On the Effects of Adding Objectives to Plateau Functions

Dimo Brockhoff, Tobias Friedrich, Nils Hebbinghaus,
Christian Klein, Frank Neumann, and Eckart Zitzler

Abstract—In this paper, we examine how adding objectives to a given optimization problem affects the computational effort required to generate the set of Pareto-optimal solutions. Experimental studies show that additional objectives may change the running time behavior of an algorithm drastically. Often it is assumed that more objectives make a problem harder as the number of different tradeoffs may increase with the problem dimension. We show that additional objectives, however, may be both beneficial and obstructive depending on the chosen objective. Our results are obtained by rigorous running time analyses that show the different effects of adding objectives to a well-known plateau function. Additional experiments show that the theoretically shown behavior can be observed for problems with more than one objective.

Index Terms—Multiobjective optimization, running time analysis, theory.

I. MOTIVATION

IN RECENT YEARS, the number of publications on evolutionary multiobjective optimization has been rapidly growing; however, most of the studies investigate problems where the number of considered objectives is low, i.e., between two and four, while studies with many objectives are rare [5]. The reason is that a large number of objectives leads to further difficulties with respect to decision making, visualization, and computation. Nevertheless, from a practical point of view it is desirable with most applications to include as many objectives as possible without the need to specify preferences among the different criteria. An open question in this context is how the inclusion of additional objectives affects the search efficiency of an evolutionary algorithm to generate the set of Pareto-optimal solutions.

There is some evidence in the literature that additional objectives can make a problem harder. Winkler [31] proved

that the number of incomparable solutions increases if further randomly generated objectives are added. Therefore, on the one hand the Pareto-optimal front may become larger and, on the other hand, the power of the dominance relation to guide the search may diminish—these are the main arguments that various researchers, e.g., [5], [7], [10], [11], [14], [27], list in favor of the assumption that the search becomes harder more objectives are involved. That, in fact, state-of-the-art evolutionary algorithms like NSGA-II and SPEA2 have problems to find a good approximation of the Pareto-optimal front for selected test problems was empirically shown in [30] and [27]. Furthermore, the investigations of Purshouse and Fleming [27] show that the behavior of a multiobjective evolutionary algorithm on a problem with few objectives cannot be generalized to a larger number of objectives.

In a contrast, a few publications point out that reformulating a problem in terms of more objective functions can reduce the computational cost of the optimization process. For example, Jensen [17] successfully used additional "helper-objectives" to guide the search of evolutionary algorithms in high-dimensional spaces. A similar approach was proposed by Knowles *et al.* [18], where single-objective problems are "multiobjectivized," i.e., decomposed into multiobjective problems which are easier to solve than the original problems. Also the idea of turning constraints of single-objective problems into additional objectives has been shown to reduce optimization cost until good solutions are found [21], although Runarsson and Yao [28] pointed out that this is not effective on all kinds of problems due to a wrong search bias. Besides these empirically oriented studies, there are theoretical results supporting the hypothesis that multiobjectivization can help. Scharnow *et al.* [29] showed that the single source shortest path problem is easier to solve for simple evolutionary algorithms (EAs) when formulated as a bi-criterion problem; Neumann and Wegener [25] proved for the minimum spanning tree problem that a formulation with two objectives leads to a lower running time complexity of simple EAs than the original single-objective version.

This discussion indicates that a general statement on the effect of increasing the number of objectives is not possible. For some problems, with a higher number of objectives it is more difficult to generate the Pareto-optimal front; for other problems, it is easier. However, given the previous work, the question arises whether one and the same problem can be made both easier and harder depending on the added objective. This paper answers this question both experimentally and

D. Brockhoff and E. Zitzler are with the Computer Engineering and Networks Laboratory, Elgenössiche Technische Hochschule Zurich, 8092 Zurich, Switzerland (e-mail: brockho@tik.ee.ethz.ch; zitzler@tik.ee.ethz.ch).

T. Friedrich is with the Algorithm Group at the International Computer Science Institute, Berkeley, CA 94704, USA (e-mail: Tobias@ICSI.Berkeley.edu).

N. Hebbinghaus, C. Klein, and F. Neumann are with the Department 1: Algorithms and Complexity at Max-Planck-Institut für Informatik, 66123 Saarbrücken, Germany (e-mail: nils.hebbinghaus@mpi-inf.mpg.de; christian.klein@mpi-inf.mpg.de; frank.neumann@mpi-inf.mpg.de).

theoretically. Based on a simple multiobjective optimizer, namely SEMO, which is known from various theoretical analyses, we show:

1) by means of running time analyses that one and the same problem can become both harder and easier to solve if different objectives are added—in contrast to [29] and [25] where the original objective is replaced by two other objectives, we here consider the case that the original objective remains in the objective set;

2) for two equally difficult single-objective functions that the combination of the two yields a bi-criterion problem that is easier to solve than either of the two single-objective problems;

3) and experimentally that problems with more objectives exist that can also be made both harder and easier, only depending on the type of the added objective.

The main observation behind both the running time analyses and the experimental studies is that problems may contain so-called plateaus. A plateau is a part of the search space where the problem does not indicate any search direction. As we will show in the reminder of this paper, an additional objective can remove or introduce those plateaus. Since an EA, be it a single- or a multiobjective one, performs a random walk on these plateaus as was shown for some of the well-known combinatorial optimization problems [13], [26], [32], the removal or introduction of plateaus by adding objectives can change the running time behavior of EAs drastically. Depending on whether the additional objective introduces the right or a deceptive search direction on a former plateau, or a good or deceptive direction is eliminated by introducing a plateau, the problem becomes harder or easier to solve for an EA.

This paper extends its conference version [2] in several ways. On one hand, the analyses have been improved and additional lower bounds on the running time behavior of the considered multiobjective evolutionary algorithms are presented. The new lower bounds make the analyses tight, as all of them match with the proven asymptotic upper bounds. On the other hand, the effect of adding objectives is investigated for problems with more than two objectives by carrying out experimental studies.

The paper is organized as follows. First, we review basic concepts such as relation graphs and objective conflicts and discuss how additional objectives can affect the dominance structure (Section II). In Section III, we detail the algorithms considered in this paper and define the setting for the running time analyses to follow. Section IV provides the proofs showing that a simple plateau function can become either harder or easier with an additional objective; Section V extends these results and demonstrates that even the combination of two equally difficult single-objective functions can yield an easier bi-criterion problem. That an additional objective can even make problems with more than one objective either harder or easier is shown experimentally in Section VI. Conclusions are presented in Section VII.

II. ADDING OBJECTIVES: FOUNDATIONS AND EFFECTS

Without loss of generality, we consider maximization problems with k objective functions $f_i: X \rightarrow \mathbb{R}$, $1 \leq i \leq k$, where

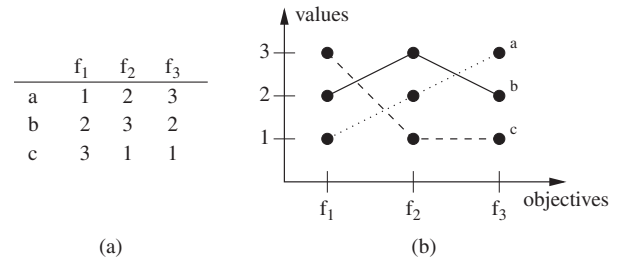|	f_1	f_2	f_3
a	1	2	3
b	2	3	2
c	3	1	1

Fig. 1. (a) Objective values and (b) corresponding parallel coordinates plot for three solutions $a, b, c \in X$.

the vector function $f := (f_1, \ldots, f_k)$ maps each solution $x \in X$ to an objective vector $f(x) \in \mathbb{R}^k$. Furthermore, we assume that the underlying dominance structure is given by the weak Pareto-dominance relation which is defined as follows: $\succeq_{\mathcal{F}'} := \{(x, y) \in X^2 \mid \forall f_i \in \mathcal{F}': f_i(x) \geq f_i(y)\}$, where $\mathcal{F}'$ is a set of objectives with $\mathcal{F}' \subseteq \mathcal{F} := \{f_1, \ldots, f_k\}$. We say x *weakly dominates y w. r. t. the objective set $\mathcal{F}'$ ($x \succeq_{\mathcal{F}'} y$)* if $(x, y) \in \succeq_{\mathcal{F}'}$ and distinguish between the following three cases:

1) the solution pair x, y is called *comparable* if x weakly dominates y and/or y weakly dominates x;

2) two solutions x, y are *incomparable* if neither weakly dominates the other one;

3) two solutions having the same objective vector are called *indifferent*.

A solution $x^* \in X$ is called *Pareto-optimal* if every $x \in X$ is either indifferent to x^* or does not weakly dominate x^* w. r. t. the set of all objectives. The set of all Pareto-optimal solutions is called *Pareto (optimal) set*, its image in the objective space is called *Pareto front*.

