TECHNISCHE UNIVERSITÄT DORTMUND

REIHE COMPUTATIONAL INTELLIGENCE

COLLABORATIVE RESEARCH CENTER 531

Design and Management of Complex Technical Processes and Systems by means of Computational Intelligence Methods

---

Runtime Analyses for Using Fairness in
Evolutionary Multi-Objective
Optimization

Tobias Friedrich, Christian Horoba
and Frank Neumann

No. CI-251/08

Technical Report   ISSN 1433-3325   July 2008

Runtime Analyses for Using Fairness in Evolutionary Multi-Objective Optimization

Tobias Friedrich

Algorithms and Complexity

Max-Planck-Institut für Informatik

Saarbrücken, Germany

Christian Horoba[*]

Fakultät für Informatik, LS 2

Technische Universität Dortmund

Dortmund, Germany

Frank Neumann

Algorithms and Complexity

Max-Planck-Institut für Informatik

Saarbrücken, Germany

July 21, 2008

Abstract

It is widely assumed that evolutionary algorithms for multi-objective optimization problems should use certain mechanisms to achieve a good spread over the Pareto front. In this paper, we examine such mechanisms from a theoretical point of view and analyze simple algorithms incorporating the concept of fairness introduced in [Laumanns, Thiele, Zitzler, IEEE TEC 2004]. This mechanism tries to balance the number of offspring of all individuals in the current population. We rigorously analyze the runtime behavior of different fairness mechanisms and present showcase examples to point out situations, where the right mechanism can speed up the optimization process significantly. We also indicate drawbacks for the use of fairness by presenting instances, where the optimization process is slowed down drastically.

1 Introduction

Evolutionary algorithms (EAs) evolve a set of solutions called the population during the optimization process. In multi-objective optimization one usually does not search for a single optimal solution but a set of solutions representing the possible trade-offs when dealing with

---

conflicting objective functions. Hence, multi-objective evolutionary algorithms (MOEAs) seem to be in a natural way well suited for dealing with these problems.

Many MOEAs give priority to regions in the decision or objective space that have been rarely explored. This leads to the use of fairness in evolutionary multi-objective optimization. The idea behind using fairness is that the number of offspring generated by individuals with certain properties should be balanced. Different mechanisms for spreading the individuals in the population over the Pareto front have been proposed. In NSGA-II [1] a uniform spread over the Pareto front should be achieved by using a crowded comparison operator that gives individuals in less crowded regions a higher priority. SPEA2 [11] uses a density estimator such that the fitness of an individual is given by its objective vector and a density value which depends on the other individuals in the population. The goal of the density estimator is also to give individuals in less crowded regions a higher priority. Our aim is to get a theoretical understanding how such fairness mechanisms influence the optimization process.

The theoretical understanding of the runtime behavior of MOEAs is far behind their practical success. The first rigorous runtime analyses of such algorithms have been carried out by Laumanns et al. [7] on some pseudo-Boolean functions. They have investigated a mutation-based MOEA called Simple Evolutionary Multi-objective Optimizer (SEMO) that searches locally by flipping in each mutation step a single bit. In addition, they have considered a MOEA called Fair Evolutionary Multi-objective Optimizer (FEMO) and shown that this algorithm slightly outperforms SEMO on a particular pseudo-Boolean function called *LOTZ* (*Leading Ones, Trailing Zeroes*). Giel [5] has investigated SEMO with a mutation operator that searches globally and called the algorithm Global SEMO. Global SEMO has also been considered for some well-known combinatorial optimization problems [3, 8, 9].

In this paper, we want to put forward the runtime analysis of MOEAs and consider how the use of fairness can influence the runtime behavior. We investigate the concept of fairness introduced by Laumanns et al. [7]. The implementation of this concept relies on several counters, where each individual in the population corresponds to one of these counters. The counters measure the number of offspring that the corresponding group of individuals has created. Fairness means to balance these counters to achieve that all groups have been granted the same chance to create a better individual. There are two basic ideas to link individuals with counters. The first idea is that individuals with the same decision vector share a counter and the second idea is that individuals with the same objective vector share a counter. Our goal is to compare the runtime behavior of these two variants. Additionally, we consider Global SEMO for our comparisons to examine situations, where the use of fairness in evolutionary multi-objective optimization is helpful.

The outline of this paper is as follows. A short introduction into multi-objective optimization and the algorithms that are subject of our analyses are presented in Section 2. The differences between the two variants of fairness are worked out in Sections 3 and 4. Section 5 points out situations, where both variants using fairness are not successful within polynomial time with high probability, while Global SEMO is highly efficient. Finally, we finish with some concluding remarks.

2 Algorithms

We start with some basic notations and definitions that will be used throughout the paper. We denote the set of all Boolean values by $\mathbb{B}$ and the set of all real numbers by $\mathbb{R}$ and investigate the maximization of functions $f : \mathbb{B}^n \to \mathbb{R}^m$. We call f *objective function*, $\mathbb{B}^n$

Algorithm 1 GLOBAL SEMO

> choose $x \in \mathbb{B}^n$ uniformly at random
> set $P := \{x\}$
> **repeat**
>   choose $x \in P$ uniformly at random
>   create an offspring x' by flipping each bit of x with probability $1/n$
>   **if** there is no $y \in P$ with $f(y) \succ f(x')$ **then**
>   set $P := (P \setminus \{y \in P \mid f(x') \succeq f(y)\}) \cup \{x'\}$
>   **end if**
> **until** stop

---

decision space, and $\mathbb{R}^m$ *objective space*. The elements of $\mathbb{B}^n$ are called *decision vectors* and the elements of $\mathbb{R}^m$ *objective vectors*. We define that y *weakly dominates* y', denoted by $y \succeq y'$, if and only if $y_i \geq y_i'$ for all $i \in \{1, \ldots, m\}$, and y *dominates* y', denoted by $y \succ y'$, if and only if $y \succeq y'$ and $y \neq y'$, where $y = (y_1, \ldots, y_m) \in \mathbb{R}^m$ and $y' = (y_1', \ldots, y_m') \in \mathbb{R}^m$ are two objective vectors.

