Real-valued evolutionary multi-modal multi-objective optimization by hill-valley clustering

3 authors, including:

Stef Maree
Academisch Medisch Centrum Universiteit van Amsterdam

11 PUBLICATIONS   **32** CITATIONS

Real-valued Evolutionary Multi-modal Multi-objective Optimization by Hill-valley Clustering

S.C. Maree
Amsterdam UMC
University of Amsterdam
Amsterdam, The Netherlands
s.c.maree@amc.uva.nl

T. Alderliesten
Amsterdam UMC
University of Amsterdam
Amsterdam, The Netherlands
t.alderliesten@amc.uva.nl

P.A.N. Bosman
Centrum Wiskunde & Informatica
Amsterdam, The Netherlands
peter.bosman@cwi.nl

ABSTRACT

In model-based evolutionary algorithms (EAs), the underlying search distribution is adapted to the problem at hand, for example based on dependencies between decision variables. Hill-valley clustering is an adaptive niching method in which a set of solutions is clustered such that each cluster corresponds to a single mode in the fitness landscape. This can be used to adapt the search distribution of an EA to the number of modes, exploring each mode separately. Especially in a black-box setting, where the number of modes is a priori unknown, an adaptive approach is essential for good performance. In this work, we introduce multi-objective hill-valley clustering and combine it with MAMaLGaM, a multi-objective EA, into the multi-objective hill-valley EA (MO-HillVallEA). We empirically show that MO-HillVallEA outperforms MAMaLGaM and other well-known multi-objective optimization algorithms on a set of benchmark functions. Furthermore, and perhaps most important, we show that MO-HillVallEA is capable of obtaining and maintaining multiple approximation sets simultaneously over time.

CCS CONCEPTS

• **Mathematics of computing** → *Evolutionary algorithms*; *Continuous optimization*;

KEYWORDS

Multi-objective optimization, Multi-modal optimization, Niching

1 INTRODUCTION

A multi-objective optimization problem comprises two or more objective functions that need to be optimized simultaneously. When these objectives are conflicting, instead of a single optimal solution, multiple Pareto-optimal solutions exist. Without further information, none of these solutions is better than any other. The aim of multi-objective optimization is to obtain as many diverse Pareto-optimal solutions as possible to be presented to the user for decision making. Multi-objective evolutionary algorithms (MOEAs) are aimed to find a set of Pareto-optimal solutions with different trade-offs in the objectives [1, 4, 20]. Distinct solutions that have (almost) equivalent objective values often get lost during optimization, as there is no mechanism to maintain all of these solutions [1, 4]. Maintaining these could however provide insight that can be used during decision making, and could improve performance.

The aim of multi-modal multi-objective optimization is to obtain a good approximation of the set of *all* Pareto-optimal solutions [8, 17, 19]. In other words, while multi-objective optimization is aimed to approximate the *Pareto front* in objective space, multi-modal multi-objective optimization is aimed to approximate the *Pareto set* in decision space.

Recently, a number of MOEAs for multi-modal optimization (MMOEAs) have been introduced, by applying niching techniques to existing MOEAs. These MMOEAs are then aimed to approximate the Pareto set with a diverse set of high-quality solutions in decision space. DN-NSGA-II is a niched NSGA-II using crowding, which shows better decision-space diversity compared to standard NSGA-II [9]. MO_Ring_PSO_SCD is a niched particle swarm optimizer using an index-based ring topology [19]. MOEA/D-AD is a niched MOEA/D, which was shown to outperform MO_Ring_PSO_SCD in terms of decision-space diversity [17].

The contribution of this work is twofold. We introduce a novel niching technique, hill-valley clustering, for multi-objective optimization in Section 2. Hill-valley clustering, previously introduced for single-objective optimization [11, 12], is an adaptive clustering method in which a set of solutions is clustered into niches by spending additional function evaluations to determine whether two solutions belong to the same niche. Second, in Section 3, we combine hill-valley clustering with the MOEA MAMaLGaM [15] into an MMOEA, which we refer to as MO-HillVallEA. In Section 4, MO-HillVallEA is benchmarked against MAMaLGaM and other (M)MOEAs on different (multi-modal) multi-objective optimization problems. We discuss this work in Section 5, and conclude in Section 6.

2 MULTI-OBJECTIVE NICHING

We define multi-modal multi-objective optimization by a to-be-minimized function $\mathbf{f} : \mathcal{X} \to \mathbb{R}^m$, where $\mathbf{f} = [f_0, \ldots, f_{m-1}]$ is an m-dimensional objective function and where $\mathcal{X}$ is the n-dimensional

decision space $X \subseteq \mathbb{R}^n$. A solution $x \in X$ is said to *dominate* another solution $y \in X$, when x is better than y in at least one objective, and not worse in the others. The set of all non-dominated solutions is called the *Pareto set*. The image of the Pareto set under f is called the *Pareto front*. The aim of multi-objective optimization is to find an approximation set $\mathcal{A}$ of solutions that approximates the Pareto front, while the aim of multi-modal multi-objective optimization is to approximate the Pareto set.

In optimization, a *niche* is a subset of the decision space where only one mode resides. In this work, we consider multi-objective niching as the partitioning of the decision space into the minimum number of niches required so that all objectives are uni-modal within a niche. Each niche then contains one local Pareto set. The Pareto set is a subset of the union of all local Pareto sets. When a local Pareto set maps to (part of) the global Pareto front, we refer to it as a global Pareto set. For a formal definition of local Pareto sets, or locally efficient sets, we refer the reader to [7].

Let us demonstrate this using the minimum distance (MinDist) problem [1, 6], which is based on multiple objective functions of the form, $f_i(x|c_0, c_1) = \min\{\|x - c_0\|, \|x - c_1\|\}$, for center points $c_0, c_1 \in X$. For the MinDist problem with two objectives, we use the objectives $f = [f_0(c_0, c_1), f_1(c_2, c_3)]$ and center points $c_0 = [-2, -1, 0, \ldots, 0]$, $c_1 = [2, 1, 0, \ldots, 0]$, $c_2 = [-2, 1, 0, \ldots, 0]$, and $c_3 = [2, -1, 0, \ldots, 0]$ and the Euclidean distance. MinDist with $m = 2$ objectives is visualized in Figure 1. The overlapping niches in the two objectives result in four multi-objective niches. Two local Pareto sets map to the global Pareto front, while the other two map to local Pareto fronts. A local Pareto set can be understood as a line in the decision space, connecting optima of different objectives. However, parts of these the lines resulting in the local Pareto sets are dominated by the global Pareto sets. Note that, even though both objectives have only global optima, the multi-objective problem has both local and global Pareto sets.

