DBSCAN-Based Multi-Objective Niching to Approximate Equivalent Pareto-Subsets

Oliver Kramer
International Computer Science Institute
Algorithms Group
Berkeley CA 94704, USA
okramer@icsi.berkeley.edu

Holger Danielsiek
Technische Universität Dortmund
Department of Computer Science
44221 Dortmund, Germany
holger.danielsiek@tu-dortmund.de

ABSTRACT

In systems optimization and machine learning multiple alternative solutions may exist in different parts of decision space for the same parts of the Pareto-front. The detection of equivalent Pareto-subsets may be desirable. In this paper we introduce a niching method that approximates Pareto-optimal solutions with diversity mechanisms in objective *and* decision space. For diversity in objective space we use rake selection, a selection method based on the distances to reference lines in objective space. For diversity in decision space we introduce a niching approach that uses the density-based clustering method DBSCAN. The clustering process assigns the population to niches while the multi-objective optimization process concentrates on each niche independently. We introduce an indicator for the adaptive control of clustering processes, and extend rake selection by the concept of adaptive corner points. The niching method is experimentally validated on parameterized test function with the help of the $\mathcal{S}$-metric.

Categories and Subject Descriptors

G.1.6 [**Optimization**]: Global optimization; G.4 [**Math. Software**]: Algorithm design and analysis

General Terms

Algorithms

Keywords

Hybrid Evolutionary Multiobjective Algorithm, Local Search, Memetic Algorithms, Hybrid Metaheuristics

1. INTRODUCTION

The optimization of conflictive objectives belongs to one of the most challenging tasks in optimization. E.g., in machine learning conflictive objectives are prediction accuracy and model complexity, and can be handled with regularization techniques such as penalty functions. In this paper we will concentrate on the detection and approximation of equivalent Pareto-subsets. We will introduce a niching approach that is based on evolution strategies (ES) [2], a variant of evolutionary algorithms (EA) with operators like self-adaptive Gaussian mutation that are well-appropriate to numerical optimization. Furthermore, our approach is based on a recently proposed multi-objective technique called rake selection [12] that makes use of reference lines in decision space. Solutions are assigned to niches with a data analysis technique, i.e., the density-based clustering method DBSCAN [7]. Most multi-objective algorithms that have been proposed in the past concentrate on diversity in objective space. Only a few approaches also consider diversity in decision space. Our niching technique maintains diversity in objective space with rake selection, and diversity in decision space with the clustering approach.

1.1 Test Problems

In this paper we focus on multi-objective optimization problems with multiple alternative solutions in different parts of decision space for the same parts of the Pareto-front, i.e., equivalent Pareto-subsets. We use two parameterized functions, i.e., *TWO-ON-ONE* by Preuss *et al.* [13] and *SYM-PART* by Rudolph *et al.* [14]. The two-objective problem *TWO-ON-ONE* consists of two objective functions [13]:

$$f = (f_1, f_2) : \mathbb{R}^2 \to \mathbb{R}^2 : \tag{1}$$

with

$$\begin{aligned} f_1(x_1, x_2) &= x_1^4 + x_2^4 - x_1^2 + x_2^2 - cx_1x_2 + dx_1 + 20, \\ f_2(x_1, x_2) &= (x_1 - k)^2 + (x_2 - l)^2. \end{aligned} \tag{2}$$

Function f_1 has two optima while the spheric function f_2 has only one optimum at (k, l). Figure 1 shows the contour lines of *TWO-ON-ONE*, and the influence of the four parameters c, k, and l, with $d = 0$. The behavior of our niching approach on *TWO-ON-ONE* will be analyzed in Section 4.1. The experiments will show that the problem is relatively easy to solve with our approach.

The second problem *SYM-PART* has been introduced by Rudolph *et al.* [14]. It is more difficult to optimize than *TWO-ON-ONE*, and hence well appropriate to analyze the interactions between clustering and multi-objective optimization. *SYM-PART* has one Pareto-front with nine disconnected Pareto-sets. Each set covers the whole Pareto-front. It consists of nine sections in decision space for that the function has been defined. The points of the Pareto-set of each section lie on a line. The complexity of *SYM-PART* can be

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
GECCO'10, July 7–11, 2010, Portland, Oregon, USA.
Copyright 2010 ACM 978-1-4503-0072-8/10/07 ...$10.00.

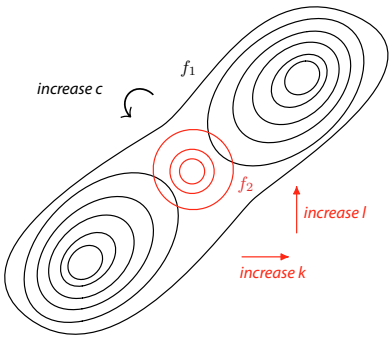

Figure 1: Contour lines and influence of parameters of *TWO-ON-ONE*, f_1 is plotted in black, f_2 in red.

controlled with the following manipulations: 1. Scaling of each section, 2. shift of each section, 3. rotation of each section by ω°, and 4. transformation, i.e., control of the individual size of each section. *SYM-PART* is described in detail by Rudolph *et al.* [14]. In the following, we summarize the derivation shortly. A simple version of the test problem consists of two objective functions and a two-dimensional decision space.

$$f(x_1, x_2) = \begin{pmatrix} (x_1 + a)^2 + x_2^2 \\ (x_1 - a)^2 + x_2^2 \end{pmatrix} \text{ for arbitrary } a > 0, \quad (3)$$

with Pareto-set:

$$\mathcal{P}^* = \{x \in \mathbb{R}^2 : x = \begin{pmatrix} x_1 \\ 0 \end{pmatrix} \text{ mit } x_1 \in [-a, a]\}, \quad (4)$$

and Pareto-front:

$$\mathcal{PF}^* = \{z \in \mathbb{R}^2 : z = \begin{pmatrix} 4a^2\nu^2 \\ 4a^2(1-\nu)^2 \end{pmatrix} | \nu \in (0,1)\}. \quad (5)$$