The set $\mathcal{F}_f := \{y \in f(\mathbb{B}^n) \mid \nexists y' \in f(\mathbb{B}^n) \colon y' \succ y\}$ is called the *Pareto front of f* and the set $\mathcal{P}_f := f^{-1}(\mathcal{F}_f) = \{x \in \mathbb{B}^n \mid \nexists x' \in \mathbb{B}^n \colon f(x') \succ f(x)\}$ the *Pareto set of f*. The elements of $\mathcal{F}_f$ and $\mathcal{P}_f$ are called *Pareto optimal*. The set $\{(x, f(x)) \mid x \in \mathcal{P}_f\}$ constitutes the canonical solution of an optimization problem of the considered kind. In the literature a set $\{(x, f(x)) \mid x \in X\}$ with $X \subseteq \mathcal{P}_f$ is also considered as a valid solution if $f(X) = \mathcal{F}_f$. This means that it is sufficient to determine for all Pareto optimal objective vectors $y \in \mathcal{F}_f$ at least one decision vector $x \in \mathbb{B}^n$ with $f(x) = y$.

We first consider Algorithm 1 called Global Simple Evolutionary Multi-objective Optimizer (Global SEMO). It has already been discussed for the optimization of pseudo-Boolean functions [5] and different kinds of spanning tree problems [8, 9]. Global SEMO starts with an initial individual that is chosen uniformly at random from the underlying decision space. In each generation an individual x is chosen randomly from the population P to produce an offspring x' by mutation. In the mutation step each bit of x is flipped with probability $1/n$ to produce x'. After that, x' is added to P if it is not dominated by any individual in P. If x' is added to P, all individuals in P that are dominated by x' or have the same objective vector as x' are removed from P.

Laumanns et al. [7] argue that it can be beneficial when all individuals in the population have created roughly the same number of offspring and introduced an algorithm called Fair Evolutionary Multi-objective Optimizer (FEMO). This algorithm works with a local mutation operator and uses a counter for each individual in the population to measure the number of offspring the corresponding individual has created. We investigate generalized variants of FEMO. Our algorithms apply a global mutation operator and additionally accept individuals with the same objective vector as an individual in the population. The use of a global mutation operator seems more appropriate as the ability to flip two or more bits in a single mutation step is essential to escape from a local optimum. The relaxed acceptance rule also tends to improve the optimization, since it allows the exploration of plateaus, i.e., regions in the decision space whose decision vectors are mapped to the same objective vector. We distinguish two kinds of fairness depending on whether the fairness is ensured in the decision or objective space.

The algorithm Global FEMO$_{ds}$ (see Algorithm 2) maintains fairness with respect to the

---

Algorithm 2 GLOBAL FEMO$_{ds}$

---
   choose $x \in \mathbb{B}^n$ uniformly at random
   set $c(x) := 0$ {initialize offspring counter}
   set $P := \{x\}$
   repeat
      choose $x \in \{y \in P \mid c(z) \geq c(y) \text{ for all } z \in P\}$ uniformly at random
      set $c(x) := c(x) + 1$ {increment offspring counter}
      create an offspring x' by flipping each bit of x with probability $1/n$
      if there is no $y \in P$ with $f(y) \succ f(x')$ **then**
         if $x' \notin P$ **then** set $c(x') := 0$ **end if** {initialize offspring counter}
         set $P := (P \setminus \{y \in P \mid f(x') \succeq f(y)\}) \cup \{x'\}$
      end if
   until stop

---

decision space. Resetting a counter to 0 in the algorithm depends on the individuals in the current population. This implies that the algorithm forgets about counter values for decision vectors that have been seen during the optimization process but are not part of the current population. This phenomenon is of relevance if a decision vector re-enters the population which has been replaced in the meantime by another decision vector which is mapped to the same objective vector. However, we think that this is a natural way of implementing this idea of fairness as EAs are usually limited to the knowledge of the individuals that are contained in the current population. Note that Global FEMO$_{ds}$ coincides with Global SEMO [3, 9], when the counter values do not influence the search process, i.e., $c(x) = 0$ holds for each search point at each time step.

The goal in multi-objective optimization is to find the Pareto front. Thus the question arises whether it might be more beneficial to associate each counter with an objective vector rather than a decision vector, since the latter approach emphasizes the exploration of the objective space. The algorithm called Global FEMO$_{os}$ (see Algorithm 3) implements fairness in the objective space.

For our theoretical investigations carried out in the following sections, we count the number of iterations until a desired goal has been achieved. Since we are interested in the

---

Algorithm 3 GLOBAL FEMO$_{os}$

---
   choose $x \in \mathbb{B}^n$ uniformly at random
   set $c(f(x)) := 0$ {initialize offspring counter}
   set $P := \{x\}$
   repeat
      choose $x \in \{y \in P \mid c(f(z)) \geq c(f(y)) \text{ for all } z \in P\}$ uniformly at rand.
      set $c(f(x)) := c(f(x)) + 1$ {increment offspring counter}
      create an offspring x' by flipping each bit of x with probability $1/n$
      if there is no $y \in P$ with $f(y) \succ f(x')$ **then**
         if $f(x') \notin f(P)$ **then** set $c(f(x')) := 0$ **end if** {initialize offspring counter}
         set $P := (P \setminus \{y \in P \mid f(x') \succeq f(y)\}) \cup \{x'\}$
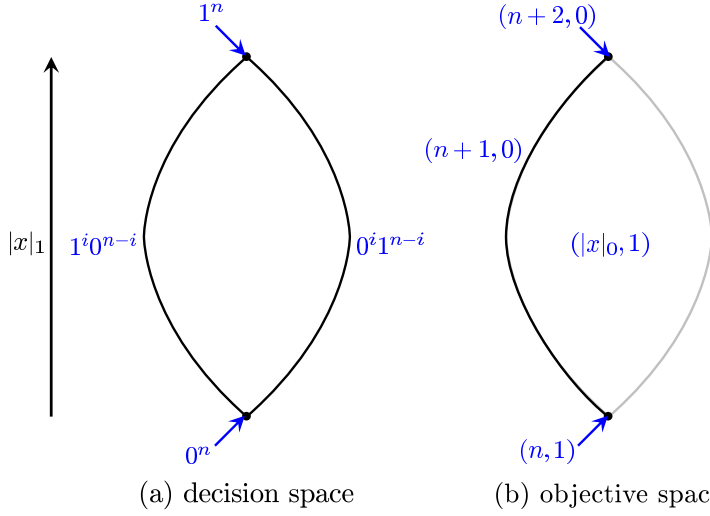   **end if**
   until stop

Figure 1: An illustration of the explored function PL. The left picture shows the decision space and the right picture shows the corresponding objective vectors.

discovery of all Pareto optimal objective vectors, we count the number of iterations until an individual for each objective vector of $\mathcal{F}_f$ has been included into the population and call it the optimization time of the algorithm. The expectation of this value is called the expected optimization time.