2.1 Single-objective hill-valley clustering

Hill-valley clustering is a niching approach that was introduced for single-objective multi-modal optimization [11, 12]. It can be used to cluster a set of solutions such that each cluster resides in a single niche. To determine whether two solutions belong to the same niche, the *hill-valley test* is used [12, 18]. The idea behind this test is that, when there is a *hill* between two solutions, they belong to different *valleys* (niches). For this, a number of solutions are evaluated along the line segment connecting the two solutions in decision space. If any of these test solutions has worse fitness than the two solutions, the two solutions belong to different niches.

To cluster an entire population of solutions, an approach inpsired by nearest-better clustering [13] is used. First, the population is sorted on fitness, starting with the best solution in the population, which forms the first cluster. Then, iteratively, each next-best solution is tested to determine whether it belongs to the niche of the nearest solution that has better fitness. When a solution-pair is rejected, the next-nearest solution with better fitness is considered. It could be that the nearest solution belongs to a different niche, but a next-nearest solution that is located slightly further, but in a different direction, belongs to the same niche. If a solution does not
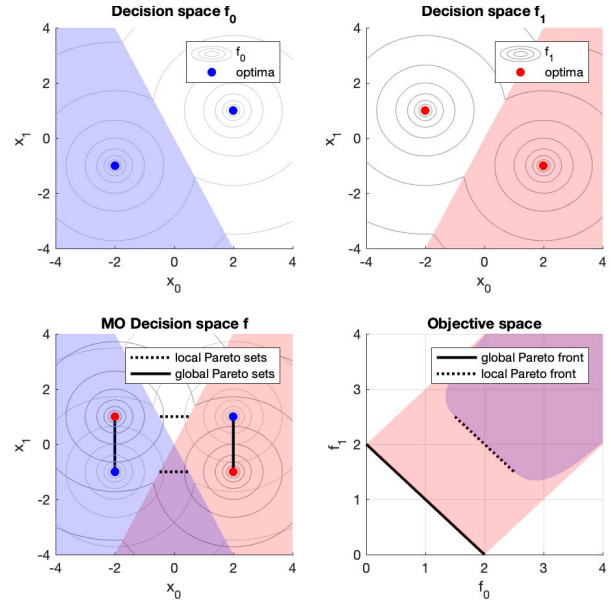

Figure 1: Niches for the MinDist problem with two objectives. The top subfigures show the single-objective niches for both objectives in different colors. The lower left subfigure show the four niches for the multi-objective problem $f = [f_0, f_1]$, together with the four local Pareto sets. The lower right subfigure shows the objective space, where the red and purple niches are shown. The blue niche overlaps exactly with red niche, and white with purple.

belong to the same niche as its $N_n = n + 1$ nearest-better neighbors, a new cluster is formed from that solution. This is repeated until the entire population is clustered. A detailed description of single-objective hill-valley clustering can be found in [12].

2.2 Multi-objective hill-valley clustering

We extend hill-valley clustering to the multi-objective case using insight gained from Figure 1. Single-objective hill-valley clustering is performed once for each objective. The result is m (different) clusterings of the population. These are then reduced to a single cluster set by taking intersections, similar as the differently colored regions illustrated in Figure 1. That is, solutions that belong to the same niches for all objectives are grouped together. See pseudo code in Algorithm 1.

To improve efficiency, test solutions evaluated by the hill-valley test are stored to prevent evaluating them twice for the clustering of different objectives. Furthermore, test solutions for solution-pairs that ended up in the same cluster are added to that cluster as well. Test solutions for solution-pairs that ended up in different clusters are discarded, as it is not clear to which cluster they should be added without further testing.

Hill-valley clustering uses at least N function evaluations to cluster a population of size N, as at least one neighbor needs to be checked for each solution. An upper bound is more difficult to formulate because the number solutions used in the hill-valley test

---

Algorithm 1: Hill-valley Clustering

function : [boolean] = HillValleyTest($\mathbf{x}, \mathbf{y}, N_t, f$)
input   : Solutions $\mathbf{x}, \mathbf{y}$, Number of test points N_t, Objective
             function f (to be minimized)
output : $\mathbf{x}, \mathbf{y}$ belong to the same niche? (boolean)

for $k = 0, \ldots, N_t - 1$ **do**
  $\mathbf{x}_k = \mathbf{x} + \frac{k}{N_t+1}(\mathbf{x} - \mathbf{y})$;
  if $\max\{f(\mathbf{x}), f(\mathbf{y})\} < f(\mathbf{x}_k)$ **then**
    return false;
  end
end
return true;

---

function : $\mathbb{C}$ = HillValleyClustering($\mathcal{P}, \mathbf{f}$)
input   : Population of solutions $\mathcal{P}$,
             Objective function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$
output : Set of clusters $\mathbb{C} = \{C_0, \ldots, C_{s-1}\}$

$\Delta = \sqrt[n]{V_{\mathcal{P}}/|\mathcal{P}|}$;   // $V_{\mathcal{P}}$ = volume of bounding box of $\mathcal{P} \subset \mathbb{R}^n$
for $l = 0, \ldots, m - 1$ **do**
  // Single-objective hill-valley clustering for objective l
  Sort $\mathbf{x} \in \mathcal{P}$ on fitness value of l-th objective, fittest first;
  $C = \{\mathbf{x}_0\}; \mathbb{K}_l = \{C\}$;   // Init cluster from best solution
  for $i = 1, \ldots, |\mathcal{P}| - 1$ **do**
    // Compute Euclidean distances to all better $\mathbf{x}_p$
    $\delta_p = \|\mathbf{x}_i - \mathbf{x}_p\|$, for $p = 0, \ldots, i - 1$;
    for $j = 0, \ldots, \min\{i - 1, n\}$ **do**
      k = index of j-th nearest better solution according
        to $\{\delta_p\}_{p=0}^{i-1}$;
      if *HillValleyTest*($\mathbf{x}_k, \mathbf{x}_i, 1 + \lfloor \delta_k/\Delta \rfloor, f_l$) **then**
        $C_{(\mathbf{x}_k)} = C_{(\mathbf{x}_k)} \cup \{\mathbf{x}_i\}$;   // Add to cluster of $\mathbf{x}_k$
        break;
      end
    end
    if $\mathbf{x}_i$ *was not added to any cluster* **then**
      $C := \{\mathbf{x}_i\}; \mathbb{K}_l = \mathbb{K}_l \cup C$;   // Init new cluster
    end
  end
end
$\mathbb{C} = \bigcap\{\mathbb{K}_l\}_{l=0}^{m-1}$;   // all intersections of all clusters in $\mathbb{K}_l$

---

depends on the distance between nearest better solutions, which is problem- and sample-dependent. As at most $n + 1$ neighbors are considered per solution, the number of function evaluations is roughly $O((n + 1)N)$ in practice.

