Plateaus Can Be Harder in Multi-Objective Optimization

Tobias Friedrich, Nils Hebbinghaus, Frank Neumann

Department 1: Algorithms and Complexity
Max-Planck-Institut für Informatik
66123 Saarbrücken, Germany

***Abstract*—In recent years a lot of progress has been made in understanding the behavior of evolutionary computation methods for single- and multi-objective problems. Our aim is to analyze the diversity mechanisms that are implicitly used in evolutionary algorithms for multi-objective problems by rigorous runtime analyses. We show that, even if the population size is small, the runtime can be exponential where corresponding single-objective problems are optimized within polynomial time. To illustrate this behavior we analyze a simple plateau function in a first step and extend our result to a class of instances of the well-known SETCOVER problem.**

I. INTRODUCTION

Using evolutionary computation methods to solve multi-objective optimization problems has become very popular during the last ten years [3, 4]. In contrast to single-objective problems where often much more is known about the structure of a given problem, multi-objective problems seem to be more complicated and harder to understand. By increasing the number of objectives, one has to optimize several (often conflicting) functions instead of a single one. This implies that there is often not a single optimum, but a set of incomparable optima, known as the Pareto front. The number of such optima may increase with the number of objectives that are considered [20], but even optimizing only two objective functions may lead to a Pareto front that is exponential in the input size [7].

Due to the problem of computing several optima instead of a single one, multi-objective optimization is often considered as at least as difficult as single-objective optimization. However, there are examples where adding additional objectives can speed up the optimization process of a single-objective problem [2, 14]. In addition, it has been shown that some combinatorial optimization problems such as minimum spanning trees or different covering problems may be easier in a multi-objective model than in a single-objective one [8, 19]. Often it is assumed that a multi-objective model for a single-objective optimization problem should have the structure that the set of incomparable objective vectors is always small. The results obtained in [19] and [8] mainly rely on this property as it implies that the algorithms considered in these papers work with a small population size.

In this paper, we want to point out a different obstacle when using multi-objective models for single-objective optimization problems. To the best of our knowledge, there is so far no rigorous analysis of a problem on which the multi-objective approach is slower by more than a factor bounded by the population size compared to the respective single-objective one. Our aim is to show that a multi-objective model may lead to a totally inefficient optimization process (in comparison to a single-objective one) even if the population size is always small. The reason for this is that the population used to approximate the Pareto set may prevent the algorithm from obtaining optimal solutions. Evolutionary algorithms for multi-objective optimization problems such as NSGA-II [5] or SPEA2 [22] make use of different diversity mechanisms to obtain for each Pareto optimal search point a good approximation. For simple single-objective problems it has been shown in [9] that, depending on the diversity strategy, the individuals either help or block each other from developing the population closer to the optimum. There, the right diversity measure may make the difference between a polynomial and an exponential optimization time. The simplest strategy in the case of multi-objective optimization is to keep in the population at each time step only solutions that are not dominated by any other solution produced during the optimization process. The positive effect of using such a population (compared with one consisting always of a single individual) has already been pointed out in [11].

We show that such a natural strategy may have problems to cope with plateaus of constant fitness. Plateaus are regions in the search space where all search points have the same fitness. Often, the number of different objective values for a given function is polynomially bounded while the number of different search points is exponential. This implies an exponential number of solutions with the same objective value. The behavior of a simple evolutionary algorithm on different plateau functions has already been investigated in [13] where it has been shown that evolutionary algorithms may be efficient on such functions by doing a random walk on the plateau. The same holds for some single-objective combinatorial optimization problems [12, 17] for which it has been proven that evolutionary algorithms have to cope with plateaus of a similar structure. We point out that in the case of multi-objective problems such a random walk may be prevented by other individuals in the population.

We compare the (1+1) EA [6, 13, 18] with its multi-objective counterpart Global SEMO [10, 11, 19] and describe

1-4244-1340-0/07/$25.00

situations where Global SEMO is exponentially slower even if the population size is always small. First, we illustrate this by considering the optimization of a well-known artificial plateau function. Afterwards, the ideas are used to construct a class of SETCOVER problems where Global SEMO with polynomially bounded population size fails to produce an optimal solution within expected polynomial time while the (1+1) EA has a polynomially bounded expected optimization time.

The outline of the paper is as follows. In Section II, we introduce the algorithms that are subject of our analyses. In Sections III and IV, we compare them on an artificial function and an instance of SETCOVER, respectively. We finish with conclusions and some topics for future research.

II. ALGORITHMS

In the following, we will define the setting for our theoretical investigations. We consider the search space $X = \{0,1\}^n$ and a pseudo Boolean function $f\colon \{0,1\}^n \rightarrow \mathbb{R}^k$ with k objectives. Concerning the algorithms, we examine simple single-objective EA and compare it with its multi-objective counterpart. We define both algorithms for problems where all objectives should be maximized. Minimization problems can be considered in a similar way by interchanging the roles of "$\geq$" and "$\leq$" in the algorithms.

For single-objective optimization problems (where $k = 1$), our analyses are carried out for the (1+1) EA which has been considered in theoretical investigations on pseudo Boolean functions [6] as well as some of the best-known combinatorial optimization problems [12, 18, 21]. The algorithm works with a population of size 1 together with elitism-selection and creates in each iteration one offspring by flipping each bit with probability $1/n$:

Algorithm 1: (1+1) EA

1) Choose an initial solution $x \in \{0,1\}^n$
2) Repeat
   - Create x' by flipping each bit of x with probability $1/n$.
   - If $f(x') \geq f(x)$, set $x := x'$.

Analyzing single-objective randomized search heuristics with respect to their runtime behavior, we are interested in the number of constructed solutions until an optimal one has been created for the first time. This is called the runtime or optimization time of the considered algorithm. Often, the expectation of this value is considered and called the expected optimization time or expected runtime.