Given these basic terms, we will now illustrate on the basis of a simple example what happens if objectives are added. To this end, we recapitulate some concepts introduced in [3], [4]. Assume the search space X consists of three solutions a, b, and c and $\mathcal{F}$ consists of three objective functions f_1, f_2, and f_3. In Fig. 1, the objective functions are shown and the solutions are depicted in a parallel coordinates plot. To see what happens when merging, e.g., the objectives f_1 and f_2 into a bi-criterion problem, the visualization of the dominance relations $\succeq_{f_1}$, $\succeq_{f_2}$, and $\succeq_{f_1 \cup f_2}$ as *relation graphs* is useful. In such a relation graph, each solution corresponds to a vertex and a directed edge from vertex v to vertex w is drawn iff $v \succeq_{\mathcal{F}} w$. Fig. 2 shows the relation graphs for $\succeq_{f_1}$, $\succeq_{f_2}$, $\succeq_{f_3}$, and their corresponding combinations. With a single objective only, the three solutions are pairwise comparable, see Fig. 2(a)–(c). When merging f_1 and f_2 to a bi-criterion problem, comparabilities disappear, Fig. 2(d). For example, the edge between a and c is not present in the resulting relation graph of $\succeq_{\{f_1, f_2\}}$. Because $f_1(c) > f_1(a)$, solution c weakly dominates a with respect to f_1 but c does not weakly dominate a. With respect to f_2, solution a weakly dominates c because $f_2(a) > f_2(c)$. When taking both f_1 and f_2 into account, neither does a weakly dominate c nor does c weakly dominate a by definition of $\succeq$; the solution pair (a, c) is incomparable, i.e., no edge between a and c is drawn in $\succeq_{\{f_1, f_2\}}$. The same holds for the solution pair (b, c).
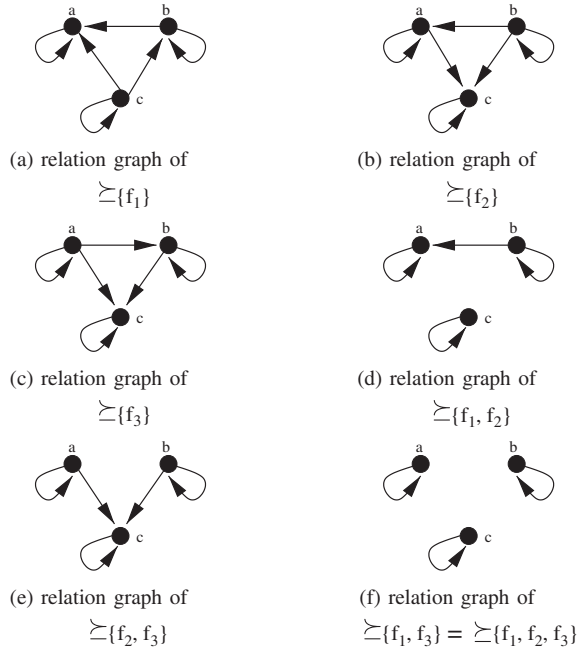
(a) relation graph of
$\succeq_{\{f_1\}}$

(b) relation graph of
$\succeq_{\{f_2\}}$

(c) relation graph of
$\succeq_{\{f_3\}}$

(d) relation graph of
$\succeq_{\{f_1, f_2\}}$

(e) relation graph of
$\succeq_{\{f_2, f_3\}}$

(f) relation graph of
$\succeq_{\{f_1, f_3\}} = \succeq_{\{f_1, f_2, f_3\}}$

Fig. 2. Relation graphs for the three solutions *a*, *b*, and *c* and different objective subsets.

What happens now if f_3 is added to the bi-criterion problem, described by f_1 and f_2? The comparable solutions *a* and *b* become also incomparable because $f_3(a) > f_3(b)$, but *b* weakly dominates *a* with respect to $\{f_1, f_2\}$, i.e., $f_1(b) > f_1(a)$ and $f_2(b) > f_2(a)$. The edge between *a* and *b* is also removed in the relation graph of $\succeq_{\{f_1, f_2, f_3\}}$, Fig. 2(f).

We observe that additional objectives result in the disappearing of edges in the relation graphs—new edges cannot appear if objectives are added. On the one hand, if a solution pair is comparable with respect to all objectives, i.e., an edge is drawn, the two solutions are comparable with respect to any subset of objectives and the edge is already included in the relation graphs for all objective subsets. On the other hand, if a solution *x* is better than a solution *y* with respect to the objectives in $\mathcal{F}_1$, i.e., an edge in $\succeq_{\mathcal{F}_1}$ is only drawn from *x* to *y* but not the other way round, and *y* is better than solution *x* with respect to the objective set $\mathcal{F}_2$, i.e., $(y, x) \in \succeq_{\mathcal{F}_2}$, but $(y, x) \notin \succeq_{\mathcal{F}_1}$, the solution pair becomes incomparable with respect to $\mathcal{F}_1 \cup \mathcal{F}_2$; the edges between *x* and *y* disappear in $\succeq_{\mathcal{F}_1 \cup \mathcal{F}_2}$. Considering our example, the edges of the new relation graphs can always be derived from the relation graphs of the smaller objective sets: an edge is drawn iff the edge is present in the relation graphs of both objective subsets; the new edge set is the intersection of the previous edge sets. This observation can be summarized as follows, see [3] for details.

Let $\mathcal{F}' \subseteq \mathcal{F} = \{f_1, \dots, f_k\}$ be a set of objective functions. Then $\bigcap_{i \in \mathcal{F}'} \succeq_i = \succeq_{\mathcal{F}'}$.

Based on this result, one can define two sets of objectives as *conflicting* according to [4] if the relation graphs are different.

Let $\mathcal{F}_1, \mathcal{F}_2 \subseteq \mathcal{F}$ be two sets of objectives. $\mathcal{F}_1$ is *conflicting* with $\mathcal{F}_2$ iff $\succeq_{\mathcal{F}_1} \neq \succeq_{\mathcal{F}_2}$.

Note that the addition of an objective to a problem can, therefore, affect the running time of a dominance relation-based EA, e.g., simple evolutionary multiobjective optimized (SEMO), only if the additional objective is conflicting with the set of objectives, defining the original problem. As the example in Figs. 1 and 2 shows, the addition of f_2 to the problem defined by f_1 and f_3 does not change the underlying dominance relation and, therefore, does not change the running time of evolutionary algorithms which consider the dominance relation solely. Now, the question arises, how the addition of a conflicting objective affects the complexity of a problem, or more precisely, how an additional objective changes the running time of an evolutionary algorithm.

Addressing the above-mentioned question, we sketch the fundamental idea of this paper. When adding an objective f_i to an objective set $\mathcal{F}'$, there can be two situations:

1) comparable solutions can become incomparable;
2) an indifferent relation between solutions can become a comparable one.

Of course, both cases can occur simultaneously, if an objective is added.[1] Surprisingly, in both cases, a problem can become easier or harder to solve as is shown analytically in the following sections.

Generally speaking, case 1) turns a region with given search space direction into a plateau of incomparable solutions, whereas case 2) turns a plateau of indifferent solutions into a region where the weak Pareto dominance indicates a direction. The different behavior of additional objectives in both cases depends on the direction in which the weak Pareto dominance points. In case 1), where comparable solutions become incomparable, the comparability between solutions can either lead to the Pareto front or be deceptive. The addition of an objective will cause a new plateau of incomparable solutions, but in the latter case the incomparability will help to solve the problem, whereas in the former case the incomparability will make the problem harder. In case 2), the problem can either become harder or easier when changing the dominance structure from a plateau of indifferent solutions into a region of comparable solutions. Depending on whether the newly introduced comparability will lead to the Pareto front or behave deceptively, the computational effort to identify the Pareto optima may decrease or increase.

III. ALGORITHMS

This section defines the setting for the running time analyses to follow. As to the search space, we consider pseudo boolean functions $f: \{0, 1\}^n \to \mathbb{R}^k$, i.e., $X = \{0, 1\}^n$. Concerning the algorithms, we examine both a single-objective EA and a multiobjective EA.

For single-objective optimization problems (where $k = 1$), our analyses are based on the $(1 + 1)$ EA (Algorithm 1) which has been considered in theoretical investigations on pseudo boolean functions [9] as well as some of the best known combinatorial optimization problems [13], [26], [32].

---

[1]The other way around, an omission of an objective can make incomparable solutions comparable and comparable solutions indifferent.

Algorithm 1 (1+1) EA

Choose $x \in \{0, 1\}^n$ uniformly at random
repeat
   Create x' by flipping each bit of x with probability $1/n$
   if $f(x') \geq f(x)$ **then** set $x := x'$
until stop

Algorithm 2 GLOBAL SEMO

Choose $x \in \{0, 1\}^n$ uniformly at random
Determine $f(x)$
$P \leftarrow \{x\}$
repeat
   Choose $x \in P$ uniformly at random
   Create x' by flipping each bit of x with probability $1/n$
   Determine $f(x')$
   if x' is not dominated by any other search point in P **then**
   include x' into P and delete all solutions dominated by x'
   or with objective vector $f(x')$ from P
until stop

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 $1/n$.

Analyzing single-objective randomized search heuristics with respect to their running time 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 running time or optimization time of the considered algorithm. Often, the expectation of this value is considered and called the expected optimization time or expected running time.

We compare the $(1 + 1)$ EA with its multiobjective counterpart called Global SEMO (Algorithm 2) [12], [20], which has been investigated in the context of different multiobjective problems, e.g., spanning tree problems [23], [25]. Global SEMO starts with an initial population P that consists of one single randomly chosen 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. If x' is added to P, all individuals of P that are dominated by x' or have the same objective vector as x' are removed from P. In detail, Global SEMO is defined in the Algorithm 2.

Analyzing multiobjective evolutionary algorithms with respect to their running time 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 expectation value of the optimization time.

Let $|x|_1$ denote the number of 1s and $|x|_0$ denote the number of 0s in a given bitstring x. We are also interested in variants of the introduced algorithms using the following asymmetric mutation operator proposed in [15].

Algorithm 3 ASYMMETRIC MUTATION OPERATOR

Create x' by flipping each bit x_i of x with probability $1/(2|x|_1)$ if $x_i = 1$ and with probability $1/(2|x|_0)$ otherwise

---

We denote by $(1 + 1)$ EA$_{asy}$ and Global SEMO$_{asy}$ the algorithms that differ from the $(1 + 1)$ EA and Global SEMO by using the mutation operator given in Algorithm 3.

