Runtime Analyses for Using Fairness in Evolutionary Multi-Objective Optimization

Tobias Friedrich[1], Christian Horoba[2], and Frank Neumann[1]

[1] Max-Planck-Institut für Informatik, Saarbrücken, Germany
[2] Fakultät für Informatik, LS 2, TU Dortmund, Dortmund, Germany

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 by Laumanns et al [7]. This mechanism tries to balance the number of descendants of all individuals of the current population. We rigorously analyze the runtime behavior of different fairness mechanism and present showcase examples to point out situations where the right mechanism can speed up the optimization process significantly.

1 Introduction

Evolutionary algorithms (EAs) evolve a set of solutions called the population during the optimization process. As 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, 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 descendants generated by individuals with certain properties should be balanced. Different mechanisms for spreading the individuals of a 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 decision vectors in less crowded regions a higher priority. SPEA2 [10] 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 for such algorithms have been carried out by Laumanns et al. [7]. They have investigated a simple mutation-based MOEA called SEMO that searches locally by flipping in each mutation step a single bit. In addition, they have considered a fair MOEA called FEMO and shown that this algorithm outperforms SEMO on a particular pseudo Boolean function called LOTZ.

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 model of fairness introduced by Laumanns et al. [7]. The algorithms that are subject to our analyses count the number of descendants that have been generated by the individuals in the population. The first idea is to count the number of descendants with respect to the decision space, i. e., a separate counter is dedicated to each decision vector. The descendants are generated by individuals that have not produced many descendants in order to discover new regions of the decision space. This prevents individuals that have achieved less progress towards other non-dominated decision vectors from producing additional descendants. The other idea we examine is the usage of a counter with respect to the objective space. This implies that many decision vectors potentially depend on the same counter. Our goal is to compare the runtime behavior of these two variants.

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. 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 $f\colon \mathbb{B}^n \rightarrow \mathbb{R}^m$. We call f *objective function*, $\mathbb{B}^n$ *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*. Let $y = (y_1, \ldots, y_m) \in \mathbb{R}^m$ and $y' = (y'_1, \ldots, y'_m) \in \mathbb{R}^m$ be two 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 $y' \not\succeq y$.

The set $\mathcal{PF}(f) := \{y \in f(\mathbb{B}^n) \mid \forall y' \in f(\mathbb{B}^n)\colon y' \not\succ y\}$ is called the *Pareto front of* f and the set $\mathcal{P}(f) := f^{-1}(\mathcal{PF}(f)) = \{x \in \mathbb{B}^n \mid \forall x' \in \mathbb{B}^n\colon f(x') \not\succ f(x)\}$ *Pareto set of* f. 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 of the form $\{(x, f(x)) \mid x \in X\}$ with $X \subseteq \mathcal{P}(f)$ is also considered as a valid solution if $f(X) = \mathcal{PF}(f)$. This means that it is sufficient to determine for all non-dominated objective vectors $y \in \mathcal{PF}(f)$ at least one decision vector $x \in \mathbb{B}^n$ with $f(x) = y$.

Laumanns et al. [7] al. argue that it can be beneficial when all individuals in the population have created roughly the same number of descendants and therefore introduced the algorithm FEMO. This algorithm works with a local mutation operator and uses a counter for each individual in the population to measures the number of descendants 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 is more common as the ability to flip two or more bits in a single mutation step tends to improve the optimization process. 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 following Global FEMO$_{ds}$ uses fairness in the decision space.

Algorithm 1 *Global FEMO$_{ds}$*

1. *Choose* $x \in \mathbb{B}^n$ *uniformly at random.*
2. *Set* $c(x) := 0$.
3. *Set* $P := \{x\}$.
4. *Repeat*
   - *Choose* $x \in \{y \in P \mid c(z) \geq c(y)$ *for all* $z \in P\}$ *uniformly at random.*
   - *Set* $c(x) := c(x) + 1$.
   - *Create a descendant* 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* $c(x') := 0$.
     - *Set* $P := (P \setminus \{y \in P \mid f(x') \succeq f(y)\}) \cup \{x'\}$.