In this simple definition the Pareto-set covers a line of length $2 \cdot a$, the vertical distances between the sections are specified by parameter b, horizontal distances by parameter c. The problem is distributed to 3×3-section in decision space. The section size is specified by width $2 \cdot a + c$ and height b. The function is symmetric and the zero-point lies in the middle of the Pareto-set of section $(0, 0)$. Hence, the whole decision space lies between $((-3a - \frac{3}{2}c), -\frac{3}{2}b)$ and $((3a + \frac{3}{2}c), \frac{3}{2}b)$. In the following, we will use the fixed parameterization of $a = 1$, $b = 10$, $c = 8$, and a decision space between $(-15, -15)$ and $(15, 15)$. Based on the preliminary equation, function *SYM-PART1* is defined by:

$$f^{(1)}(x_1, x_2) = f(x_1 - t_1(c + 2a), x_2 - t_2 b) \quad (6)$$

To rotate the function, the following rotation matrix is used:

$$r(x) = \begin{pmatrix} \cos\omega & -\sin\omega \\ \sin\omega & \cos\omega \end{pmatrix} \quad (7)$$

with rotation angle ω. The function *SYM-PART2* sounds as follows:

$$f^{(2)}(x_1, x_2) = f^{(1)}(r_1(x), r_2(x)). \quad (8)$$

A further extension of *SYM-PART2* makes the optimization problem even more difficult. The transformation

$$d(x_1, x_2) = x_1 \times \left(\frac{x_2 - L + \varepsilon}{U - L}\right)^{-1} \quad (9)$$

for small $\varepsilon > 0$, U as upper bound and L as lower bound is the basis of *SYM-PART3*, which is defined as follows:

$$f^{(3)}(x_1, x_2) = f^{(2)}(d(x_1, x_2), x_2). \quad (10)$$

The two introduced functions, and their variants will be the basis of the experimental analyses in Section **??**.

1.2 Related Work

Our niching approach is based on ES, which have been introduced by Rechenberg and Schwefel [2], and make use of self-adaptive Gaussian mutation. Evolutionary multi-objective optimization algorithms (EMOAs) have shown outstanding success in the last decade. Algorithms like NSGA-II by Deb *et al.* [5], SPEA by Zitzler [19] and the SMS-EMOA by Beume *et al.* [1] are able to generate Pareto-sets of solutions in non-linear and multimodal scenarios. A comprehensive introduction to evolutionary multi-objective optimization is given in the book of Coello *et al.* [4]. We will use the $\mathcal{S}$-metric in the experimental analyses as indicator for the ability to approximate the Pareto-front. The $\mathcal{S}$-metric is an indicator for the approximation of the Pareto-front by computing the dominated hypervolume of a population. Niching is an evolutionary technique to maintain diversity in a population. Niching in ES is described in Shir [15]. Goldberg [9] was the first who introduced domination as selection criterion. To maintain diversity he introduced sharing [10] as a niching-based approach. Also Horn *et al.* [11] as well as Fonseca *et al.* [8] proposed a niching approach. Many Pareto-sampling techniques have been introduced, e.g., MOGA by Fonseca and Fleming [8] or SPEA2 by Zitzler *et al.* [18]. One of the most famous approaches in this line of research is the non-dominated sorting genetic algorithm (NSGA) by Srinivas and Deb [17] and its successor NSGA-II by Deb *et al.* [5]. A combined consideration of objective and decision space has been introduced by Chan and Ray [3]. It considers the location of solutions in decision space. Also Deb and Tiwari [6] proposed the omni-optimizer, which could also handle objective and decision space. A further niching-based approach has been introduced by Shir *et al.* [16]. Their *Niching-CMA-ES* combines a multi-objective variant of the CMA-ES with a special niching-based selection operator.

2. RAKE SELECTION

Rake-Selection [12] has been introduced as approach to approximate the Pareto-front with the help of reference lines called rakes that can arbitrarily be distributed in objective space. For $m = 2$ they can be distributed in parallel, and equidistantly in objective space yielding approximately uniformly distributed solutions on the Pareto-front. This section introduces the rake principle for the generation of a Pareto-set.

Let m be the number of objective functions $f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})$ with $\mathbf{x} \in \mathbb{R}^N$. The basis of rake selection is to define k reference lines r_j, $1 \leq j \leq k$ in decision space that we call *rakes* in the following. The rakes can be placed arbitrarily, but for $m = 2$ we propose to place them in parallel and distribute them uniformly in objective space to guide the search. Uniformly distributed rakes lead to approximately uniformly distributed solutions on the Pareto-front. The rakes can be arranged orthogonally and equidistantly on the surface defined by the Pareto-optimal corner points $\mathbf{c}_i$, $1 \leq i \leq m$ on a $(m-1)$-dimensional hyperplane. This

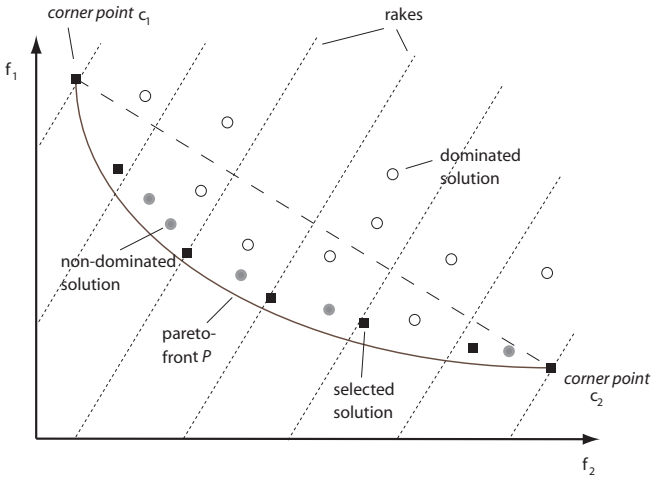

Figure 2: Illustration of Rake Selection: The rakes lie equidistantly in the objective space and guide the evolutionary selection process. Among the set of non-dominated individuals $\mathfrak{N}$ (black squares and grey circles) for each rake line r_j the closest solution (black squares) is selected.

