Do Additional Objectives Make a Problem Harder?

Dimo Brockhoff*   Tobias Friedrich†   Nils Hebbinghaus†

Christian Klein†   Frank Neumann†   Eckart Zitzler*

† Algorithms and Complexity Group
Max-Planck-Institut für Informatik
Saarbrücken, Germany
firstname.lastname@mpi-inf.mpg.de

* Computer Engineering and Networks Laboratory
ETH Zurich
8092 Zurich, Switzerland
lastname@tik.ee.ethz.ch

ABSTRACT

In this paper, we examine how adding objectives to a given optimization problem affects the computation effort required to generate the set of Pareto-optimal solutions. Experimental studies show that additional objectives may change the runtime behavior of an algorithm drastically. Often it is assumed that more objectives make a problem harder as the number of different trade-offs 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 runtime analyses that show the different effects of adding objectives to a well-known plateau-function.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity

General Terms

Theory, Algorithms, Performance

Keywords

Multi-objective optimization, Running time analysis

1. MOTIVATION

In recent years, the number of publications on evolutionary multi-objective 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, cf. [4]. 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 [25] proved that the number of incomparable solutions increases, if further randomly generated objectives are added. Thereby, 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. [9, 12, 5, 4, 8], list in favor of the assumption that the search becomes harder the 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 [24].

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 [15] 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. [16] where single-objective problems are "multi-objectivized", i.e., decomposed into multi-objective problems which are easier to solve than the original problems. Besides these empirically oriented studies, there are theoretical results supporting the hypothesis that multi-objectivization can help: Scharnow et al. [23] showed that the Single Source Shortest Path problem is easier to solve for simple EAs when formulated as a bi-criterion problem; Neumann and Wegener [22] proved for the Minimum Spanning Tree problem that a formulation with two objectives leads to a lower runtime 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, the opposite is the case. However, given the previous work, the question arises whether one and the same problem can be made both easier and harder by adding adequate objectives. So far, the issue of adding objectives (which is different from decomposing an objective function into several ones) has only been investigated empirically,

and only a recent study [3] considered both directions (less or more difficult). Inspired by the experimental results presented in [3], this paper for the first time answers this question theoretically. Based on a simple multi-objective optimizer, namely SEMO which is known from various theoretical analyses, we show by means of runtime analyses that

(i) one and the same problem can become both harder and easier to solve if an objective is added—in contrast to [23] and [22], where the original objective is replaced by two other objectives, we here consider the case that the original objective remains in the objective set;

(ii) the combination of two equally difficult single-objective functions can yield a bi-criterion problem that is easier to solve than any of the two single-objective problems.

The basis of the running time analyses to follow is provided by a well-known plateau function. As objective functions with plateaus of a similar kind appear in many well-known combinatorial optimization problems, the proposed analyses can be seen as a first, but general, investigation of how additional objectives influence the running time behavior of evolutionary algorithms.

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 (Sec. 2). In Sec. 3, we detail the algorithms considered in this study and define the setting for the runtime analyses to follow. Sec. 4 provides the proofs showing that a simple plateau function can become both harder and easier with an additional objective; Sec. 5 extends these results and demonstrates that even the combination of two equally difficult single-objective functions can yield an easier bi-criterion problem. Conclusions are presented in Sec. 6.

2. ADDING OBJECTIVES: FOUNDATIONS AND EFFECTS

Without loss of generality, we consider maximization problems with k objective functions $f_i \colon X \to \mathbb{R}$, $1 \leq i \leq k$, where 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: $\succeq_{\mathcal{F}'} := \{(x, y) \in X^2 \mid \forall f_i \in \mathcal{F}' \colon f_i(x) \geq f_i(y)\}$, 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:

- The solution pair x, y is called *comparable* if x weakly dominates y and/or y weakly dominates x;

- Two solutions x, y are *incomparable* if neither weakly dominates the other one;

- Two solutions having the same objective vector are called *indifferent*.