3 Advantages of Fairness in the Decision Space

The goal of the next two sections is to point out the differences that the use of different fairness mechanisms might have. Therefore, we examine situations, where the runtime behavior of the two variants differs significantly. To ease the notation in the following sections we will refer to the number of 0- and 1-bits in a decision vector $x \in \mathbb{B}^n$ as $|x|_0$ and $|x|_1$, respectively. We start with the examination of a situation, where Global FEMO_{ds} is efficient while Global FEMO_{os} is inefficient, and therefore investigate the bi-objective function PL (*PLateau*) [4]. The function is similar to the well-known single-objective function SPC (*Short Path with Constant values*) [6]. The function PL is illustrated in Figure 1 and defined as follows:

$$PL(x) := \begin{cases} (|x|_0, 1) & x \notin \{1^i 0^{n-i} \mid 1 \le i \le n\}, \\ (n+1, 0) & x \in \{1^i 0^{n-i} \mid 1 \le i < n\}, \\ (n+2, 0) & x = 1^n. \end{cases}$$

The function features the following properties. The decision space is partitioned into a short path $SP := \{1^i 0^{n-i} \mid 1 \le i \le n\}$ and its complement $\mathbb{B}^n \setminus SP$. The second objective of the function ensures that decision vectors from one of the mentioned sets are comparable while decision vectors from different sets are incomparable. The Pareto front of PL is $\mathcal{F}_{PL} = \{(n, 1), (n+2, 0)\}$ and the Pareto set of PL is $\mathcal{P}_{PL} = \{0^n, 1^n\}$. The set $SP \setminus \{1^n\}$ constitutes a plateau, since all decision vectors are mapped to the objective vector $(n+1, 0)$, while $\mathbb{B}^n \setminus SP$ features a richer structure. Since $PL(x) \succ PL(x')$ for $x, x' \in \mathbb{B}^n \setminus SP$ iff

$|x|_0 > |x'|_0$, the algorithms are directed to the Pareto optimal decision vector 0^n. This function has already been considered by Friedrich et al. [4] who have shown that Global SEMO is inefficient on PL. The next theorem shows that Global FEMO$_{os}$ is also not efficient on this function.

Theorem 1. *The optimization time of Global FEMO$_{os}$ on PL is $2^{\Omega(n^{1/4})}$ with probability $1 - 2^{-\Omega(n^{1/3})}$.*

Proof. We show that the decision vector 1^n is not created with probability $1 - 2^{-\Omega(n^{1/3})}$ within a phase of $2^{\Omega(n^{1/4})}$ steps. The initial individual $x \in \mathbb{B}^n$ is not in SP with probability $1 - |SP|/2^n = 1 - 2^{-\Omega(n)}$, as it is chosen uniformly at random. In addition, $|x|_1 \le (2/3)n$ holds with probability $1 - 2^{-\Omega(n)}$ using Chernoff bounds. In the remainder of the proof we consider a typical run consisting of phases of length $n^{3/2}$.

Claim 1.1 *Within the first $n^{3/2}$ steps with probability $1 - 2^{-\Omega(n^{1/3})}$, the population P never contains 1^n and at one time the population $P = \{0^n, 10^{n-1}\}$ is reached.*

Proof of Claim 1.1. The probability that a mutation flips at least i bits is upper bounded by

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

Therefore, the probability that a mutation flips at least $n^{1/3}$ bits is upper bounded by $(e/n^{1/3})^{n^{1/3}} = 2^{-\Omega(n^{1/3} \log n)}$. This implies that none of the first $n^{3/2}$ mutations flips more than $n^{1/3}$ bits with probability $1 - 2^{-\Omega(n^{1/3} \log n)}$.

The probability to create and accept an offspring x' with more 1-bits than its parent is at most $1/n$, since x is required to be in SP. Hence, the expected number of such steps is upper bounded by $n^{1/2}$. Due to Chernoff bounds this happens at most $2n^{1/2}$ times with probability $1 - 2^{-\Omega(n^{1/2})}$. Hence, the number of 1-bits increases by at most $2n^{1/2} \cdot n^{1/3} = o(n)$ which implies that the decision vector 1^n has not been found.

As at most $(1/2)n^{3/2}$ mutation trials are allocated to $c((n+1, 0))$, the individuals from $\mathbb{B}^n \setminus SP$ are chosen at least $(1/2)n^{3/2}$ times for mutation. We consider the first $(1/4)n^{3/2}$ of these mutation steps and show that the search point 0^n is included into the population. The probability that an offspring x' of an individual $x \in \mathbb{B}^n \setminus SP$ contains less 1-bits than x and is not in SP is lower bounded by $(|x|_1 - 1)/en$ if $|x|_1 \ge 2$ and $1/en$ if $|x|_1 = 1$. Therefore, the decision vector 0^n is found after an expected number of

$$en + \sum_{i=2}^{n-1} \frac{en}{i-1} \le en + en(\ln(n-2) + 1) \le en(\ln n + 2)$$

individuals from $\mathbb{B}^n \setminus SP$ have been chosen for mutation. Using Markov's inequality the probability to discover the decision vector 0^n within $2en(\ln n + 2)$ steps is at least $1/2$. Dividing $(1/4)n^{3/2}$ steps into $(1/(8en(\ln n + 2)))n^{3/2} = \Omega(n^{1/3})$ phases of length $2en(\ln n + 2)$ the decision vector 0^n is reached with probability at least $1 - 2^{-\Omega(n^{1/3})}$. The remaining $(1/4)n^{3/2}$ of these mutation steps affect 0^n. Therefore, the search point 10^{n-1} is included into the population with probability $1 - 2^{-\Omega(n^{1/2})}$ using similar arguments. $\square$

After having proven Claim 1.1, we now consider an additional phase of length $n^{3/2}$. Within this phase a search point with more than $n/2$ 1-bits is not included into the population using

6

previous arguments. Additionally, a situation is reached, where $c(n, 1) = c(n + 1, 0)$ holds. From this point of time the two individuals with the objective vectors $(n, 1)$ and $(n + 1, 0)$ are alternately selected for mutation. We consider the situation when $c(n, 1) = c(n + 1, 0)$ holds for the first time and show the following invariant to complete the proof.

