

# Additive Approximations of Pareto-Optimal Sets by Evolutionary Multi-Objective Algorithms

Christian Horoba\*  
Fakultät für Informatik, LS 2  
Technische Universität Dortmund  
Dortmund, Germany

Frank Neumann  
Algorithms and Complexity  
Max-Planck-Institut für Informatik  
Saarbrücken, Germany

## ABSTRACT

Often the Pareto front of a multi-objective optimization problem grows exponentially with the problem size. In this case, it is not possible to compute the whole Pareto front efficiently and one is interested in good approximations. We consider how evolutionary algorithms can achieve such approximations by using different diversity mechanisms. We discuss some well-known approaches such as the density estimator and the  $\varepsilon$ -dominance approach and point out how and when such mechanisms provably help to obtain good additive approximations of the Pareto-optimal set.

## Categories and Subject Descriptors

F.2.2 [Analysis of Algorithms and Problem Complexity]: Nonnumerical Algorithms and Problems

## General Terms

Algorithms, Performance, Theory

## Keywords

Evolutionary algorithm, density estimator, diversity mechanism,  $\varepsilon$ -dominance, multi-objective optimization, runtime analysis

## 1. INTRODUCTION

Multi-objective optimization problems are often difficult to solve as the task is not to compute a single optimal solution but a set of solutions representing the different trade-offs with respect to the given objective functions. The number of these trade-offs can be exponential with regard to the problem size, which implies that not *all* trade-offs can be computed efficiently. In this case, one is interested in good approximations of the Pareto front consisting of a not too large set of Pareto-optimal solutions.

\*This author was supported by the Deutsche Forschungsgemeinschaft (DFG) as part of the Collaborative Research Center "Computational Intelligence" (SFB 531).

Evolutionary algorithms (EAs) form a class of randomized algorithms that is popular with practitioners. They are easy to implement and often achieve good results without having much knowledge of the problem under consideration. It has been observed empirically that multi-objective evolutionary algorithms (MOEA) are able to obtain good approximations for a wide range of multi-objective optimization problems. The aim of this paper is to contribute to the theoretical understanding of MOEAs in particular with respect to their approximation behavior.

One field of research that has gained increasing interest during recent years is the rigorous analysis of simple randomized algorithms with respect to their runtime behavior. The advantage of this line of research is that it provides rigorous results, which often show that even simple EAs are quite successful. On the other hand, EAs are composed of several components that influence their behavior. The investigation of simplified EAs often can not capture important features of more sophisticated EAs. We want to put forward this line of research by considering MOEAs that incorporate diversity mechanisms that are frequently used in successful applications and analyzing their approximation behavior on different multi-objective optimization problems.

Most rigorous studies of MOEAs investigate a simple MOEA called GSEMO [5]. These studies comprise rigorous statements about the runtime of MOEAs, which increase the theoretical understanding of when and how MOEAs are able to solve multi-objective optimization problems. For instance, [6] introduces an example problem where a population-based MOEA outperforms several algorithms that are based on a single individual, and [1] shows that the addition of objectives can be advantageous as well as disadvantageous. In recent years, even the analysis of simple MOEA for combinatorial optimization problems has become possible [4, 11, 12].

However, GSEMO is a quite simple MOEA with some disadvantages in comparison to the MOEAs used in practice. One disadvantage of GSEMO is that the population size grows with the number of discovered non-dominated individuals since the population achieves *all* non-dominated individuals found so far. Most MOEAs used in practice are based on a population of fixed size. When dealing with large Pareto fronts, these MOEAs try to spread the individuals in the population over the whole Pareto front. The application of a wide range of diversity mechanism can help to achieve this

goal [3]. A popular diversity strategy is to use a density estimator to favor individuals in less crowded regions of the objective space [10] (density estimator approach). Another well-known diversity strategy is to partition the objective space into boxes and to restrict the population to at most one individual per box [9] ( $\delta$ -dominance approach). We concentrate in this paper on the density estimator approach (using the density estimator proposed for SPEA2 [13]) and the  $\delta$ -dominance approach.

The goal of this paper is to better understand how such diversity mechanisms influence the approximation ability of MOEAs. We present example problems that allow rigorous statements about the usefulness of such diversity mechanisms. In particular, we point out for each diversity mechanism a typical situation, which explains when and how the considered diversity mechanism is crucial to obtain a good approximation of the Pareto front of the given problem.

The outline is as follows. In Section 2, we introduce the basic definitions and the algorithms that are relevant to this paper. We present in Section 3 a problem where a proper diversity strategy is crucial for obtaining good approximations. We show that the density estimator as well as the  $\delta$ -dominance approach lead to a significant performance boost in comparison to GSEMO. In Section 4, we present two problems that point out the differences between both diversity mechanisms. Finally, we discuss our results and finish with some conclusions.

## 2. DEFINITIONS AND ALGORITHMS

In multi-objective optimization the goal is to optimize several objective functions simultaneously. The different objective functions are often conflicting, which implies that there is no single optimal solution but a set of solutions that represents the possible trade-offs with respect to the objective functions. We consider problems defined on binary strings of fixed length, i. e., the search space is  $\mathbb{B}^n$  where  $\mathbb{B} = \{0, 1\}$ . The objective function  $f: \mathbb{B}^n \rightarrow \mathbb{R}^m$  maps search points to a vector-valued objective space. An objective vector  $u$  *weakly dominates*  $v$  ( $u \succeq v$ ) iff  $u_i \geq v_i$  for all  $i \in \{1, \dots, m\}$ , and  $u$  *dominates*  $v$  ( $u \succ v$ ) iff  $u \succeq v$  and  $u \neq v$ . The concept of dominance directly transfers from the objective vectors to the corresponding search points, e. g.,  $x \succeq y$  holds for two search points  $x$  and  $y$  iff  $f(x) \succeq f(y)$ .

The MOEA Global Simple Evolutionary Multi-objective Optimizer (GSEMO) (see Algorithm 1) has been analyzed with respect to runtime behavior on various problems [1, 4, 5, 6, 11, 12]. GSEMO maintains a population  $P$  of variable size, which serves as an archive for the discovered non-dominated individuals as well as a pool of possible parents.  $P$  is initialized with a single individual that is drawn uniformly at random from the decision space. In each generation an individual  $x$  is drawn uniformly at random from  $P$ , and an offspring  $y$  is created by applying a mutation operator to  $x$ . We resort to the global mutation operator that flips each bit of  $x$  with probability  $1/n$  throughout this paper. If  $y$  is not dominated by any individual of  $P$ ,  $y$  is added to  $P$ . All other individuals that are weakly dominated by  $y$  are in turn deleted from  $P$ . The last step ensures that  $P$  stores for each discovered non-dominated objective vector  $u$  just the most recently created decision vector  $x$  with  $f(x) = u$ .