In the case of multi-objective optimization problems ($k \geq 2$) the search space becomes higher-dimensional. As there is no canonical complete order on $\mathbb{R}^k$, one compares the quality of search points with respect to the canonical partial order on $\mathbb{R}^k$, namely $f(x) \geq f(x')$ iff $f_i(x) \geq f_i(x')$ for all $i \in \{1,\ldots,k\}$. We consider the algorithm called Global SEMO (Global Simple Evolutionary Multi-objective Optimizer) [10, 15] which has been investigated in the context of different multi-objective problems, e. g., spanning tree problems [16, 19] and covering problems [8]. This algorithm equals the (1+1) EA for the case $k = 1$.

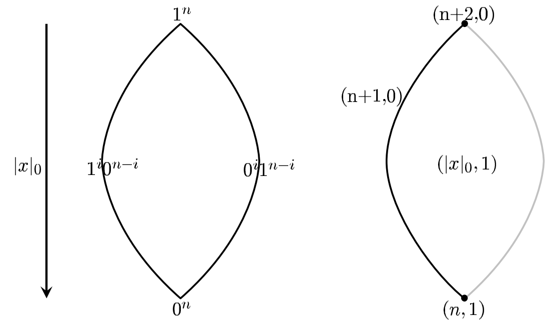

Fig. 1: An illustration of the explored function *PL*.

Global SEMO starts with an initial population P that consists of one single individual. In each generation, an individual x of P is chosen randomly to produce one child x' by mutation. In the mutation step, each bit of x is flipped with probability $1/n$ to produce the offspring x'. After that, x' is added to the population if it is not dominated by any individual in P (i. e., there is no $x \in P$ with $f(x) \geq f(x')$ and $f(x) \neq f(x')$). If x' is added to P all individuals of P that are dominated by x' or have the same fitness vector as x' are removed from P. In detail, Global SEMO is defined as follows.

Algorithm 2: Global SEMO

1) Choose an initial solution $x \in \{0,1\}^n$
2) Determine $f(x)$.
3) $P \leftarrow \{x\}$.
4) Repeat
   - Choose $x \in P$ uniformly at random.
   - Create x' by flipping each bit of x with probability $1/n$.
   - Determine $f(x')$.
   - If no $x \in P$ dominates x', exclude all z where $f(z) \leq f(x')$ from P and add x' to P.

Analyzing multi-objective evolutionary algorithms with respect to their runtime behavior, we consider the number of constructed solutions until for each Pareto optimal objective vector a solution has been included into the population and call this the optimization time of the algorithm—the expected optimization time refers to the expected value of the optimization time.

Throughout this paper we consider two very popular alternatives for choosing the initial solution in our defined algorithms. On the one hand, we consider the case $x = 0^n$. This is quite typical, e. g., for simulated annealing. On the other hand, we consider the case where the initial solution x is chosen uniformly at random from the search space $\{0,1\}^n$. This is the most popular choice for evolutionary algorithms.

III. ANALYSIS OF A PLATEAU FUNCTION

The behavior of the (1+1) EA on plateaus of different structures has been studied in [13] by a rigorous runtime analysis. We want to examine the optimization times of multi-objective plateau functions in contrast to their single-objective counterparts. [2] introduced the function

$$\textsc{plateau}_1(x) := \begin{cases} |x|_0 & : \ x \notin \{1^i 0^{n-i} \mid 1 \le i \le n\} \\ n+1 & : \ x \in \{1^i 0^{n-i} \mid 1 \le i < n\} \\ n+2 & : \ x = 1^n. \end{cases}$$

which is similar to the well-known function SPC [13]. We consider a simple multi-objective extension *PL* of the function $\textsc{plateau}_1$ by adding a second objective that may only attain the two objective values 0 and 1. The function *PL* is defined as follows.

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

Adding the second objective in the defined way has the consequence that there are two Pareto optimal search points namely 0^n and 1^n. As in the case of $\textsc{plateau}_1$ the multi-objective extension consists of a plateau given by the search point of $SP := \{1^i 0^{n-i} \mid 1 \le i < n\}$. All search points of *SP* attain the objective vector $(n+1, 0)$. Figure 1 shows an illustration of this function. The (1+1) EA maximizes *PL* with respect to the lexicographic order $\prec_{lex}$, i. e., we define

$$(x_1, x_2) \prec_{lex} (y_1, y_2) \ \text{ iff } x_1 < y_1 \lor (x_1 = y_1 \land x_2 < y_2).$$

It is easy to see that

$$PL(x) \prec_{lex} PL(y) \ \text{ iff } \textsc{plateau}_1(x) < \textsc{plateau}_1(y).$$

Figure 2 shows the relation graph for the lexicographically sorted multi-objective function *PL*. Note that this is equivalent to the relation graph for $\textsc{plateau}_1$. Therefore, all results which only use the relative structure of $\textsc{plateau}_1$ also hold for *PL* with respect to the lexicographic order $\prec_{lex}$. As [2] showed an expected runtime of the (1+1) EA on $\textsc{plateau}_1$ of $\Theta(n^3)$, the following theorem holds.

Theorem 1: The expected optimization time of the (1+1) EA on *PL* is $O(n^3)$ independently of the chosen initial solution.

This shows that the (1+1) EA is efficient on *PL*. We will now prove that Global SEMO requires an exponential runtime to optimize *PL* and make use of some ideas given in Theorem 2 of [9].

Theorem 2: The optimization time of Global SEMO on *PL* is $2^{\Omega(n^{1/24})}$ with probability $1 - e^{-\Omega(n^{1/24})}$ if the initial solution is 0^n or has been chosen uniformly at random.