variant of rake placement would result in k^{m-1} rakes in objective space. Each corner point is the optimum with regard to one objective. If $\mathbf{x}_i^* \in \mathbb{R}^N$ is the optimal solution minimizing f_i, the corner point $\mathbf{c}_i$ is computed by insertion $\mathbf{c}_i = (f_1(\mathbf{x}_i^*), \ldots, f_i(\mathbf{x}_i^*), \ldots, f_m(\mathbf{x}_i^*))^T$. For this sake rake selection has to compute the corner points $\mathbf{c}_i$ in the first stage by minimizing the single objectives $f_i, 1 \leq i \leq m$, unlike the corner points are known in advance. Another approach to compute the corner points is to evolve them during the optimization process by taking into account the so far best solutions with regard to each objective f_m.

To define each rake r_j a vector $\mathbf{n}$ that is orthogonal to the hyperplane $\mathbf{h}$ of corner points $\mathbf{c}_i$ has to be computed, e.g., using Gram-Schmidt orthogonalization. Due to the equidistant distribution of the intercept points of the rake the lines lie equidistantly parallel to each other in the objective space. Consequently, they cut the Pareto-front $\mathcal{PF}^*$ equidistantly with respect to a projection of the Pareto-front onto the hypergrid, and *approximately* equidistantly on the Pareto-front hypersurface. Figure 2 illustrates the situation in $N = 2$ dimensions. Parameter k is the number of rakes along the connection between the two corner points $\mathbf{c}_1$ and $\mathbf{c}_2$ that are optimal with regard to f_1 and f_2 respectively. The rakes are uniformly distributed on the connecting line between $\mathbf{c}_1$ and $\mathbf{c}_2$. The intercept points

$$\mathbf{p}_i = \mathbf{c}_1 + (i-1) \cdot \|\mathbf{c}_1 - \mathbf{c}_2\|/(k-1), \quad 1 \leq i \leq k \qquad (11)$$

define the *rake* lines $r_j, \quad 1 \leq j \leq k$. The direction vector $\mathbf{n}$, orthogonal to $\mathbf{c}_1 - \mathbf{c}_2$, is the same for each rake.

In each generation the algorithm produces population $\mathcal{P}$ consisting of λ offspring solutions with the help of intermediate recombination and self-adaptive Gaussian mutation [2]. After λ offspring solutions have been generated, non-dominated sorting is applied as the first step of the selection process. For each individual $\mathbf{x}_i$ the number d_i of solutions is computed that dominate $\mathbf{x}_i$. The set of non-dominated solutions $\mathfrak{N} = \{\mathbf{x}_i \in \mathcal{P} | d_i = 0\}$ is subject to the rake selection procedure. The core of rake selection is to select the closest solution (black squares) to each rake r_j among the set of non-dominated solutions (black squares and grey circles), i.e.:

$$\mathbf{x}_i = \arg \min_{\mathbf{x}_n \in \mathfrak{N}} \text{dist}(\mathbf{x}_n, r_j), \qquad (12)$$

if $\text{dist}(\mathbf{x}_n, r_j)$ measures the distance between point $\mathbf{x}_n$ and line r_j in $\mathbb{R}^m$. One solution may be selected multiple times by different rakes, in particular at the beginning of the search. The rakes guide the optimization process to establish equidistant solutions in objective space. If the number of selected solutions δ is smaller than μ, we add the $\mu - \delta$ solutions to the population $\mathcal{P}_{t+1}$ that are dominated least to maintain diversity.

Algorithm 1 RAKE-SELECTION (see [12])

```
Initialize population P_1
Set t := 0
Minimize each objective function f_1(x), ..., f_m(x)
Compute corner points c_1, ..., c_m
Compute hyperplane h (rake base)
Compute orthogonal vector b
Compute intercept points of rake lines r_1, ..., r_{k^{m-1}}
while NOT termination condition do
    Set t := t + 1
    Initialize offspring set O_t := ∅
    for μ = 1 to λ do
        Produce offspring x_i
        Add offspring x_i to O_t (O_t := O_t ∪ {x_i})
    end for
    Select set 𝔑 of non-dominated solutions from P_t ∪ O_t
    Compute distance matrix D_ij with distances between
    x_i and rake r_j
    Initialize set of selected solutions P_{t+1} = ∅
    for all rake r_j do
        Find closest solution x_i to rake r_j,
        i.e. (x_i = arg min_{x_n ∈ 𝔑} D(x_n, r_j))
        Add x_i to P_{t+1} (P_{t+1} := P_{t+1} ∪ {x_i})
    end for
    for i = 0 to |P_{t+1}| − μ do
        Find solution x of lowest rank in not selected solu-
        tions ((P_t ∪ O_t) \ P_{t+1})
        Add x to P_{t+1} (P_{t+1} := P_{t+1} ∪ x)
    end for
end while
```

Table 1 shows a comparison of $\mathcal{S}$-metric values between rake selection and the SMS-EMOA with the same number of fitness function evaluations on three ZDT problems from literature, see [4]. Although the SMS-EMOA directly maximizes the hypervolume, the rake selection results are competitive. On ZDT2 the standard deviation of the 25 runs of rake selection is remarkably small, while it even achieves the best $\mathcal{S}$-metric result on ZDT6.