IV. ADDING OBJECTIVES TO A PLATEAU

Our aim is to examine the effect of adding different objectives to a well-known plateau function. Plateaus are regions in the search space where all search points have the same objective vectors. Consider a function $f: \{0, 1\}^n \rightarrow \mathbb{R}$ and assume that the number of different objective values for that function is V. Then there are at least $2^n/V$ search points with the same objective value. Often, the number of different objective values for a given function is polynomially bounded. This implies an exponential number of solutions with the same objective value. Nevertheless, such functions where V is polynomially bounded are easy to optimize for EAs if for each nonoptimal solution there is a better Hamming neighbor, which means that an improvement can be reached by flipping a single bit of a nonoptimal solution. If this is not the case, the search for a randomized search heuristic may become much harder. In the extreme case, we end up with the function NEEDLE where only one single solution has objective value 1 and the remaining ones get an objective value of 0 [16]. The behavior of the $(1 + 1)$ EA on plateaus of different structures has been studied in [16] by a rigorous running time analysis.

The function PLATEAU$_1$, which we examine in the following, contains a set of $n - 1$ search points that form a plateau having objective value $n+1$. We denote by $SP_1 := \{1^i 0^{n-i}, 1 \leq i < n\}$ this set of search points and define PLATEAU$_1$ as

$$\text{PLATEAU}_1(x) := \begin{cases} |x|_0 & : \quad x \notin SP_1 \\ n + 1 : & x \in SP_1 \\ n + 2 : & x = 1^n. \end{cases}$$

Note, that this function is similar to the function SPC$_n$ already investigated in [16].

The relation graph of PLATEAU$_1$ for $n = 4$ is shown in Fig. 3. The search is directed to the all-zero string as long as no search point with objective value at least $n + 1$ has been produced. This has the effect for simple randomized search heuristics such as the $(1 + 1)$ EA that after having reached the plateau the Hamming distance to the optimal search point is large. Nevertheless, the structure of the plateau admits a fair random walk. The following theorem shows an expected optimization time of $\Theta(n^3)$.

Theorem 1: The expected running time of the $(1 + 1)$ EA on PLATEAU$_1$ is $\Theta(n^3)$.

Proof: As the relative structure of PLATEAU$_1$ and SPC$_n$ (as defined in [16]) are identical besides the inclusion of 0^n in the plateau or not, we can reuse all ideas used in the proof of [16] for the expected running time $O(n^3)$ of the $(1 + 1)$ EA on SPC$_n$. Therefore, also on PLATEAU$_1$ the expected running time of the $(1 + 1)$ EA can be bounded by $O(n^3)$. To the
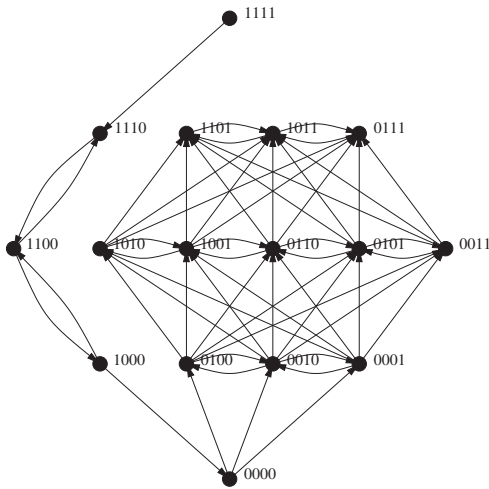
Fig. 3.   Relation graph for the objective function $\text{PLATEAU}_1 : \{0, 1\}^4 \rightarrow \mathbb{R}$. Reflexive and transitive edges are omitted for clarity.
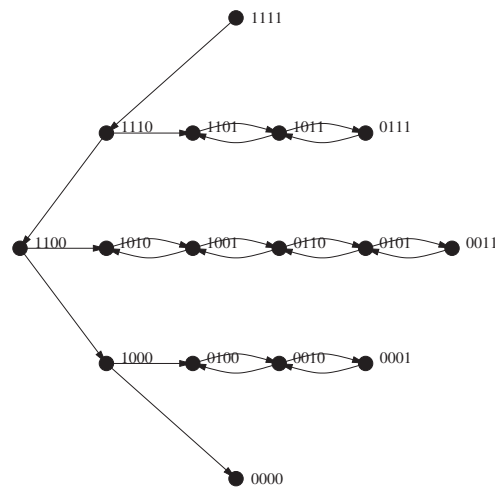

Fig. 4.   Relation graph for the bi-criterion problem $\text{PLOM}: \{0, 1\}^4 \rightarrow \mathbb{R}^2$. Reflexive and transitive edges are omitted for clarity.

best of our knowledge, there is, up to now, no matching lower bound in the literature.

We will now prove a lower bound of $\Omega(n^3)$. In the initialization step of the $(1 + 1)$ EA , a solution $x \in \{0, 1\}^n$ is produced that fulfills $|x|_1 \leq (2/3)n$ with probability $1 - o(1)$ by Chernoff bounds. As long as the current solution is not in SP_1 and not equal to 0^n, the value $|x|_1$ is nonincreasing. Thus, the first individual x chosen by the $(1 + 1)$ EA that is in the set SP_1 has the property $|x|_1 \leq (2/3)n$ with probability $1 - o(1)$. Once the current search point is in the set SP_1, only children also from the set SP_1 are accepted. Hence, only the following mutations are allowed for an accepted mutation step. The first components of x that are 0's or the last components of x that are 1's can be flipped. The probability to flip four or more components in an accepted step is at most $\sum_{i=4}^{n} 2(1/n)^i (n - 1/n)^{n-i} = O(n^{-4})$. Thus, with probability $1 - o(1)$ no such mutation will be accepted in time $\Theta(n^3)$. The probability for a mutation step consisting of three flips to be accepted is at most $2(1/n)^3 (n - 1/n)^{n-3} = O(n^{-3})$. With probability $1 - o(1)$ there will be only a constant number of such mutation steps in time $\Theta(n^3)$. By the same arguments, there are only $O(n)$ accepted mutation steps with exactly two flips and only $O(n^2)$ accepted mutation steps with exactly one flipped bit in time $\Theta(n^3)$. Therefore, in time $\Theta(n^3)$ the two and three-bit flip mutations can only decrease the Hamming distance of the current search point x to the point 1^n by at most $O(n^{1/2})$ with probability $1 - o(1)$, since the two bit flip mutations and the three-bit flip mutations both perform a random walk on the line SP_1. Thus, the search point has to cover a distance of order $\Theta(n)$ by one-bit flip mutations. This takes $\Theta(n^2)$ accepted one-bit flips with probability $1 - o(1)$ using similar arguments as in [8]. Since the expected time for an accepted one-bit flip is $\Theta(n)$, the time until the $(1 + 1)$ EA has reached the search point 1^n is $\Omega(n^3)$. ∎

The analyses of variants of the $(1 + 1)$ EA in [8], [13], [24] point out that some of the well-known combinatorial optimization problems such as maximum matching or Eulerian cycle have natural objective functions where plateaus have

a similar structure as in PLATEAU_1. Therefore, this function plays a key role when considering the behavior of randomized search heuristics on plateaus and understanding the effect of adding objectives to that function may lead to more efficient search heuristics by using additional objectives.

We investigate the effect of adding two of the simplest nontrivial objective functions to the problem and consider the behavior of Global SEMO on these functions.

Namely, we consider the bi-objective problems

$$\text{PLOM}(x) := (\text{PLATEAU}_1(x), |x|_1)$$
$$\text{PLZM}(x) := (\text{PLATEAU}_1(x), |x|_0)$$

and show that Global SEMO is faster (Theorem 2) on PLOM and exponentially slower on PLZM (Theorem 4) compared to the $(1 + 1)$ EA on PLATEAU_1. Note that the optimum of PLATEAU_1 is included in the Pareto-optimal sets of PLOM and PLZM. In addition, the Pareto fronts of the bi-objective problems PLOM and PLZM are of constant size 1, and 2, respectively. According to [18], multiobjectivization only makes sense if the single-objective optimum is included in the not too large Pareto front of the new problem which is given for both PLOM and PLZM.

We now consider Global SEMO on the problem PLOM. The first observation is that all $x \in SP_1$ are comparable in PLOM while they are indifferent in PLATEAU_1. The second objective $|x|_1$ of PLOM also gives the Global SEMO the "right direction" to move on the former plateau $(n + 1, \cdot)$ up to the only Pareto optimum 1^n. This can be seen nicely in the relation graph of PLOM in Fig. 4. The following theorem shows that Global SEMO is indeed significantly faster on PLOM than $(1 + 1)$ EA on PLATEAU_1.

Theorem 2: The expected optimization time of Global SEMO on PLOM is $\Theta(n^2 \log n)$.