A solution $x^* \in X$ is called *Pareto optimal* if any $x \in X$ 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*.
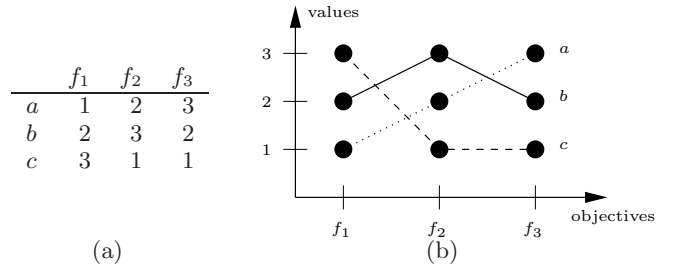

	f_1	f_2	f_3
a	1	2	3
b	2	3	2
c	3	1	1

(a)

(b)

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

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 [1, 2]. 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 direct edge from vertex v to vertex w is drawn iff $v \succeq_{\mathcal{F}} w$. Figure 2 shows the relation graphs for $\succeq_{f_1}$, $\succeq_{f_2}$, $\succeq_{f_3}$ and the corresponding combinations of them. With only a single objective, the three solutions are pairwisely comparable, see Fig. 2(a), 2(b), and 2(c). When merging f_1 and f_2 to a bi-criterion problem, comparabilities disappear, cf. 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 the opposite does not hold. 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 a is weakly dominating c nor c is weakly dominating 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).

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 $\succeq_{\{f_1, f_2, f_3\}}$, cf. Fig. 2(f).

We observe that additional objectives result in the omission 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 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 $(x, y) \in \succeq_{\mathcal{F}_1}$, the solution pair becomes incomparable with respect to $\mathcal{F}_1 \cup \mathcal{F}_2$; the edges between x and y disappear in $\succeq_{\mathcal{F}_1 \cup \mathcal{F}_2}$. Resumed, in our example, the edges of the new relation graphs can always be derived from
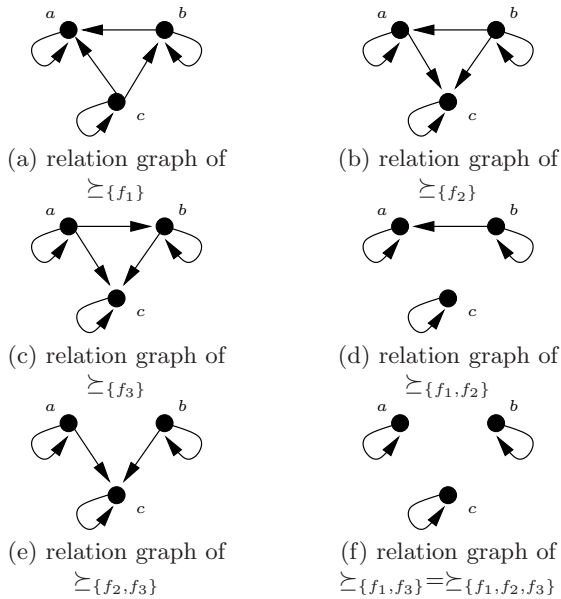
(a) relation graph of $\succeq_{\{f_1\}}$

(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 $\succeq_{\{f_1,f_3\}}=\succeq_{\{f_1,f_2,f_3\}}$

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

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 [1] for details:

Let $\mathcal{F}' \subseteq \mathcal{F} := \{f_1, \ldots, f_k\}$ 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 [2] 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 runtime of a dominance relation based evolutionary algorithm, e.g., SEMO, only if the additional objective is conflicting with the set of objectives, defining the original problem. As the example in Fig. 1 and Fig 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 runtime of evolutionary algorithms which consider the dominance relation solely. Now, the question arises, how the addition of a non-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:

(i) Comparable solutions can become incomparable, and

(ii) 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

[1]The other way round, an omission of an objective can

become easier and harder to solve, as was shown experimentally in [3] and is proved analytically in the following sections.

Generally speaking, case (i) turns a region with given search space direction into a plateau of incomparable solutions, whereas case (ii) 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 (i), where comparable solutions become incomparable, the comparability between solutions can either lead to the Pareto-optimal 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 (ii), the problem can both 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-optimal front or behave deceptively, the computational effort to identify the Pareto optima may decrease or increase.

3. ALGORITHMS

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

For single-objective optimization problems (where $k = 1$), our analyses are based on the $(1+1)$ EA which has been considered in theoretical investigations on pseudo boolean functions [7] as well as some of the best-known combinatorial optimization problems [11, 21, 26]. 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 $x \in \{0,1\}^n$ uniformly at random.

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.

We compare the $(1+1)$ EA with its multi-objective counterpart called Global SEMO (Global Simple Evolutionary Multi-objective Optimizer) [17, 10] which has been investigated in the context of different multi-objective problems, e.g., spanning tree problems [19, 22]. Global SEMO starts

make incomparable solutions comparable and comparable solutions indifferent.

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 fitness vector as x' are removed from P. In detail, Global SEMO is defined as follows.

