Diss. ETH No. 18316

Many-Objective Optimization and Hypervolume Based Search

A dissertation submitted to
ETH ZURICH

for the degree of
Doctor of Sciences

presented by
DIMO BROCKHOFF
Dipl.-Inform., Universität Dortmund, Germany
born December 5, 1979
citizen of Germany

accepted on the recommendation of
Prof. Dr. Eckart Zitzler, examiner
Prof. Dr. Kaisa Miettinen, co-examiner

2009

Institut für Technische Informatik und Kommunikationsnetze
Computer Engineering and Networks Laboratory

TIK-SCHRIFTENREIHE NR. 105

Dimo Brockhoff

Many-Objective Optimization and Hypervolume Based Search

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

A dissertation submitted to
ETH Zurich
for the degree of Doctor of Sciences

Diss. ETH No. 18316

Prof. Dr. Eckart Zitzler, examiner
Prof. Dr. Kaisa Miettinen, co-examiner

Examination date: April 3, 2009

Contents

Abstract

Multiobjective optimization problems occur frequently in practice where multiple objectives have to be optimized simultaneously and the goal is to find or approximate the set of Pareto-optimal solutions. Multiobjective evolutionary algorithms (MOEAs) are one type of randomized search heuristics that are well-suited for multiobjective optimization problems due to their ability of computing a set of trade-off solutions in one run. However, current state-of-the-art MOEAs are known to have difficulties if the number of objectives is high, i.e., larger than 4. This thesis tackles such *many-objective optimization* problems in terms of theoretical investigations to better understand why classical MOEAs have difficulties with many objectives. New approaches and techniques are provided that enhance the search capabilities of MOEAs for many-objective problems. Furthermore, we investigate *hypervolume-based MOEAs* in more depth that have been proposed recently especially for many-objective optimization scenarios.

In particular, we generally investigate the question of what happens if objectives are added to a problem formulation—both with respect to the Pareto dominance relation in general and with respect to the running time of specific MOEAs. This includes the proposition of a general measure to quantify the changes occurring in the Pareto dominance relation if the objective set is changed. Based on this measure, we propose the term of δ-non-conflicting objectives: if two objective sets are δ-non-conflicting with each other, the two sets can be interchangeably optimized without changing the resulting Pareto set approximation by an additional term of δ in each objective, i.e., we make an error of at most δ.

Based on these theoretical foundations, objective reduction techniques are developed that automatically reduce the number of objectives while the Pareto dominance relation is preserved or only slightly changed. While reducing the number of objectives, we distinguish between two problems. We either predefine a maximal size of the sought objective subset or a maximal δ-error allowed and then ask for the objective subset of the original objectives that meets these goals and which optimizes the above mentioned δ-error and the size of the resulting objective set respectively. Both exact algorithms and greedy heuristics are proposed and analyzed with respect to their running time for each of the two problems. Furthermore, the $\mathcal{NP}$-hardness of both problems is proved. Besides objective reduction by omitting objectives, we also consider the aggregation of objectives and investigate what happens with respect to the Pareto dominance relation if objectives are aggregated. Algorithms that aggregate objectives automatically are proposed and compared to their objective omission counterparts. The usefulness of the objective reduction algorithms for a decision maker in an a posteriori scenario, i.e., after the search, is presented for several test problems and a real-world application of radar waveform optimization.

In the last part of the thesis, we investigate several aspects of hypervolume-based MOEAs. The hypervolume indicator has become popular to guide the search of MOEAs in the past years due to its properties of being a refinement of the Pareto dominance relation. This allows for a better guidance towards the Pareto-optimal solutions which is especially useful if many objectives are to be optimized. However, the running time for computing the hypervolume indicator exactly is exponential in the number of objectives which results in the need for further research in this area to provide more powerful and better applicable MOEAs if many objectives are to be optimized. The first contribution of this thesis part is the first rigorous running time analysis of a hypervolume-based MOEA showing that the approach of optimizing the hypervolume with a $(\mu+1)$-strategy can find the set of Pareto-optimal solutions for a specific problem in a reasonable time. To increase the applicability of hypervolume-based MOEAs to real-world problems, two further studies are presented. First, we generalize the hypervolume indicator to a weighted version and show how this generalized indicator allows to incorporate user preferences into the search. Second, we apply the developed objective reduction algorithms within a hypervolume-based MOEA which is shown to reduce the running time needed for the hypervolume calculation and which at the same time produces solutions of higher quality.

Zusammenfassung

In der Praxis auftretende Optimierungsprobleme sind häufig Mehrzielprobleme, bei denen mehrere Zielfunktionen gleichzeitig optimiert werden müssen und üblicherweise die Menge der Pareto-optimalen Lösungen oder eine Approximation dieser gesucht ist. Evolutionäre Algorithmen sind eine spezielle Art von randomisierten Suchheuristiken, die für Mehrzielprobleme gut geeignet sind, da sie die Möglichkeit bieten eine Menge von Kompromisslösungen in einem Programmlauf zu erzeugen.

Gängige evolutionäre Algorithmen für die Mehrzieloptimierung haben jedoch Schwierigkeiten Mehrzielprobleme mit vielen, d. h. mehr als 4 Zielfunktionen zu lösen. Diese Arbeit beschäftigt sich mit solchen Problemen mit vielen Zielfunktionen; theoretische Untersuchungen versuchen dabei aufzuzeigen, warum klassische evolutionäre Mehrzielalgorithmen Schwierigkeiten mit vielen Zielfunktionen haben. Es werden zudem neue Ansätze und Techniken vorgestellt, die das Suchverhalten von evolutionären Mehrzielalgorithmen für Probleme mit vielen Zielfunktionen verbessern. Weiterhin werden im Detail so genannte hypervolumenbasierte evolutionäre Mehrzielalgorithmen untersucht, die vor kurzem inbesondere für Mehrzielprobleme mit vielen Zielfunktionen vorgeschlagen wurden.