Proof: We prove the theorem for the case of a uniformly at random chosen initial solution. As the proof mainly relies on proving that the search point 0^n has been obtained before the search point 1^n, the results also hold for starting with the initial search point 0^n.

The maximal population size is two as there are only two different values for the second fitness value. The initial solution x consists with probability $1 - e^{-\Omega(n)}$ of at most $2n/3$ ones using Chernoff bounds. As long as no solution of

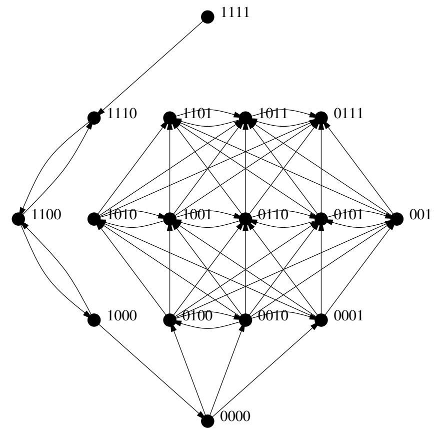

Fig. 2: Relation graph for the objective function $PL\colon \{0,1\}^4 \to \mathbb{R}^2$ with respect to the lexicographic order $\prec_{lex}$. Reflexive and transitive edges are omitted for clarity.

SP has been obtained, only solutions with at most $|x|_1$ ones are accepted. This implies that with probability at least $1 - n^{-n/3}$, there is no step producing the optimal search point 1^n until a first solution in *SP* is discovered. Moreover, this first solution in *SP* has at most $3n/4$ ones as the probability of flipping at least $n/12$ bits in a single mutation step is $e^{-\Omega(n)}$.

We now consider a phase of $2n^{3/2}$ steps of the algorithm after for the first time a solution in *SP* has been produced. Roughly speaking, we will show that within such a phase the random walk of the solution $y \in SP$ reaches the optimal search point 1^n only with very small probability while at the same time the other solution x quickly becomes $x = 0^n$ and produces a descendant on *SP* (both in at most $n^{3/2}$ steps) with high probability and therewith sets back y to small $|y|_1$, which moves it further away from the optimal search point 1^n.

Let $y = 1^i 0^{n-i}$ be the solution on *SP*. We call a step relevant iff it produces a solution $z \in SP$ with $z \neq y$. To achieve this the bit y_i or y_{i+1} has to flip. Therefore, the probability of not having a relevant step is at least $1 - 2/n$ and the expected number of non-relevant steps during this phase is at least $(1 - 2/n)2n^{3/2} = 2n^{3/2} - 4n^{1/2}$. There are at least

$$(1 - n^{-2/3}) \cdot (2n^{3/2} - 4n^{1/2}) \ge 2n^{3/2} - 3n^{5/6}$$

non-relevant steps with probability

$$1 - e^{\left(-n^{3/2} \cdot \frac{n^{-4/3}}{2}\right)} = 1 - e^{-\Omega(n^{1/6})}$$

using Chernoff bounds.

The probability that at least $n^{1/12}$ bits flip in a single accepted mutation step is at most $n^{-n^{1/12}}$. Such an event happens in the phase of $2n^{3/2}$ steps only with probability at most $2n^{3/2 - n^{1/12}} = n^{-\Omega(n^{1/12})}$. Therefore, within this phase the Hamming distance to the optimal search point decreases by at most $3n^{5/6} n^{1/12} = 3n^{11/12}$ and an optimal search point has not been obtained with probability $1 - e^{-\Omega(n^{1/12})}$.

In the following we show that after $n^{3/2}$ steps, the solution 0^n is inserted into the population and in a second phase of $n^{3/2}$ steps a solution $x \in SP$ (setting back the random walk) is produced from 0^n with high probability. We consider in each step the solution x with the largest number of zeros in the population P. As an optimal search point will not be produced within $n^{3/2}$ steps with probability $1 - e^{-\Omega(n^{1/12})}$ such a solution will never be removed from P in this phase. Assume $|x|_1 = k$. Then the probability of producing in the next step a solution z with $|z|_0 > |x|_0$ is at least $(k/(2en))$. Summing up over the different values of k, the search point 0^n is included into P after an expected number of at most $en \log n$ steps. After an expected number of $O(n)$ steps a solution with fitness value $(n+1, 0)$ is included afterwards. Hence, after an expected number of $2en \log n$ steps $P = \{x, 0^n\}$ where $x \in SP$ and $4en \log n$ steps are enough with probability at least $1/2$. The probability of not having obtained these solutions within $n^{3/2}$ steps is upper bounded by $e^{-\Omega(n^{1/2}/\log n)} \leq e^{-\Omega(n^{1/4})}$ considering $n^{1/2}/(4e \log n)$ phases of length $4en \log n$.

The probability to produce from 0^n a search point $x \in SP$ is at least $1/(en)$ as this can be achieved by flipping the first bit of 0^n. The probability to select 0^n in the next mutation step is $1/2$. Using Markov's inequality the probability that such an x has not been produced during $4en$ steps is bounded above by $1/2$ and the probability that this has not happened during $n^{3/2}$ steps is $2^{-\Omega(n^{1/2})}$. We already know that, with probability $1 - e^{-\Omega(n^{1/12})}$ a phase of $2n^{3/2}$ steps does not lead to an optimal solution. Considering $2^{\Omega(n^{1/24})}$ steps the probability of obtaining an optimal solution is still upper bounded by $e^{-\Omega(n^{1/24})}$ which proves the theorem. ■

IV. Analysis of a SetCover Instance

We now show that the behavior observed in the previous section may also occur when applying multi-objective models to single-objective combinatorial optimization problems. We consider the well-known NP-hard SetCover problem for which the use of a multi-objective model has already been examined in [8]. There, it has been shown that using a multi-objective model for the SetCover problem leads to a better approximation ratio for Global SEMO than for the (1+1) EA in a corresponding single-objective setting. The problem can be stated as follows.