---

### Algorithm 1 GSEMO

---

```

1: choose an initial population  $P \subseteq \mathbb{B}^n$  with  $|P| = 1$  uniformly at random
2: repeat
3:   choose a parent  $x \in P$  uniformly at random
4:   create an offspring  $y$  by flipping each bit of  $x$  with probability  $1/n$ 
5:   if  $\nexists z \in P: z \succ y$  then
6:     set  $P \leftarrow (P \setminus \{z \in P \mid y \succeq z\}) \cup \{y\}$ 
7:   end if
8: until stop

```

---



---

### Algorithm 2 GDEMO

---

```

1: choose an initial population  $P \subseteq \mathbb{B}^n$  with  $|P| = 1$  uniformly at random
2: repeat
3:   choose a parent  $x \in P$  uniformly at random
4:   create an offspring  $y$  by flipping each bit of  $x$  with probability  $1/n$ 
5:   if  $\nexists z \in P: z \succ y \vee b(z) \succ b(y)$  then
6:     set  $P \leftarrow (P \setminus \{z \in P \mid b(y) \succeq b(z)\}) \cup \{y\}$ 
7:   end if
8: until stop

```

---

Often the number of Pareto-optimal objective vectors grows exponentially with the problem size. In this case, it is not possible to obtain the whole front efficiently. Hence, we are interested in the time to obtain a good approximation of the Pareto front and want to examine in which situations the use of a diversity mechanism can help to achieve this goal. In this paper, we use the additive  $\varepsilon$ -dominance measure (see e. g. [9]) to judge the quality of an approximation. An objective vector  $u$   $\varepsilon$ -dominates  $v$  ( $u \succeq_\varepsilon v$ ) iff  $u_i + \varepsilon \geq v_i$  for all  $i \in \{1, \dots, m\}$ . A set of objective vectors  $T$  (or a set of corresponding search points) is called an  $\varepsilon$ -approximation of  $f$  iff there is for each objective vector  $v \in f(\mathbb{B}^n)$  at least one objective vector  $u \in T$  that  $\varepsilon$ -dominates  $v$ .

If we are satisfied with an approximation of the Pareto front, it might be beneficial to avoid storing similar individuals in the population of GSEMO. We partition the objective space into boxes and store at most one individual per box in the population. To do this, we map an individual  $x$  to the box index vector  $b(x) = (b_1(x), \dots, b_m(x))$  with  $b_i(x) := \lfloor f_i(x)/\delta \rfloor$  where  $\delta \in \mathbb{R}^+$  determines the size of the boxes. Global Diversity Evolutionary Multi-Objective Optimizer (GDEMO) (see Algorithm 2) is a MOEA that incorporates this concept.

Often evolutionary algorithms work with a population of fixed size and try to spread the individuals in the population over the Pareto front by increasing the distance in the objective space between the individuals. To keep a population of size  $\mu$  in each generation either the offspring has to be skipped or one individual in the current population has to be deleted to make room for the offspring. We investigate a simplification of the diversity mechanism incorporated into SPEA2 [13], which relies on a so-called density estimator.

Let  $Q$  be a given set of search points. The  $\text{rank}_Q(x)$  of a search point  $x \in Q$  is given by the number of search points in  $Q$  that dominate  $x$ , i. e.,  $\text{rank}_Q(x) := |\{y \in Q \mid y \succ x\}|$ .

---

**Algorithm 3** SELECTION FOR REMOVAL

---

**Input:** set of search points  $Q$ **Output:** search point  $z \in Q$ 

- 1: set  $Q' \leftarrow \arg \max_{x \in Q} \text{rank}_Q(x)$
  - 2: set  $Q'' \leftarrow \arg \min_{x \in Q'} \text{dist}_Q(x)$
  - 3: **return**  $z \in Q''$  chosen uniformly at random
- 

---

**Algorithm 4** RADEMO

---

- 1: choose an initial population  $P \subseteq \mathbb{B}^n$  with  $|P| = \mu$  uniformly at random
  - 2: **repeat**
  - 3:   choose a parent  $x \in P$  uniformly at random
  - 4:   create an offspring  $y$  by flipping each bit of  $x$  with probability  $1/n$
  - 5:   choose an individual  $z \in P \cup \{y\}$  for removal using Algorithm 3
  - 6:   set  $P \leftarrow (P \cup \{y\}) \setminus \{z\}$
  - 7: **until** stop
- 

$x\}$ . Additionally, a metric on the objective space is taken into account. We consider the maximum metric  $d(u, v) := \max_{i \in \{1, \dots, m\}} |u_i - v_i|$  where  $u$  and  $v$  are objective vectors. Let  $\text{dist}_Q(x) := (\text{dist}_Q^0(x), \dots, \text{dist}_Q^{|Q|-1}(x))$  where  $\text{dist}_Q^k(x)$  denotes the distance  $d(f(x), f(y))$  from  $x \in Q$  to its  $k$ -th nearest neighbor  $y \in Q$  with respect to  $d$ .

The archive truncation procedure of SPEA2 selects a search point  $x \in Q$  with the lowest  $\text{dist}_Q(x)$  value with respect to the lexicographic order from the search points with the highest  $\text{rank}_Q(x)$  value for removal (see Algorithm 3). Using this selection procedure, we obtain a simplified version of SPEA2, which we refer to as Rank- And Distance-based Evolutionary Multi-objective Optimizer (RADEMO) (see Algorithm 4).

For theoretical investigations, we count the number of generations until a desired goal has been achieved. This number is called the runtime of the considered algorithm. The subject of our investigations is the runtime that an algorithm needs to achieve an additive  $\varepsilon$ -approximation of a given problem where  $\varepsilon \in \mathbb{R}^+$ .

### 3. APPROXIMATION OF LARGE PARETO FRONTS

In this section, we discuss how diversity mechanisms can be provably helpful to achieve an  $\varepsilon$ -approximation of an exponentially large Pareto front. All the problems examined in this paper depend on a parameter  $\varepsilon$  and the goal is to examine whether our algorithms are able to achieve an  $\varepsilon$ -approximation of the Pareto-optimal set in polynomial time.