3 MO-HILLVALLEA

We use hill-valley clustering to construct a multi-modal multi-objective evolutionary algorithm that we refer to as the multi-objective hill-valley evolutionary algorithm (MO-HillVallEA). The idea behind the algorithm is that every generation, the population is clustered using hill-valley clustering. Then, each cluster is explored with a generation of a core optimization algorithm. For this we

use different versions of MAMaLGaM in this work. MAMaLGaM is an estimation-of-distribution algorithm, i.e., a type of model-based evolutionary algorithm that learns a probability distribution to subsequently sample new solutions from [15].

The population $\mathcal{P}$ is initialized by uniform sampling, after which hill-valley clustering is used to obtain clusters $\mathbb{C} = \{C_0, C_1, \ldots\}$. For each cluster C_i, a generation of the core optimization algorithm is performed to generate offspring O_i. Core optimization algorithms of the model-building type, such as MAMaLGaM, generally estimate model-parameters over the course of multiple generations. Therefore, a set of model parameters ρ_i is maintained for each cluster C_i. To smoothly transfer model parameters over generations, clusters are linked to the nearest cluster from the previous generation. This distance is measured by the Euclidean distance between the cluster means in decision space. Note that this supports having a variable number of clusters every generation.

Besides the main population, an elitist archive $\mathbb{E}$ is maintained that contains the best solutions over time. To allow for multi-modal optimization, all non-dominated solutions within a niche need to be maintained, even if they are dominated by a solution from a different niche. Therefore, a local elitist archive, or subarchive, $\mathcal{E}_i$ is generated from each cluster C_i by maintaining only the non-dominated solutions within that cluster. The (global) elitist archive $\mathbb{E}$ is then the set of all subarchives, i.e., $\mathbb{E} = \{\mathcal{E}_0, \mathcal{E}_1, \ldots\}$. Note that in this way, dominated solutions can end up in the elitist archive, but only if the dominating solution is from a different subarchive, and thus presumably from a different niche. In order to incorporate subarchives from the previous generation into the clusters of the next generation, one must know to which cluster each subarchive corresponds. Therefore, all elites are added to the population before hill-valley clustering is applied. In that way, clusters contain all new offspring and all elites. After the clustering process and the construction of the elitist archive, the elites are removed from the clusters, but maintained in the elitist archives. Note that MAMaLGaM, within core_opt_generation($C, \mathcal{E}, \rho$), does again add a few elites back in the clusters to improve convergence.

To reduce the number of function evaluations spent during hill-valley clustering, whenever two elites are considered in the hill-valley test that originate from the same subarchive, they are said to be part of the same niche, without further testing. However, when two elites originate from different subarchives, the hill-valley test is performed. Due to the discrete nature of the hill-valley test, solutions close to the boundaries of a niche can be clustered incorrectly, resulting in small, low-fitness clusters. By testing elites from different niches every generation, these clusters are more often merged with the correct neighboring clusters.

If, at the end of the generation, the elitist archive size exceeds a user-defined target size $N_{\mathbb{E}}$, adaptive objective-space discretization is performed, to reduce computational cost while maintaining diversity in the archive [10, 15]. All subarchives are discretized using the same grid size, which is adapted until the total archive size is less than the given target size $N_{\mathbb{E}}$. Note that this maintains diversity within subarchives, but does not focus on total diversity over all archives. The main generational loop for MO-HillVallEA is inspired by MAMaLGaM, and pseudo code is given in Algorithm 2.

Algorithm 2: MO-HillVallEA

---

function : $[\mathbb{E}] = $ MO-HillVallEA$(\mathbf{f}, N, N_\mathbb{E})$

input : Objective function $\mathbf{f}$, Population size N, Elitist archive target size $N_\mathbb{E}$

output : Elitist archive $\mathbb{E} = \{\mathcal{E}_0, \mathcal{E}_1, \ldots\}$

$\mathcal{P} = $ UniformSampling(N);　　　// Also evaluates solutions
$\mathbb{C} = $ HillValleyClustering$(\mathcal{P}, \mathbf{f})$;　　　// See Algorithm 1
$\mathbb{E} = $ ConstructLocalElitistArchives$(\mathbb{C}, N_\mathbb{E})$;
$\rho = $ InitModelParameters$(\mathbb{C})$;　// For each C_i, modelparams ρ_i

while *budget remaining* **do**

$\quad \mathcal{P} = \bigcup_i \{\mathcal{E}_i\}$;　　　// Copy all elites to the population
$\quad$ **for** $(C_i, \mathcal{E}_i, \rho_i) \in (\mathbb{C}, \mathbb{E}, \rho)$ **do**
$\quad\quad (O_i, \rho_i) = $ core_opt_generation $(C_i, \mathcal{E}_i, \rho_i)$;
$\quad\quad \mathcal{P} = \mathcal{P} \cup O_i$;　　　// Collect offspring O_i
$\quad$ **end**
$\quad \mathbb{C}_{\text{prev}} = \mathbb{C}$;　　　// Backup old clusters
$\quad \mathbb{C} = $ HillValleyClustering$(\mathcal{P}, \mathbf{f})$;　　// See Algorithm 1
$\quad \mathbb{E} = $ ConstructLocalElitistArchives$(\mathbb{C}, N_\mathbb{E})$;
$\quad \mathbb{C} = $ RemoveElitesFrom$(\mathbb{C})$
$\quad (\mathbb{C}, \rho) = $ LinkClusters$(\mathbb{C}, \mathbb{C}_{\text{prev}}, \rho)$;

end

3.1 MAMaLGaM

We briefly describe a generation of MAMaLGaM, and necessary adaptations required for its use as a core optimization algorithm.