Claim 1.2 *Assume that $0^n \in P$ and $\max_{x \in P} |x|_1 \le (n/2)$. Consider a non-empty phase of at most $n^{3/2}$ steps. Then with probability $1 - 2^{-\Omega(n^{1/3})}$, the population never contains 1^n and at one time a population P with $0^n \in P$ and $\max_{x \in P} |x|_1 \le (n/2)$ is reached.*

Proof of Claim 1.2. The search point 0^n will not be removed from the population once it has been included. From the proof of the previous claim, we already known that the decision vector 1^n is not obtained within a phase of $n^{3/2}$ steps with probability $1 - 2^{-\Omega(n^{1/3})}$. The decision vector 0^n is selected at least $(1/2)n^{3/2} - 1$ times for mutation within the considered phase. With probability at least $1/(en)$ such a mutation produces the search point 10^{n-1}. Hence, within the considered phase of length $n^{3/2}$ this holds with probability $1 - 2^{-\Omega(n^{1/2})}$. Having produced the search point 10^{n-1}, it replaces the previous search point of SP in the population. Hence, the assumption of the claim is fulfilled again. $\square$

We are now well-equipped to finish the proof of Theorem 1. Considering the invariant of Claim 1.2 at most $2^{n^{1/4}}$ times, Global FEMO_{os} does not create the decision vector 1^n with probability $1 - 2^{-\Omega(n^{1/3})}$. This proves Theorem 1 as all failure probabilities are bounded by $1 - 2^{-\Omega(n^{1/3})}$. $\square$

We will see that Global FEMO_{ds} performs much better on PL than its counterpart Global FEMO_{os}. The main reason for this is that after a while the Pareto optimal decision vector 0^n is prevented from generating additional offspring that can stop the random walk on the plateau.

Theorem 2. *The expected optimization time of Global FEMO_{ds} on PL is $\mathcal{O}(n^3 \log n)$.*

Proof. Before showing that Global FEMO_{ds} quickly creates the decision vectors 0^n and 1^n we summarize some results concerning PL. On one hand, the decision vector 0^n is created with probability at least $1/2$ if at least $\gamma n \log n$ individuals not from SP are chosen for mutation, where $\gamma > 0$ is a constant (see proof of Theorem 1). On the other hand, the decision vector 1^n is created with probability at least $1/2$ if at least δn^3 individuals from SP are chosen for mutation and all offspring of individuals not contained in SP do not belong to SP, where $\delta > 0$ is an appropriate constant (see [6]).

We show that the expected time until one decision vector of $\{0^n, 1^n\}$ is introduced into the population is $\mathcal{O}(n^3 \log n)$. We observe a phase of length

$$\ell := (2\gamma \log n + 1) \cdot (\delta n^3 + \gamma n \log n) = \mathcal{O}(n^3 \log n)$$

and distinguish two cases. If at least $\gamma n \log n$ individuals not from SP are chosen for mutation, the probability to find the decision vector 0^n is lower bounded by $1/2$ according to the first statement. The probability that an offspring of an individual not from SP belongs to SP is upper bounded by $1/n$. Therefore, otherwise at most $2\gamma \log n$ offspring of individuals not from SP belong to SP with probability at least $1/2$ according to Markov's inequality. Assuming that this has happened and applying the pigeonhole principle we can be sure that the phase contains a sub-phase of length

$$\delta n^3 + \gamma n \log n,$$

7

where no offspring of an individual not contained in SP belongs to SP. The mentioned sub-phase fulfills the second statement, since at least δn^3 individuals from SP are selected for mutation. Hence, the decision vector 1^n is created with probability at least $1/4$. Since the probability to create the decision vector 0^n or 1^n in a phase of length ℓ is lower bounded by $1/4$, an expected number of at most $4\ell = \mathcal{O}(n^3 \log n)$ steps suffices.

We now consider the situation, where the decision vector 0^n has been found and the decision vector 1^n is still missing. Observe a phase of length

$$\ell' := (2e \ln(2\delta n^3) + 1) \cdot (\delta n^3 + en \ln(2\delta n^3)) = \mathcal{O}(n^3 \log n).$$

If 0^n is selected at most $en \ln(2\delta n^3)$ times, then the probability that at most $2e \ln(2\delta n^3)$ offspring of 0^n are from SP is lower bounded by $1/2$ using Markov's inequality. Assuming that this has happened the phase contains a sub-phase of length

$$\delta n^3 + en \ln(2\delta n^3)$$

in which at least δn^3 individuals from SP are chosen for mutation and all offspring of the individual 0^n do not belong to SP. Hence, the probability that the missing decision vector 1^n is found or the counter value $c(0^n)$ exceeds $en \ln(2\delta n^3)$ is lower bounded by $1/4$. One of the mentioned events occurs after an expected number of most $4\ell' = \mathcal{O}(n^3 \log n)$ steps. If the individual 1^n still has not been found, we observe a phase of length $2en^2 + \delta n^3$. The probability to add a new individual from SP to the population is lower bounded by $1/(en^2)$ as at most 2 specific bits have to flip. This worst case occurs if 0^n is selected for mutation and 10^{n-1} is already contained in the population. Hence, the probability that in the first $2en^2$ steps of the phase a new individual from SP with an initial counter value of 0 is added to the population is lower bounded by $1/2$ due to Markov's inequality. Assuming that this has happened the probability that the individual 0^n is selected in the following δn^3 steps can be upper bounded as follows. The probability to reset the counter of the individual from SP is lower bounded by $1/en$. The probability that this does not happen in $en \ln(2\delta n^3)$ consecutive steps is upper bounded by

$$\left(1 - \frac{1}{en}\right)^{en \ln(2\delta n^3)} \leq e^{-\ln(2\delta n^3)} = \frac{1}{2\delta n^3}.$$

The probability that this does not happen in a phase of length δn^3 is upper bounded by $\delta n^3 \cdot 1/(2\delta n^3) \leq 1/2$. We conclude that the counter value of the actual individual from SP does not exceed $en \ln(2\delta n^3)$ with probability at least $1/2$ and therefore the individual 0^n is not chosen for mutation. Assuming that this has happened the probability that the decision vector 1^n is found is lower bounded by $1/2$. Hence, the decision vector 1^n is found in an expected number of $8 \cdot (2en^2 + \delta n^3) = \mathcal{O}(n^3)$ steps.