For $m = 2$ objectives a parallel distribution of rakes r_i orthogonal to the connection of corner points is sufficient to reach all points on the Pareto-front. In case of more objectives, i.e., $m > 2$, this condition cannot be guaranteed. Adaptive corner points solve this problem, see Section 4.

problem	EMOA	best	median	worst	dev
ZDT1	rake	99.6558	99.65	99.64	0.003
	SMS	**99.6572**	99.65	99.65	0.0002
ZDT2	rake	99.3233	99.32	99.31	**0.003**
	SMS	**99.3235**	99.32	90.00	3.901
ZDT6	rake	**96,7411**	96.72	96.69	0.014
	SMS	95.6742	95.43	95.29	0.153

Table 1: Comparison of $\mathcal{S}$-metric values between rake selection and SMS-EMOA with $N = 30$ and the same number of fitness function evaluations $(20,000)$. The SMS-EMOA applies SBX recombination and polynomial mutation. Rake selection uses 50 rakes and a population of $(50 + 100)$-ES as well as $\tau_0 = 2.0$ and $\tau_1 = 2.0$.

3. CLUSTERING WITH DBSCAN

In this section we introduce the clustering technique DBSCAN (***D**ensity **B**ased **S**patial **C**lustering of **A**pplications with **N**oise*) by Ester *et al.* [7]. DBSCAN is based on the density of data samples in the search space and has two essential advantages that are useful in the context of the niching procedure. First, it allows to cluster data samples that are non-convex and intertwined, second, the number of clusters does not have be known in advance. In the following, we give a brief introduction to density-based clustering. Density-based clustering methods are based on the assumption that a certain least number of data samples have to lie in the radius of a reference data sample. The density of data samples has to exceed a given threshold. A further advantage in comparison to *squared-error*-methods like k-means is the ability to adapt the number of clusters in the course of the clustering process. Hence, the number of clusters does not have to be known in advance. Furthermore, structures of arbitrary shape like non-convex or intertwined data sets can be identified, see Figure 3.

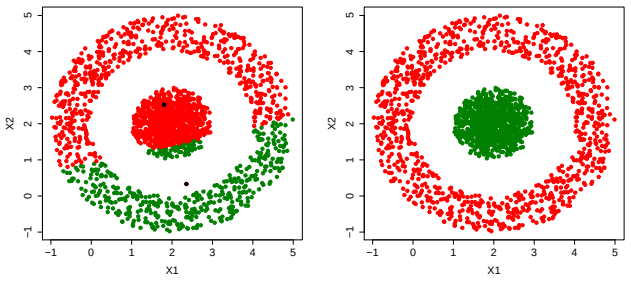

Figure 3: Comparison of clustering results of k-means (left) and DBSCAN (right). The clustering method k-means is not able to detect both clusters properly. The right part shows the clustering result of DBSCAN, and shows that both donut clusters can be detected.

At first, we introduce the most important terms of density-based clustering. Let $\mathfrak{H}$ be the set of data elements. Two parameters are essential: the neighborhood radius ε and the least number of neighbors ν. The discovery of core points plays a fundamental role. For this sake the concept of ε-neighborhood of a feature vector $\mathbf{x}$ has to be defined that comprises all data samples within an ε-distance of $\mathbf{x}$. The *ε-neighborhood* $N_\varepsilon(\mathbf{x})$ of point $\mathbf{x}$ is defined by (see [7]):

$$N_\varepsilon(\mathbf{x}) = \{\mathbf{y} \in \mathfrak{H} \mid \|\mathbf{y} - \mathbf{x}\| \leq \varepsilon\}. \tag{13}$$

Points at the border of a cluster usually have less points in their neighborhood than points in the cluster center. Hence, we distinct between core and border points of a cluster in the following. Core points exhibit at least ν neighbors in their neighborhood. Border points that are *density-connected* with a core point belong to one cluster.
Point $\mathbf{x}$ is a core point iff:

$$|N_\varepsilon(\mathbf{x})| \geq \nu, \tag{14}$$

(see [7]). Density-based clusters exhibit at least one core point, but can also exhibit border points that do not fulfill the condition to have ν points in their neighborhood. They are direct density-reachable from at least one core point, see [7]: A point $\mathbf{x}$ is *direct density-reachable* from a point $\mathbf{y}$ with regard to ε and ν if:

$$\mathbf{x} \in N_\varepsilon(\mathbf{y}) \text{ and} \tag{15}$$

$$|N_\varepsilon(\mathbf{y})| \geq \nu \ (\mathbf{y} \text{ is a core point}). \tag{16}$$

The relation *Direct Density-Reachable* is only symmetric, if both points are core points. A transitive extension is the relation *Density-Reachable*. A point $\mathbf{x}$ is *density-reachable* from a point $\mathbf{y}$ with regard to ε and ν, if a chain $\mathbf{x}_1, \ldots, \mathbf{x}_n$ of data points exists with $\mathbf{x}_1 = \mathbf{y}$ and $\mathbf{x}_n = \mathbf{x}$, such that $\mathbf{x}_{i+1}$ is directly density-reachable from $\mathbf{x}_i$. This relation is asymmetric, if one point is a core- and one point is a border-point, but symmetric for two core points. For two border-points of one cluster the relation *Density-Connected* exists. A point $\mathbf{x}$ is *density-connected* with a point $\mathbf{y}$ with regard to ε and ν, if a point $\mathbf{w}$ exists, from that $\mathbf{x}$ and $\mathbf{y}$ are density-reachable. Because the relation *density-connected* is symmetric, transitive and reflexive, it is an equivalence relation. With the help of this relation, the density-based clusters can be expressed as follows. For a data set $\mathfrak{H}$ a *density-based cluster* $\mathcal{K}$ with regard to ε and ν is defined as non-empty subset of $\mathfrak{H}$ that fulfills the following conditions:

$$\forall\, \mathbf{x}, \mathbf{y} : \mathbf{x} \in \mathcal{K} \wedge \mathbf{y} \text{ density-reachable from } \mathbf{x} \Rightarrow \mathbf{y} \in \mathcal{K}, \tag{17}$$

and

$$\forall\, \mathbf{x}, \mathbf{y} \in \mathcal{K} : \mathbf{x} \text{ density-connected with } \mathbf{y}. \tag{18}$$

