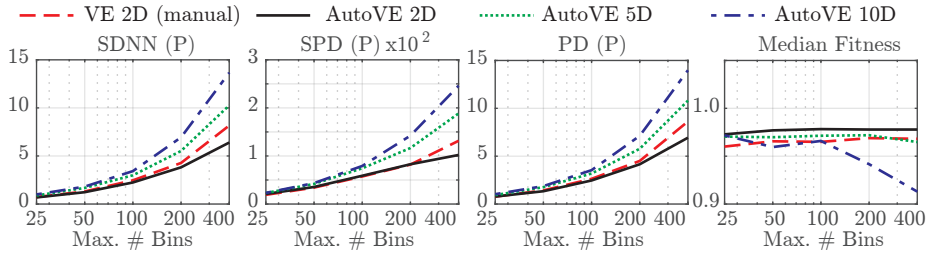

Fig. 7. Phenotypic diversity and fitness of manually crafted features (VE) compared to using an autoencoder (AutoVE) with 2, 5 or 10 latent dimensions.

be expected, as lower-fitness optima are protected in their own niches. Finally, the diversity of higher-dimensional AutoVE is around 50% higher, according to most metrics, than any of the other methods tested in this work.

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 that is similar to some older methods but much simpler and self-spreading. 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 all 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 maximize/minimize 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.

Future Work. The three paradigms have advantages and disadvantages, so some research efforts should be focused on hybridization. MOO and QD are already connected, as the boundary of valid solutions in the phenotypic archive is close to the Pareto front, yet we might want to be able to increase exploration towards it. To connect MMO and QD would mean to use a local search method in QD, which needs to overcome the genetic neutrality problem and deal with the fact that the phenotypic space cannot be searched directly. We cannot search close to a solution in genetic space and expect the newly created solutions to be close in phenotypic space.

Final Words. 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. EVOLVE 2013-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computing (2013), 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 Transactions on Evolutionary Computation **22**(2), 245–259 (2017)
5. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE transactions on evolutionary computation **6**(2), 182–197 (2002)
6. DeJong, K.: Analysis of the behavior of a class of genetic adaptive systems. Dept. Computer and Communication Sciences, University of Michigan, Ann Arbor (1975)
7. Gaier, A., Asteroth, A., Mouret, J.B.: Are quality diversity algorithms better at generating stepping stones than objective-based search? In: Proceedings of the Genetic and Evolutionary Computation Conference Companion. pp. 115–116 (2019)
8. Hamming, R.W.: Error detecting and error correcting codes. The Bell system technical journal **29**(2), 147–160 (1950)
9. Holland, J.H.: Adaptation in natural and artificial systems. MIT press (1975)
10. Hu, T., Payne, J.L., Banzhaf, W., Moore, J.H.: Robustness, evolvability, and accessibility in linear genetic programming. In: European Conference on Genetic Programming. pp. 13–24. Springer (2011)
11. Kingma, D., Ba, L., et al.: Adam: A method for stochastic optimization (2015)
12. Lehman, J., Stanley, K.O.: Abandoning objectives: Evolution through the search for novelty alone. Evolutionary computation **19**(2), 189–223 (2011)
13. 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)
14. Li, M., Yao, X.: Quality evaluation of solution sets in multiobjective optimisation: A survey. ACM Computing Surveys (CSUR) **52**(2), 1–38 (2019)
15. Li, X., Epitropakis, M.G., Deb, K., Engelbrecht, A.: Seeking multiple solutions: an updated survey on niching methods and their applications. IEEE Transactions on Evolutionary Computation **21**(4), 518–538 (2016)
16. Meinl, T., Ostermann, C., Berthold, M.R.: Maximum-score diversity selection for early drug discovery. Journal of chemical information and modeling **51**(2), 237–247 (2011)
17. Pošík, P., Huyer, W.: Restarted local search algorithms for continuous black box optimization. Evolutionary Computation **20**(4), 575–607 (2012)
18. Preuss, M.: Improved topological niching for real-valued global optimization. In: European Conference on the Applications of Evolutionary Computation. pp. 386–395. Springer (2012)
19. Pugh, J.K., Soros, L.B., Stanley, K.O.: Quality diversity: A new frontier for evolutionary computation. Frontiers in Robotics and AI **3**, 40 (2016)
20. 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)

21. 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)
22. Solow, A.R., Polasky, S.: Measuring biological diversity. Environmental and Ecological Statistics **1**(2), 95–103 (1994)
23. Tian, Y., Cheng, R., Zhang, X., Jin, Y.: Platemo: A matlab platform for evolutionary multi-objective optimization. IEEE Computational Intelligence Magazine **12**(4), 73–87 (2017)
24. 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 Computational Intelligence Magazine **14**(3), 61–74 (2019)
25. Törn, A., Viitanen, S.: Topographical global optimization. Recent advances in global optimization pp. 384–398 (1992)
26. Ursem, R.K.: Multinational evolutionary algorithms. In: Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406). vol. 3, pp. 1633–1640. IEEE (1999)
27. 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)
28. Vassiliades, V., Chatzilygeroudis, K., Mouret, J.B.: Using centroidal voronoi tessellations to scale up the multidimensional archive of phenotypic elites algorithm. IEEE Transactions on Evolutionary Computation **22**(4), 623–630 (2017)
29. 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. The Journal of Physical Chemistry A **101**(28), 5111–5116 (1997)
30. Wang, H., Jin, Y., Yao, X.: Diversity assessment in many-objective optimization. IEEE transactions on cybernetics **47**(6), 1510–1522 (2016)
31. Wessing, S.: Two-stage methods for multimodal optimization. Ph.D. thesis, Universitätsbibliothek Dortmund (2015)
32. Wessing, S., Preuss, M.: On multiobjective selection for multimodal optimization. Computational Optimization and Applications **63**(3), 875–902 (2016)