We also have to examine the situation that the decision vector 1^n has been found and the decision vector 0^n is still missing. We wait until the population contains an additional individual not contained in SP and the counter value $c(1^n)$ is at least as big as the counter value of this individual. Afterwards we observe a phase of length $2\gamma n \log n$. We can be sure that at least $\gamma n \log n$ steps are allocated to individuals not from SP as $c(1^n)$ is never set to 0. Hence, after an expected number of $\mathcal{O}(n \log n)$ additional steps the decision vector 0^n is added to the population. $\qquad\square$
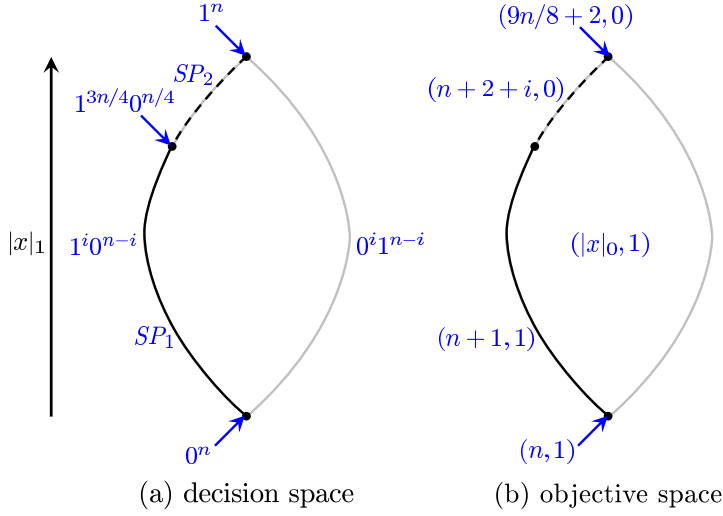
$$1^n \qquad (9n/8 + 2, 0)$$

Figure 2: An illustration of the explored function PLG. The left picture shows the decision space and the right picture shows the corresponding objective vectors.

4 Advantages of Fairness in the Objective Space

In this section, we point out situations, where the use of fairness in the objective space favors over fairness in the decision space. We have already seen that the latter fairness mechanism enables a random walk on a plateau of constant fitness, where the former fairness mechanism does not allow this kind of exploration. During the random walk the counter of the individual on the plateau is set to 0 whenever a new individual on the plateau is created. This can also be a drawback of fairness in the decision space as it might prevent the algorithm from improvements that are harder to obtain than finding a new individual on the plateau.

The function that is used to point out the mentioned behavior is similar to the function PL that has been examined in Section 3. To ease the following definition we assume $n = 8m$, $m \in \mathbb{N}$, and define

$$SP_1 := \{1^i 0^{n-i} \mid 1 \le i < 3n/4\} \quad \text{and} \quad SP_2 := \{1^{3n/4 + 2i} 0^{n/4 - 2i} \mid 0 \le i \le n/8\}.$$

The function PLG (*PLateau and Gaps*) is illustrated in Figure 2 and defined as follows:

$$PLG(x) := \begin{cases} (|x|_0, 1) & x \notin SP_1 \cup SP_2, \\ (n+1, 1) & x \in SP_1, \\ (n+2+i, 0) & x = 1^{3n/4 + 2i} 0^{n/4 - 2i}. \end{cases}$$

Note that $\mathcal{F}_{PLG} = \{(n+1, 1), (9n/8 + 2, 0)\}$ and $\mathcal{P}_{PLG} = SP_1 \cup \{1^n\}$. The short path SP is divided into a plateau and a short path with little gaps that leads to the second Pareto optimal objective vector $(9n/8 + 2, 0)$.

The next theorem shows that Global FEMO$_{os}$ performs well on PLG.

Theorem 3. *The expected optimization time of Global FEMO$_{os}$ on PLG is $\mathcal{O}(n^3)$.*

9

Proof. An individual of $SP_1 \cup SP_2$ is added to the population after an expected number of $\mathcal{O}(n \log n)$ steps, since before the achievement of such a situation the population contains one individual and the algorithm behaves like (1+1) EA on ONEMAX (see [2]).

We first consider the situation, where this individual belongs to SP_1. After an expected number of $\mathcal{O}(n^3)$ steps an individual of SP_2 is introduced into the population (see [6]). The probability to find a better individual of SP_2 under the condition that the individual of SP_2 has been selected for mutation is lower bounded by $(1/n)^2(1 - 1/n)^{n-2} \geq 1/(en^2)$, as it suffices to flip its two leftmost 0-bits. Hence, in expectation at most en^2 attempts per non-optimal individual of SP_2 are needed to improve it. The counter of the Pareto optimal individual of SP_1 is never reset. Hence, the individual of SP_2 is chosen at least once in two consecutive iterations. Therefore, an expected number of at most $2 \cdot n/8 \cdot en^2 = \mathcal{O}(n^3)$ steps is needed to obtain the missing decision vector 1^n.

In the case that the first individual of $SP_1 \cup SP_2$ belongs to SP_2 an individual of $\mathbb{B}^n \setminus SP_2$ is created with probability at least $1/e$ in a mutation step as it suffices to flip a single bit. Hence, after an expected number of $e = O(1)$ steps the population contains besides a solution of SP_2 an additional solution of $\mathbb{B}^n \setminus SP_2$. A decision vector of SP_1 is reached by allocating an expected number of $O(n \log n)$ mutation trials to the individuals of $\mathbb{B}^n \setminus SP_2$. We already know that $O(n^3)$ mutation trials allocated to the individuals of SP_2 are enough to reach the decision vector 1^n which completes the proof. $\square$

The next theorem states that Global FEMO$_{ds}$ is inefficient on *PLG*. We will see that the random walk on the plateau prevents the algorithm from following the short path to the second Pareto optimal decision vector 1^n.

Theorem 4. *The optimization time of Global FEMO$_{ds}$ on PLG is $2^{\Omega(n^{1/2})}$ with probability $1 - 2^{-\Omega(n^{1/2})}$.*

Proof. For the initial individual x holds $|x|_1 > 5n/8$ with probability $e^{-\Omega(n)}$ due to Chernoff bounds. One of the first $2^{n^{1/2}}$ mutations flips more than $n^{1/2}$ bits with probability $2^{-\Omega(n^{1/2} \log n)}$ (cf. proof of Theorem 1). We assume that these events have not happened and show that 1^n is not found within a phase of length $2^{n^{1/2}}$ with high probability.