MAMaLGaM is initialized with a cluster $C_i \in \mathbb{C}$ as its population, from which offspring are generated. For this, C_i is clustered once more. To prevent confusion with hill-valley clustering, we refer to these subclusters within MAMaLGaM as *subsets*. MAMaLGaM combines two clustering approaches to obtain a predefined number of subsets k. First, for each of the m objectives, a subset is formed by the N_c best solutions in that objective. Second, the population is sorted based on domination rank, and balanced k-leader-means (BKLM) clustering [15] is used to cluster the τN best-ranked solutions into the remaining $k - m$ subsets, where τ is the selection fraction. This results in a set of k overlapping subsets. A Gaussian distribution is then estimated for each subset, from which new offspring are sampled. For this, mechanisms from different evolutionary algorithms have been used previously, such as, CMA-ES, iAMaLGaM, and AMaLGaM [15]. Subsets are registered to subsets from the previous generation and a one-to-one subset registration is applied. Besides sampling new solutions every generation, a maximum of τN elites is put back into the population, selected from the elitist archive based on objective-space diversity.

To initialize MAMaLGaM, the offspring size needs to be set, for which we use $N_i = \lfloor N/|\mathbb{C}| \rfloor$, i.e., we sample an equal number of offspring for each cluster, with the purpose of better maintaining smaller niches. Furthermore, we fix the total number of subsets k over all clusters within MO-HillVallEA. For an instance of MA-MaLGaM initialized with $C_i \in \mathbb{C}$ as population, we set the number of subsets k_i by $k_i = \lfloor k \cdot |C_i| / \sum_j |C_j| \rfloor$, with $k_i \geq 1$. Only when $k_i > m$, m out of k_i subsets are formed by performing single-objective selection. Because k_i can now vary per generation, the one-to-one subset registration that was previously used is no longer

applicable. It is therefore replaced by simply registering each subset to the nearest subset from the linked cluster of the previous generation. As subsets are formed based on objective-space distances, subset distances are computed by the distance of the subset means in objective space as well. Additionally, single-objective subsets are linked with each other if they existed in both generations. Note that this approach is significantly faster than the original one-to-one subset registration.

3.2 Post-processing the approximation set

In practice, a user can often process only a limited number of solutions during decision making, and the size of the resulting approximation set $\mathcal{A}$ that an algorithm obtains must thus be limited. Directly restricting the size of the elitist archive $\mathbb{E}$ during optimization has a risk of deteriorating performance, especially when the desired number of solutions is small. Therefore, a post-processing step is applied when the desired approximation set size $N_\mathcal{A}$ is smaller than the elitist archive target size $N_\mathbb{E}$. In that case, the approximation set is formed by combining all local elitist archives that contain at least one solution that is non-dominated within the global elitist archive. If the approximation set still exceeds $N_\mathcal{A}$, a greedy scattered subset selection is performed to reduce the archive size while preserving decision-space diversity as good as possible. This is the same subset selection algorithm as was used for BKLM clustering [15]. A similar post-processing step was used for MOEA/D-AD in [17], with the difference that MO-HillVallEA can maintain dominated solutions as long as they are from different subarchives.

3.3 Multi-start scheme

To overcome the need for parameter tuning of the population size N, a multi-start scheme is applied, similar to [3], where multiple instances of MO-HillVallEA are run simultaneously, each with a larger population size. Recursively, after 8 generations of an instance of MO-HillVallEA with population size N, one generation of an instance with population size $2N$ is performed. The first instance of MO-HillVallEA is initialized with population size $N_{\text{base}} = 10 \cdot (1 + m) \cdot (1 + \log n)$ and $k_{\text{base}} = 1 + m$ subsets, where m is the number of objectives and n the decision-space dimensionality. Each subsequent instance has a population size increased by a factor 2 and number of subsets increased by a factor 1.5. The number of subsets increases slower than the population size, so that the subset size increases over time. A single global elitist archive $\mathbb{E}$ is maintained for all instances. Whenever a new instance is initialized, it is counted how many solutions the smaller instances contributed to the elitist archive in their most recent generation. An instance that is responsible for less than 25% of the total contribution is terminated, except for the latest instance.

4 EXPERIMENTS

We empirically benchmark MO-HillVallEA against comparable (M)MOEAs. For this, MAMaLGaM was implemented in C++, and MO-HillVallEA was implemented in the same framework to reduce behavioral differences due to implementation choices. Source code of MO-HillVallEA is available at github.com/scmaree/MOHillVallEA. Results of other (M)MOEAs were provided by the authors of [17].

4.1 Test problems

We consider the MinDist problem with $m = 2$ objectives described in Section 2 with $n = \{10, 20\}$ decision variables, initialized in the box $[-4, 4]^n$. Additionally, we consider the MinDist problem with $m = 3$ objectives, given by $\mathbf{f} = [f_0(\mathbf{c}_0, \mathbf{c}_1), f_1(\mathbf{c}_2, \mathbf{c}_3), f_2(\mathbf{c}_4, \mathbf{c}_5)]$ and center points $\mathbf{c}_0 = [-4, -4, 0]$ and $\mathbf{c}_1 = [2, 2, 0]$ for the first objective, $\mathbf{c}_2 = [-2, -4, 0]$ and $\mathbf{c}_3 = [4, 2, 0]$ for the second objective, and $\mathbf{c}_4 = [-3, -2, 1]$, and $\mathbf{c}_5 = [3, 4, 1]$ for the third objective, resulting in two triangle-shaped global Pareto sets. Similarly for the $m = 2$ MinDist problem, zeros are appended to the centers if the number of decision parameters is larger.

Furthermore, we consider six bi-objective multi-modal test problems with $n = 2$ decision variables that are frequently considered in multi-modal literature: Two-On-One [14], SYM-PART{1,2,3} [16], and SSUF{1,3} [19]. The Two-On-One problem and the SSUF1 problem both have two symmetrical global Pareto sets that are connected in decision space and that map onto the same Pareto front in objective space. The SYM-PART problems have nine global Pareto sets, where SYM-PART2 has rotated global Pareto sets and SYM-PART3 has non-linear global Pareto sets. The SSUF3 problem has two global Pareto sets that are shifted, and many local Pareto sets.

All problems have a bounded decision space, and boundary repair is performed when solutions violate the boundary conditions. Initial populations are sampled uniformly in the entire domain.