Given a ground set S and a collection $C_1, \ldots, C_n$ of subsets of S with corresponding positive costs $c_1, \ldots, c_n$. The goal is to find a minimum-cost selection $C_{i_1}, \ldots, C_{i_k}$, $1 \leq i_j \leq n$ and $1 \leq j \leq k$, of subsets such that all elements of S are covered.

Considering the algorithms introduced in Section II, a search point $x \in \{0,1\}^n$ encodes a selection of subsets. $w(x) = \sum_{i=1}^{n} c_i x_i$ measures the total cost of the selection and $u(x)$ denotes the number of elements of S that are uncovered. Considering RLS and the (1+1) EA for the SetCover problem, the fitness of a search point x is given by the vector $f(x) = (u(x), u(x) + w(x))$ which should be minimized with respect to the lexicographic order. In our multi-objective setting, we would like to minimize $u(x)$ and $p(x) := u(x) + w(x)$ at the same time. Using $p(x)$ as the second objective instead of just $w(x)$ as done in [8] has the effect that the number of incomparable elements for the multi-objective approach becomes smaller which leads to a smaller population size during the optimization process.

Solution x	$p(x)$
$\mathcal{B} \cup \mathcal{C}$ (optimum)	$2n-4$
$\mathcal{B} \cup \{D_i\} \cup \{C_j \mid 1 \leq j < i\}$ for all i	$2n-2$
$\mathcal{B} \cup \{A_i\}$ for all i	$2n-1$
$\{A_i, A_j\}$ for all $i \neq j$	$2n$

Table 1: All set covers x with $p(x) \leq 2n$.

Our aim is to show that even such a model which tends to work with a small population may prevent the algorithm from being efficient. The class of instances under consideration can be defined as follows. Let $k \in \mathbb{N}$ be a constant. Furthermore, set $n := 4k+3$ and $S := [n] := \{j \in \mathbb{N} \mid 1 \leq j \leq n\}$. We define the collection $\mathcal{S} := \mathcal{A} \cup \mathcal{B} \cup \mathcal{C} \cup \mathcal{D}$, where $\mathcal{A}$ consists of the sets $A_i := [n] \setminus \{4k+4-i\}$ $(i \in [2k+2])$, $\mathcal{B}$ contains exactly the set $B := \{2k+1, 2k+2, \ldots, n\}$, $\mathcal{C}$ consists of the sets $C_i := \{2k+1-2i, 2k+2-2i\}$ $(i \in [k])$, and $\mathcal{D}$ consists of the sets $D_i := [2k+3-2i]$ $(i \in [k])$. Thus, the set system $\mathcal{S}$ has cardinality $n = 4k+3$. We define the cost function $w\colon \mathcal{S} \to \mathbb{R}^+$ by

$$w(M) := \begin{cases} n & : \quad M \in \mathcal{A}, \\ n-1 & : \quad M = B, \\ 4 & : \quad M \in \mathcal{C}, \\ n+3-4i & : \quad M = D_i. \end{cases}$$

The case $k = 4$ is shown in Figure 3.

An optimal solution consists of all subsets in $\mathcal{B} \cup \mathcal{C}$ and has price $(n-1) + 4k = 2n-4$. There are 3 additional possibilities to have covers x with $p(x) \leq 2n$. If the current solution consists of two subsets of $\mathcal{A}$, it has cost $2n$. A solution with exactly one set of $\mathcal{A}$ and the set B has cost $2n-1$ and all solutions with the set B, a single set of $\mathcal{D}$ and suitable subsets of $\mathcal{C}$ have cost $2n-2$ (cf. Table 1).

Theorem 3: The expected optimization of the (1+1) EA on *SC* is $O(n^5)$ independently of the chosen initial solution.

Proof: The number of subsets in $\mathcal{A}$ is $2k+2$ and the total number of all subset is $4k+3$. If the current solution x contains at least two subset of $\mathcal{A}$, $u(x) = 0$ holds, i.e., it represents a set cover. The expected time to produce a solution x with $u(x) \leq 1$ is $O(1)$ as there are $\Theta(n)$ subsets covering exactly $n-1$ elements. If the current solution x fulfills $u(x) = 1$ introducing an additional subset of $\mathcal{A}$ leads to a cover. As the there are $\Theta(n)$ subset of $\mathcal{A}$ that are unchosen the expected waiting time to obtain a cover is $O(1)$.

As long as the price p of the cover x is greater than $2n$ an improvement can be obtained by removing a single subset of the current solution and a solution of price at most $2n$ can be obtained by removing a suitable subset of the elements chosen in the current cover. We apply the method of the expected