To simplify the following function definitions, we use the common functions

$$|x|_1 := \sum_{i=1}^n x_i \quad \text{and} \quad |x|_2 := \sum_{i=1}^n 2^{n-i} \cdot x_i.$$

We assume w.l.o.g. that the number of bits is even and refer to the first (second) half of a bit string  $x$  as  $x'$  ( $x''$ ). We consider the bi-objective example function  $\text{LF}'_\varepsilon(x) =$

$(\text{LF}'_{\varepsilon,1}(x), \text{LF}'_{\varepsilon,2}(x))$  (Large Front) where

$$\text{LF}'_{\varepsilon,1}(x) := \begin{cases} (2 \cdot |x'|_1 + 2^{-n/2} \cdot |x''|_2) \cdot \varepsilon & \min\{|x'|_1, |\overline{x'}|_1\} \geq \sqrt{n} \\ 2 \cdot |x'|_1 \cdot \varepsilon & \text{otherwise} \end{cases}$$

$$\text{LF}'_{\varepsilon,2}(x) := \begin{cases} (2 \cdot |\overline{x'}|_1 + 2^{-n/2} \cdot |\overline{x''}|_2) \cdot \varepsilon & \min\{|x'|_1, |\overline{x'}|_1\} \geq \sqrt{n} \\ 2 \cdot |\overline{x'}|_1 \cdot \varepsilon & \text{otherwise.} \end{cases}$$

Figure 1(a) shows the objective space of  $\text{LF}'_\varepsilon$ .

The above-defined function  $\text{LF}'$  induces the same dominance relation on the search space as the similar function  $\text{LF}_\varepsilon(x) = (\text{LF}_{\varepsilon,1}(x), \text{LF}_{\varepsilon,2}(x))$  [7] with

$$\text{LF}_{\varepsilon,1}(x) := \begin{cases} (1+\varepsilon)^{2 \cdot |x'|_1 + 2^{-n/2} \cdot |x''|_2} & \min\{|x'|_1, |\overline{x'}|_1\} \geq \sqrt{n} \\ (1+\varepsilon)^{2 \cdot |x'|_1} & \text{otherwise} \end{cases}$$

$$\text{LF}_{\varepsilon,2}(x) := \begin{cases} (1+\varepsilon)^{2 \cdot |\overline{x'}|_1 + 2^{-n/2} \cdot |\overline{x''}|_2} & \min\{|x'|_1, |\overline{x'}|_1\} \geq \sqrt{n} \\ (1+\varepsilon)^{2 \cdot |\overline{x'}|_1} & \text{otherwise} \end{cases}$$

shown in Figure 1(b). In particular,

$$\text{LF}'_\varepsilon(x) = \varepsilon \cdot \log_{1+\varepsilon}(\text{LF}_\varepsilon(x))$$

holds for each  $x \in \mathbb{B}^n$ . This allows to transfer the results for GSEMO and GDEMO on LF provided in [7] directly to  $\text{LF}'$ . Hence, we can state the following results.

**THEOREM 1.** *The time until the algorithm GSEMO has achieved an  $\varepsilon$ -approximation of  $\text{LF}'_\varepsilon$  is  $2^{\Omega(n^{1/4})}$  with probability  $1 - 2^{-\Omega(n^{1/4})}$ .*

**THEOREM 2.** *Choosing  $\delta = \varepsilon$  as box size, the algorithm GDEMO has achieved an  $\varepsilon$ -approximation of  $\text{LF}'_\varepsilon$  in expected time  $O(n^2 \log n)$ .*

The reason for the stated negative result for GSEMO on  $\text{LF}'$  is that the algorithm produces many Pareto-optimal objective vectors with roughly  $n/4$  1-bits in the first half of the bitstring. However, to achieve an  $\varepsilon$ -approximation it is necessary that for each  $i$ ,  $0 \leq i \leq n/2$ , a solution with  $i$  1-bits in the first half of the bitstring is obtained. This implies that at least  $n/2 + 1$  search points are necessary to achieve an  $\varepsilon$ -approximation. In contrast to the negative result for GSEMO, the algorithm GDEMO is able to obtain an  $\varepsilon$ -approximation efficiently when choosing the value of  $\delta$ , which determines the size of the boxes, as in the definition of  $\text{LF}'$ . This has the effect that the algorithm keeps for each fixed number of 1-bits in the first half exactly one individual in the population.

In the extreme case when the parameter  $\delta$  in the GDEMO algorithm becomes small, the algorithm collapses to GSEMO. This shows that the right choice of  $\delta$  is crucial for dealing with large Pareto fronts. In the following, we show that the density estimator ensures in a natural way a spread over the Pareto front of  $\text{LF}'$ . We already know that  $n/2 + 1$  points are necessary to achieve an  $\varepsilon$ -approximation of  $\text{LF}'$ . Therefore, we assume that the population size of RADEMO is at least  $n/2 + 1$  and show that the algorithm constructs an  $\varepsilon$ -approximation efficiently.

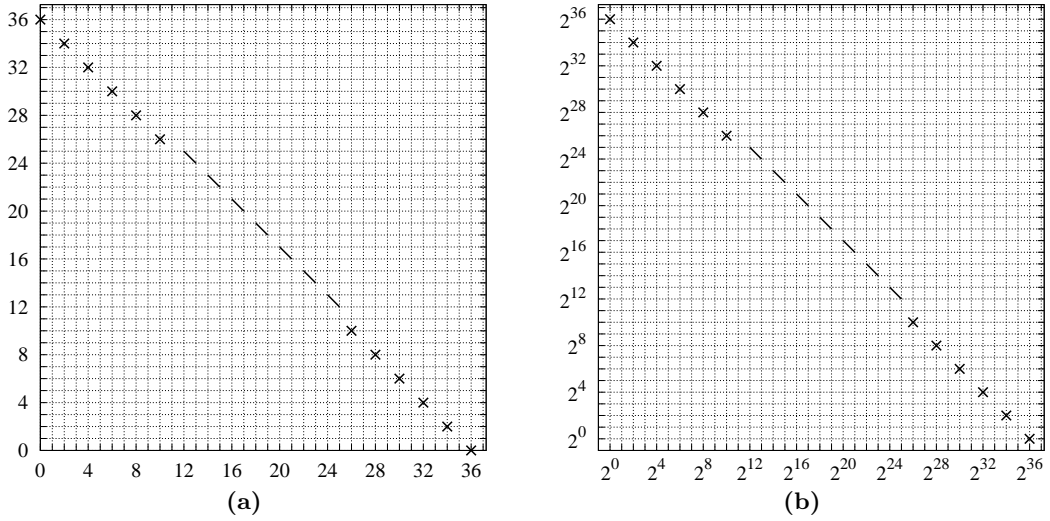


Figure 1: Objective space of  $LF'_\varepsilon$  (see (a)) and  $LF_\varepsilon$  (see (b)) for  $\varepsilon = 1$  and  $n = 36$ .