ALGORITHM 2. *Global SEMO*

1. *Choose $x \in \{0,1\}^n$ uniformly at random.*

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 x' is not dominated by any other search point in P, include x' into P and delete all solutions dominated by x' or with fitness vector $f(x')$ from 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 expectation value of the optimization time.

Let $|x|_1$ denote the number of ones and $|x|_0$ denote the number of zeros in a given bitstring x. We are also interested in variants of the introduced algorithms using the following asymmetric mutation operator proposed in [13].

ALGORITHM 3. *Asymmetric Mutation Operator*

- *Create x' by flipping each bit 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.

4. 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 fitness. 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 vector. 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 are easy for evolutionary algorithms if for each non-optimal solution there is a better Hamming neighbor, which means that an improvement can be reached by flipping a single bit
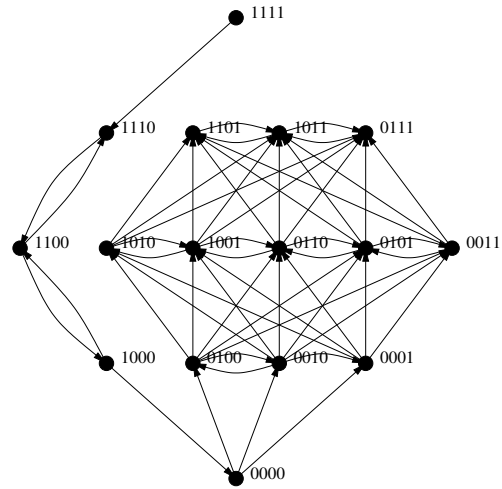

Figure 3: Relation graph for the objective function $\text{PLATEAU}_1 : \{0,1\}^4 \rightarrow \mathbb{R}$. Reflexive and transitive edges are omitted for clarity.

of a non-optimal 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, cf. [14]. The behavior of the (1+1) EA on plateaus of different structures has been studied in [14] by a rigorous runtime analysis.

We consider the function PLATEAU_1 similar to the function SPC_n already investigated in [14], which is defined as

$$\text{PLATEAU}_1(x) := \begin{cases} |x|_0 & : & x \notin \{1^i 0^{n-i}, 1 \leq i \leq n\} \\ n+1 & : & x \in \{1^i 0^{n-i}, 1 \leq i < n\} \\ n+2 & : & x = 1^n. \end{cases}$$

The relation graph of PLATEAU_1 for $n = 4$ is shown in Fig. 3. PLATEAU_1 contains a set of $n-1$ search points that form a short path having fitness value $n+1$. We denote by $SP := \{1^i 0^{n-i}, 1 \leq i < n\}$ the set of all these search points. In addition, 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 runtime 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 [14]) are identical besides the inclusion of 0^n in the plateau or not, we can reuse all ideas used in the proof of [14] for the expected runtime $O(n^3)$ of the (1+1) EA on SPC_n. Therefore, also on PLATEAU_1 the expected runtime of the (1+1) EA can be bounded by $O(n^3)$. To the 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 $|x|_1 \leq \frac{2}{3}n$ with probability $1 - o(1)$

by Chernoff bounds. As long as the current solution is not in SP and not equal to 0^n, the value $|x|_1$ is non-increasing. Thus, the first individual x chosen by the (1+1) EA that is in the set SP has the property $|x|_1 \leq \frac{2}{3}n$ with probability $1 - o(1)$. Once the current search point is in the set SP, only children also from the set SP 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 4 or more components in an accepted step is at most $\sum_{i=4}^{n} 2(\frac{1}{n})^i (\frac{n-1}{n})^{n-i} = O(n^{-4})$. 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 3 flips to be accepted is at most $2(\frac{1}{n})^3 (\frac{n-1}{n})^{n-3} = O(n^{-3})$. 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 $O(n^{\frac{1}{2}})$ 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. 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 $1 - o(1)$. 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 $\Omega(n^3)$. $\qquad\square$