Note that the algorithm resets the counter to 0 depending on the individuals in the current population. This implies that it forgets about counter values for decision vectors that have been seen during the optimization process but are not part of the current population. 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. Global FEMO$_{ds}$ collapses to the Global SEMO algorithm [3, 9] in the case that the counter value does 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 explore the objective space. Therefore, the question arises whether it might be more beneficial to associate each counter with an objective vector rather than a decision vector. The following algorithm implements fairness in the objective space.

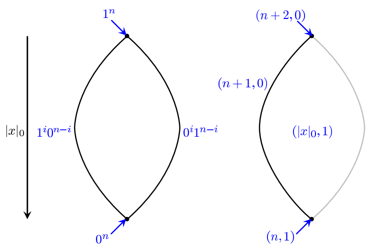
Algorithm 2 *Global FEMO$_{os}$*

1. *Choose* $x \in \mathbb{B}^n$ *uniformly at random.*
2. *Set* $c(f(x)) := 0$.
3. *Set* $P := \{x\}$.
4. *Repeat*
   - *Choose* $x \in \{y \in P \mid c(f(z)) \geq c(f(y))$ *for all* $z \in P\}$ *uniformly at random.*
   - *Set* $c(f(x)) := c(f(x)) + 1$.
   - *Create a descendant* 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* $c(f(x')) := 0$.
     - *Set* $P := (P \setminus \{y \in P \mid f(x') \succeq f(y)\}) \cup \{x'\}$.

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 discovery of all non-dominated objective vectors, we count the number of iterations until an individual for each objective vector of $\mathcal{PF}(f)$ 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 by examining situations where Global FEMO$_{ds}$ is efficient while Global FEMO$_{os}$ is inefficient. For this, we define a bi-objective function *PL* which is similar to the well-known single-objective function *SPC* [6] and has been introduced in [4]. It is illustrated in the right figure and defined as follows:



(a) decision space (b) objective space

$$PL(x) := \begin{cases} (|x|_0, 1) & x \notin \{1^i0^{n-i} \mid 1 \le i \le n\}, \\ (n+1, 0) & x \in \{1^i0^{n-i} \mid 1 \le i < n\}, \\ (n+2, 0) & x = 1^n. \end{cases}$$

The function features the following properties. The decision space is partitioned into a short path $SP := \{1^i0^{n-i} \mid 1 \le i \le n\}$ 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 $\mathcal{PF}(PL) = \{(n,1), (n+2,0)\}$ and the Pareto set of *PL* is $\mathcal{P}(PL) = \{0^n, 1^n\}$. $SP \setminus \{1^n\}$ 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 neighborhood structure that gives hints directing to the non-dominated 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 lower bounded by $2^{\Omega(n^{1/2-\varepsilon})}$ with probability $1 - 2^{-\Omega(n^{1/2-\varepsilon})}$ for all constants $\varepsilon > 0$.*

Proof. We show that the decision vector 1^n is not produced within a phase of length $2^{n^{1/2-\varepsilon}}$ with high probability. As the initial individual $x \in \mathbb{B}^n$ is chosen uniformly at random, it does not belong to *SP* with probability $1 - |SP|/2^n = 1 - 2^{-\Omega(n)}$. In addition, $|x|_1 \leq 3n/4$ holds with probability $1 - 2^{-\Omega(n)}$ using Chernoff bounds. The probability that a mutation flips at least i bits is upper bounded by

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

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

We show that the decision vector 1^n is not found while the decision vector 0^n is found within a phase of length $4en^{3/2}(\ln n + 2)$ with high probability. The probability to produce and accept a decision vector $x' \in \mathbb{B}^n$ with $|x'|_1 > \max_{y \in P} |y|_1$ is upper bounded by $1/n$, since this event requires $x' \in$ *SP*. Hence, we expect this to happen at most $4en^{1/2}(\ln n + 2)$ times. Due to Chernoff bounds this happens at most $6en^{1/2}(\ln n + 2)$ times with probability $1 - 2^{-\Omega(n^{1/2} \log n)}$. At most $n^{1/2-\varepsilon}$ bits flip per mutation. Hence,

$$\max_{y \in P} |y|_1 \leq 3n/4 + 6en^{1/2}(\ln n + 2) \cdot n^{1/2-\varepsilon} = 3n/4 + 6en^{1-\varepsilon}(\ln n + 2) < n$$