Feature vectors that cannot be assigned to a density-based cluster, because no core point lies in the ε-neighborhood, are classified as noise according to Equation 19. Let $\mathcal{K}_1, \ldots, \mathcal{K}_k$ be classes of data set $\mathfrak{H}$ with regard to ε_i and ν_i, $i = 1, \ldots, k$, then all points that are not assigned to density-based clusters $\mathcal{K}_i$ are noise:

$$\text{noise} = \{\mathbf{x} \in \mathfrak{H} \mid \forall i : \mathbf{x} \notin \mathcal{K}_i\}. \tag{19}$$

Of course, noise and density-based clusters depend on parameter ν. The pseudocode of DBSCAN can be found in Algorithm 2. Parameters of DBSCAN are the data set $\mathfrak{H}$, radius ε, and the number ν of points that have to lie in the neighborhood of $\mathbf{x}$ so that it will be classified as core point.

At the beginning, *ClusterNumber* is initialized with 1, and each point is assigned to label -1, i.e., it is not assigned to a cluster yet. In the main loop of DBSCAN each unclassified point is called with method *ExpandCluster*, see

Algorithm 2 DBSCAN (see [7])

Require: Data set $\mathfrak{H}$, radius ε, density ν
Ensure: Label set $\mathfrak{L} := \{l_1, \ldots, l_n\}$
1: $ClusterNumber := 1$
2: **for** $i = 1$ **to** n **do**
3: $l_i := -1$ {-1 means not classified}
4: **end for**
5: **for all** $\mathbf{x} \in \mathfrak{H}$ **do**
6: **if** $l_x = -1$ **then**
7: **if** $ExpandCluster(\mathfrak{H}, \mathbf{x}, ClusterNumber, \varepsilon, \nu)$ **then**
8: $ClusterNumber := ClusterNumber + 1$
9: **end if**
10: **end if**
11: **end for**

Algorithm 3. The method returns $True$ if the point is classified as core point with regard to ε and ν. Furthermore, $ExpandCluster$ detects the complete density-connected cluster and assigns the current $ClusterNumber$.

To assign a label for the next cluster, parameter $ClusterNumber$ is increased after each iteration. If the point is no core point, it will be classified as noise and parameter $ClusterNumber$ is not increased. Method $ExpandCluster$ is called with the whole data set $\mathfrak{H}$, point $\mathbf{x}$, the current cluster number $ClusterNumber$, radius ε, and density ν. At the beginning a neighborhood request is called that saves all feature vectors $\mathbf{y}$ in the set $neighbors$ that lie within radius ε with regard to metric $\|\cdot\|$. In the following, we will use the Euclidean distance.

If the number of neighbors of feature vector $\mathbf{x}$ is lower than ν, point $\mathbf{x}$ will be classified as noise, l_x is assigned to 0 (noise), and the method returns $False$. If set $neighbors$ exhibits at least ν data samples, $\mathbf{x}$ is classified as core point and the lower part of Algorithm 3 (from line 6) assigns each density-reachable point $\mathbf{x}$ to the current $ClusterNumber$. For this sake two sets are used: Set $neighbors$ contains the assigned feature vectors, while set $areal$ contains new identified data samples. For each point of set $neighbors$ the set of neighbored points $areal$ is computed. If $|areal|$ is smaller than ν, $\mathbf{x}$ is classified as border point and does not have to be considered any more. Otherwise, i.e., for $|areal| \geq \nu$ all points $\mathbf{y}$ in $areal$, which are not classified yet, will be added to set $neighbors$ and assigned to the current $ClusterNumber$. Points $\mathbf{y}$ in $areal$ that are already classified as noise are assigned to the current $ClusterNumber$, and will not be given attention anymore, as they are border points. The algorithm has to assure that no core point is classified as noise. Before a new iteration starts, the current feature vector $\mathbf{x}$ is deleted from set $neighbors$. If $neighbors$ is empty, the method terminates and returns $True$.

The result of DBSCAN is deterministic, and guarantees the neighborhood conditions with regard to ε and ν, i.e., it guarantees that the same feature vectors are classified as core points. The difference that may result in different runs on the same data set is the assignment of border points that are directly density-connected from two core points of different clusters. As long as the data set is processed in the same order, the result is deterministic and the mentioned points will be assigned to the first processed cluster.

Algorithm 3 DBSCAN (ExpandCluster)(see [7])

Require: Data set $\mathfrak{H}$, data point $\mathbf{x}$, $ClusterNumber$, radius ε, density ν
Ensure: $True$, if new cluster is found
Ensure: $False$, if $\mathbf{x}$ is classified as $noise$
1: $neighbors := \{\mathbf{y} \in \mathfrak{H} \mid \|\mathbf{y} - \mathbf{x}\| \leq \varepsilon\}$
2: **if** $|neighbors| < \nu$ **then**
3: $l_i := 0$ {0 means noise}
4: **return** $False$
5: **else**
6: **for all** $\mathbf{y} \in neighbors$ **do**
7: $l_y := ClusterNumber$ {Points belong to a new cluster}
8: **end for**
9: $neighbors := neighbors \setminus \mathbf{x}$ {Delete $\mathbf{x}$ from set}
10: **while** $neighbors \neq \emptyset$ **do**
11: $\mathbf{x} :=$ first element from $neighbors$
12: $areal := \{\mathbf{y} \in \mathfrak{H} \mid \|\mathbf{y} - \mathbf{x}\| \leq \varepsilon\}$
13: **if** $|areal| \geq \nu$ **then**
14: **for all** $\mathbf{y} \in areal$ **do**
15: **if** $l_y < 1$ {l_y is not classified or noise} **then**
16: **if** $l_y = -1$ {l_y is not classified} **then**
17: $neighbors := neighbors \cup \mathbf{y}$
18: **end if**
19: $l_y := ClusterNumber$
20: **end if**
21: **end for**
22: **end if**
23: $neighbors := neighbors \setminus \mathbf{x}$ {Delete $\mathbf{x}$ from set}
24: **end while**
25: **return** $True$
26: **end if**