**THEOREM 3.** *Choosing  $\mu \geq n/2 + 1$  as population size, the algorithm RADEMO has achieved an  $\varepsilon$ -approximation of  $LF'_\varepsilon$  in expected time  $O(\mu n \log n)$ .*

*Proof.* Let  $x$  and  $y$  be two individuals. If  $|x'|_1 = |y'|_1$ , the distance between the two search points in the objective space is small, i. e.,  $d(LF'_\varepsilon(x), LF'_\varepsilon(y)) < \varepsilon$  holds. Otherwise,  $d(LF'_\varepsilon(x), LF'_\varepsilon(y)) > \varepsilon$  holds. This implies that once a solution with a specific number of 1-bits in the first half of the bitstring is obtained, such a solution will remain in the population during the whole optimization process as  $\mu \geq n/2 + 1$ . We therefore have to consider the different times to produce the individuals with  $i$ ,  $0 \leq i \leq n/2$ , 1-bits in the first half of the bitstring. An offspring  $y$  with  $|y'|_1 = i$  is created with probability at least

$$\min\{n/2 - i + 1, i + 1\} \cdot \frac{1}{\mu} \cdot \frac{1}{en}$$

if the population contains an individual  $x$  with  $|x'|_1 = i - 1$  or  $|x'|_1 = i + 1$ . Note that such an  $i$  always exists as long as an  $\varepsilon$ -approximation has not been obtained.

Therefore, we can sum up the waiting times for the different values of  $i$ , which leads to an upper bound of

$$\sum_{i=0}^{n/2} \frac{\mu \cdot en}{\min\{n/2 - i + 1, i + 1\}} = O(\mu n \log n)$$

for the time needed to obtain an  $\varepsilon$ -approximation of  $LF'$ .  $\square$

#### 4. COMPARISON OF ADDITIVE $\delta$ -DOMINANCE WITH THE DENSITY ESTIMATOR

In the previous section, we have shown that both diversity mechanisms may help to achieve a good approximation of an exponentially large Pareto front. The goal of this section is to work out the difference between the additive  $\delta$ -dominance approach and the use of the density estimator in a rigorous way. To do this, we examine functions that illustrate the

different behavior. Our functions are simple in the sense that they have a small Pareto front which can be easily computed by the GSEMO algorithm. However, we will show that the examined diversity mechanisms may even have difficulties to achieve a good approximation of the Pareto-optimal set.

##### 4.1 The Right Parameter for the $\delta$ -Dominance Approach

In the following, we want to point out how the choice of  $\delta$  in GDEMO influences the ability of this algorithm to achieve a good approximation. We have already seen in Section 3 that the right choice of  $\delta$  may help to achieve good approximations for an exponentially large Pareto front. Now, we illustrate that the right value of  $\delta$  is crucial for the success of the algorithm. In particular, we point out that GDEMO may fail on problems with a small Pareto front which may be easily solved by GSEMO.

For this reason, we consider the bi-objective example function  $SF'_\varepsilon(x) = (SF'_{\varepsilon,1}(x), SF'_{\varepsilon,2}(x))$  (Small Front) where

$$\begin{aligned} SF'_{\varepsilon,1}(x) &:= (|x|_1/n + \lfloor |x|_1/n \rfloor) \cdot \varepsilon \\ SF'_{\varepsilon,2}(x) &:= (\lfloor \bar{x} \rfloor_1/n + \lfloor \lfloor \bar{x} \rfloor_1/n \rfloor) \cdot \varepsilon. \end{aligned}$$

Figure 2(a) shows the objective space of  $SF'_\varepsilon$ . To obtain an  $\varepsilon$ -approximation of  $SF'$  the search points  $0^n$  and  $1^n$  have to be obtained.

The above-defined function  $SF'$  induces the same dominance relation on the search space as the similar function  $SF_\varepsilon(x) = (SF_{\varepsilon,1}(x), SF_{\varepsilon,2}(x))$  [7] with

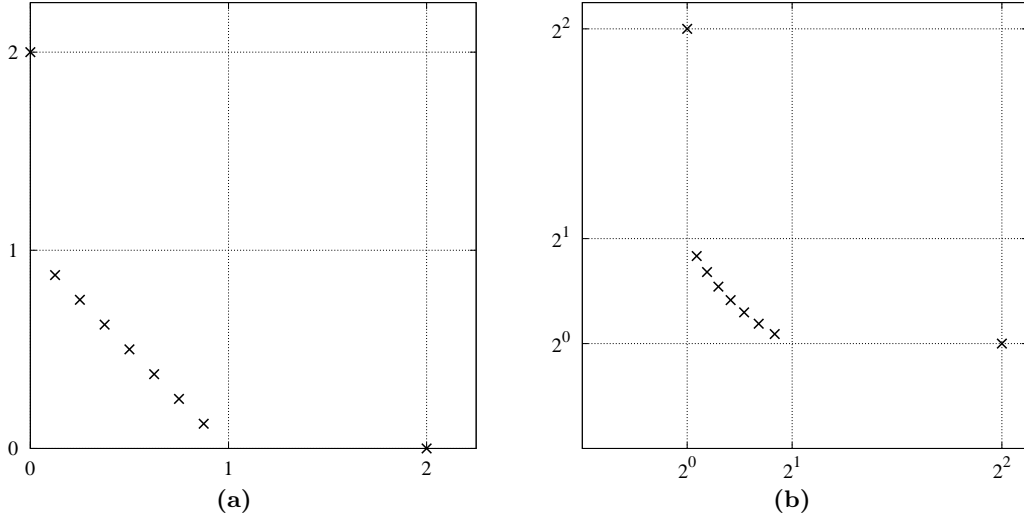
$$\begin{aligned} SF_{\varepsilon,1}(x) &:= (1 + \varepsilon)^{|x|_1/n + \lfloor |x|_1/n \rfloor} \\ SF_{\varepsilon,2}(x) &:= (1 + \varepsilon)^{\lfloor \bar{x} \rfloor_1/n + \lfloor \lfloor \bar{x} \rfloor_1/n \rfloor} \end{aligned}$$

shown in Figure 2(b). In particular,

$$SF'_\varepsilon(x) = \varepsilon \cdot \log_{1+\varepsilon}(SF_\varepsilon(x))$$

holds for each  $x \in \mathbb{B}^n$ .

All search points  $\notin \{0^n, 1^n\}$  are mapped to the box index



**Figure 2: Objective space of  $SF'_\varepsilon$  (see (a)) and  $SF_\varepsilon$  (see (b)) for  $\varepsilon = 1$  and  $n = 8$ .**

vector  $(0, 0)$  if  $\delta \geq \varepsilon$  is chosen in the GDEMO algorithm. Therefore, we can transfer the results for GSEMO on SF provided in [7] directly to  $SF'$  and state the following theorem.