Proof: The single Pareto optimum of PLOM is 1^n with the corresponding objective vector $(n + 2, n)$. The population size is bounded by $O(n)$ as each objective function attains at most $n + 3$ different values. If the initial random $x \in \{0, 1\}^n$ is in SP_1, Global SEMO will walk along the objective vectors

$(n + 1, \cdot)$ up to 1^n in expected $O(n^2 \log n)$ steps. This follows from the Coupon Collector's Problem [22] and the fact that in each step the algorithm chooses with probability $\geq 1/n$ the uppermost search point of SP_1. If the initial solution is not in SP_1, Global SEMO produces solutions that tradeoff between the number of 1s and 0s. In this case, we consider the number of steps until a solution with objective vector $(n + 1, \cdot)$ is included or solution 1^n is found. Since the population size is bounded by $O(n)$, the expected number of steps to go from an x with $|x|_1 = k$ to an x' with $|x'|_1 = k + 1$ is $O(n \cdot n/(n - k))$. Therefore, after $O(n^2 \sum_{k=1}^{n} 1/k) = O(n^2 \log n)$ steps, the single Pareto-optimal search point 1^n is found. For the proof of the lower bound we make a slight modification to the Global SEMO model and argue afterwards why this is admissible. We assume that every newly generated child is accepted by Global SEMO . This is indeed the case in the phase until Global SEMO has determined the first solution x with $\text{PLATEAU}_1(x) > n$. We will show that with probability $1 - o(1)$ the modified model is not different from Global SEMO in the phase we are analyzing. Since the initial individual is uniformly distributed in $\{0, 1\}^n$ and the mutation step produces from a uniformly distributed parent a uniformly distributed child, every element x having $|x|_1 = i$ that is newly generated in our modified model is uniformly distributed in $\{x \in \{0, 1\}^n, |x|_1 = i\}$. For every such element x the probability that it is mutated to an element x' with $|x'|_1 = i$ and $\text{PLATEAU}_1(x') \geq n$ is exactly $1/\binom{n}{|x'|_1}$. Thus, the probability that every produced $x' \in \{0, 1\}^n$ with $3 \leq |x'|_1 \leq n - 3$ in $\Theta(n^2 \log n)$ steps fulfills $\text{PLATEAU}_1(x') < n$ is at least

$$\left(1 - \frac{1}{\binom{n}{3}}\right)^{O(n^2 \log n)} = 1 - o(1).$$

In other words, with probability $1 - o(1)$ our new model behaves in $\Theta(n^2 \log n)$ steps exactly like Global SEMO and produces no solution x' with $\text{PLATEAU}_1(x') > n$ as long as every element x of the current population fulfills $3 \leq |x|_1 \leq n - 3$. For the lower bound proof it is enough to restrict ourselves to this fraction of cases.

Let $a_{\min} := \min_{x \in P} \min\{|x|_0, |x|_1\}$, where P denotes the current population. So $a_{\min}$ is the minimal number of 1s respectively 0s of an individual in the current population. Until the first individual $x \in \{0, 1\}^n$ with $\text{PLATEAU}_1(x) > n$ is produced, the value $a_{\min}$ is decreasing and the population size is increasing. After the initialization, $a_{\min} \geq n/3$ holds with high probability (w. h. p.) using Chernoff bounds [22]. We regard the phase where $a_{\min}$ is in the range between $n/3$ and $n/4$ and show that the population size after this phase is of order $\Theta(n)$ with probability at least $1/2$.

Let us consider only steps that decrease $a_{\min}$. We show that the expected decrease of $a_{\min}$ in all such steps in the phase $a_{\min} \in [n/4, n/3]$ is bounded by 2. To obtain from a step that decreases $a_{\min}$ by i a step that decreases $a_{\min}$ by $i + 1$, one of the remaining (at most $n/3$) 1s respectively 0s has to be flipped. The probability for this extra flip is at most $(n/3)/n = 1/3$. Thus, the expected decrease of $a_{\min}$ in such steps is at most 2 (geometric series). Therefore, the average decrease of $a_{\min}$ in the phase $a_{\min} \in [n/4, n/3]$ is larger than 4

with probability less than $1/2$. It follows that with probability at least $1/2$ the population size is $\Theta(n)$ when having obtained for the first time a solution with at most $n/4$ 1s respectively 0s. With high probability, $a_{\min}$ is greater or equal $2 n^{1/4}$ at this time. In other words, we can assume that there are at least $2 n^{1/4}$ 1s respectively 0s left in every element of the current population of size $\Theta(n)$.

For every x in the current population, we define $a(x) := \min\{|x|_0, |x|_1\}$. Now we consider the time to reduce $a_{\min}$ from $n^{1/4}$ to 3. The probability to produce from a solution y with $a(y) > a_{\min} + n^{1/4}$ an improving z is of order $O(n^{-n^{1/4}})$ and therefore such an event does not happen within a polynomial number of steps with probability close to 1. We call a step a k-step iff it creates a solution z with $|z|_1 > |x|_1$ by flipping k of the remaining 0-bits respectively the remaining 1-bits. The probability to flip k of these bits in a single mutation step of a solution y with $a(y) \leq a_{\min} + n^{1/4}$ is upper bounded by $((a_{min} + n^{1/4})/n)^k = O(n^{-3k/4})$. Since the probability that a y with $a_{\min} \leq |y|_1 \leq a_{\min} + n^{1/4}$ will be chosen for the mutation from the current population is of order $O(n^{-3/4})$, the probability for a k-step mutating a y from that region is $O(n^{-3(k+1)/4})$. Hence, for $k \geq 2$ this does not happen within $\Theta(n^2 \log n)$ steps with probability $1 - o(1)$ using Markov's inequality. This implies that with probability $1 - o(1)$ a solution z with $a(z) < a_{\min}$ can only be produced by mutating the at most 2 elements of the population with a–value $a_{\min}$. The expected time to reduce the current $a_{\min}$ to $a_{\min} - 1$ by one step under the condition that an x with $a(x) = a_{\min}$ has been chosen for mutation is n/a_{min}. Thus, the expected time to reduce the value $a_{\min}$ from $n^{1/4}$ to 3 is of the order

$$\Theta(n) \sum_{r=4}^{n^{1/4}} \frac{n}{r} = \Theta(n^2 \log n).$$

This shows that the expected time until the first $x \in \{0, 1\}^n$ with $\text{PLATEAU}_1(x) > n$ is determined by Global SEMO is $\Omega(n^2 \log n)$, which completes the proof. ∎

Using the asymmetric mutation operator, the function PLATEAU_1 becomes much harder. Jansen and Sudholt [15] have shown that the probability that $(1 + 1)$ EA_{asy} optimizes PLATEAU_1 in $2^{O(n^{1/4})}$ steps is bounded above by $2^{-\Omega(n^{1/4})}$. In contrast to this, the search gets easier for Global SEMO_{asy} on PLOM.

Theorem 3: The expected optimization time of Global SEMO_{asy} on PLOM is $\Theta(n^2)$.

Proof: First assume that the population contains an element $x \in \{1^i 0^{n-i}, 1 \leq i \leq n\}$. For such an element x, Global SEMO_{asy} behaves on PLOM like the $(1 + 1)$ EA_{asy} on $|x|_1$. According to [15], $(1 + 1)$ EA_{asy} needs an expected time of $O(n)$ to optimize $|x|_1$. As the population size is at most $O(n)$, the optimum is reached after an expected number of $O(n^2)$ steps.

Now assume that we start with an element $x \notin \{1^i 0^{n-i}, 1 \leq i \leq n\}$. We will analyze the expected number of steps to reach the optimum assuming that no element from $\{1^i 0^{n-i}, 1 \leq i \leq n\}$ enters the population. Otherwise, we already know that we need at most an additional number of $O(n^2)$ steps in expectation to reach the optimum. To mutate an element x

towards the optimum, a mutation which flips no one-bit and at least one zero-bit can be used. The probability that such a mutation happens for a given x is

$$p(x) := \left(1 - \frac{1}{2|x|_1}\right)^{|x|_1} \left(1 - \left(1 - \frac{1}{2|x|_0}\right)^{|x|_0}\right).$$

Since

$$\frac{1}{2} \leq \left(1 - \frac{1}{2k}\right)^k \leq e^{-1/2}$$

we can bound this probability by $p(x) \geq ((1 - e^{-1/2})/2)$. As two elements $x, y \in (\{0,1\}^n \setminus \{1^i 0^{n-i}, 1 \leq i \leq n\})$ with $|x|_0 \neq |y|_0$ do not dominate each other, as soon as a mutation creates an element with k 1s, the population will contain one such element until the end of the algorithm. Hence, we need an expected number of

$$O\left(n \cdot \sum_{i=0}^{n-1} \frac{2}{1 - e^{-1/2}}\right) = O(n^2)$$

steps to reach the optimum, as a specific element of the population is picked with probability $\Omega(1/n)$.

The proof of the lower bound largely follows the proof of Theorem 2. Again we show that the population size is linear after $a_{\min}$ first leaves $[n/4, n/3]$ by ensuring that the expected decrease of $a_{\min}$ is constant in this interval. For this, observe that to obtain from a step that decreases $a_{\min}$ by i a step that decreases it by $i + 1$, one of the remaining 1s (respectively 0s) has to be flipped. The probability for this flip is at most $(n/3)(1/2(n/4)) = 2/3$, which then leads to a constant expected decrease in each step. This in turn shows a linear population size after this phase.