4. CLUSTERING-BASED NICHING

In this section we introduce a clustering-based niching approach based on DBSCAN and rake selection. We will introduce various algorithmic variations complemented by an experimental evaluation on two problems, i.e., *TWO-ON-ONE* and *SYM-PART*.

4.1 Single Population Niching and Adaptive Corner Points

First, we show a simple algorithmic variant on the problem *TWO-ON-ONE*. Unlike *SYM-PART*, the Pareto-set of *TWO-ON-ONE* is connected. For the experimental analysis we use the *TWO-ON-ONE* parameterization ($c = 20$, $d = k = l = 0$), see Section 1.1. As the Pareto-set is connected, we use k-means as DBSCAN would only find one connected cluster. We set $k = 2$. Preuss *et al.* [13] have shown that famous multi-objective optimization algorithms like NSGA-II and SPEA2 neglect about half of the Pareto-set after few generations and concentrate on only one niche. Our experiments, see Figure 4, show that our approach is able to maintain diversity in both niches.

Algorithm 4 Simple Niching

1: Initialize population
2: Optimize population with rake selection
3: Divide population into ($k = 2$) niches with k-means
4: Optimize each niche with reduced mutation rate σ and number k of rakes

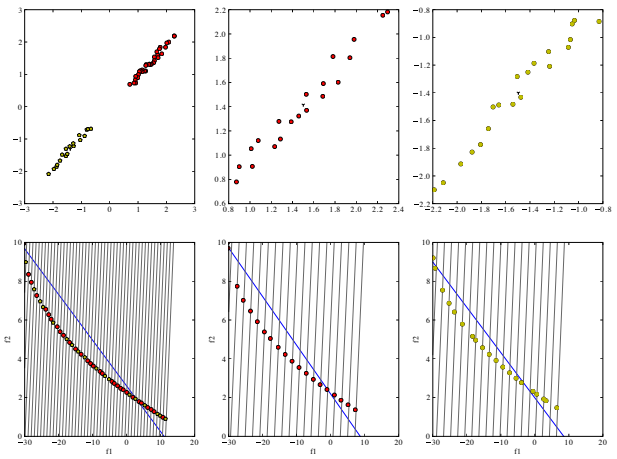

Figure 4: The clustering-based niching approach with k-means and rake selection, see Algorithm 4 on problem *TWO-ON-ONE*. The upper plots show the Pareto-set in decision space while the lower plots show the Pareto-fronts in objective space with rakes (grey) and rake base (blue). The left column shows the whole population, while the center column indicates the first cluster and the right column shows the second cluster with less individuals. Notice that the pareto-front is well approximated.

Algorithm 4 shows the simple k-means and rake selection hybrid that is used for *TWO-ON-ONE*. To avoid the necessity of preliminary problem knowledge, we propose an evolvement of the rake corner points during the optimization process. The idea of adaptive rake corner points is to use the m optimal solutions $\mathbf{x}_j$, $1 \leq j \leq m$ of population $\mathcal{P}_t$ at generation t with regard to each objective $\mathbf{f}_j$ as corner points for the rake base, i.e., $\mathbf{c}_j = \mathbf{f}_j$. First experiments have shown that this idea is not sufficient to guarantee that the corner points are distributed over the objective space, so that the rakes cut the whole Pareto-front. As the rakes are only distributed between the corner points, the original approach has no explorative character and only a part of the Pareto-front can be found. To increase the explorative capabilities the best solutions are shifted to the outside, i.e., $\mathbf{c}_j = \mathbf{f}_j + \epsilon$. The outer rakes in the neighborhood of the corner points have an explorative character, while the inner points exploit the knowledge about the location of the already explored Pareto-front. After the first optimization phase the Pareto-set is clustered and the two niches in the decision space are identified. Each niche is optimized independently with rake selection and a reduced mutation rate σ (see [2] for mutation strengths in ES). The optimization within the single niches allows to evolve a Pareto-set that covers the whole Pareto-front in each niche. Figure 4 shows the results of Algorithm 4 on *TWO-ON-ONE*. A $(50+100)$-ES is used for pre-optimization, and a $(20+100)$-ES for the post-optimization process. The upper part shows that both equivalent subsets in decision space have been found and assigned to different clusters (red and yellow circles). The lower part shows that both niches cover the Pareto-front. It also shows that each solution lies very close to a rake reference line.

4.2 Multiple Populations and Recluster Indicator

In the following, we analyze the use of multiple populations, one for each niche. The niching approach with multiple populations can be found in Algorithm 5. Let q be the number of niches. Each subpopulation makes use of a (μ, λ) population scheme. Hence, the total number of individuals is $q \cdot \mu$. The high population size increases the probability that new niches can be detected. Clustering and optimization take place alternately. We propose a recluster indicator that allows an automatic recognition of new potential niches. The *recluster indicator* ϕ, see Definition 4.1, considers the genotypic and phenotypic distance between two individuals. A threshold θ_ϕ for ϕ decides, whether two solutions lie in the same or in different niches. Again, we use the Euclidean distance.