4.2 Performance Measures

We use the inverted generational distance (IGD) [2] to measure objective-space diversity and the IGDX [21] to measure decision-space diversity, which are given by,

$$
\begin{aligned}
\text{IGD}_{\mathcal{A}^*}(\mathcal{A}) &= \frac{1}{|\mathcal{A}^*|} \sum_{\mathbf{y} \in \mathcal{A}^*} \min_{\mathbf{x} \in \mathcal{A}} \|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\|, \\
\text{IGDX}_{\mathcal{A}^*}(\mathcal{A}) &= \frac{1}{|\mathcal{A}^*|} \sum_{\mathbf{y} \in \mathcal{A}^*} \min_{\mathbf{x} \in \mathcal{A}} \|\mathbf{x} - \mathbf{y}\|,
\end{aligned}
\tag{1}
$$

where $\|\cdot\|$ is the Euclidean norm, and $\mathcal{A}^*$ is the reference Pareto set, for which 5000 Pareto-optimal solutions are sampled using the analytical expression of the Pareto set. For the Two-On-One problem, an approximation of the Pareto set is used [14]. Both measures should be minimized, and a perfect score of 0 is achieved when each solution in $\mathcal{A}^*$ is also in $\mathcal{A}$. Note that for multi-modal fitness landscapes, a low IGD does not imply a low IGDX. E.g., for the MinDist function in Figure 1, IGD = 0 can be achieved for IGDX = 2, when one of the global Pareto sets is perfectly approximated, while no solution has been obtained in the other set. Best achievable scores for the IGD and IGDX depend on the maximum number of solutions $N_{\mathcal{A}}$ desired in the approximation set. To compute achievable limits for these scores, the reference Pareto set is compared to a subset of it that contains $N_{\mathcal{A}}$ solutions. This subset is generated using greedy scattered subset selection [1]. Subset selection has been performed with objective-space distances for the IGD and decision-space distances for the IGDX.

Additionally, we introduce a novel performance measure that we refer to as the *mode ratio* (MR), which is the ratio of attained modes. We say that a Pareto set, or mode, is attained if the IGDX for that mode is smaller than a predefined threshold ε, which we set to $\varepsilon = 0.05$ for problems with two objectives, and $\varepsilon = 0.1$ for

problems with three objectives. To compute the MR, the reference Pareto set $\mathcal{A}^*$ is partitioned up into different modes by clustering it with hill-valley clustering. The MR should be maximized, and the best score is 1.0, when all modes are attained.

We aim in this work to improve decision space diversity, measured by the IGDX and MR. To indicate how this affects objective space diversity, the IGD is also included. It should be noted, however, that it is not the primary aim of the multi-modal approaches in this work to optimize it.

4.3 Visualization of hill-valley clustering

Hill-valley clustering is visualized for the 2D benchmark problems in Figure 2 with initial population sizes $N = 250$ and $N = 10\,000$. Tree-like clusters are formed because test solutions that are used by the hill-valley test are added to the clusters. A well-structured clustering can be observed for the Two-On-One, SYM-PART3, and three-objective MinDist problems. For the SYM-PART3 problem, low-fitness clusters with only a few solutions are formed on local Pareto sets that exist in between the global Pareto sets, which is caused by the limited number of test points in the hill-valley test. For SSUF1, the domain boundary results in that Pareto sets are connected only by a single point, resulting in more clusters than niches. For the SSUF3 problem, hill-valley clustering is not always able to connect the long stretched local Pareto sets. This can be reduced by increasing the number of neighbors considers in the clustering process.

4.4 Comparing core optimization algorithms

We compare different core optimization algorithms and their performance within MO-HillVallEA in combination with the multi-start scheme. All parameters of MO-HillVallEA and MAMaLGaM are set according to literature [15]. For the bi-objective problems, the elitist archive target size is set to $N_{\mathbb{E}} = 1000$ and for the three-objective problems to $N_{\mathbb{E}} = 2500$. To show the full potential of MO-HillVallEA, post processing is disabled, by setting the approximation set equal to the union of all subarchives $\mathcal{A} = \bigcup_i \mathcal{E}_i$.

We compare four versions of MAMaLGaM as core search algorithm. MAMaLGaM (MAM) and MAMaLGaM-univariate (MAMu) estimate a Gaussian distribution with respectively a full-rank and univariate covariance matrix. Similarly, iMAMaLGaM (iMAM) and iMAMaLGaM-univariate (iMAMu) estimate respectively a full-rank and univariate covariance matrix, but the mean and covariance matrix are estimated incrementally over the course of multiple generations. Limiting the search distribution to a univariate covariance matrix or estimating it incrementally typically requires a smaller population size, but results in a worse fit of the fitness landscape.

4.4.1 Results. Results in terms of the three performance measures for MO-HillVallEA with different core optimization algorithms and MAMaLGaM are shown in Figure 3. In terms of the IGD, all versions of MO-HillVallEA perform similar, and a similar rate of convergence can be observed for MO-HillVallEA and MAMaLGaM. MAMaLGaM ultimately achieves a better IGD score for all problems, which is mainly due to the limited elitist archive size. When the archive is full, discretization of the archive takes place. MO-HillVallEA aims to maintain parameter space diversity within the archive while reducing its size, while MAMaLGaM aims
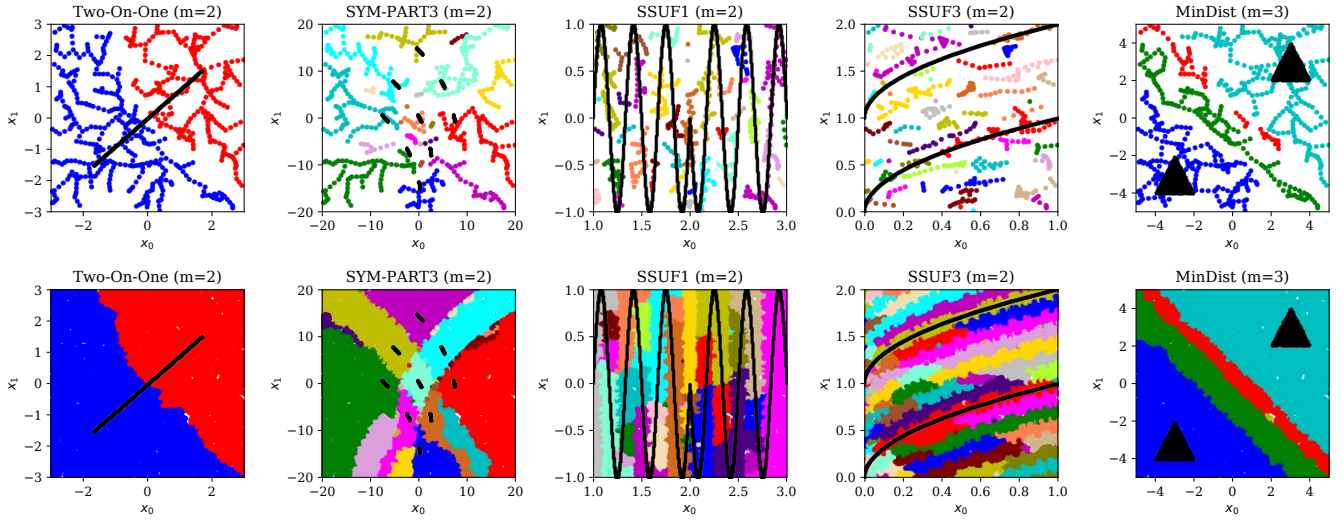
Figure 2: Hill-valley clustering of the decision space for different problems, with population size of $N = 250$ (first row) and $N = 10\,000$ (second row). The global Pareto sets are illustrated in black. For the SYM-PART3 and SSUF problems, the number of clusters obtained in the first row differed from the second row, resulting in a different color mapping.