**THEOREM 4.** *Choosing  $\delta \geq \varepsilon$  as box size, the time until the algorithm GDEMO has achieved an  $\varepsilon$ -approximation of  $SF'_\varepsilon$  is  $2^{\Omega(n)}$  with probability  $1 - 2^{-\Omega(n)}$ .*

The proof given in [7] relies on the phenomenon that an offspring of an individual  $x$  with  $|x|_1 < n/2$  ( $|x|_1 > n/2$ ) tends to have more (less) 1-bits inducing a drift towards the middle of the front. Since GDEMO is limited to at most one individual per box, it takes a long time to reach the outskirts of the Pareto front, which are necessary for an  $\varepsilon$ -approximation.

For the middle part of the Pareto front of  $SF'$  holds that all distances between neighboring objective vectors are equal. In addition, the objective vectors corresponding to the search points  $0^n$  and  $1^n$  have a large distance to all other objective vectors. This helps the algorithm RADEMO to achieve an  $\varepsilon$ -approximation of  $SF'$  as the density estimator enforces the algorithm to produce solutions that have a large distance in the objective space. The next theorem shows that RADEMO obtains an  $\varepsilon$ -approximation efficiently if the population size is at least 2.

**THEOREM 5.** *Choosing  $\mu \geq 2$  as population size, the algorithm RADEMO has achieved an  $\varepsilon$ -approximation of  $SF'_\varepsilon$  in expected time  $O(\mu n \log n)$ .*

*Proof.* Consider the potential  $\text{pot}(P) := \max\{|x|_1 - |y|_1| \mid x, y \in P\}$ . It holds  $0 \leq \text{pot}(P) \leq n$ . The optimization goal has been reached if  $\text{pot}(P) = n$ . Let  $x, y \in P$  be two individuals leading to the potential and assume that  $|x|_1 \leq |y|_1$ . Flipping a single 1-bit in  $x$  or a single 0-bit

in  $y$  increases the potential. Therefore the probability of increasing the potential is at least

$$\frac{1}{\mu} \cdot \left( \frac{|x|_1}{n} + \frac{n - |x|_1 - \text{pot}(P)}{n} \right) \cdot \left( 1 - \frac{1}{n} \right)^{n-1} \geq \frac{n - \text{pot}(P)}{e\mu n}.$$

The upper bound  $O(\mu n \log n)$  is the result of the summation of the expected waiting times for increasing the different potential values.  $\square$

The following theorem shows that GSEMO is able to compute the Pareto front of  $SF'_\varepsilon$  efficiently. The theorem stems from the result for GSEMO on SF provided in [7].

**THEOREM 6.** *The expected time until GSEMO has computed the Pareto front of  $SF'_\varepsilon$  is  $O(n^2 \log n)$ .*

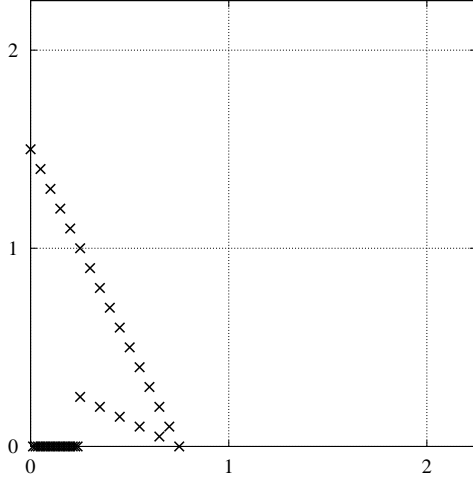
## 4.2 The Distance Measure of the Density Estimator

In the following, we showcase a simple function, which exemplifies how the diversity mechanism of RADEMO might hamper the optimization process if the population is not large enough.

Let  $SP := \{1^i 0^{n-i} \mid 0 \leq i \leq n\}$ . We consider the bi-objective example function  $\text{TF}'_\varepsilon(x) = (\text{TF}'_{\varepsilon,1}(x), \text{TF}'_{\varepsilon,2}(x))$  (Two Fronts) where

$$\text{TF}'_{\varepsilon,1}(x) := \begin{cases} |\bar{x}|_1 \cdot \varepsilon / (4n) & x \notin SP \\ \varepsilon / 4 + i \cdot 2\varepsilon / n & x = 1^i 0^{n-i}, 0 \leq i \leq n/4 \\ 3\varepsilon / 4 - (i - n/4) \cdot \varepsilon / n & x = 1^i 0^{n-i}, n/4 \leq i \leq n \end{cases}$$

$$\text{TF}'_{\varepsilon,2}(x) := \begin{cases} 0 & x \notin SP \\ \varepsilon / 4 - i \cdot \varepsilon / n & x = 1^i 0^{n-i}, 0 \leq i \leq n/4 \\ (i - n/4) \cdot 2\varepsilon / n & x = 1^i 0^{n-i}, n/4 \leq i \leq n. \end{cases}$$



**Figure 3: Objective space of  $TF'_\varepsilon$  for  $\varepsilon = 1$  and  $n = 20$ .**

Figure 3 shows the objective space of  $TF'_\varepsilon$ . The Pareto optimal set of  $TF'_\varepsilon$  is

$$\{1^i 0^{n-i} \mid n/4 \leq i \leq n\}$$

and the Pareto front consists of the objective vectors of

$$\{(3\varepsilon/4 - i \cdot \varepsilon/n, i \cdot 2\varepsilon/n) \mid 0 \leq i \leq 3n/4\}.$$

The next technical lemma, which is used in the proof of the following theorem, describes how RADEMO spreads the individuals in the population over a linear front. The lemma considers the following abstract scenario. At the beginning,  $\mu \geq 2$  tiles are placed within a discrete interval  $\{0, 1, \dots, n\}$ . In each step, a new tile is added and afterwards a most closely surrounded tile is removed. How often do we need to add a new tile next to an existing tile until the minimal distance between neighboring tiles exceeds a certain threshold  $m$ ?

**LEMMA 1.** *Let  $n \in \mathbb{N}$ ,  $I = \{0, 1, \dots, n\}$ ,  $2 \leq \mu \in \mathbb{N}$ , and  $(P_t)_{t \in \mathbb{N}}$  be a sequence of multisets  $P_t = \{p_{t,1}, \dots, p_{t,\mu}\} \subseteq I$  where  $P_{t+1} = (P_t \cup \{x\}) \setminus \{y\}$  with  $x \in I$  and*

$$y \in \arg \min_{z \in P_t \cup \{x\}} (\text{dist}_{P_t \cup \{x\}}^1(z), \text{dist}_{P_t \cup \{x\}}^2(z)).$$

*Assume w. l. o. g. that  $p_{t,i+1} \geq p_{t,i}$ . In addition, let  $D(P_t) = \{d_{t,1}, \dots, d_{t,\mu-1}\}$  be the set of all distances*