Hence, when $a_{\min}$ leaves this interval, the population P is w. h. p. of size $|P| = \Theta(n)$. Also w. h. p., we have $a_{\min} \geq n/5$. Now consider the probability p_{ij} to produce in the next mutation step from x with $|x|_0 = i$ a solution x' with $|x'|_0 = j$ where $i < j$. Jansen and Sudholt [15] have shown that $p_{ij} \leq 2^{j-i}$. Let $x \in P$ be the solution with the largest number of zeros. Denote by $D = |x|_0$ the distance of x to 0^n. Consider a solution $y \in P$ with $|y|_0 = |x|_0 + k$, where $k \in [0, |x|_1]$. A mutation step of y reduces D in expectation by at most

$$\sum_{i=k+1}^{n} 2^{-i}(i - k) < 2^{-k} \sum_{i=1}^{\infty} 2^{-i} i = 2^{-k+1}.$$

Then the expected decrease of D in the next mutation step is at most

$$\frac{1}{|P|} \sum_{k=0}^{n} 2^{-k+1} < \frac{2}{|P|} \sum_{k=0}^{\infty} 2^{-k} = \frac{4}{|P|}$$

as P contains at most one individual with k 0s for each k. Since $|P| = \Theta(n)$ holds, the expected decrease of D in each iteration is at most $O(1/n)$. Hence, $\Omega(n^2)$ iterations are necessary to reduce the value of D by $\Theta(n)$ which completes the proof. ∎

It remains to examine the problem PLZM. An exponential deceleration comes from the $x \in SP_1$. These search points are now comparable in PLZM, but this time the second objective $|x|_0$ of PLZM is leading Global SEMO and Global SEMO$_{asy}$ in

the opposite direction of the Pareto optimum 1^n. The following theorem shows the more than clear effect of adding the "wrong objective."

Theorem 4: The optimization times of Global SEMO and Global SEMO$_{asy}$ on PLZM are $e^{\Omega(n)}$ with probability $1 - e^{-\Omega(n)}$.

Proof: The objective vectors $(n + 2, 0)$, (n, n), and $(n + 1, n - 1)$ with the corresponding search points 1^n, 0^n, and 10^{n-1} are the three Pareto optima of PLZM. We show that the claimed lower bound holds for obtaining the search point 1^n. The initial search point consists with probability $1 - e^{-\Omega(n)}$ of at most $2n/3$ 1s using Chernoff bounds. Accepted steps increasing the number of 1s have to produce a solution of $SP_1 \cup \{1^n\}$. The probability to reach 1^n directly from a search point $x \notin SP_1$ is upper bound by $2^{-n/3}$ for both algorithms as all 0-bits have to be flipped. The other opportunity to obtain the search point 1^n is to produce it from a search point of SP_1. The first solution of SP_1 found during the run of the algorithm has with probability $1 - e^{-\Omega(n)}$ at most $3n/4$ 1-bits as the probability of flipping $\Theta(n)$ bits in a single mutation step is $e^{-\Omega(n)}$ for both algorithms. Afterwards, the number of 1s can only be increased by producing the search point 1^n directly. As each individual in the population has at most $3n/4$ 1s with probability $1 - e^{-\Omega(n)}$, the probability of obtaining 1^n is upper bounded by $2^{-n/4}$ for both algorithms. Hence, overall the time to achieve the search point 1^n is $e^{\Omega(n)}$ with probability $1 - e^{-\Omega(n)}$. ∎

V. COPING WITH TWO PLATEAUS

In Section IV, the added objectives were easy to solve individually for the $(1 + 1)$ EA. The main reason for the smaller running time of PLOMAS compared to PLATEAU$_1$ is that both functions have the same global optimum. The question arises whether combining two objectives may result in a faster optimization process than optimizing the different objective functions separately. We show that the combination of two equally complex problems yields an easier problem if both functions are optimized as a bi-criterion problem.

We know from Theorem 1 that Global SEMO has an expected running time of $\Theta(n^3)$ on PLATEAU$_1$. Let $SP_2 := \{0^i 1^{n-i}, 1 \leq i < n\}$; then this result also holds for the function

$$\text{PLATEAU}_2(x) = \begin{cases} |x|_1 & : \ x \notin SP_2 \\ n + 1 & : \ x \in SP_2 \\ n + 2 & : \ x = 0^n \end{cases}$$

due to the symmetry with PLATEAU$_1$. We now consider the multiobjective function

$$\text{PLATEAUS} = (\text{PLATEAU}_1(x), \text{PLATEAU}_2(x))$$

where Global SEMO has to cope with a plateau in each objective and show that this is easier than solving the single-objective problems separately.

Theorem 5: The expected optimization time of Global SEMO on PLATEAUS is $\Theta(n^2 \log n)$.

Proof: The objective vectors $(n + 2, n)$ and $(n, n + 2)$ with the corresponding search points 1^n and 0^n are Pareto-optimal, as they are the optima of the two objective functions

PLATEAU$_1$ and PLATEAU$_2$. There does not exist an objective vector $(n+1, n+1)$ for the considered problem which shows that the search points 1^n and 0^n are the only Pareto-optimal ones.

The population size is always bounded by $O(n)$ as each objective function attains at most $n + 3$ different values. We consider the number of steps until solutions with objective vectors $(n + 1, \cdot)$ and $(\cdot, n + 1)$ have been included into the population and assume that the Pareto-optimal solutions with objective vectors $(n + 2, n)$ and $(n, n + 2)$, respectively, have not been obtained before. We investigate the case to obtain $(n + 1, \cdot)$. As long as such a solution has not been obtained, we consider the solution x with the largest PLATEAU$_1$ value in the population. This is determined by the number of zeros in x. Assume that $|x|_0 = k$ holds. Then, the probability to produce from x a solution x' with a higher number of zeros is at least $(n-k)/(en)$. The probability of choosing x in the next step is $\Omega(1/n)$. Hence, the number of zeros increases after an expected number of $O(n^2/(n - k))$ steps. Summing up over the different values of k, the search point 0^n with objective vector $(n, n + 2)$ has been obtained after $O(n^2 \log n)$ steps if no solution with objective vector $(n + 1, \cdot)$ has been produced before. Flipping the first bit in 0^n leads to a solution with objective vector $(n+1, \cdot)$ and can be obtained in an additional phase of $O(n^2)$ steps. The expected time to obtain a solution with objective vector $(\cdot, n+1)$ can be bounded by $O(n^2 \log n)$ using the same arguments.

After P includes solutions with objective vectors $(n + 1, \cdot)$ and $(\cdot, n + 1)$ or a subset of Pareto-optimal solutions dominating these vectors, the population size is always bounded by 2. We consider how to obtain the search point 1^n. Let x be the search point with objective vector $(n + 1, k)$ in the population. Flipping the bit x_{k+1} in x leads to a solution x' with objective vector $(n + 1, k + 1)$. The population size is at most 2 and the probability of flipping one single specific bit is at least $1/(en)$, which implies that the expected waiting time for such a step is $O(n)$. The value of k will be increased at most $n - 1$ times until the search point 1^n has been included into P. Hence, the expected time until this solution has been obtained is $O(n^2)$. The same holds for including the search point 0^n using the same arguments. Altogether, the expected optimization of Global SEMO on PLATEAUS is $O(n^2 \log n)$.

The lower bound proof is analogous to the lower bound proof of Theorem 2, since the functions PLATEAUS and PLOM are the same on the set $\{0, 1\}^n \setminus (\{0^i 1^{n-i}, 0 \le i \le n\} \cup \{1^i 0^{n-i}, 0 \le i \le n\})$. ∎

Jansen and Sudholt [15] have shown that the $(1 + 1)$ EA$_{asy}$ is totally inefficient on PLATEAU$_1$. The same arguments hold for PLATEAU$_2$ as it differs from PLATEAU$_1$ only by exchanging the roles of zeros and ones. Surprisingly, this does not hold for Global SEMO$_{asy}$ and PLATEAUS. In the following, we show that Global SEMO$_{asy}$ is quite efficient on PLATEAUS.

Theorem 6: The expected optimization time of Global SEMO$_{asy}$ on PLATEAUS is $\Theta(n^2)$.

Proof: As in the proof of Theorem 5, we first bound the expected number of steps until the population includes search points with objective vectors $(n + 1, \cdot)$ and $(\cdot, n + 1)$ and assume that the Pareto-optimal objective vectors $(n+2, n)$

respectively $(n, n + 2)$ have not been obtained before. For obtaining $(n + 1, \cdot)$, consider the search point x with the largest PLATEAU$_1$ value. Assume that it has $|x|_0 = k$ zeros. The probability to obtain from x a solution with more zeros can be bounded by $(1 - e^{-1/2})/2$, as shown in the proof of Theorem 3. Summing this up for all values of k and using the fact that the population size is always bounded by $O(n)$, a solution with objective vector $(n + 1, \cdot)$ is obtained after an expected number of $O(n^2)$ steps. By symmetry, the same holds for obtaining a search point with objective vector $(\cdot, n + 1)$.

Now assume that two search points with objective vectors $(n + 1, \cdot)$ and $(\cdot, n + 1)$ are included in the population. Since they dominate all other points, the population size is upper bounded by 2 in this case. If the objective vector of the first search point is $(n + 1, k)$, it consists of k ones followed by $n - k$ zeros. Its objective vector can be improved by flipping the $(k + 1)$th zero to one. The probability for this to happen is

$$p(x) = \left(1 - \frac{1}{2k}\right)^k \left(\frac{1}{2(n - k)}\right) \left(1 - \frac{1}{2(n - k)}\right)^{n-k-1}$$

which can be bounded by

$$p(x) \ge \frac{1}{2} \frac{1}{2(n - k)} \left(1 - \frac{1}{2(n - k)}\right) \frac{1}{2} = \Omega\left(\frac{1}{n}\right).$$

Hence, after an expected number of $O(n^2)$ steps the objective vector will reach $(n+2, n)$. By symmetry, the same holds for obtaining the search point with objective vector $(n, n+2)$.

The lower bound proof can be done analogously to the lower bound proof of Theorem 3, since the functions PLATEAUS and PLOM are the same on the set $\{0, 1\}^n \setminus (\{0^i 1^{n-i}, 0 \le i \le n\} \cup \{1^i 0^{n-i}, 0 \le i \le n\})$. ∎

VI. EXPERIMENTAL STUDIES

In the previous sections, we investigated plateaus of indifferent solutions in single-objective problems and examined how an additional objective changes the dominance relation on this plateau and therefore influences the running time for simple algorithms like the Global SEMO and the $(1 + 1)$ EA.