to maintain objective-space diversity. This results in stagnating IGD convergence for MO-HillVallEA. Initially, performance of MO-HillVallEA on MinDist ($m = 2, n = 20$) is better in terms of the IGD, which can be attributed to the observation that MO-HillVallEA splits the search distribution into two, each focusing on converging locally, while MAMaLGaM tries to approach both modes with a single search distribution, resulting in a slower convergence. As soon as one of the modes is discarded due to generational drift, MAMaLGaM achieves a higher accuracy. In terms of decision-space performance, measured by the IGDX and MR, MO-HillVallEA outperforms MAMaLGaM for all problems. This holds especially for SYM-PART and MinDist problems, where MAMaLGaM ultimately obtains only a single mode, while MO-HillVallEA obtains all modes. For MinDist ($m = 2, n = 20$), due to the dimensionality of the decision space, the two modes appear to be close to each other, and distinguishing them becomes hard with a small population size. Therefore, restarts are required to obtain both modes, resulting in the step-like IGDX curves.

Differences between core optimization algorithms within MO-HillVallEA can be observed for the SYM-PART1 and the bi-objective MinDist problems in IGDX and MR. The Pareto sets for these problems are linear and non-rotated, so the univariate MAMu and iMAMu outperform the full-covariance MAM and iMAM, as can be seen in terms of MR performance and a lower IGDX. For SYM-PART3, with rotated Pareto sets, the full-covariance MAM and iMAM outperform the univariate core optimization algorithms by achieving a higher MR.

4.5 Benchmark comparison with budget

Next, we benchmark MO-HillVallEA in the same experiment setup and budget as in [17], which allows 30 000 function evaluations and approximation set size of at most $N_{\mathcal{A}} = 100$ solutions. From the proposed problems, the Omni-Test [5] was omitted, as it has

$3^5 = 243$ global Pareto sets, which is more than the number of solutions allowed in the current experiment setup. Therefore, higher IGDX scores would be achieved when each solution is a non-Pareto optimal solution that is in between Pareto sets. This is not in line with the purpose of multi-modal optimization. The IGD and IGDX are used as performance measures. The MR is omitted because the limited approximation set size $N_{\mathcal{A}}$ does not allow all modes to be obtained simultaneously with a sufficient number of solutions to achieve the desired accuracy of $\varepsilon = 0.05$. Therefore, higher MR scores would be obtained by algorithms that obtain only a subset of the modes, which is again not the purpose of this experiment. MO-HillVallEA is equipped with MAM, based on its better IGDX performance on the MinDist problems shown in Figure 3. The MMOEAs MO_Ring_PSO_SCD [19] and MOEA/D-AD [17], and the MOEAs MAMaLGaM, NSGA-II [4], and MOEA/D [17] are used for comparison. We set the elitist archive size to $N_{\mathbb{E}} = 1000$. The approximation set is post-processed using greedy scattered subset selection in *decision* space. The multi-start scheme was not used for MO-HillVallEA in this experiment as it is used to prevent the need to set the population size, but generally lowers performance in a limited budget setting as considered. Similar to the compared algorithms, MO-HillVallEA was ran with the standard population size recommended in literature. The population size N and number of clusters is fixed according to the recommendations in [15] to $k = 20$ clusters and population size $N = \frac{1}{2}k\lfloor 17 + 3n^{\frac{3}{2}}\rfloor = 250$, where n is the number of decision variables. For MAMaLGaM, the same setup as MO-HillVallEA is used, but the final elitist archive is post-processed with greedy scattered subset selection in *objective* space. Obtained scores are tested for significance with the Wilcoxon rank-sum test with Bonferroni correction for the 60 tests performed, resulting in a significance level of $\alpha = \frac{0.01}{60} = 0.00017$.
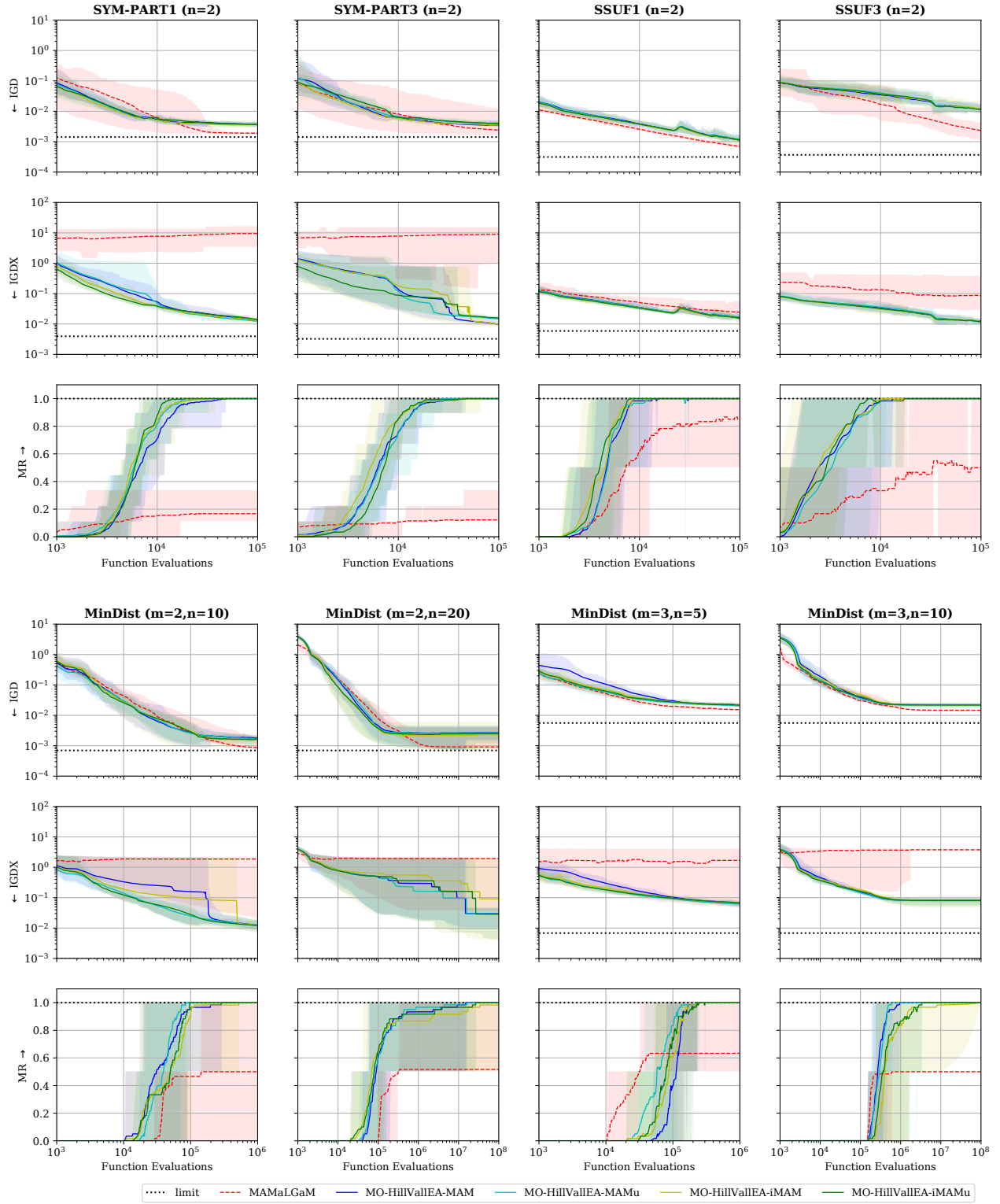
Figure 3: IGD, IGDX, and MR over time for MO-HillVallEA with different core optimization algorithms, compared to MAMaL-GaM. Results are averaged over 30 runs, shaded areas are min/max scores. Note that the MinDist problems have been run with a larger budget.