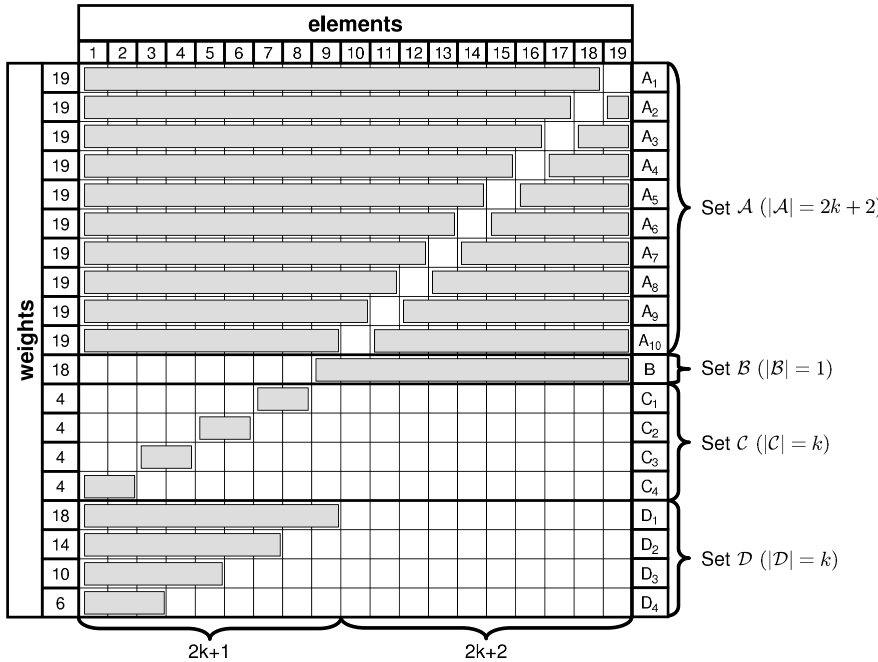

Fig. 3: The examined set system for $k = 4$ and $n = 19$.

multiplicative weight decrease [18] to upper bound the time until a cover of price at most $2n$ has been obtained. Denote by $D = p - 2n$ the amount by which the price of the current solution exceeds the value $2n$. We consider all 1–bit flips that lead to a cover of smaller price. The sum of all these price reductions is at least D. For simplicity all other 1–bit flips that are not accepted reduce the price by 0. Hence, the expected price after such a step is at most $2n+(1-1/n)\cdot D$ and after t such steps the expected price is at most $2n+(1-1/n)^t\cdot D$. As the price of a solution is always an integer and $D = O(n^2)$ holds after having obtained a cover for the first time, $t = cn \log n$ such steps, c an appropriate constant, lead to a price of at most $2n$. The expected waiting time for a 1–bit flip is upper bounded by e which implies that a cover of price at most $2n$ is obtained after an expected number of $O(n \log n)$ steps.

The expected time to obtain from a cover of price $2n$ (i. e., two sets of $\mathcal{A}$) a cover of price at most $2n-1$ is $O(n^2)$ as one of the chosen subset of $\mathcal{A}$ has to be removed and the set B introduced. Similarly a solution of cost $2n-2$ can be obtained from a solution of cost $2n-1$ by removing the chosen set of $\mathcal{A}$ and introducing the largest set of $\mathcal{D}$ in time $O(n^2)$.

Having obtained a solution with cost $2n-2$ the algorithm has to cope with a plateau containing $O(n)$ solutions. The solutions on the plateau differ by the number of subsets of $\mathcal{C}$ that are chosen. The number of subsets of $\mathcal{C}$ can be increased (and also decreased) by a mutation step flipping three specific bits. The expected waiting time for such a step is $O(n^3)$ and the expected number of steps needed to obtain the optimal solution where all subsets of $\mathcal{C}$ (and none of $\mathcal{D}$) are chosen is $O(n^2)$ using arguments similar to [13] for the function SPC_n. Altogether, this leads to the upper bound $O(n^5)$ stated in the theorem. ■

In the case of the multi-objective approach, Global SEMO works with a population of the different trade-offs with respect to the two objective functions. This may have the effect that a single-solution can not cope with the plateau given by the instance *SC*. In fact the optimization time of Global SEMO on *SC* is exponential with probability asymptotically close to 1 if the initial solution is chosen as the empty set.

Theorem 4: The optimization time of Global SEMO on *SC* is $2^{\Omega(n)}$ with probability $1 - o(1)$ if it starts with the initial solution 0^n.

For the proof of this theorem we need the following lemma.

Lemma 5: In the first n mutation steps Global SEMO chooses the empty solution 0^n at least $\frac{1}{2}\ln n$ times for mutation with probability at least $1 - n^{-1/8}$.

Proof: We use the following generalized Chernoff bound [1]: Let $p_1, \ldots, p_n \in [0,1]$ and $X_1, \ldots, X_n$ be mutually independent random variables with $\mathrm{P}[X_i = 1 - p_i] = p_i$ and $P[X_i = -p_i] = 1 - p_i$. Set $X := X_1 + \ldots + X_n$ and $p := (p_1 + \ldots + p_n)/n$. Then

$$\mathrm{P}[X < -a] < e^{-a^2/(2pn)}$$

for any $a > 0$.

Thus, we have to define random variables that give an estimation to the behavior of Global SEMO in the first n steps. Global SEMO starts with the empty solution 0^n. Since we have $w(M) > |M|$ for every set $M \in \mathcal{S}$, the function $(u+w)$ attains its unique minimum for 0^n. Hence, 0^n remains in the population forever. The population size of Global SEMO before the k–th step is at most k. Therefore, the probability that Global SEMO chooses the 0^n for mutation in the k–th step is at least $1/k$. Let $p_i := \frac{1}{i}$ for all $1 \leq i \leq n$. We set $p := (p_1 + \ldots + p_n)/n$ and define random variables $\widetilde{X}_i$ with $\mathrm{P}[\widetilde{X}_i = 1] = p_i$ and $P[\widetilde{X}_i = 0] = 1 - p_i$ for all $1 \leq i \leq n$. Then the random variable $\widetilde{X} = \sum_{i=1}^{n} \widetilde{X}_i$ is a lower bound for the random variable describing the number of mutation steps of 0^n in the first n steps. To use the generalized Chernoff bound, we have to subtract the mean of $\widetilde{X}_i$ from the random variable $\widetilde{X}_i$ for all i. We define $X_i := \widetilde{X}_i - \frac{1}{i}$ and $X := \sum_{i=1}^{n} X_i = \widetilde{X} - \sum_{i=1}^{n} \frac{1}{i}$. The mean of all X_i and thus also the mean of X is 0. We set $a := \frac{1}{2}pn \geq \frac{1}{2}\ln n$ and apply the generalized Chernoff bound to the random variable X. We have

$$\mathrm{P}[\widetilde{X} < \tfrac{1}{2}\ln n] \leq \Pr[X < -a] \leq e^{-\frac{a^2}{2pn}} \leq e^{-\frac{pn}{8}} = n^{-1/8}.$$

As discussed above, this proves that Global SEMO chooses the solution 0^n at least $\frac{1}{2}\ln n$ times for mutation in the first n steps with probability at least $1 - n^{-1/8}$. ∎

Proof of Theorem 4: As a first step we show that with high probability after $2n$ steps of Global SEMO

- the population size is 3,
- there is a set cover with a p–value less or equal $2n$ in the current population,
- the optimum is not determined.