With the following experimental study, we want to tackle three questions that remain open after our theoretical investigations: 1) can the asymptotical results also be observed for small instances; 2) can the effect of making a problem harder or easier by adding an objective be reported for a multiobjective problem instead of the single-objective PLATEAU$_1$ and; 3) can we observe the same behavior also on other types of plateaus, e.g., sets of incomparable solutions?

In the following, we investigate experimentally for both multiobjective problems and plateaus of incomparable solutions whether the running time of Global SEMO can be increased and decreased with an additional objective. First, we investigate a bi-objective problem with the same plateau SP_1 that was considered above, whereas Section VI-B shows that an addition of objectives can increase or decrease the running time of Global SEMO also for other kinds of plateaus. The general explanation of what happens remains the same as in the previous sections: if plateaus are introduced by an additional objective, the running time increases if a good

direction on the search points vanishes and decreases if a deceptive direction vanishes; if plateaus are eliminated by adding a direction to the corresponding search space region, the new direction increases or decreases the running time depending on whether the introduced direction is deceptive or not.

A. Similar Plateaus With More Objectives

First, we investigate the influence of the addition of a third objective to a bi-objective problem, based on the two functions

$$\text{LEADINGONES}(x) = \sum_{i=1}^{n} \prod_{j=1}^{i} x_j$$

and

$$\text{TRAILINGZEROS}(x) = \sum_{i=1}^{n} \prod_{j=i}^{n} (1 - x_j)$$

which were first investigated in [19] as the problem LOTZ. Here, we consider the slightly changed functions

$$f_1(x) = \begin{cases} \text{LEADINGONES}(x) & \text{if } x \notin SP_1 \\ n+1 & \text{if } x \in SP_1 \\ n+2 & \text{if } x = 1^n \end{cases}$$

and

$$f_2(x) = \begin{cases} \text{TRAILINGZEROS}(x) & \text{if } x \notin SP_1 \\ n+1 & \text{if } x \in SP_1 \\ n+2 & \text{if } x = 1^n \end{cases}$$

that are to be maximized at the same time leading to the modified LOTZ problem (f_1, f_2) where $SP_1 = \{1^i 0^{n-i}, 1 \leq i < n\}$ as defined above. Note, that $f_1(x) = f_2(x)$ holds if $x \in SP_1$. In this case, we have to cope with the same plateau as given by the function PLATEAU$_1$. The only difference between the bi-objective problem (f_1, f_2) and the function PLATEAU$_1$ is given by the search points not on the plateau. Here, the population of Global SEMO may grow due to a number of incomparable solutions.

Starting with the modified LOTZ problem (f_1, f_2), we investigate the effect of adding either the function $|x|_1$ or the function $|x|_0$ to the problem. Adding $|x|_1$ decreases the running time of Global SEMO, whereas adding $|x|_0$ increases it. The effect is caused by the same principle observed in Section IV. Right before finding the Pareto-optimal front, Global SEMO has to overcome the plateau $1^i 0^{n-i}$ $(1 \leq i < n)$ of indifferent solutions. If $|x|_1$ is added, this third objective induces a direction to the optimum on this plateau; if $|x|_0$ is added, the generated direction on the plateau is deceptive.

Fig. 5 shows the box plots of the running times of 31 independent Global SEMO runs on all three problems for different bitstring lengths ($n \in \{5, 10, 15, 20, 25, 30\}$).[2] The

[2]The boxplots have been produced by the built-in boxplot command of MATLAB showing the lower quartile, median, and upper quartile values. The default maximum whisker length of 1.5 times the interquartile range has been used. Data points lying beyond the ends of the whiskers are marked by a "+."
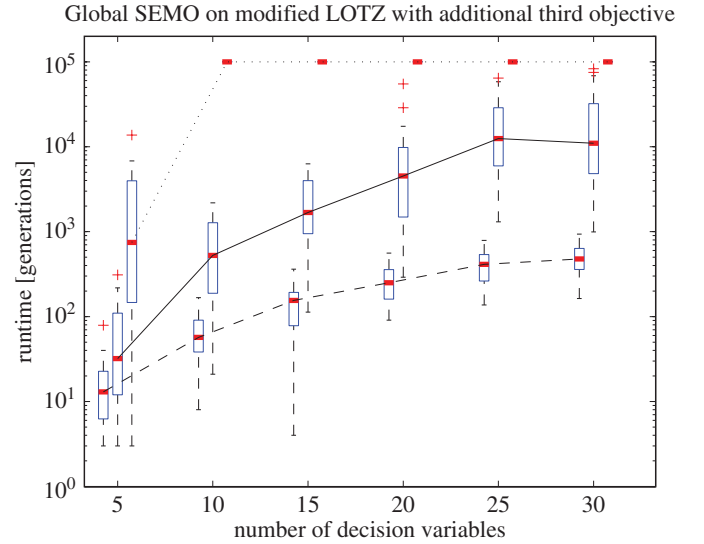

Fig. 5. Comparison of the running times for Global SEMO if a third objective is added to the modified LOTZ problem: original problem (solid line) (f_1, f_2), (dashed line) $(f_1, f_2, |x|_1)$, and (dotted line) $(f_1, f_2, |x|_0)$. Note, that the runs are aborted if no Pareto-optimal point has been found in the first 100 000 generations. For clarity, the three boxplots corresponding to a specific number of decision variables have been slightly shifted horizontally.

nonparametric Kruskal–Wallis test with the extension to multiple comparisons[3] has been performed to support the above-stated hypotheses that Global SEMO needs more time for optimizing $(f_1, f_2, |x|_0)$ than for (f_1, f_2) and that Global SEMO needs less time for optimizing $(f_1, f_2, |x|_1)$ than for the original problem (f_1, f_2). The null hypothesis of equal distributions was rejected at the significance level of 0.01 for all considered decision space sizes supporting the visual illustration of Fig. 5. Note that the runs were aborted if no Pareto-optimal search point has been found in the first 100 000 generations. This and the large variance of the single runs explain the unexpected decrease of the median between the original problem with 25 and the one with 30 decision variables.

B. Different Kinds of Plateaus

In addition to plateaus of indifferent solutions which occur frequently in single-objective problems, multiobjective problems may exhibit plateaus of incomparable solutions as well. In this section, we investigate the running time changes of Global SEMO for both kinds of plateaus if an objective is added.

1) Plateaus of Indifferent Solutions: The basis bi-objective problem we use for the investigation of plateaus of indifferent solutions is the original LOTZ of [19]. All solutions with the same number of leading 1s and trailing 0s are mapped to the same objective function values yielding a plateau of indifferent solutions. In the following, we will refer to the decision variables that neither belong to the leading 1s nor to the trailing 0s of a solution x as the middle block x_M. In addition, $|x|$ denotes the length of the bitstring x. Adding

[3]As implemented in the PISA performance assessment toolbox [1] and described in [6, p.290].
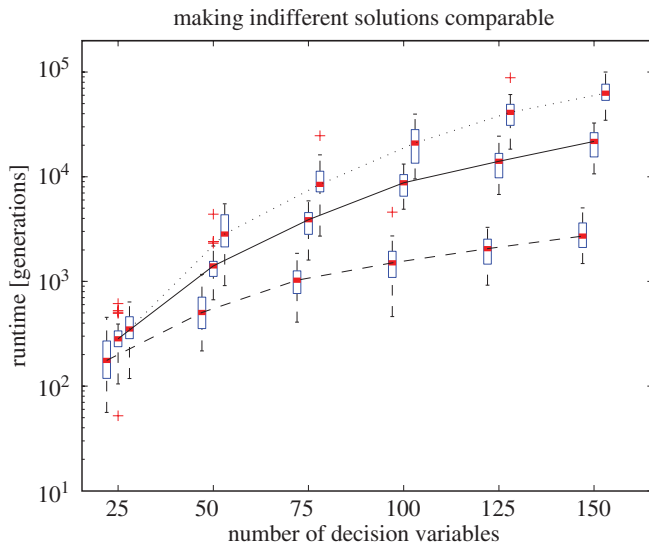
making indifferent solutions comparable

Fig. 6. How a problem can become either easier or harder if indifferent solutions are made comparable, i.e., if a plateau is removed by an additional objective. Here, the boxplots for the running times until the first Pareto-optimal point is reached are shown. Note, that the runs are aborted if no Pareto-optimal point has been found in the first 100 000 generations. For clarity, the three boxplots corresponding to a specific number of decision variables have been slightly shifted horizontally.
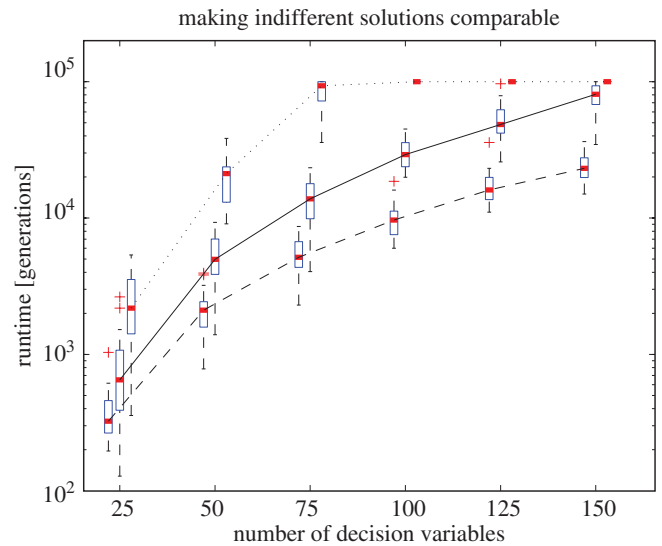
making indifferent solutions comparable

Fig. 7. How a problem can become either easier or harder if comparable solutions are made incomparable, i.e., if a plateau is introduced by an additional objective. Here, the boxplots for the running times until the first Pareto-optimal point is reached are shown. Note that the runs are aborted if no Pareto-optimal point has been found in the first 100 000 generations. For clarity, the three boxplots corresponding to a specific number of decision variables have been slightly shifted horizontally.

objectives that take into account only the bits in the middle blocks of solutions will give a direction to these plateaus of indifferent solutions. Depending on whether this direction is deceptive or not, the running times of Global SEMO on the corresponding three-objective problem will be higher or lower than for the original bi-objective problem.