Table 1: Mean IGD and IGDX for MMOEAs and MOEAs over 31 runs (standard deviation in brackets). Bold scores are best per problem (not significantly worse than any other). In the _limit_ column, best scores achievable with $N_{\mathcal{A}} = 100$ are shown.

	Problem	limit	MO-HillVallEA-MAM	MOEA/D-AD	MO_Ring_PSO_SCD	MAMaLGaM	NSGA-II	MOEA/D
IGD (objective space)	Two-On-One	0.004	0.006 (0.000)	0.064 (0.008)	0.061 (0.006)	**0.004** (0.000)	0.049 (0.002)	0.045 (0.001)
	SYM-PART1	0.018	0.047 (0.009)	0.030 (0.002)	0.028 (0.001)	**0.018** (0.000)	0.021 (0.001)	0.047 (0.001)
	SYM-PART2	0.018	0.044 (0.006)	0.031 (0.002)	0.031 (0.002)	**0.018** (0.000)	0.023 (0.001)	0.047 (0.008)
	SYM-PART3	0.018	0.043 (0.005)	0.031 (0.002)	0.032 (0.003)	**0.019** (0.001)	0.023 (0.001)	0.045 (0.005)
	SSUF1	0.004	0.008 (0.001)	0.007 (0.001)	0.006 (0.001)	**0.004** (0.000)	0.006 (0.000)	0.006 (0.003)
	SSUF3	0.004	0.012 (0.001)	0.019 (0.006)	0.011 (0.002)	**0.005** (0.001)	0.007 (0.002)	0.063 (0.052)
IGDX (decision space)	Two-On-One	0.013	**0.026** (0.001)	0.035 (0.003)	0.037 (0.002)	0.043 (0.012)	0.148 (0.118)	0.281 (0.164)
	SYM-PART1	0.051	**0.073** (0.007)	**0.069** (0.003)	0.148 (0.024)	9.427 (1.520)	7.929 (2.343)	9.155 (2.748)
	SYM-PART2	0.052	**0.070** (0.006)	0.078 (0.003)	0.161 (0.026)	9.410 (1.082)	5.371 (1.964)	9.483 (2.191)
	SYM-PART3	0.042	**0.053** (0.004)	0.148 (0.209)	0.491 (0.369)	8.335 (3.050)	5.841 (1.892)	7.397 (1.965)
	SSUF1	0.055	**0.057** (0.001)	0.076 (0.008)	0.086 (0.006)	0.142 (0.040)	0.132 (0.022)	0.244 (0.065)
	SSUF3	0.008	**0.016** (0.003)	0.030 (0.009)	0.020 (0.006)	0.162 (0.077)	0.071 (0.048)	0.308 (0.109)

4.5.1 Results. Table 1 shows that MAMaLGaM comes very close to the best achievable IGD for all problems, outperforming all other algorithms. This suggests that MAMaLGaM is a sensible choice to use as a starting point for MO-HillVallEA. MO-HillVallEA outperforms all algorithms in terms of the IGDX on all problems but SYM-PART1. In all cases, MO-HillVallEA obtained all nine Pareto sets of the SYM-PART problems, which becomes increasingly harder for respectively SYM-PART1, 2, and 3. The reference Pareto set of the Two-On-One problem is an approximation that is accurate up to 0.0045 [14], and multiple algorithms have IGD and IGDX scores close to this accuracy, which may obfuscate true performance.

5 DISCUSSION

We showed that hill-valley clustering succesfully distinguishes multi-objective niches, which are then explored separately in MO-HillVallEA. By considering Pareto domination per niche, local Pareto sets can be maintained. However, for problems with a large number of niches, (e.g. Omni-Test [5], SSUF1, SSUF3), this approach results in a large number of clusters, and a very large population is required to explore all niches simultaneously. In that case, controlling the maximum number of clusters, or a serial search, as in [12], might be preferred.

Visualizing hill-valley clustering on problems with two decision variables provided useful insight in the behavior of hill-valley clustering. Performance of MO-HillVallEA on the MinDist problem with up to 20 variables shows that this technique extends to higher-dimensional problems. However, its performance and scalability has yet to be shown for more complex problems and even higher-dimensional decision spaces.

The current problem formulation of multi-modal optimization, where one is interested in only locating global Pareto sets, poses a fundamental difficulty. In practice, especially when a problem contains noise, the global Pareto front is never exactly attained, which makes it difficult or impossible to distinguish between global and high-quality local Pareto sets.

It would furthermore be interesting to see how hill-valley clustering can be applied to improve performance when the aim is

objective-space diversity in a multi-modal fitness landscape, rather than decision space diversity.