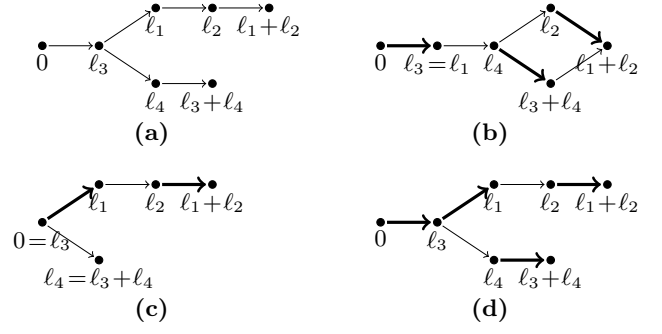
$$d_{t,i} = d(p_{t,i+1}, p_{t,i}) = p_{t,i+1} - p_{t,i}$$

*between neighboring points. Then it suffices to add successively  $O(m^3 \mu + m \mu^3)$  certain points next to a point in the actual multiset to increase the minimal distance  $\min D(P_t)$  to at least  $m$  if*

$$n > L := \sum_{i=1}^{\mu-1} (m + i - 2) = \Theta(\mu^2 + m)$$

*holds.*

*Proof.* The addition of a point  $x$  to  $P_t$  can have two consequences.



**Figure 4: Illustration for changing interval lengths. A thin arrow from  $a$  to  $b$  stands for  $a \leq b$  and a thick arrow from  $a$  to  $b$  stands for  $a < b$**

1. If  $x < p_{t,1}$  or  $x > p_{t,\mu}$ , a new interval of length  $p_{t,1} - x$  or  $x - p_{t,\mu}$  is created.
2. If  $p_{t,1} \leq x \leq p_{t,\mu}$ , an existing interval of length  $\ell_1 + \ell_2$  is divided creating two neighboring intervals of length  $\ell_1$  and  $\ell_2$ .

Thereafter two neighboring intervals of length  $\ell_3$  and  $\ell_4$  are merged creating an interval of length  $\ell_3 + \ell_4$ .

We consider the first case. If  $p_{t,\mu} - p_{t,1} \leq L$  then  $p_{t,1} > 0$  or  $p_{t,\mu} < n$ . Therefore adding  $x = p_{t,1} - 1$  or  $x = p_{t,\mu} + 1$  to  $P_t$  increases  $p_{t,\mu} - p_{t,1}$ . Hence, at most  $L + 1 = O(\mu^2 + m)$  steps are sufficient to achieve  $p_{t,\mu} - p_{t,1} > L$ .

In the second case, at most three intervals are affected. In particular,  $D(P_{t+1}) = D(P_t) \cup \{\ell_1, \ell_2, \ell_3 + \ell_4\} \setminus \{\ell_1 + \ell_2, \ell_3, \ell_4\}$  holds. We distinguish four cases where we assume w. l. o. g. that  $\ell_1 \leq \ell_2$  and  $\ell_3 \leq \ell_4$ . Note that the selection of  $y$  ensures that  $(\ell_3, \ell_4)$  is at most  $(\ell_1, \ell_2)$  with respect to the lexicographic order. The situation is illustrated in Figure 4(a).

- If  $0 = \ell_3 = \ell_1$  then  $D(P_{t+1}) = D(P_t)$ .
- If  $0 < \ell_3 = \ell_1$  then  $D(P_{t+1}) = D(P_t) \cup \{\ell_2, \ell_3 + \ell_4\} \setminus \{\ell_1 + \ell_2, \ell_4\}$  with  $|\ell_2 - (\ell_3 + \ell_4)| \leq |(\ell_1 + \ell_2) - \ell_4|$  (see Figure 4(b)).
- If  $0 = \ell_3 < \ell_1$  then  $D(P_{t+1}) = D(P_t) \cup \{\ell_1, \ell_2\} \setminus \{\ell_1 + \ell_2, \ell_3\}$  with  $|\ell_1 - \ell_2| < |(\ell_1 + \ell_2) - \ell_3|$  (see Figure 4(c)).
- If  $0 < \ell_3 < \ell_1$  then  $D(P_{t+1}) = D(P_t) \cup \{\ell_1, \ell_2, \ell_3 + \ell_4\} \setminus \{\ell_1 + \ell_2, \ell_3, \ell_4\}$  with  $\min\{\ell_1, \ell_2, \ell_3 + \ell_4\} > \min\{\ell_1 + \ell_2, \ell_3, \ell_4\}$  (see Figure 4(d)).

Consider the functions

$$f_1(P_t) := \max\{(m - 1 - \min_{1 \leq i < \mu} d_{t,i}) \cdot (\mu - 1) + |\arg \min_{1 \leq i < \mu} d_{t,i}, 0\},$$

$$f_2(P_t) := g(d_{t,1}) + \dots + g(d_{t,\mu-1})$$

where  $g(d_{t,i}) := \sum_{j=1}^{m+\mu-2-d_{t,i}} j < (m + \mu)^2$ . An inspection of the above cases shows that  $f_1(P_t)$  is monotone decreasing

and that  $f_2(P_t)$  does not increase in all cases except the last where  $f_2(P_{t+1}) - f_2(P_t)$  is

$$\begin{aligned} & (g(\ell_1) + g(\ell_2) + g(\ell_3 + \ell_4)) - (g(\ell_1 + \ell_2) + g(\ell_3) + g(\ell_4)) \\ &= (g(\ell_1) - g(\ell_1 + \ell_2)) + (g(\ell_2) - g(\ell_3)) + (g(\ell_3 + \ell_4) - g(\ell_4)) \\ &\leq g(\ell_1) - g(\ell_1 + \ell_2) \leq g(\ell_1) < (m + \mu)^2. \end{aligned}$$

Hence, the potential

$$\text{pot}(P_t) := f_1(P_t) \cdot (m + \mu)^2 + f_2(P_t)$$

is monotone decreasing. Additionally,

$$\begin{aligned} & \text{pot}(P_t) \\ &< ((m - 1) \cdot (\mu - 1) + \mu - 1) \cdot (m + \mu)^2 + (\mu - 1) \cdot (m + \mu)^2 \\ &= O(m^3 \mu + m \mu^3) \end{aligned}$$

holds.