holds at the end of the phase implying that the decision vector 1^n has not been found. Since at most $2en^{3/2}(\ln n + 2)$ mutation trials are allocated to $c((n + 1, 0))$, the individuals from $\mathbb{B}^n \setminus$ *SP* are chosen at least $2en^{3/2}(\ln n + 2)$ times for mutation. The probability that a descendant x' of an individual $x \in \mathbb{B}^n \setminus$ *SP* contains less 1-bits than x and does not belong to *SP* is lower bounded by $(|x|_1 - 1)/en$ if $|x|_1 \geq 2$ and $1/en$ if $|x|_1 = 1$. Therefore, we expect that the decision vector 0^n has been found having

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

mutation trials allocated to individuals from $\mathbb{B}^n \setminus$ *SP*. Using Markov's inequality the probability to discover the decision vector 0^n within $2en(\ln n + 2)$ steps is at least $1/2$. Considering $2en^{3/2}(\ln n + 2)$ steps organized into $n^{1/2}$ phases of length $2en(\ln n + 2)$ the decision vector 0^n is reached with probability at least $1 - 2^{n^{1/2}}$.

We are now in the situation right after the individual 0^n has been added to the population. If the population contains an individual of *SP*, we wait until $c((n, 1)) = c((n + 1, 0))$. Otherwise we wait until such an individual is added to the population. This happens in $en^{3/2}$ steps with probability $1 - 2^{n^{1/2}}$. Afterwards we wait until $c((n + 1, 0)) = c((n, 1))$. Note, that in this waiting phase of length at most $en^{3/2}$ the individual 1^n is not produced with probability $1 - 2^{-\Omega(n^{1/2})}$ using to same argumentation as above.

In this situation we examine a phase of length $2en^{3/2}$ and conclude that the random walk on *SP* does not reach the decision vector 1^n with probability $1 - 2^{-\Omega(n^{1/2})}$ resorting once again to above arguments. We can be sure that the decision vector 0^n is selected $en^{3/2}$ times for mutation, since the mutation trials are equally divided between $c((n, 1))$ and $c((n + 1, 0))$. Hence, within such a phase the decision vector 0^n produces the decision vector 10^{n-1} with probability $1-2^{-n^{1/2}}$ which implies that the random walk on *SP* has to start again. Dividing a phase of length $2^{n^{1/2-\varepsilon}}$ into $2^{n^{1/2-\varepsilon}}/(2en^{3/2})$ such sub-phases gives that Global FEMO*os* does not produce the decision vector 1^n with probability $1-2^{-\Omega(n^{1/2})}$ which completes the proof. $\square$

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 descendants that can stop the random walk on the plateau.

Theorem 2. *The expected optimization time of Global FEMOds 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 the one hand, the decision vector 0^n is created with probability at least $1/2$ if at least $cn \log n$ individuals not from *SP* are chosen for mutation, where $c > 0$ 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 $c'n^3$ individuals from *SP* are chosen for mutation and all descendants of individuals not contained in *SP* do not belong to *SP*, where $c' > 0$ is an appropriate constant (see [6]).

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

$$\ell := (2c \log n + 1) \cdot (c'n^3 + cn \log n)$$

and distinguish two cases. If at least $cn \log n$ 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. Otherwise at most $2c \log n$ descendants of individuals not from *SP* lead to individuals of *SP* with probability at least $1/2$ according to Markov's inequality, since the probability that a descendant of an individual not from *SP* belongs to *SP* is upper bounded by $1/n$. Assuming that this has happened and applying the pigeonhole principle we can be sure that the phase contains a sub-phase of length

$$c'n^3 + cn \log n,$$

where no descendants of individuals not contained in *SP* belong to *SP*. The mentioned sub-phase fulfills the second statement, since at least $c'n^3$ 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

$$\ell' := (2e \log(2c'n^3) + 1) \cdot (c'n^3 + en \log(2c'n^3)).$$

If the decision vector 0^n is selected at most $en \log(2c'n^3)$ times then the probability that at most $2e \log(2c'n^3)$ descendants of the decision vector 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