By aiming MO-HillVallEA for decision-space diversity, objective-space diversity deteriorated for most problems, compared to MAMaLGaM. This deterioration was found to relate mainly to the size limit of the elitist archive. If a balance between objective- and decision space diversity is desired, different size-control mechanisms could be applied to the elitist archive.

MOEA/D-AD [17] is equipped with a population size growing scheme by which the population size is adapted to the problem at hand. In this work, a population-sizing scheme was used, but adapting this scheme or the population size itself on the number of detected niches could further improve performance. Both MOEA/D-AD and MO-HillVallEA use a post-processing step to construct a limited-size approximation set while maintaining a larger elitist archive during optimization, which seems to be beneficial.

6 CONCLUSION

In this work, we introduced hill-valley clustering for clustering multi-objective optimization problems into niches. We combined hill-valley clustering with MAMaLGaM into the multi-modal multi-objective evolutionary algorithm MO-HillVallEA. We empirically show that MO-HillVallEA outperforms MAMaLGaM and other multi-objective optimization algorithms in multi-modal optimization on a set of multi-modal benchmark functions. Furthermore, and perhaps most importantly, we show that MO-HillVallEA is capable of obtaining, maintaining, and exploiting multiple local Pareto sets simultaneously over time.

REFERENCES

[1] P. A. N. Bosman. 2010. The Anticipated Mean Shift and Cluster Registration in Mixture-based EDAs for Multi-Objective Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference - GECCO-2010*. ACM Press, 351–358.

[2] P. A. N. Bosman and D. Thierens. 2002. Multi-objective Optimization with Diversity Preserving Mixture-based Iterated Density Estimation Evolutionary Algorithms. *International Journal of Approximate Reasoning* 31, 3 (2002), 259–289.

[3] A. Bouter, N.H Luong, T. Alderliesten, C. Witteveen, and P. A. N. Bosman. 2017. The multi-objective real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference - GECCO 2017*. ACM Press, 537–544.

[4] K Deb, A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (2002), 182–197.

[5] K. Deb and S. Tiwari. 2005. Omni-optimizer: A Procedure for Single and Multi-objective Optimization. In *Evolutionary Multi-Criterion Optimization*. Springer Berlin Heidelberg, Berlin, Heidelberg, 47–61.

[6] H. Ishibuchi, Y. Hitotsuyanagi, N. Tsukamoto, and Y. Nojima. 2010. Many-objective Test Problems to Visually Examine the Behavior of Multiobjective Evolution in a Decision Space. In *Proceedings of the 11th International Conference on Parallel Problem Solving from Nature: Part II (PPSN'10)*. Springer-Verlag, Berlin, Heidelberg, 91–100.

[7] Pascal Kerschke and Christian Grimme. 2017. An Expedition to Multimodal Multi-objective Optimization Landscapes. In *9th International Conference on Evolutionary Multi-Criterion Optimization - Volume 10173 (EMO 2017)*. Springer-Verlag, Berlin, Heidelberg, 329–343.

[8] X. Li, M. G. Epitropakis, K. Deb, and A. Engelbrecht. 2017. Seeking Multiple Solutions: An Updated Survey on Niching Methods and Their Applications. *IEEE Transactions on Evolutionary Computation* 21, 4 (2017), 518–538.

[9] J. J. Liang, C. T. Yue, and B. Y. Qu. 2016. Multimodal Multi-objective optimization: a preliminary study. In *Proceedings of the IEEE Congress on Evolutionary Computation - CEC 2016*. IEEE Press, 2454–2461.

[10] N. H. Luong and P. A. N. Bosman. 2012. Elitist Archiving for Multi-Objective Evolutionary Algorithms: To Adapt or Not to Adapt. In *Parallel Problem Solving from Nature - PPSN XII*. Springer Berlin Heidelberg, Berlin, Heidelberg.

[11] S. C. Maree, T. Alderliesten, D. Thierens, and P. A. N. Bosman. 2018. Benchmarking the Hill-Valley Evolutionary Algorithm for the GECCO 2018 Competition on Niching Methods Multimodal Optimization. arXiv:1807.00188

[12] S. C. Maree, T. Alderliesten, D. Thierens, and P. A. N. Bosman. 2018. Real-Valued Evolutionary Multi-Modal Optimization driven by Hill-Valley Clustering. In *Proceedings of the Genetic and Evolutionary Computation Conference - GECCO 2018*. ACM Press, 857–864.

[13] M. Preuss. 2010. Niching the CMA-ES via nearest-better clustering. In *Proceedings of the Genetic and Evolutionary Computation Conference - GECCO 2010*. ACM Press, 1711–1718.

[14] M. Preuss, B. Naujoks, and G. Rudolph. 2006. Pareto Set and EMOA Behavior for Simple Multimodal Multiobjective Functions. In *Parallel Problem Solving from Nature - PPSN 2006*. Springer Berlin Heidelberg, Berlin, Heidelberg, 513–522.

[15] S. Rodrigues, P. Bauer, and P. A. N. Bosman. 2014. A Novel Population-based Multi-objective CMA-ES and the Impact of Different Constraint Handling Techniques. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2014)*. ACM, New York, NY, USA, 991–998.

[16] G. Rudolph, B. Naujoks, and M. Preuss. 2007. Capabilities of EMOA to Detect and Preserve Equivalent Pareto Subsets. In *Proceedings of the 4th International Conference on Evolutionary Multi-criterion Optimization*. Springer-Verlag, Berlin, Heidelberg, 36–50.

[17] R. Tanabe and H. Ishibuchi. 2018. A decomposition-based Evolutionary Algorithm for Multi-modal Multi-objective Optimization. In *Parallel Problem Solving from Nature - PPSN 2018*, Vol. 5199. Springer, Berlin, Heidelberg.

[18] R. K. Ursem. 1999. Multinational evolutionary algorithms. In *Proceedings of the IEEE Congress on Evolutionary Computation - CEC 1999*. IEEE Press, vol. 3, 1633–1640.

[19] C.T. Yue, B.Y. Qu, and J.J. Liang. 2018. A Multi-objective Particle Swarm Optimizer Using Ring Topology for Solving Multimodal multi-objective problems. *IEEE Transactions on Evolutionary Computation* 22, 5 (2018), 805–817.

[20] Q. Zhang and H. Li. 2007. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation* 11, 6 (2007), 712–731.

[21] A. Zhou, Q. Zhang, and Y. Jin. 2009. Approximating the Set of Pareto-Optimal Solutions in Both the Decision and Objective Spaces by an Estimation of Distribution Algorithm. *IEEE Transactions on Evolutionary Computation* 13, 5 (2009), 1167–1189.