We wait until the algorithm has generated for the first time an individual $x \in SP_2$ with $|x|_1 \geq 3n/4 + n^{1/2} - 1$. As at most $n^{1/2}$ bits flip per mutation, we can be sure that $|x|_1 \leq 3n/4 + 2n^{1/2} - 2$ holds in the next step and that the population contains an additional individual of SP_1. The probability to generate a better individual of SP_2 under the condition that the individual of SP_2 has been selected for mutation is upper bounded by $1/n^2$ since at least the two leftmost 0-bits of x have to be flipped. The probability that $n^2 - 1$ trials to find a better individual of SP_2 fail is lower bounded by $(1 - 1/n^2)^{n^2 - 1} \geq 1/e$. As at most $n^{1/2}$ bits flip per mutation, the algorithm is at least

$$\frac{n/4 - 2n^{1/2} + 2}{n^{1/2}} = \frac{n^{1/2}}{4} - 2 + \frac{2}{n^{1/2}} \geq \frac{n^{1/2}}{8}$$

times in the above situation. Hence, the probability that there is an individual $x^* \in SP_2$ for which the first $n^2 - 1$ trials to find a better individual of SP_2 fail is at least

$$1 - \left(1 - \frac{1}{e}\right)^{n^{1/2}/8} \geq 1 - 2^{-\Omega(n^{1/2})}.$$

We upper bound the counter value of the individual of SP_1 which shows that the algorithm is not able to find an individual with more 1-bits than x^*. Note that there is at least one Hamming neighbor for the individual of SP_1 that is mapped to the same objective vector. Hence, the probability to reset the counter value of the individual of $P \cap SP_1$ is lower bounded by $1/en$. Therefore, the probability that the counter value of an individual of SP_1 reaches n^2 is upper bounded by

$$\left(1 - \frac{1}{en}\right)^{n^2-1} = \left(1 - \frac{1}{en}\right)^{en \cdot n/e} \cdot \frac{en}{en-1} \le e^{-n/e} \cdot \frac{en}{en-1} = 2^{-\Omega(n)}.$$

As the probability that this happens in the observed phase is upper bounded by $2^{n^{1/2}} \cdot 2^{-\Omega(n)} = 2^{-\Omega(n)}$, the statement of the theorem follows. □

5 Drawbacks of Fairness

The goal of this section is to point out the drawbacks that our fair selection mechanisms might have in comparison to the uniform selection. In particular, we are interested in describing situation, where the use of fairness slows down the optimization process significantly.

The function that is used to point out such situations is based on paths through the Boolean hypercube. The main idea is that two paths lead to the Pareto front: a long one and a short one with little gaps. Global SEMO is able to follow the short path and reaches the Pareto front in expected polynomial time. Due to their fairness mechanisms Global FEMO_{ds} and Global FEMO_{os} are not able to follow the short path. Hence, the last-mentioned algorithms have to follow the long path to reach the Pareto front which requires an exponential number of steps.

Before we can define the function DP (Dual Path), we have to define long paths through the Boolean hypercube based on a similar definition that can be found in [10]. These paths have been used to construct a unimodal function that serves as an example on which $(1+1)$ EA needs an exponential number of steps to find the global optimum [2].

Definition 1. *Let $n \in \mathbb{N}$ and $k \in \mathbb{N}$ with $n/k \in \mathbb{N}$. The long k-path P_k^n of dimension n is a finite sequence of bit-strings of length n. The long k-path P_k^k of dimension k is defined as*

$$(0^k, 0^{k-1}1^1, \ldots, 0^1 1^{k-1}, 1^k).$$

Let $P_k^{ik} = (p_1, \ldots, p_\ell)$. The long k-path $P_k^{(i+1)k}$ of dimension $(i+1)k$ is defined as

$$(0^k p_1, \ldots, 0^k p_\ell, 0^{k-1}1^1 p_\ell, \ldots, 0^1 1^{k-1} p_\ell, 1^k p_\ell, \ldots, 1^k p_1).$$

To illustrate the last definition we present the long 2-path P_2^4 of dimension 4:

$$P_2^4 = (0000, 0001, 0011, 0111, 1111, 1101, 1100).$$

The long k-path $P_k^n = (p_1, \ldots, p_\ell)$ of dimension n features the following basic properties. By definition all points on the path are different. The length ℓ of P_k^n is given by

$$\ell = (2^{n/k} - 1)k + 1.$$

The last statement can be proven by induction on k. The following statement highlights the neighborhood structure of the points on P_k^n with respect to the Hamming distance $H(\cdot, \cdot)$. Let $p_i \in P_k^n$ and $p_j \in P_k^n$ be two points on P_k^n. For the Hamming distance of p_i and p_j holds $H(p_i, p_j) = |i - j|$ if $|i - j| < k$ and $H(p_i, p_j) \geq k$ if $|i - j| \geq k$. The last statement which can also be proven by induction on k implies that following the path by flipping less than k bits per step it is impossible to take shortcuts.

In addition, we define a short path with little gaps.

Definition 2. *Let $n \in \mathbb{N}$ with $n/2 \in \mathbb{N}$. The* short path Q^n *of dimension n is a finite sequence of bit-strings of length n. The* short path Q^n *of dimension n is defined as*

$$(0^n, 1^2 0^{n-2}, \ldots, 1^{n-2} 0^2, 1^n).$$

To illustrate the last definition, we present the short path Q^4 of dimension 4:

$$Q^4 = (0000, 1100, 1111).$$

In the following, we assume $n = 2(2m)^2$, $m \in \mathbb{N}$. We partition the bit-string $x = (x_1, \ldots, x_n) \in \mathbb{B}^n$ into 2 blocks $(x_1, \ldots, x_{n/2})$ and $(x_{n/2+1}, \ldots, x_n)$ of length $n/2$ and use the symbol $\circ$ to denote the concatenation of bit-strings. In the function definition we resort to the long $\sqrt{n/2}$-path

$$P_{\sqrt{n/2}}^{n/2} = (p_1, \ldots, p_\ell)$$

of dimension $n/2$, the short path

$$Q^{n/2} = (q_1, \ldots, q_{\ell'})$$

of dimension $n/2$, and a weighted ZeroMax function

$$z(x) := n^3 - n^2 \sum_{i=1}^{\sqrt{n/2}} x_i - \sum_{i=\sqrt{n/2}+1}^{n/2} x_i - n \sum_{i=n/2+1}^{n} x_i$$

to simplify the proof of Theorem 6 below. Furthermore, we call the bit-strings contained in the set

$$W := \{p_i \circ q_1 \mid 1 < i \leq \ell\} \cup \{p_1 \circ q_j \mid 1 < j \leq \ell'\}$$

well-formed. Now, we define the bi-objective function DP that is illustrated in Figure 3:

$$DP(x) := \begin{cases} (z(x)/n^3, 2 + z(x)/n^3) & x \notin W, \\ (1 + i/\ell, 1 + i/\ell) & x = p_i \circ q_1, 1 < i < \ell, \\ (2 + j/\ell', j/\ell') & x = p_1 \circ q_j, 1 < j < \ell', \\ (3, 3) & x = p_\ell \circ q_1 \vee x = p_1 \circ q_{\ell'}. \end{cases}$$