Minimizing the objective $g_{(i)}$ in addition to LOTZ $= (g_1, g_2)$ (as defined in Fig. 8) increases the running time due to its deceptive behavior. On the one hand, the number of leading 1s and trailing 0s has to be maximized to reach the Pareto-optimal front; on the other hand, the additional objective $g_{(i)}$ rewards a higher number of leading 0s in the middle block as well as a higher number of trailing 1s. This forces Global SEMO to flip more or less all bits in the middle block at least once instead of benefiting from already correctly set bits, i.e., the running time increases. In contrast, the additional minimization of objective $g_{(ii)} = |x_M|_1$, or in other words the maximization of zeros in the middle block, will flip bits of the middle block to zeros also if they do not contribute directly to the maximization of g_1 and g_2. However, with the middle block's bits that are already set to zero, Global SEMO is able to perform big jumps in the objective function value of g_2 in future steps, i.e., the running time decreases.

To support the above-mentioned hypothesis that the addition of $g_{(i)}$ increases and the addition of $g_{(ii)}$ decreases the running time of Global SEMO in comparison with the original problem (g_1, g_2), 31 independent runs of Global SEMO were performed for different numbers of decision variables ($n \in \{25, 50, 75, 100, 125, 150\}$). Note that we measured the number of generations until the first Pareto-optimal point has been found by Global SEMO instead of the normal running time. The reason for that is the already high number of generations that are needed to find the first Pareto-optimal

point which forced us to restrict the number of generations to 100 000: if Global SEMO did not find any Pareto-optimal point within the first 100 000 generations, we stopped the run and noted 100 000 as the run's optimization time. Fig. 6 shows the corresponding box plots. The nonparametric Kruskal–Wallis test for multiple comparisons of [6] again rejects the null hypothesis of equal distributions for all comparisons at a significance level of 0.01 except for the comparison between the running times for $(g_1, g_2, g_{(i)})$ and (g_1, g_2) with 25 decision variables where the p-value is approximately 0.033.

2) Plateaus of Incomparable Solutions: It remains to show that also for problems with plateaus of incomparable solutions, the addition of objectives can change the running time of an EA in both ways. To this end, the problem (h_1, h_2) as defined in Fig. 8 is investigated. The objective space of (h_1, h_2) can be illustrated as the objective space of the original LOTZ problem where the levels 2, 3, 6, 7, 10, 11, and so forth are mirrored at the origin and then translated. Fig. 9 illustrates this problem exemplary for a small number of decision variables.

The change of the original LOTZ problem to (h_1, h_2) turns around the Pareto dominance relation between the mirrored levels: where the Pareto dominance relation is indicating the direction to the optimum in LOTZ, the new search space direction is deceptive. Global SEMO has to jump out of the newly introduced local optima by at least a two-bit flip. This is where a third objective can help. By making the solutions within the region with deceptive Pareto dominance relation incomparable, Global SEMO is able to perform a random walk on newly introduced plateaus of incomparable solutions. If on the other hand, solutions where the Pareto dominance relation points in direction to the Pareto-optimal front are made incomparable, Global SEMO needs more time to find the Pareto-optimal front than for the original problem (h_1, h_2).

base function g [19]

$$\max \quad g_1(x) = \text{LEADINGONES}(x)$$
$$\max \quad g_2(x) = \text{TRAILINGZEROS}(x)$$

slower with

$$\min \quad g_{(i)}(x) = |x_M| - \text{LEADINGZEROS}(x_M) - \text{TRAILINGONES}(x_M)$$

faster with

$$\min \quad g_{(ii)}(x) = |x_M|_1$$

base function h

$$\max \quad h_1(x) = \begin{cases} \text{LEADINGONES}(x) & \begin{array}{l} \text{iff} \quad 0 \equiv (n - |x_M|) \bmod 4 \\ \text{or} \quad 1 \equiv (n - |x_M|) \bmod 4 \end{array} \\ -\text{LEADINGONES}(x) - n \cdot \left\lfloor \dfrac{\left\lfloor \frac{n-|x_M|}{2} \right\rfloor}{2} \right\rfloor & \text{else} \end{cases}$$

$$\max \quad h_2(x) = \begin{cases} \text{TRAILINGZEROS}(x) & \begin{array}{l} \text{iff} \quad 0 \equiv (n - |x_M|) \bmod 4 \\ \text{or} \quad 1 \equiv (n - |x_M|) \bmod 4 \end{array} \\ -\text{TRAILINGZEROS}(x) - n \cdot \left\lfloor \dfrac{\left\lfloor \frac{n-|x_M|}{2} \right\rfloor}{2} \right\rfloor & \text{else} \end{cases}$$

slower with

$$\min \quad h_{(i)}(x) = \frac{n}{2} - \left| \frac{n}{2} - |x_M| \right|$$

faster with

$$\min \quad h_{(ii)}(x) = |x_M|$$

Fig. 8. Definitions of the functions illustrating the changes of running time with respect to making indifferent solutions comparable [problems $(g_1, g_2, g_{(\cdot)})$] and making comparable solutions incomparable [problems $(h_1, h_2, h_{(\cdot)})$].
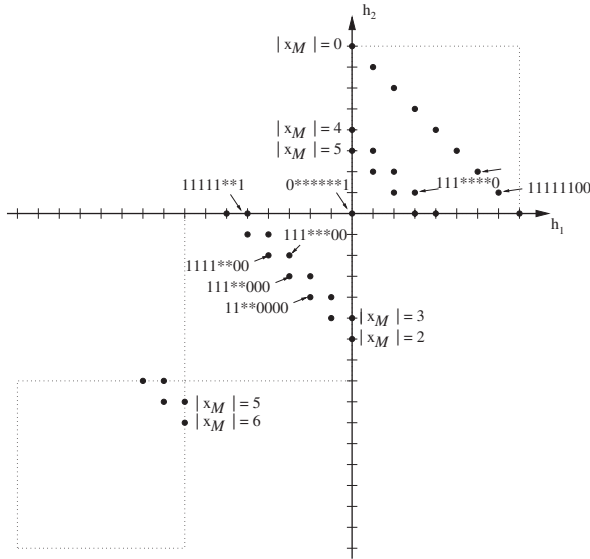

Fig. 9. Illustration of the objective space for the modified LOTZ problem (h_1, h_2) and $n = 8$ decision variables. For some objective vectors, the corresponding solutions in decision space are indicated, where a "*" denotes either a "1" or a "0" on the corresponding bit string position.

The objectives $h_{(i)}$ and $h_{(ii)}$ defined in Fig. 8 are introducing these incomparabilities either on the mirrored levels of LOTZ only ($h_{(ii)}$) or in both the first and third quadrant ($h_{(i)}$). The expected behavior is that the addition of $h_{(i)}$ will increase and the addition of $h_{(ii)}$ will decrease the running time of Global SEMO in comparison to the bi-objective problem.

Fig. 7 shows the boxplots of 31 independent Global SEMO runs for different numbers of decision variables ($n \in \{25, 50, 75, 100, 125, 150\}$). As before, we count the number of generations until the first Pareto-optimal point is found or count 100 000 if no Pareto-optimal point is found within the first 100 000 generations. The visual inspection of the boxplots in Fig. 7 indicates that Global SEMO has a higher average running time on $(h_1, h_2, h_{(i)})$ and a lower average

running time on $(h_1, h_2, h_{(ii)})$, which is supported by the same Kruskal–Wallis test as mentioned before at a significance level of 0.01 for all tested decision space sizes.

VII. CONCLUSION

We have investigated the question of how additional objectives affect 1) the structure, i.e., the dominance relation, of a given optimization problem and 2) the search behavior of evolutionary algorithms applied to this problem. Motivated by previous studies on the relationship between single-objective and multiobjective versions of particular problems ([25], [29]), we have shown that one and the same problem can be made both easier and more difficult solely by adding different objectives, i.e., without changing the search space or one of the existing objectives. In particular, we have provided rigorous running time analyses in order to prove that in the extreme case the effect of adding objectives can make the difference between a polynomial and an exponential running time.

The changes in the running times are due to changes in the dominance structure: whenever a new objective interferes with the existing ones, plateaus of indifferent or incomparable solutions may emerge or vanish, respectively, and may be enlarged or reduced. Therefore, useful or misleading information may be implanted or removed. So, additional objectives can decrease the complexity of a problem, although in general this effect is less likely if more objectives are involved as indicated in [17] and [31].

The presented results have different implications. On the one hand, they can help with the design and the classification of multiobjective benchmark problems according to different categories of hardness. On the other hand, they indicate that domain knowledge may not only be incorporated in terms of problem-specific algorithmic components, but also in the form of additional objective functions. Finally, the insights may be lead to general guidelines on problem transformations from M to N objectives.