The minimal distance  $\min D(P_t)$  is at least  $m$  if  $\text{pot}(P_t)$  is at most  $(\mu - 1) \cdot \sum_{j=1}^{\mu-2} j$ . If  $\text{pot}(P_t)$  is greater than  $(\mu - 1) \cdot \sum_{j=1}^{\mu-2} j$  then there is an  $d_{t,i} \leq m + \mu - 3$  with  $d_{t,i-1} \geq d_{t,i} + 2$  or  $d_{t,i+1} \geq d_{t,i} + 2$  due to the pigeonhole principle. Therefore adding  $x = p_{t,i} - 1$  or  $x = p_{t,i+1} + 1$  decreases the potential. Hence,  $O(m^3 \mu + m \mu^3)$  such steps are sufficient to increase the minimal distance  $\min D(P_t)$  to at least  $m$ .  $\square$

The next theorem shows that RADEMO does not achieve an  $\varepsilon$ -approximation of  $\text{TF}'_\varepsilon$  within polynomial time w. h. p. if the size of the population is not too large. The main idea of the proof is that the individuals spread out over  $\{(\varepsilon/4 + i \cdot 2\varepsilon/n, \varepsilon/4 - i \cdot \varepsilon/n) \mid 0 \leq i \leq n/4\}$  in an almost equally spaced manner before the Pareto front is reached. Thereafter RADEMO's diversity mechanism prevents the algorithm from spreading out on the Pareto front. Hence, RADEMO does not obtain the objective vectors in the top left part of the Pareto front, which are necessary to achieve an  $\varepsilon$ -approximation.

**THEOREM 7.** *Choosing  $2 \leq \mu = O(n^{1/3-c})$  as population size where  $0 \leq c \leq 1/3$  is a constant, the time until the algorithm RADEMO has achieved an  $\varepsilon$ -approximation of  $\text{TF}'_\varepsilon$  is  $2^{\Omega(n^c)}$  with probability  $1 - 2^{-\Omega(n^c)}$ .*

*Proof.* We use the method of considering a typical run of the algorithm and regard the first  $2^{\Theta(n^c)}$  steps of the algorithm. Our goal is to show that the probability of obtaining an  $\varepsilon$ -approximation within this phase is  $2^{-\Omega(n^c)}$ .

The probability of flipping at least  $i$  bits in a single step is at most

$$\binom{n}{i} \cdot \left(\frac{1}{n}\right)^i \leq \left(\frac{en}{i}\right)^i \cdot \left(\frac{1}{n}\right)^i = \left(\frac{e}{i}\right)^i.$$

Therefore, at least  $n^c$  bits flip with probability at most  $(e/n^c)^{n^c} = 2^{-\Omega(n^c \log n)}$ . This implies that the probability of flipping more than  $n^c$  bits in a single step within the considered phase is at most  $2^{-\Omega(n^c \log n)}$ .

Since the initial population  $P$  is chosen uniformly at random,  $x \notin SP$  and  $n/12 < |x|_1 < 11n/12$  hold for all  $x \in P$

with probability at least  $1 - 2^{-\Omega(n)}$ . Let  $x$  be an individual of the initial population with  $|x|_1 = i$ . This search point is uniformly distributed in  $\{y \in \mathbb{B}^n \mid |y|_1 = i\}$ . Consider an offspring  $z$  that is accepted before for the first time an individual of  $SP$  has been obtained. This individual is uniformly distributed in  $\{y \in \mathbb{B}^n \mid |y|_1 = |z|_1\}$ . Hence, the probability of obtaining a solution of  $SP$  whose number of 1-bits is at least  $n/12$  is upper bounded by  $1/\binom{n}{n/12} = 2^{-\Omega(n)}$ . It follows that  $0 \leq |x|_1 < n/12$  applies for the first individual  $x \in SP$  that is added to  $P$  w. h. p. Afterwards,  $\max_{x \in P} |x|_1 < n/12 + (\mu - 1) \cdot n^c$  holds. Hence, when the first individual  $x \in SP$  with  $n/6 < |x|_1 \leq n/6 + n^c$  is added to  $P$ ,  $P$  consists of different individuals from  $SP$ .

Thereafter, the probability to create an offspring  $y$  with  $|y|_1 > \max_{x \in P} |x|_1$  is at most  $O(\mu^{-1} n^{-1})$ . Consider a phase until an individual  $1^i 0^{n-i}$  with  $n/4 - n^c < i \leq n/4$  is added to the population. The phase involves  $\Omega(\mu n^{2-c})$  steps with probability at least  $1 - 2^{-\Omega(n^{1-c})}$  due to Chernoff bounds.

Let  $m$  be the minimal distance between two different individuals in the population. The probability to increase  $m$  to at least  $6n^c$  within  $\Omega(\mu n^{2-2c})$  consecutive steps is at least  $\Omega(1)$  due to Lemma 1 since a certain bit of a certain individual is flipped with probability at least  $\Omega(\mu^{-1} n^{-1})$ . Therefore, this event occurs at least once in the considered phase with probability at least  $1 - 2^{-\Omega(n^c)}$ . Hence, at the end of the phase  $m \geq 6n^c$  holds w. h. p.

Now, we regard the situation after an individual  $1^i 0^{n-i}$  with  $n/4 - n^c < i \leq n/4$  has been added to  $P$ . It holds  $j \leq n/4 - 6n^c$  for the nearest individual  $1^j 0^{n-j}$  of  $1^i 0^{n-i}$ . Each individual  $1^k 0^{n-k}$  with  $n/4 + 2n^c \leq k < n/4 + 3n^c$  does not dominate any individual in the population since

$$\begin{aligned} \text{TF}'_{\varepsilon,1}(1^k 0^{n-k}) &= 3\varepsilon/4 - (k - n/4) \cdot \varepsilon/n \\ &\leq 3\varepsilon/4 - (n/4 + 2n^c - n/4) \cdot \varepsilon/n \\ &= \varepsilon/4 + (n/4 - n^c) \cdot 2\varepsilon/n \\ &< \varepsilon/4 + i \cdot 2\varepsilon/n = \text{TF}'_{\varepsilon,1}(1^i 0^{n-i}) \end{aligned}$$

and

$$\begin{aligned} \text{TF}'_{\varepsilon,2}(1^k 0^{n-k}) &= (k - n/4) \cdot 2\varepsilon/n \\ &< (n/4 + 3n^c - n/4) \cdot 2\varepsilon/n \\ &= \varepsilon/4 - (n/4 - 6n^c) \cdot \varepsilon/n \\ &\leq \varepsilon/4 - j \cdot \varepsilon/n = \text{TF}'_{\varepsilon,2}(1^j 0^{n-j}). \end{aligned}$$

Therefore these individuals are rejected since accepting such an individual would decrease  $m$ . Hence, more than  $n^c$  bits have to be flipped in a single mutation step to approach the top left part of the Pareto front, which shows the theorem.  $\square$

We have seen that RADEMO does not compute an  $\varepsilon$ -approximation of  $\text{TF}'_\varepsilon$  within polynomial time w. h. p. if the size of the population is at most  $O(n^{1/3-c})$ . Note that restricting the population size to  $O(n^{1/3-c})$  does not seem to be too limited since an  $\varepsilon$ -approximation of  $\text{TF}'_\varepsilon$  can be obtained by choosing a single search point of  $\{1^i 0^{n-i} \mid n/2 \leq i \leq n\}$ .