Note that two decision vectors of one of the sets $\mathbb{B}^n \setminus W$, $\{p_i \circ q_1 \mid 1 < i < \ell\}$, and $\{p_1 \circ q_j \mid 1 < j < \ell'\}$ are comparable, whereas two decision vectors of different sets are incomparable. We also point out that the decision vectors $p_\ell \circ q_1$ and $p_1 \circ q_{\ell'}$ dominate all other decision vectors. Hence, $\mathcal{F}_{DP} = \{(3, 3)\}$ and $\mathcal{P}_{DP} = \{1^{\sqrt{n/2}} 0^{n-\sqrt{n/2}}, 0^{n/2} 1^{n/2}\}$.

Before we state and prove our theorems, we explain the function DP informally. All
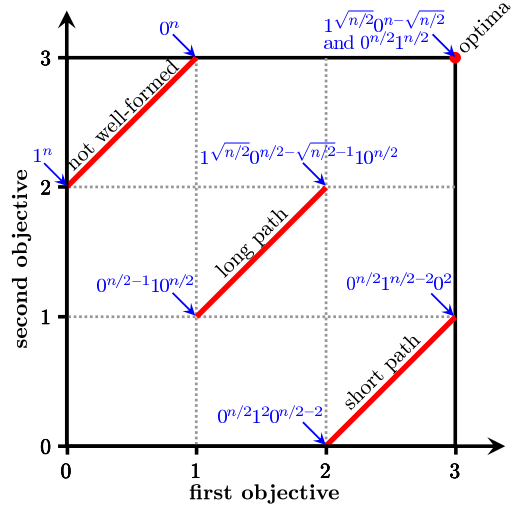
Figure 3: An illustration of the explored function DP. The possible objective vectors are shown in red. Some exemplary decision vectors are marked in blue.

decision vectors that are not well-formed, i.e., not of the form $p_i \circ q_1$, $1 < i \leq \ell$, or $p_1 \circ q_j$, $1 < j \leq \ell'$, are mapped on the line $(0, 2) + \lambda(1, 1)$, $0 \leq \lambda \leq 1$, in the objective space. In this case, a decision vector with less 1-bits than another decision vector dominates the latter one. After a while, the decision vector $0^n = p_1 \circ q_1$ is created. From now on, the considered algorithms create decision vectors of the form $p_i \circ q_1$, $1 < i < \ell$, or $p_1 \circ q_j$, $1 < j < \ell'$. In these cases, a decision vector that is located closer to the end of the respective path than another decision vector dominates the latter one. The function definition guarantees that the two types of decision vectors are incomparable, as they are mapped on the line $(1, 1) + \lambda(1, 1)$, $0 < \lambda < 1$, and $(2, 0) + \lambda(1, 1)$, $0 < \lambda < 1$, respectively. As a consequence, the considered algorithms follow both paths in parallel until one of the decision vectors $p_\ell \circ q_1$ and $p_1 \circ q_{\ell'}$ is found. The properties of both paths are complementary to each other: The first path is long while the second one is short and it is relatively easy to find a better decision vector on the first path while it is relatively hard to find a better one on the second path.

The following theorems show that the optimization time can increase significantly by using a fair selection rule instead of a uniform selection rule. We begin with a theorem that shows that Global SEMO is able to optimize DP efficiently.

Theorem 5. *The expected optimization time of Global SEMO on DP is $\mathcal{O}(n^3)$.*

Proof. We upper bound the time needed to create the individual $0^{n/2}1^{n/2}$. If all individuals in the population are well-formed, then exactly one half of the decision vector of each individual in the population equals $0^{n/2}$. Since the probability to flip a single bit in this half is lower bounded by

$$\frac{n}{2} \cdot \frac{1}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \geq \frac{1}{2e},$$

we have to wait an expected number of at most $2e = \mathcal{O}(1)$ steps until a not well-formed individual is introduced into the population. Let x be the not well-formed individual in the

population. If $|x|_1 > 0$, then it is always possible to flip a single 1-bit of x to create a not well-formed individual x' with $|x'|_1 = |x|_1 - 1$. Since the population size is always upper bounded by 3, the probability to create such an individual is at least

$$\frac{1}{3} \cdot \frac{1}{n} \cdot \left(1 - \frac{1}{n}\right)^{n-1} \geq \frac{1}{3en}.$$

Hence, we have to wait an expected number of at most $n \cdot 3en = 3en^2 = \mathcal{O}(n^2)$ steps until the individual 0^n is introduced into the population. The probability to introduce a new individual of the form $p_1 \circ q_j$, $1 < j \leq \ell'$, into the population is lower bounded by

$$\frac{1}{3} \cdot \left(\frac{1}{n}\right)^2 \cdot \left(1 - \frac{1}{n}\right)^{n-2} \geq \frac{1}{3en^2},$$

since a particular individual $x \in P$ with $x = p_1 \circ q_j$, $1 \leq j < \ell'$, has to be chosen for mutation and its two leftmost 0-bits have to be flipped. Hence, we have to wait an expected number of at most $n/4 \cdot 3en^2 = 3en^3/4 = \mathcal{O}(n^3)$ steps until the individual $0^{n/2}1^{n/2}$ is introduced into the population. $\square$

The next theorem shows that a fair selection mechanism can slow down the optimization process drastically.

Theorem 6. *The optimization time of Global FEMO$_{ds}$ and Global FEMO$_{os}$ on DP is $2^{\Omega(n^{1/2})}$ with probability $1 - 2^{-\Omega(n^{1/2})}$.*

Proof. Let $k := \sqrt{n/2}$. For the initial individual x we have $\sum_{i=1}^{k}(1 - x_i) < k/4$ with probability $2^{-\Omega(n^{1/2})}$ and $\sum_{i=n/2+1}^{n} x_i > 3n/8$ with probability $2^{-\Omega(n)}$ due to Chernoff bounds. The probability that one of the first $2^{\sqrt{n/2}}$ mutations flips at least $k/4$ bits is upper bounded by $2^{-\Omega(n^{1/2}\log n)}$ (see proof of Theorem 1). We assume that these events have not happened.