The analyses of variants of the (1+1) EA in [6, 11, 20] 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 non-trivial 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 (cf. Thm. 2) on PLOM and exponentially slower on PLZM (cf. Thm. 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 [16], multi-objectivization 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 $x \in SP$ 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 optima 1^n. This can be seen nicely in the relation graph of PLOM in Fig. 4. The following theorem
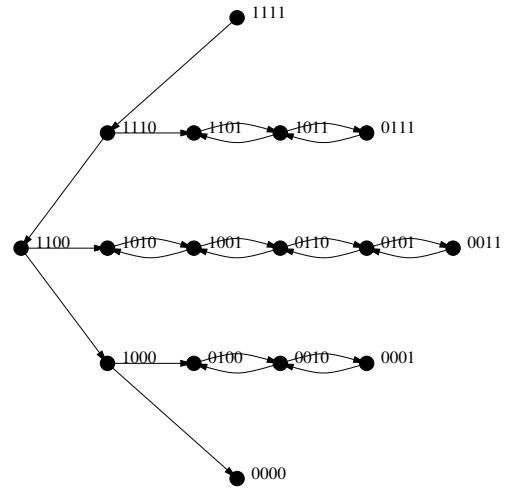

Figure 4: Relation graph for the bi-criterion problem PLOM : $\{0,1\}^4 \rightarrow \mathbb{R}^2$. Reflexive and transitive edges are omitted for clarity.

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* $O(n^2 \log n)$.

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, 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 [18] and the fact that in each step the algorithm chooses with probability $\geq 1/n$ the uppermost search point of SP. If the initial solution is not in SP, Global SEMO produces solutions that trade off between the number of ones and zeros. 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. $\qquad\square$

Using the asymmetric mutation operator, the function PLATEAU$_1$ becomes much harder. Jansen and Sudholt [13] 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* $O(n^2)$

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 ONEMAX. According to [13], (1+1) EA$_{asy}$ needs an expected time of $O(n)$ to optimize ONEMAX. 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 $x \notin \{1^i0^{n-i}, 1 \le i \le n\}$. We will analyze the expected number of steps to reach the optimum assuming that no element from $\{1^i0^{n-i}, 1 \le i \le n\}$ 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

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

we can bound this probability by $p(x) \ge \frac{1 - e^{-1/2}}{2}$. As two elements $x, y \in (\{0,1\}^n \setminus \{1^i0^{n-i}, 1 \le i \le n\})$ with $|x|_0 \ne |y|_0$ do not dominate each other, as soon as a mutation creates an element with k ones, 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 $\Omega(1/n)$. $\square$

It remains to examine the problem PLZM. An exponential deceleration comes from the $x \in SP$. 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 exponential with high probability.*

Proof. The objective vectors (n, n) and $(n + 2, 0)$ with the corresponding search points 1^n and 0^n are the two Pareto optima of PLZM. The population size is at each time step at most three as the two objectives are not in conflict for search points $x \notin SP \cup \{1^n\}$. Search points of SP are comparable such that always the one with the largest $|x|_0$ is kept in the population. Considering in addition the remaining search point 1^n the claim on the population size follows. Using the ideas of the proof of Thm. 2 and 3 together with the bound on the population size, the expected time to obtain the solution 0^n is $O(n \log n)$ for Global SEMO and $O(n)$ for Global SEMO$_{asy}$. Now we show that the time to find the other Pareto optimal search point 1^n is exponential. Starting with a random $x \in \{0,1\}^n$, both algorithms only accept solutions x' with $|x'|_0 \ge |x|_0$ or the search point 1^n. With probability exponentially close to 1, the initial random start point x satisfies $|x|_1 < \frac{2}{3}n$. Hence, the only chance to produce the search point 1^n is to do one big jump by flipping at least $\frac{n}{3}$ 0-bits. The corresponding probabilities $O(n^{-n/3})$ for Global SEMO and $O(2^{-n/3})$ for Global SEMO$_{asy}$ are

exponentially low. Therefore, the optimization times are exponential with high probability. $\square$

5. COPING WITH TWO PLATEAUS

In Sec. 4, the added objectives were easy to solve individually for the (1+1) EA. The main reason for the smaller runtime of PLOM as 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 Thm. 1 that Global SEMO has an expected running time of $\Theta(n^3)$ on PLATEAU$_1$. The same holds for the following function

$$\text{PLATEAU}_2(x) = \begin{cases} |x|_1 & : \quad x \notin \{0^i1^{n-i}, 1 \le i \le n\} \\ n+1 & : \quad x \in \{0^i1^{n-i}, 1 \le i < n\} \\ n+2 & : \quad x = 0^n \end{cases}$$

due to the symmetry with PLATEAU$_1$. We now consider the multi-objective 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 may be easier than solving the single-objective problems separately.

THEOREM 5. *The expected optimization time of Global SEMO on* PLATEAUS *is* $O(n^2 \log n)$.