	GSEMO	GDEMO	RADEMO
$LF'_\varepsilon$	$\searrow$	$\nearrow$	$\nearrow$
$SF'_\varepsilon$	$\nearrow$	$\searrow$	$\nearrow$
$TF'_\varepsilon$	$\nearrow$	$\nearrow$	$\searrow$

**Table 1: Overview of the performance of the algorithms GSEMO, GDEMO, and RADEMO on the functions  $LF'_\varepsilon$ ,  $SF'_\varepsilon$ , and  $TF'_\varepsilon$ .  $\searrow$  means exponential time w. h. p. and  $\nearrow$  expected polynomial time.**

We will show next that GDEMO with the right choice of  $\delta$  performs much better.

**THEOREM 8.** *Choosing  $\delta = \varepsilon$  as box size, the algorithm GDEMO has achieved an  $\varepsilon$ -approximation of  $TF'_\varepsilon$  in expected time  $O(n^3)$ .*

*Proof.* If there is no solution of  $SP$  in the population, the population size is 1 and the algorithm maximizes the number of 0-bits. Note that no steps increasing the number of 0-bits are accepted in this case as such search points are dominated by the current one constituting the population. This implies that after an expected number of  $O(n \log n)$  steps the population consists of an individual from  $SP$ . Afterwards, the individual starts a random walk on  $SP$ . The population has converged to an  $\varepsilon$ -approximation if an individual of the second box has been obtained. This happens after an expected number of  $O(n^3)$  steps (see [8]).  $\square$

The next theorem shows that the quite small Pareto front of  $TF'_\varepsilon$  can also be efficiently computed by the simple algorithm GSEMO.

**THEOREM 9.** *The expected time until GSEMO has computed the Pareto front of  $TF'_\varepsilon$  is  $O(n^3)$ .*

*Proof.* Since the population size is 1 as long as  $SP$  has not been found, the expected time until the first individual from  $SP$  is created is at most  $O(n \log n)$  following the proof ideas in [2] for the (1+1) EA and the function  $|\cdot|_1$ . As long as the Pareto front has not been found, the population size is at most  $n/4$ . The probability to create an individual  $y$  with  $|y|_1 > \max_{x \in P} |x|_1$  is therefore at least

$$\frac{1}{n/4} \cdot \frac{1}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \geq \frac{1}{n/4} \cdot \frac{1}{en}.$$

Hence, the first Pareto-optimal individual is added to the population after an expected number of at most  $n/4 \cdot n/4 \cdot en = O(n^3)$  steps. Finally, the expected time until the last Pareto-optimal search point is revealed is at most

$$\frac{3n}{4} \cdot \frac{3n}{4} \cdot en = O(n^3)$$

using similar arguments.  $\square$

## 5. DISCUSSION AND CONCLUSIONS

We have pointed out how different diversity strategies used by MOEAs can help to achieve a good approximation of the Pareto-optimal set. Table 1 gives an overview of the different results.

For problems where many search points map to different Pareto-optimal objective vectors (see function  $LF'$ ), we have shown that MOEAs that do not incorporate any diversity mechanism have difficulties to obtain a good approximation as the different individuals may only cover parts of the Pareto front that are close to each other. In contrast to this both diversity mechanisms examined in this paper lead to a good additive approximation of the Pareto-optimal set.

Afterwards, we compared the two diversity strategies. It turned out that the investigated density estimator may prevent the MOEA from obtaining new Pareto-optimal search points even if these search points are Hamming neighbors of search points already contained in the population (see function  $TF'$ ). The reason for this is that the search points already contained in the population may have a relatively large distance to each other and may not be dominated by the search points that can be introduced into the population. In such cases, the algorithm rejects the new search points of the Pareto-optimal set if their insertion into the population would lead to a worsening with respect to the distance measure. Finally, this has the effect that it is not possible to achieve a good approximation of the problem at hand.

To obtain good approximations by using the  $\delta$ -dominance approach, it is necessary to choose the right value for  $\delta$ . We have pointed out that even problems with a small Pareto front that is easy to compute without any diversity mechanism (see function  $SF'$ ) may not be optimized by the mentioned approach as it loses the information obtained by sampling Pareto-optimal search points.

The results obtained in this paper are a first step to understand how simple randomized algorithms may achieve good approximations for multi-objective optimization problems. The goal for future research is to analyze such algorithms on classical problems from multi-objective combinatorial optimization. As such problems may have Pareto fronts of exponential size diversity strategies are clearly necessary for the success of such algorithms.

## 6. REFERENCES

- [1] D. Brockhoff, T. Friedrich, N. Hebbinghaus, C. Klein, F. Neumann, and E. Zitzler. Do additional objectives make a problem harder? In *Proc. of GECCO '07*, pages 765–772. ACM Press, 2007.
- [2] S. Droste, T. Jansen, and I. Wegener. On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276(1–2):51–81, 2002.
- [3] C. M. Fonseca and P. J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1):1–16, 1995.
- [4] T. Friedrich, J. He, N. Hebbinghaus, F. Neumann, and C. Witt. Approximating covering problems by randomized search heuristics using multi-objective



- models. In *Proc. of GECCO '07*, pages 797–804. ACM Press, 2007.
- [5] O. Giel. Expected runtimes of a simple multi-objective evolutionary algorithm. In *Proc. of CEC '03*, pages 1918–1925. IEEE Press, 2003.
- [6] O. Giel and P. K. Lehre. On the effect of populations in evolutionary multi-objective optimization. In *Proc. of GECCO '06*, pages 651–658. ACM Press, 2006.
- [7] C. Horoba and F. Neumann. Benefits and drawbacks for the use of  $\varepsilon$ -dominance in evolutionary multi-objective optimization. In *Proc. of GECCO '08*, pages 641–648. ACM Press, 2008.
- [8] T. Jansen and I. Wegener. Evolutionary algorithms — how to cope with plateaus of constant fitness and when to reject strings of the same fitness. *IEEE Transactions on Evolutionary Computation*, 5(6):589–599, 2001.
- [9] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation*, 10(3):263–282, 2003.
- [10] M. Laumanns, E. Zitzler, and L. Thiele. On the effects of archiving, elitism, and density based selection in evolutionary multi-objective optimization. In *Proc. of EMO '01*, pages 181–196. Springer, 2001.
- [11] F. Neumann. Expected runtimes of a simple evolutionary algorithm for the multi-objective minimum spanning tree problem. *European Journal of Operational Research*, 181(3):1620–1629, 2007.
- [12] F. Neumann and I. Wegener. Minimum spanning trees made easier via multi-objective optimization. *Natural Computing*, 5(3):305–319, 2006.
- [13] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In *Proc. of EUROGEN '01*, pages 95–100. CIMNE, 2002.