We consider a phase of length $2^{\sqrt{n/2}}$ and show that Global FEMO$_{ds}$ and Global FEMO$_{os}$ do not reach the Pareto front with high probability. The weights in the definition of z ensure that $\sum_{i=1}^{k}(1 - x_i) \geq k/4$ and $\sum_{i=n/2+1}^{n} x_i \leq 3n/8 + 3k/4$ always hold for the individual $x \in P \setminus W$. Since less than $k/4$ bits flip per mutation, the first k bits of p_i and the bits of q_j of a well-formed offspring $p_i \circ q_j$ of the individual $x \in P \setminus W$ contain more than 0 0-bit and less than $3n/8 + 3k/4 + k/4$ 1-bits, respectively. Hence, we can be sure that p_i and q_j are not located too close to the end of the respective path. More precisely, the indices i and j can be upper bounded by

$$i \leq \frac{\ell+1}{2} \quad \text{and} \quad j < \left\lfloor \frac{1}{2} \cdot \left(\frac{3n}{8} + \frac{3k}{4} + \frac{k}{4}\right)\right\rfloor + 1 \leq \frac{7n}{32}.$$

The Pareto front can be reached via the short path or via the long one. The traversal of the long path is impossible, since its second half consists of $(\ell-1)/2$ points and no shortcuts can be taken, since less than k bits flip per mutation, and therefore

$$\frac{(\ell-1)/2}{k/4-1} = \frac{(2^{\sqrt{2n}}-1)\sqrt{n/2}/2}{\sqrt{n/2}/4-1} > 2^{\sqrt{n/2}}$$

14

advances are needed which exceeds the length of the considered phase.

Although the traversal of the short path is possible it is unlikely. The probability to progress on the short path is upper bounded by $1/n^2$, since at least the 2 leftmost 0-bits of the so far best individual on the short path have to flip. As at least $(n/32)/(k/4-1)$ advances on the short path are necessary and $n^2 - 1$ attempts to generate an onward individual on the path fail with probability $(1 - 1/n^2)^{n^2-1} \geq 1/e$, one individual on the short path fails for $n^2 - 1$ times to create a better individual on the path with probability

$$1 - \left(1 - \frac{1}{e}\right)^{(n/32)/(k/4-1)} = 1 - 2^{-\Omega(n^{1/2})}.$$

We show that the mentioned individual never gets an additional chance to create a better offspring. It can only get a n^2-th chance if the lowest counter value of an individual in the population reaches $n^2 - 1$. In the considered phase the population always contains an individual with a better Hamming neighbor. The probability to reach this Hamming neighbor is lower bounded by $1/en$. Hence, the probability that such an individual fails for $n^2 - 1$ times to generate a better individual with an initial counter value of 0 is at most

$$\left(1 - \frac{1}{en}\right)^{n^2-1} \leq e^{-n/e} \cdot \frac{en}{en-1} = 2^{-\Omega(n)}.$$

Therefore, the probability that the lowest counter value in a phase of length $2^{\sqrt{n/2}}$ reaches $n^2 - 1$ is upper bounded by $2^{-\Omega(n)}$. Altogether, the statement of the theorem follows. $\square$

6 Conclusions

Popular variants of MOEAs such as NSGA-II or SPEA2 use specific modules to explore the Pareto front of a given problem by favoring solutions belonging to regions in the decision or objective space that are rarely covered. With this paper, we have taken a first step to understand such mechanisms by rigorous runtime analyses. We have shown that there are simple plateau functions which cannot be optimized without fairness or with fairness in the objective space, but with a MOEA which implements fairness in the decision space (cf. Section 3). We also proved that for certain "perforated" plateaus the impact of fairness can be the other way around (cf. Section 4). Our analyses point out that a fair MOEA has a marked preference for accepting quick small improvements. This can help to find new solutions close to the current population quicker. On the other hand, this effect can prevent a fair MOEA from exploring parts of the Pareto front that are harder to find (cf. Section 5). Hence, depending on the function the use of fairness significantly influences the runtime behavior of MOEAs.

After having gained some insight into MOEAs using fairness mechanisms, future work should concentrate on analyzing the mechanisms of NSGA-II or SPEA2. Analyses for these mechanisms seem to be more complicated. However, we think that it may be possible to point out the advantages and disadvantages of these mechanisms (in combination with simplified algorithms) by rigorous analyses. These insights hopefully contribute to the design of improved algorithms.

References

[1] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *Proc. of the International Conference on Parallel Problem Solving From Nature (PPSN VI)*, volume 1917 of *LNCS*, pages 849–858. Springer, 2000.

[2] S. Droste, T. Jansen, and I. Wegener. On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276:51–81, 2002.

[3] 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 the annual Conference on Genetic and Evolutionary Computation (GECCO '07)*, volume 1, pages 797–804. ACM Press, 2007.

[4] T. Friedrich, N. Hebbinghaus, and F. Neumann. Plateaus can be harder in multi-objective optimization. In *Proc. of the IEEE Congress on Evolutionary Computation (CEC '07)*, pages 2622–2629. IEEE Press, 2007.

[5] O. Giel. Expected runtimes of a simple multi-objective evolutionary algorithm. In *Proc. of the IEEE Congress on Evolutionary Computation (CEC '03)*, pages 1918–1925. IEEE Press, 2003.

[6] 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.

[7] M. Laumanns, L. Thiele, and E. Zitzler. Running time analysis of multiobjective evolutionary algorithms on pseudo-boolean functions. *IEEE Transactions on Evolutionary Computation*, 8(2):170–182, 2004.

[8] 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.

[9] F. Neumann and I. Wegener. Minimum spanning trees made easier via multi-objective optimization. *Natural Computing*, 5(3):305–319, 2006.

[10] G. Rudolph. *Convergence properties of evolutionary algorithms.* Number 35 in Forschungsergebnisse zur Informatik. Verlag Dr. Kovač, Hamburg, 1997.

[11] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In *Proc. of EUROGEN 2001*, pages 95–100. International Center for Numerical Methods in Engineering (CIMNE), 2002.