$$c'n^3 + en\log(2c'n^3)$$

in which at least $c'n^3$ individuals from *SP* are chosen for mutation and all descendants 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 $en\log(2c'n^3)$ is lower bounded by $1/4$. We expect that one of the mentioned events occurs after at 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 $2en^2 + c'n^3$. 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 $c'n^3$ 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 $en\log(2c'n^3)$ consecutive steps is upper bounded by

$$\left(1 - \frac{1}{en}\right)^{en\log(2c'n^3)} \leq \frac{1}{2c'n^3}.$$

The probability that this does not happen in a phase of length $c'n^3$ is upper bounded by $c'n^3 \cdot 1/(2c'n^3) \leq 1/2$. We conclude that the counter value of the actual individual from *SP* does not exceed $en\log(2c'n^3)$ 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 $8 \cdot (2en^2 + c'n^3) = \mathcal{O}(n^3)$ 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 $2cn\log n$. We can be sure that at least $cn\log n$ 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. □

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 the ability to perform a random walk on a plateau of constant fitness where the former fairness mechanism is not able to do so. During the random walk the counter of the individual on the plateau is set to 0 whenever a new individual on the plateau is produced. This can also be a drawback of fairness in the decision space as it may prevent the algorithm from improvements that are harder to obtain than finding a new individual on the plateau.

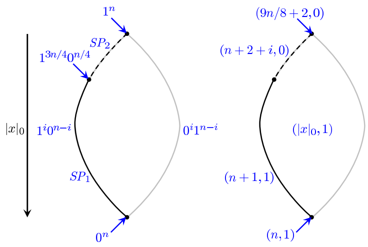
Our function that is used to point out the mentioned behavior is similar to the function *PL* that has been examined in Section 3. To ease the following definition we assume $n = 8m$, $m \in \mathbb{N}$, and define

$$SP_1 := \{1^i 0^{n-i} \mid 1 \leq i \leq 3n/4 - 1\}$$

and

$$SP_2 := \{1^{3n/4+2i} 0^{n/4-2i} \mid 0 \leq i \leq n/8\}.$$

The function *PLG* (*Plateau and gaps*) is illustrated in the figure to the right and defined by



(a) decision space (b) objective space

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

Note, that $\mathcal{PF}(PLG) = \{(n+1,1), (9n/8+2, 0)\}$ 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 non-dominated 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 $\mathcal{O}(n^3)$.*

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 having reached such a situation the population contains at most one individual and therefore 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 non-dominated 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 produced 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 non-dominated decision vector 1^n.

Theorem 4. *The optimization time of Global FEMO$_{ds}$ on PLG is lower bounded by $2^{\Omega(n^{1/2})}$ 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. The probability that one of the first $2^{n^{1/2}}$ mutations flips more than $n^{1/2}$ bits is upper bounded by $2^{-\Omega(n^{1/2}\log n)}$ (cf. proof of Theorem 1). We assume that these events have not happened. We consider a phase of length $2^{n^{1/2}}$ and show that Global FEMO$_{ds}$ does not find the decision vector 1^n 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 and 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 $(1-1/n^2)^{n^2-1} \geq 1/e$. Since 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 at least one individual $x^* \in SP_2$ for which the first n^2-1 trials to find a better individual of SP_2 fail is lower bounded by

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

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

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

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 Conclusions

Popular variants of MOEAs such as SPEA2 or NSGA-II 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.

6 Acknowledgments

The second author was supported by the Deutsche Forschungsgemeinschaft (DFG) as part of the Collaborative Research Center “Computational Intelligence” (SFB 531).

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 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. *Theor. Comput. Sci.*, 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 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 CEC ’07*, pages 2622–2629. IEEE Press, 2007.
5. O. Giel. Expected runtimes of a simple multi-objective evolutionary algorithm. In *Proc. of CEC ’03*, IEEE Press, pages 1918–1925, 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 Trans. 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 Trans. 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. In *Proc. of PPSN ’04*, volume 3242 of *LNCS*, pages 80–89, 2004.
9. F. Neumann and I. Wegener. Minimum spanning trees made easier via multi-objective optimization. In *Proc. of GECCO ’05*, pages 763–770. ACM Press, 2005.
10. 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.