REFERENCES

[1] S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler, "Pisa A platform and programming language independent interface for search algorithms," in *Proc. 2nd Int. Conf. Evol. Multi-Criterion Optimization (EMO)*, LNCS vol. 2632. New York: Springer-Verlag, 2003, pp. 494–508.

[2] D. Brockhoff, T. Friedrich, N. Hebbinghaus, C. Klein, F. Neumann, and E. Zitzler, "Do additional objectives make a problem harder?" in *Proc. 9th Annu. Conf. Genetic Evol. Comput. (GECCO)*, New York: ACM, 2007, pp. 765–772.

[3] D. Brockhoff and E. Zitzler, "Are all objectives necessary? On dimensionality reduction in evolutionary multiobjective optimization," in *Proc. 9th Int. Conf. Parallel Problem Solving From Nature (PPSN)*, LNCS vol. 4193. New York: Springer-Verlag, 2006, pp. 533–542.

[4] D. Brockhoff and E. Zitzler, "Dimensionality reduction in multiobjective optimization: The minimum objective subset problem," in *Proc. Operations Res. 2006*, New York: Springer-Verlag, pp. 423–429.

[5] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multiobjective Problems (Genetic and Evolutionary Comput.)* New York: Kluwer Academic, 2002.

[6] W. J. Conover, *Practical Nonparametric Statistics*. 3rd ed. New York: Wiley, 1999.

[7] K. Deb, *Multiobjective Optimization Using Evolutionary Algorithms*. New York: Wiley, 2001.

[8] B. Doerr, N. Hebbinghaus, and F. Neumann, "Speeding up evolutionary algorithms through asymmetric mutation operators," *Evol. Comput.*, vol. 15, no. 4, pp. 401–410, Nov. 2007.

[9] S. Droste, T. Jansen, and I. Wegener, "On the analysis of the $(1 + 1)$ evolutionary algorithm," *Theoretical Comput, Sci.*, vol. 276, no. 1–2, pp. 51–81, Apr. 2002.

[10] P. J. Fleming, R. C. Purshouse, and R. J. Lygoe, "Many-objective optimization: An engineering design perspective," in *Proc. 3rd Int. Conf. Evol. Multi-Criterion Optimization*, LNCS vol. 3410. New York: Springer-Verlag, 2005, pp. 14–32.

[11] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evol. Comput.*, vol. 3, no. 1, pp. 1–16, 1995.

[12] O. Giel, "Expected runtimes of a simple multiobjective evolutionary algorithm," in *Proc. 2003 IEEE Congr. Evol. Comput. (CEC)*, Piscataway, NJ: IEEE Press, pp. 1918–1925.

[13] O. Giel and I. Wegener, "Evolutionary algorithms and the maximum matching problem," in *Proc. 20th Annu. Symp. Theoretical Aspects Comput. Sci. (STACS)*, LNCS vol. 2607. New York: Springer-Verlag, 2003, pp. 415–426.

[14] J. Horn, "Multicriterion decision making," in *Handbook Evol. Comput.*, Oxford, U.K. Oxford Univ. Press, 1997.

[15] T. Jansen and D. Sudholt, "Design and analysis of an asymmetric mutation operator," in *Proc. 2005 IEEE Congr. Evol. Comput. (CEC)*, Piscataway, NJ: IEEE Press, pp. 190–197.

[16] 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. Evol. Comput.*, vol. 5, no. 6, pp. 589–599, Dec. 2001.

[17] M. T. Jensen, "Helper-objectives: Using multiobjective evolutionary algorithms for single-objective optimisation," *J. Math. Modelling and Algorithms*, vol. 3, no. 4, pp. 323–347, 2004.

[18] J. D. Knowles, R. A. Watson, and D. W. Corne, "Reducing local optima in single-objective problems by multiobjectivization," in *Proc. 1st Int. Conf. Evol. Multi-Criterion Optimization*, LNCS vol. 1993. New York: Springer-Verlag, 2001, pp. 269–283.

[19] M. Laumanns, L. Thiele, and E. Zitzler, "Running time analysis of evolutionary algorithms on a simplified multiobjective knapsack problem," *Natural Computing*, vol. 3, no. 1, pp. 37–51, Mar. 2004.

[20] M. Laumanns, L. Thiele, E. Zitzler, E. Welzl, and K. Deb, "Running time analysis of multiobjective evolutionary algorithms on a simple discrete optimization problem," in *Proc. 7th Int. Conf. Parallel Problem Solving from Nature*, LNCS vol. 2439. New York: Springer-Verlag, 2002, pp. 44–53.

[21] E. Mezura-Montes and C. A. Coello Coello, "Constrained optimization via multiobjective evolutionary algorithms," in *Multiobjective Problem Solving from Nature: From Concepts to Applications*, Berlin, Germany: Springer-Verlag, 2007.

[22] R. Motwani and P. Raghavan, *Randomized Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 1995.

[23] F. Neumann, "Expected runtimes of a simple evolutionary algorithm for the multiobjective minimum spanning tree problem," *Eur. J. Oper. Res.*, vol. 181, no. 3, pp. 1620–1629, 2007.

[24] F. Neumann, "Expected runtimes of evolutionary algorithms for the Eulerian cycle problem," *Comput. Operations Res.*, vol. 35, no. 9, pp. 2750–2759, Sep. 2008.

[25] F. Neumann and I. Wegener, "Minimum spanning trees made easier via multiobjective optimization," *Natural Computing*, vol. 5, no. 3, pp. 305–319, Sep. 2006.

[26] F. Neumann and I. Wegener, "Randomized local search, evolutionary algorithms, and the minimum spanning tree problem," *Theoretical Comput. Sci.*, vol. 378, no. 1, pp. 32–40, Jun. 2007.

[27] R. C. Purshouse and P. J. Fleming, "On the evolutionary optimization of many conflicting objectives," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 770–784, Dec. 2007.

[28] T. P. Runarsson and X. Yao, "Search biases in constrained evolutionary optimization," *IEEE Trans. Syst., Man, Cybern., Part C: Applicat. and Rev.*, vol. 35, no. 2, pp. 233–243, May 2005.

[29] J. Scharnow, K. Tinnefeld, and I. Wegener, "The analysis of evolutionary algorithms on sorting and shortest paths problems," *J. Math. Modelling and Algorithms*, vol. 3, no. 4, pp. 349–366, 2004.

[30] T. Wagner, N. Beume, and B. Naujoks, "Pareto-, aggregation-, and indicator-based methods in many-objective optimization," in *Proc. 4th Int. Conf. Evol. Multi-Criterion Optimization (EMO)*, LNCS vol. 4403. New York: Springer-Verlag, 2007, pp. 742–756.

[31] P. Winkler, "Random orders," *Order*, vol. 1, no. 4, pp. 317–331, Dec. 1985.

[32] C. Witt, "Worst-case and average-case approximations by simple randomized search heuristics," in *Proc. 22nd Annu. Symp. Theoretical Aspects Comput. Sci. (STACS)*, LNCS vol. 3404. New York: Springer-Verlag, 2005, pp. 44–56.

Dimo Brockhoff received the Diploma in computer science from the University of Dortmund, Germany, in 2005. He has been working toward a Ph.D. degree under Prof. Zitzler and with the Computer Engineering and Networks Laboratory, Elgenössiche Technische Hochschule Zurich, Zurich, Switzerland.

His research interests are evolutionary computation and, in particular, multiobjective optimization.

Tobias Friedrich received the M.Sc. degree in computer science from the University of Sheffield, U.K. in 2003, the Diploma in mathematics from the University of Jena, Germany, in 2005, and the Ph.D. degree in computer science from the Saarland University, Germany, in 2007.

Since 2008, he is with the Algorithm Group at the International Computer Science Institute, Berkeley, CA. The central topics of his work are randomized methods in mathematics and computer science and randomized algorithms (both classical and evolutionary).

Nils Hebbinghaus received the Diploma and the Ph.D. degree in mathematics from the University of Kiel in 2002 and 2005, respectively.

From October 2005 to September 2007, he was working as a Postdoctoral Fellow with Department 1: Algorithms and Complexity at Max-Planck-Institut für Informatik, Saarbrücken, Germany. Since October 2007, he has been a Consultant for mathematical finance. His research interests are mainly in the field of discrete mathematics. Since October 2007, he has been working as a consultant for mathematical finance.

Christian Klein received the Diploma in computer science from the Saarland University in Saarbrücken, Germany in 2004.

Since 2004, he has been a Ph.D. student with the Department 1: Algorithms and Complexity at Max-Planck-Institut für Informatik, Saarbrücken, Germany. His research interests include randomized algorithms and computational geometry.

Eckart Zitzler received the Diploma in computer science from the University of Dortmund, Germany, and the Doctor of Technical Sciences from Elgenössiche Technische Hochschule (ETH) Zurich, Switzerland.

Since 2003, he has been Assistant Professor for Systems Optimization at the Computer Engineering and Networks Laboratory at the Department of Information Technology and Electrical Engineering, ETH Zurich, Switzerland. His research focuses on bio-inspired computation, multiobjective optimization, computational biology, and computer engineering applications.

Prof. Zitzler was General Co-Chairman of the first three international conferences on evolutionary multicriterion optimization (EMO 2001, EMO 2003, and EMO 2005).

Frank Neumann received the Diploma and the Ph.D. degrees in computer science from the University of Kiel in 2002 and 2006, respectively.

Since November 2006, he has been a Researcher of the Department 1: Algorithms and Complexity Group at the Max-Planck-Institut für Informatik in Saarbrücken, Germany. In his work, he considers theoretical aspects of bio-inspired computation methods, in particular for problems from combinatorial optimization.