Using Lemma 5, Global SEMO chooses 0^n at least $\frac{1}{2}\ln n$ times for mutation in the first n steps with high probability. Now we show that in these at least $\frac{1}{2}\ln n$ mutation steps of the search point 0^n, Global SEMO produces a solution with exactly one $\mathcal{A}$-set with high probability. We call such a search point an $\mathcal{A}_1$–*solution*. The probability that a mutation of 0^n results in an $\mathcal{A}_1$–solution is at least $\frac{|\mathcal{A}|}{n}(1-\frac{1}{n})^{n-1} \geq \frac{1}{2e}$. Thus, the probability that Global SEMO produces an $\mathcal{A}_1$–solution in $\frac{1}{2}\ln n$ mutation steps of 0^n is at least

$$1 - (1 - \tfrac{1}{2e})^{\frac{1}{2}\ln n} \geq 1 - e^{-\frac{1}{10}\ln n} = 1 - n^{-1/10}.$$

As the p–value of every $\mathcal{A}_1$–solution is $n+1$ and only the empty solution 0^n has a lower p–value, such an $\mathcal{A}_1$–solution stays in the population and can only be replaced by another $\mathcal{A}_1$–solution. Moreover, all strings with u–value between 2 and $n-1$ were removed from the current population. Thus, the population size is at most 3 (0^n, $\mathcal{A}_1$–solution, and maybe a set cover) from that moment on.

We consider another round of n steps of Global SEMO. Since the population size is at most 3, 0^n will be chosen for mutation at least $n/4$ times in this phase with probability exponentially close to 1 using Chernoff bounds. The probability that such a mutation of 0^n results in a solution with exactly two sets of $\mathcal{A}$ is at least $\frac{|\mathcal{A}|(|\mathcal{A}|-1)}{2n^2}(1-\frac{1}{n})^{n-2} \geq \frac{1}{8e}$. We call such a search point an $\mathcal{A}_2$–*solution*. Thus, probability that at least one $\mathcal{A}_2$–solution is produced in $n/4$ mutation steps of 0^n is at least $1 - (1-\frac{1}{8e})^{n/4} = 1 - e^{-\Omega(n)}$. Every $\mathcal{A}_2$–solution is a set cover and has a p–value of exactly $2n$. Hence, with probability $1 - O(n^{-1/10})$ after $2n$ steps of Global SEMO the population size is 3 and there is a solution which is a set cover and has a p–value of at most $2n$.

The last thing that we have to show for the first claimed aim is that in the considered first phase of $2n$ steps, the optimum is not determined. One can easily check that the unique optimum is the solution with all sets in $\mathcal{B} \cup \mathcal{C}$ and no other set. Since Global SEMO starts with 0^n, the optimum cannot be found until every bit that corresponds to a set of $\mathcal{B} \cup \mathcal{C}$ has been flipped at least in one mutation step of Global SEMO. Using $|\mathcal{B} \cup \mathcal{C}| > n/3$, the probability that the optimum is not produced in the first $2n$ steps of Global SEMO is at least

$$1 - (1 - (1-\tfrac{1}{n})^{2n})^{n/3} \geq 1 - e^{-\Omega(n)}.$$

Table 1 shows all possible set covers with a p–value of at most $2n$. Besides the $\mathcal{A}_2$–solutions (with p–value $2n$), the optimal search point (all sets of $\mathcal{B} \cup \mathcal{C}$ with p–value $2n-4$), and the solutions with one set from $\mathcal{A}$ and the set B (with p–value $2n-1$), the only set covering solutions that can be accepted by Global SEMO are of the following form. They contain exactly the set B the sets C_1 up to C_i and the set D_{i+1} for all $0 \leq i \leq k-1$. For $i = 0$ the sets are B and D_1 (and no set from $\mathcal{C}$). The p–value of all these search points is $2n-2$. We call them RW-solutions since Global SEMO has to perform a random walk on these search points to reach the optimum. After Global SEMO has determined the first RW-solution only RW-solutions or the optimum are accepted from Global SEMO as set covers.

We now show that Global SEMO cannot perform the random walk on the RW-solutions since this random walk is reseted too frequently. If the set cover in the population after the first $2n$ steps of Global SEMO is an RW-solution, at most the first $k/3$ sets of $\mathcal{C}$ are represented in this solution with probability at least

$$1 - (1 - (1-\tfrac{1}{n})^{2n})^{k/3} \geq 1 - e^{-\Omega(k)} = 1 - e^{-\Omega(n)}.$$