DEFINITION 4.1 (RECLUSTER INDICATOR).
The recluster indicator ϕ measures the relation between the distances of two solutions in decision space and in objective space.

$$\phi(\mathbf{x}, \mathbf{y}) = \frac{\|\mathbf{x} - \mathbf{y}\|_2}{\|f(\mathbf{x}) - f(\mathbf{y})\|_2} \text{ for } \mathbf{x}, \mathbf{y} \in \text{(sub-)population.} \tag{20}$$

Algorithm 5 NICHING WITH RECLUSTER INDICATOR

Require: Initial population, ϕ
1: $recluster = TRUE$
2: **while** Number of clusters has not been reached **do**
3: **if** $recluster = TRUE$ **then**
4: Merge all individuals to one population
5: Run DBSCAN
6: Assign each individual to its cluster/niche
7: $recluster = FALSE$
8: **end if**
9: **for all** found clusters **do**
10: Optimize niches independently for t_{iso} generations
11: **if** $recluster = FALSE$ **then**
12: **for** $i = 1$ **to** |solutions of cluster| $- 1$ **do**
13: Compute recluster indicator $\phi := \phi(i, i+1)$, {assure that individuals i and $i+1$ are neighbors with regard to Pareto-front}
14: **if** $\phi > \theta_\phi$ **then**
15: $recluster = TRUE$
16: **end if**
17: **end for**
18: **end if**
19: **end for**
20: Select next populations
21: **end while**
22: Optimize each niche with reduced mutation rate σ and number k of rakes

The idea of the recluster indicator is based on the assumption that two neighbored individuals of one niche in objective space are neighbored in decision space. Hence, if the distance $\|f(\mathbf{x}) - f(\mathbf{y})\|_2$ of two neighbored solutions $\mathbf{x}$ and $\mathbf{y}$ in objective space is higher in decision space than the distance of two solutions that are more unalike in objective space, probably a novel niche has been found. In the case of rake selection, the neighborhood computation is relatively easy and can be reduced to checking neighbored rakes. If the

ES	θ_ϕ	runtime in sec	niching iterations	clustering calls	ffe	total $\mathcal{S}$-metric	∅ $\mathcal{S}$-metric
$(10+20)$	2	47.07	60.1	24.3	50,780	99.25	98.93
$(10+20)$	4	48.64	61.7	21.5	60,360	99.24	98.97
$(10+20)$	10	42.2	62.7	14.9	50,180	99.24	98.89
$(100+200)$	2	563.91	8.9	8.8	99,400	99.32	99.3
$(100+200)$	4	479.50	7.4	7.1	84,400	99.32	99.3
$(100+200)$	10	566.18	8.9	7.9	99,000	99.32	99.3

Table 2: Experimental results of rake selection with reclustering indicator and adaptive corner points on *SYM-PART2*, averaged over 10 experiments. Two different population sizes and three thresholds θ_ϕ have been tested.

ES	$[L,U]$	θ_ϕ	runtime in sec	niching iterations	clustering calls	ffe	total $\mathcal{S}$-metric	∅ $\mathcal{S}$-metric
$(100+200)$	$[-15,15]$	2	2,355	35.8	33.6	478,600	99.32	99.21
$(100+200)$	$[-15,15]$	4	2,362	34.9	31.9	466,200	99.32	99.28
$(100+200)$	$[-15,15]$	10	2,508	43.1	33.4	548,600	99.32	99.15
$(100+200)$	$[-20,20]$	2	1,076	17.9	15.5	197,400	99.32	99.28
$(100+200)$	$[-20,20]$	4	1,157	21.0	15.8	219,600	99.32	99.28
$(100+200)$	$[-20,20]$	10	1,124	21.1	13.8	216,800	99.32	99.28
$(100+200)$	$[-30,30]$	2	1,028	18.1	16.8	192,400	99.32	99.29
$(100+200)$	$[-30,30]$	4	975	19.0	15.6	210,600	99.32	99.28
$(100+200)$	$[-30,30]$	10	738	18.0	10.3	156,400	99.32	99.28
$(10+20)$	$[-15,15]$	2	566	417.1	243.2	541,340	99.23	98.63
$(10+20)$	$[-15,15]$	4	535	421.0	138.9	591,800	99.24	98.13
$(10+20)$	$[-15,15]$	10	429	395.5	69.5	514,980	99.23	97.49
$(10+20)$	$[-20,20]$	2	224	221.2	84.7	241,420	99.25	98.86
$(10+20)$	$[-20,20]$	4	180	206.0	44.4	212,640	99.24	98.88
$(10+20)$	$[-20,20]$	10	142	196.2	23.0	181,500	99.23	98.65
$(10+20)$	$[-30,30]$	2	190	184.7	78.6	202,280	99.24	98.85
$(10+20)$	$[-30,30]$	4	98	114.0	24.1	115,860	99.24	98.88
$(10+20)$	$[-30,30]$	10	77	116.8	15.0	97,240	99.23	98.84

Table 3: Results of the niching approach with recluster indicator and adaptive corner points on *SYM-PART3*, averaged over 10 runs with various parameterizations, two population sizes and three different thresholds θ_ϕ for the recluster indicator.

recluster indicator condition becomes true, the new potential niche has to be checked, if it is really new or already known. To answer this question DBSCAN is restarted and the solutions are clustered. Algorithm 5 shows the pseudo-code of rake selection, DBSCAN, and the recluster indicator. The optimization process terminates when the number of niches has been reached.