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 $O(n^2)$. 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 $O(n^2 \log n)$.　$\square$

Jansen and Sudholt [13] 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* $O(n^2)$.

Proof. As in the proof of Thm. 5, we first bound the expected number of steps until the population includes search points of fitness $(n+1, \cdot)$ 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 Thm. 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 fitness $(n+1, \cdot)$ is obtained after an expected number of $O(n^2)$ steps. By symmetry, the same holds for obtaining a search point of fitness $(\cdot, n+1)$.

Now assume that two search points of fitness $(n+1, \cdot)$ and $(\cdot, n+1)$ are included in the population. Since they dominate all other points, the population size is bounded by 2 in this case. If the fitness of the first search point is $(n+1, k)$, it consists of k ones followed by $n-k$ zeros. Its fitness 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

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

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

6. CONCLUSIONS

We have investigated the effect of adding an objective to a given problem. This can make the solutions of the original problem either incomparable or make indifferent solutions comparable. The effect of these changes in the dominance relations has been investigated via a rigorous runtime analysis. We have pointed out situations where such changes can both speed up or slow down the optimization process on one and the same problem. In the extreme case, the effect of adding objectives can make the difference between a polynomial and an exponential runtime. In our investigations, we considered a well-known plateau function. As objective functions with plateaus of a similar kind appear in some well-known combinatorial optimization problems, adding objectives might be useful for a broad class of such problems.

7. REFERENCES

[1] D. Brockhoff and E. Zitzler. Are All Objectives Necessary? On Dimensionality Reduction in Evolutionary Multiobjective Optimization. In *Proc. of PPSN '06*, vol. 4193 of *LNCS*, p. 533–542, 2006.

[2] D. Brockhoff and E. Zitzler. Dimensionality Reduction in Multiobjective Optimization: The Minimum Objective Subset Problem. In K.-H. Waldmann and U. M. Stocker, editors, *Proc. of Operations Research 2006*. Springer, 2007. to appear.

[3] D. Brockhoff and E. Zitzler. Offline and Online Objective Reduction in Evolutionary Multiobjective Optimization Based on Objective Conflicts. TIK Report 269, Institut für Technische Informatik und Kommunikationsnetze, ETH Zürich, March 2007.

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

[5] K. Deb. *Multi-objective optimization using evolutionary algorithms.* Wiley, Chichester, UK, 2001.

[6] B. Doerr, N. Hebbinghaus, and F. Neumann. Speeding up evolutionary algorithms through restricted mutation operators. In *Proc. of PPSN '06*, vol. 4193 of *LNCS*, p. 978–987, 2006.

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

[8] P. J. Fleming, R. C. Purshouse, and R. J. Lygoe. Many-Objective Optimization: An Engineering Design Perspective. In *Proc. of EMO 2005*, vol. 3410 of *LNCS*, p. 14–32. Springer, 2005.

[9] C. M. Fonseca and P. J. Fleming. An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evol. Comp.*, 3(1):1–16, 1995.

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

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

[12] J. Horn. Multicriterion decision making. In *Handbook of Evol. Comp.* CRC Press, 1997.

[13] T. Jansen and D. Sudholt. Design and analysis of an asymmetric mutation operator. In *Proc. of CEC '05*, IEEE Press, p. 497–504, 2005.

[14] 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. Comp.*, 5(6):589–599, 2001.

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

[16] J. D. Knowles, R. A. Watson, and D. W. Corne. Reducing Local Optima in Single-Objective Problems by Multi-objectivization. In *Proc. of EMO 2001*, vol. 1993 of *LNCS*, p. 269–283. Springer, 2001.

[17] 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*, vol. 2439 of *LNCS*, p. 44–53. Springer, 2002.

[18] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[19] F. Neumann. Expected runtimes of a simple evolutionary algorithm for the multi-objective minimum spanning tree problem. In *Proc. of PPSN '04*, vol. 3242 of *LNCS*, p. 80–89, 2004.

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

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

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

[23] J. Scharnow, K. Tinnefeld, and I. Wegener. The Analysis of Evolutionary Algorithms on Sorting and Shortest Paths Problems. *J. of Mathematical Modelling and Algorithms*, 3(4):349–366, 2004.

[24] T. Wagner, N. Beume, and B. Naujoks. Pareto-, Aggregation-, and Indicator-based Methods in Many-objective Optimization. In S. Obayashi et al., editors, *Proc. of EMO 2007*, vol. 4403 of *LNCS*, p. 742–756. Springer, 2007.

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

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