If the current set cover after the first $2n$ steps of Global SEMO is not an RW-solution, the first RW-solution determined by Global SEMO will also contain no more than $k/3$ sets of $\mathcal{C}$ with probability exponentially close to 1. Let us consider a phase of n^3 steps of Global SEMO. We show that with probability exponentially close to 1 the random walk is reset and also with probability exponentially close to 1 the optimum is not reached in this phase of n^3 steps. We call a mutation of 0^n that results in the solution with sets B and D_1 a *reset-step*, since this solution is accepted (until the optimum is determined) and it brings the random walk at a hamming distance of $k+1$ from the optimum. The probability for such a reset-step is at least $\frac{1}{3n^2}(1-\frac{1}{n})^{n-2} \geq \frac{1}{3en^2}$. The 3 in the denominator is caused by the population size 3. Thus, there will be a reset-step in n^3 steps of Global SEMO with probability at least

$$1 - (1 - \tfrac{1}{3en^2})^{n^3} = 1 - 2^{-\Omega(n)}.$$

Now we bound the probability that the optimum is determined in the phase of n^3 steps of Global SEMO. The probability to reduce the distance to the optimum by integrating the next set of $\mathcal{C}$ in the current set cover (plus integrating and deleting the corresponding two sets of $\mathcal{D}$) is at most $1/n^3$. Moreover, the probability to reduce the distance to the optimum by integrating the next j sets of $\mathcal{C}$ in the current set cover (and additionally integrating and deleting the corresponding two sets of $\mathcal{D}$), is at most $1/n^{j+2}$ ($j \in [k]$). For a fixed $a \in [n^3]$ there are at most k^a possible ways to achieve the optimum in exactly a such steps. And each of these ways has probability at most $1/n^{k+2a}$. Hence, the probability to determine the optimum in n^3 steps with exactly a random walk steps is at most n^{-k-a}. Altogether, the probability to reach the optimum in a phase of n^3 steps is at most

$$\sum_{a=1}^{n^3} n^{-k-a} = n^{-\Omega(n)}.$$

We have shown that with probability at least $1 - 2^{-\Omega(n)}$ in n^3 steps of Global SEMO, there is a reset-step and the optimum is not reached before this. Hence, within $2^{\Omega(n)}$ steps Global SEMO does not find the optimum with probability $1 - 2^{-\Omega(n)}$. This proves the theorem as all our statements hold with probability $1 - o(1)$. ■

In the case that the initial solution is chosen uniformly at random, the probability to obtain an exponential optimization time can only be bounded in a much weaker way. However, the probability that Global SEMO fails on *SC* in this case is still at least $1/poly(n)$, where $poly(n)$ is a polynomial in n of small degree. This implies that the expected optimization time is exponential, but leaves the opportunity to obtain the optimal solution by a small number of restarts.

Theorem 6: The expected optimization time of Global SEMO with a randomly chosen initial solution on *SC* is exponential. Precisely, the optimization time is $2^{\Omega(n)}$ with probability $\Omega(1/n^{2e-1})$.

Proof: We now start with a random initial solution. Let m be the number of $\mathcal{A}$–sets in the initial solution of Global SEMO. By Chernoff bounds we know that with probability $1 - e^{\Omega(n)}$,

$$n/6 \leq m \leq n/3.$$

Thus, the initial solution is a set cover with high probability and as long as there are no uncovered elements, the population size remains 1 and the Global SEMO behaves like the (1+1) EA.

We now consider the first $2e\,n \ln n$ steps. The probability that a specific set has been removed in this time from the initial solution is

$$p := 1 - \left(1 - \tfrac{1}{n}\right)^{2e\,n\ln n} \geq 1 - e^{2e\,\ln n} \geq 1 - 1/n^{2e}.$$

For n large enough we get the following upper bound on p.

$$p = 1 - \left(1 - \tfrac{1}{n}\right)^{2e\,\ln n}(1 - \tfrac{1}{n})^{2e\,(n-1)\ln n} \leq 1 - 1/(2n^{2e}).$$

The probability that any two sets from $\mathcal{A}$ have not been removed within $2e\,n \ln n$ steps is therefore

$$\begin{aligned}
q &= 1 - p^m - m(1-p)p^{m-1} \\
&\geq 1 - \left(1 - \frac{1}{2n^{2e}}\right)^{n/6} - \frac{n}{3n^{2e}}\left(1 - \frac{1}{2n^{2e}}\right)^{n/6} \\
&= 1 - \left(1 - \frac{1}{2n^{2e}}\right)^{n/6}\left(1 + \frac{n}{3n^{2e}}\right) \\
&\geq 1 - e^{-1/(12n^{2e-1})}\left(1 + \frac{n}{3n^{2e}}\right) \\
&= \Omega(1/n^{2e-1})
\end{aligned}$$

by the power series of the exponential function.

It remains to calculate the probability that within the first $2e\,n \ln n$ steps all sets except two $\mathcal{A}$–sets are removed under the condition that two arbitrary $\mathcal{A}$–sets are never removed. Let W be the sum of all weights of all sets. Then,

$$W := \sum_{M \in \mathcal{S}} w(M) = 10k^2 + 26k + 8 = \frac{5}{8}n^2 + \frac{11}{4}n - \frac{47}{8}.$$

We want to calculate the probability to arrive at a weight sum of the current solution of $2n$ within $2e\,n \ln n$ steps by using again the method of the expected multiplicative weight decrease [18]. For this, we now consider a single step. Let w be the weight sum before this step. The weight distance which we want to bridge to reach our aim is $D = w - 2n$. As the weight sum of all current sets is w, the expected weight decrease of a 1-bit flip is $D/(n-2)$. Therefore, one 1-bit flip decreases the weight distance by an expected factor of $\left(1 - \frac{1}{n-2}\right)$. And such a 1-bit occurs with probability $1/e$. After $2e\,n \ln n$ steps, the expected weight distance is at most

$$\left(1 - \tfrac{1}{n-2}\right)^{2\,n\ln n} W \leq W/n^2 < 1.$$

Hence, with probability at least $1/2$ we reach within $2e\,n \ln n$ steps an $\mathcal{A}_2$–solution (cf. notation used in the proof of Theorem 4) under the condition that two arbitrary $\mathcal{A}$–sets are never removed. Using the considerations above, Global SEMO attains with probability at least $\Omega(1/n^{2e-1})$ a situation where the only current individual is an $\mathcal{A}_2$-solution.