Important for the success of Algorithm 5 is the choice of parameter θ_ϕ. If θ_ϕ is chosen too small, too many cluster processes are started and the new method leads to no improvement in comparison with the algorithm without recluster indicator. If θ_ϕ is chosen too high, the algorithm may not be able to detect solutions from other niches and computes a necessary cluster reassignment too late. This would lead to an increased number of fitness function evaluations. Parameter θ_ϕ is problem-dependent. A further analysis of its influence will be subject to future work. Table 2 shows the results of the DBSCAN niching variant with recluster indicator and adaptive corner points on function *SYM-PART2*. The results are averaged over 10 runs. Two different population sizes and three different thresholds θ_ϕ for the cluster indicator have been tested. The fastest run with a satisfying $\mathcal{S}$-metric was achieved with a $(10+20)$-ES and $\theta_\phi = 2$, while the highest $\mathcal{S}$-metric was achieved with all $(100+200)$-ES variants. Table 3 shows the experimental results of the niching approach with recluster indicator and adaptive corner points on *SYM-PART3*. The results of different values for the parameter θ_ϕ do not differ much within one class of parameter settings, but a high value of $\theta_\phi = 10$ achieves the best results. Interestingly, almost all runs achieved the high $\mathcal{S}$-metric value of 99.32. Also the average $\mathcal{S}$-metric values show satisfying results, while the $\mathcal{S}$-metric of the subpopulations partly differ. E.g., for the parameter constellation $L = -15,\ U = 15$ slight deteriorations can be observed. In these cases, we recommend a further concentration on particular niches to improve the results in terms of $\mathcal{S}$-metric measures.

5. SUMMARY AND OUTLOOK

The approximation of equivalent Pareto-subsets is no easy undertaking. Without data analysis methods like clustering that support the optimization process, standard evolutionary multi-objective optimization techniques fail to detect all Pareto-subsets. Furthermore, they fail to cover these subsets in decision space in order to allow the approximation of alternative solutions in objective space. In this paper we have proposed an approach that uses density-based clustering for the detection of disjunct subsets in decision space.

Density-based approaches are well appropriate to detect and separate non-convex data sets. The number of clusters does not have to be known in advance, but the number of niches, if we apply the termination condition of this paper. It will the decision of the practitioner, if enough equivalent Pareto-optimal solutions have been detected. Our experiments have shown that rake selection and DBSCAN turn out to be a successful hybridization approach for the approximation of equivalent Pareto-subsets. The approach was successful in optimizing the simple function *TWO-ON-ONE*, and various instances of the more difficult problem *SYM-PART*.

As not many problems with equivalent Pareto-subsets exist, it must be future work to develop more test problems, also from practice. E.g., properties like non-convex and intertwined Pareto-sets have not been tested yet. As such data distributions belong to the strengths of DBSCAN, we can expect that our approach will be capable to solve this problem class as well. Further work may concentrate on a comparison of various clustering techniques, and the interaction with various EMOAs, as well as the combination with covariance matrix adaptation techniques. Parameter θ_ϕ of the recluster indicator is problem dependent. We are currently working on a parameter-free recluster indicator based on the comparisons between three points.

6. REFERENCES

[1] N. Beume, B. Naujoks, and M. Emmerich. SMS-EMOA: Multiobjective Selection based on Dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.

[2] H.-G. Beyer and H.-P. Schwefel. Evolution strategies – a comprehensive introduction. *Natural Computing*, 1(1):3–52, March 2002.

[3] K. P. Chan and T. Ray. An evolutionary algorithm to maintain diversity in the parametric and the objective space. In *Proceedings of IEEE International Conference on Computational Robotics and Autonomous Systems (CIRAS 2005)*, page CD proceedings, Los Alamitos, 2005. IEEE Computer Society Press.

[4] C. A. Coello Coello, G. B. Lamont, and D. A. van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Genetic and Evolutionary Computation Series. Springer Science+Business Media, New York, NY, USA, 2. edition, 2007.

[5] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast elitist multi-objective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2000.

[6] K. Deb and S. Tiwari. Omni-optimizer: A procedure for single and multi-objective optimization. In *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization, EMO 2005, Guanajuato, Mexico*, pages 47–61, 2005.

[7] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231. AAAI Press, 1996.

[8] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization. In *ICGA*, pages 416–423, 1993.

[9] D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.

[10] D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and their Application*, pages 41–49, Hillsdale, NJ, USA, 1987. Lawrence Erlbaum Associates Inc.

[11] J. Horn, N. Nafpliotis, and D. E. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, pages 82–87, 1994.

[12] O. Kramer and P. Koch. Rake selection: A novel evolutionary multi-objective optimization algorithm. In *Proceedings of KI 2009: Advances in Artificial Intelligence, 32nd Annual German Conference on AI*, pages 177–184. Springer, 2009.

[13] M. Preuss, B. Naujoks, and G. Rudolph. Pareto set and emoa behavior for simple multimodal multiobjective functions. In *PPSN IX: Proceedings of the 9th International Conference of Parallel Problem Solving from Nature*, pages 513–522. Springer, 2006.

[14] G. Rudolph, B. Naujoks, and M. Preuss. Capabilities of emoa to detect and preserve equivalent pareto subsets. In S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, editors, *EMO 2007: Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization*, volume 4403 of *Lecture Notes in Computer Science*, pages 36–50. Springer, 2007.

[15] O. M. Shir. *Niching in Derandomized Evolution Strategies and its Applications in Quantum Control*. PhD thesis, Leiden University, The Netherlands, 2008.

[16] O. M. Shir, M. Preuss, B. Naujoks, and M. Emmerich. Enhancing decision space diversity in evolutionary multiobjective algorithms. In *EMO '09: Proceedings of the 5th International Conference on Evolutionary Multi-Criterion Optimization*, pages 95–109, Berlin, Heidelberg, 2009. Springer.

[17] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2:221–248, 1994.

[18] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*, pages 95–100. International Center for Numerical Methods in Engineering (CIMNE), 2002.

[19] E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.