We like to apply the argumentation in the proof of Theorem 4. For this aim we show the following. Starting from the described situation, Global SEMO integrates 0^n and an $\mathcal{A}_1$-solution with probability at least $1/2e$. Moreover, if the set cover of the current population is an RW-solution, then at most the first $k/3$ sets of $\mathcal{C}$ are represented in this search point with probability $1 - e^{-\Omega(n)}$.

The next accepted mutation step of Global SEMO removes at least one of the two $\mathcal{A}$–sets. With probability $(1 - \frac{1}{n})^{n-1} \geq \frac{1}{e}$ no other bit is touched and thus an $\mathcal{A}_1$–solution is introduced in the current population. We consider a phase of $3e\,n$ steps of Global SEMO. As already shown in the proof of Theorem 4, at most the first $k/3$ sets of $\mathcal{C}$ are represented in the current set cover after $5e\,n$ steps with probability at least

$$1 - (1 - (1 - \tfrac{1}{n})^{3en})^{k/3} \geq 1 - e^{-\Omega(k)} = 1 - e^{-\Omega(n)}.$$

The last thing we have to prove is that in this phase of $3e\,n$ steps with probability at least $1/2$ the empty solution 0^n is produced by Global SEMO. Until this happens the population size is 2. Thus, with probability $1-e^{-\Omega(n)}$ by Chernoff bounds at least n times the $\mathcal{A}_1$–solution is chosen for mutation and a 1–bit flip is performed. The probability that in these at least n 1–bit flips the 0^n–string is produced is at least

$$1-(1-\tfrac{1}{n})^n \geq 1-\tfrac{1}{e} > \tfrac{1}{2}.$$

This reduces to a situation already examined in the proof of Theorem 4 and therefore finishes this proof. ■

V. Conclusions

Understanding the behavior of evolutionary algorithms for multi-objective optimization is a challenging task where many questions are still open. We have investigated how a simple multi-objective approach can cope with plateaus of constant fitness. Comparing a multi-objective EA with its single-objective counterpart, we have pointed out that even simple plateaus may be hard to optimize as the algorithm may not have the opportunity to do a random walk. In our investigations we considered a multi-objective version of a well-known pseudo-Boolean function as well as a class of instances from the SetCover problem.

We want to point out some interesting topics for future work. First, it seems interesting to compare different diversity strategies used in evolutionary algorithms for multi-objective optimization and investigate situations where using a certain strategy can make the difference between an exponential and a polynomial runtime. Second, it would be desirable to present a single-objective combinatorial optimization problem (not only a class of instances) where applying an intuitive multi-objective approach increases the runtime exponentially even if the population size is always polynomially bounded.

References

[1] N. Alon and J. H. Spencer. *The Probabilistic Method*. Wiley, 2nd edition, 2000.

[2] D. Brockhoff, T. Friedrich, N. Hebbinghaus, C. Klein, F. Neumann, and E. Zitzler. Do additional objectives make a problem harder? In D. Thierens, editor, *Proc. of GECCO '07*, London, UK, 2007. ACM. To appear.

[3] C. C. Coello, D. A. V. Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, 2002.

[4] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, Chichester, 2001.

[5] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In M. Schoenauer et al., editors, *Proc. of PPSN '00*, LNCS Vol. 1917, pages 849–858. Springer, 2000.

[6] S. Droste, T. Jansen, and I. Wegener. On the analysis of the (1+1) evolutionary algorithm. *Theor. Comput. Sci.*, 276:51–81, 2002.

[7] M. Ehrgott. *Multicriteria Optimization*. Berlin, Springer, 2nd edition, 2005.

[8] 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*, London, UK, 2007. ACM. To appear. Also available as ECCC Report TR07-027.

[9] T. Friedrich, N. Hebbinghaus, and F. Neumann. Rigorous analyses of simple diversity mechanisms. In *Proc. of GECCO '07*, London, UK, 2007. ACM. To appear.

[10] O. Giel. Expected runtimes of a simple multi-objective evolutionary algorithm. In *Proc. of CEC '03*, IEEE Press, pages 1918–1925, 2003.

[11] 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.

[12] O. Giel and I. Wegener. Evolutionary algorithms and the maximum matching problem. In *Proc. of STACS '03*, volume 2607 of *LNCS*, pages 415–426, 2003.

[13] 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.

[14] M. T. Jensen. Helper-Objectives: Using Multi-Objective Evolutionary Algorithms for Single-Objective Optimisation. *Journal of Mathematical Modelling and Algorithms*, 3(4):323–347, 2004.

[15] M. Laumanns, L. Thiele, E. Zitzler, E. Welzl, and K. Deb. Running time analysis of multi-objective evolutionary algorithms on a simple discrete optimization problem. In *Proc. of PPSN '02*, LNCS Vol. 2439, pages 44–53. Springer, 2002.

[16] 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.

[17] F. Neumann. Expected runtimes of evolutionary algorithms for the eulerian cycle problem. In *Proc. of CEC '04*, volume 1 of *IEEE Press*, pages 904–910, 2004.

[18] F. Neumann and I. Wegener. Randomized local search, evolutionary algorithms, and the minimum spanning tree problem. In *Proc. of GECCO '04*, volume 3102 of *LNCS*, pages 713–724, 2004.

[19] F. Neumann and I. Wegener. Minimum Spanning Trees Made Easier Via Multi-Objective Optimization. *Natural Computing*, 5(3):305–319, 2006.

[20] P. Winkler. Random Orders. *Order*, 1(1985):317–331, 1985.

[21] C. Witt. Worst-case and average-case approximations by simple randomized search heuristics. In *Proc. of STACS '05*, volume 3404 of *LNCS*, pages 44–56, 2005.

[22] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In K. Giannakoglou et al., editors, *Proc. of EUROGEN'01*, pages 95–100. CIMNE, 2002.