Im Folgenden beantworten wir unter anderem die Frage, wie sich die so genannte Pareto-Dominanzrelation und die Laufzeit von evolutionären Mehrzielalgorithmen verändern, wenn zusätzliche Zielfunktionen zu einer Problemformulierung hinzugefügt werden. Diese Untersuchung beinhaltet den Vorschlag eines generellen Masses um die auftretenden Änderungen der Dominanzrelation zu quantifizieren, wenn sich die Menge der Zielfunktionen ändert. Basierend auf diesem Mass entwickeln wir den Begriff von nicht im δ-Konflikt stehenden Mengen von Zielfunktionen: stehen zwei Mengen von Zielfunktionen nicht im δ-Konflikt zueinander, so kann die eine der beiden Zielfunktionsmengen durch die andere ersetzt werden, ohne dass sich die aus der Optimierung resultierende Paretomengenapproximation in jeder Zielfunktion um einen additiven Term von δ ändert; in diesem Fall sagen wir, wir machen einen δ-Fehler.

Basierend auf diesen theoretischen Untersuchungen werden Algorithmen vorgestellt, die die Anzahl der Zielfunktionen automatisch reduzieren ohne die zugrunde liegende Pareto-Dominanzrelation (stark) zu verändern. Bei der Reduzierung einer Menge von Zielfunktionen unterscheiden wir zwischen zwei Problemen. Entweder setzen wir eine gewünschte Maximalgrösse der Menge fest oder definieren einen maximal erlaubten δ-Fehler und fragen dann nach einer Teilmenge der gegebenen Zielfunktionen, die die gegebenen Vorgaben erfüllt und den δ-Fehler beziehungsweise die resultierende Mengengrösse optimiert. Für beide Probleme werden sowohl ein exakter Algorithmus als auch Heuristiken vorgestellt und deren Laufzeiten

analysiert. Zudem beweisen wir für beide Probleme, dass sie zur Menge der $\mathcal{NP}$-harten Probleme gehören. Neben dem Weglassen von Zielfunktionen berücksichtigen wir auch die Aggregation von Zielfunktionen und untersuchen, wie sich die Pareto-Dominanzrelation ändert, wenn Zielfunktionen aggregiert werden. Algorithmen zur Reduzierung der Zielfunktionenanzahl durch Aggregation werden ebenfalls vorgestellt und anschliessend mit den vorgeschlagenen Verfahren, die Zielfunktionen weglassen, verglichen. Die Nützlichkeit der vorgestellten Algorithmen zur Reduzierung der Zielfunktionsanzahl während der Entscheidungsfindung, d. h. nach der Suche, wird anhand verschiedener Testprobleme sowie der Optimierung von Wellenformen für Flugzeugradare demonstriert.

Im letzten Teil der Arbeit untersuchen wir verschiedene Aspekte von hypervolumenbasierten evolutionären Merhzielalgorithmen. Durch seine Eigenschaft, eine Verfeinerung der Pareto-Dominanzrelation zu sein, wurde der Hypervolumen-Indikator in den letzten Jahren vermehrt als Suchkriterium innerhalb von evolutionären Mehrzielalgorithmen eingesetzt. Vor allem, wenn viele Zielfunktionen gleichzeitig optimiert werden sollen, induziert die Optimierung des Hypervolumens neue Suchrichtungen in Richtung besserer Kompromisslösungen, die durch die alleinige Optimierung der Pareto-Dominanzrelation nicht gegeben sind. Allerdings steigt die Komplexität der exakten Berechnung des Hypervolumen-Indikators exponentiell mit der Anzahl der Zielfunktionen an, was die Entwicklung von schnelleren und besser anwendbaren evolutionären Mehrzielalgorithmen für den Fall von vielen Zielfunktionen erfordert. Der erste Beitrag dieses Teils der Arbeit ist die erste Laufzeitanalyse eines hypervolumenbasierten evolutionären Mehrzielalgorithmus, die zeigt, dass der Ansatz, den Hypervolumen-Indikator innerhalb einer $(\mu + 1)$-Selektion zu optimieren, funktioniert und dass die optimalen Kompromisslösungen innerhalb vernünftiger Zeit gefunden beziehungsweise approximiert werden können. Um die Anwendbarkeit von hypervolumenbasierten evolutionären Mehrzielalgorithmen in der Praxis zu verbessern, werden zwei weitere Studien präsentiert. Zuerst verallgemeinern wir die Definition des Hypervolumen-Indikators auf eine gewichtete Version und zeigen auf, wie es der verallgemeinerte gewichtete Indikator erlaubt, beliebige Benutzerpräferenzen in die Suche einzubeziehen. Danach wenden wir die zuvor entwickelten Methoden zur Reduzierung der Zielfunktionenanzahl auf hypervolumenbasierte evolutionäre Mehrzielalgorithmen an um die lange Berechnungszeit für den Hypervolumen-Indikator zu verringern, was es erlaubt, Kompromisslösungen von höherer Qualität in derselben Zeit zu erzeugen.

Statement of Contributions

Most of the work presented in this thesis has already been published in conference proceedings, journal articles, and a book chapter with various co-authors. However, the text has been completely rewritten, some proofs are corrected, and the results are integrated into one thesis. The implementation and realization of all experiments presented in this work have been carried out by myself whereas for writing the papers, I was assisted by my co-authors. Except for (Zitzler et al., 2007) where I contributed less than 1/3 to the writing, I had at least a contribution of $1/n$ if the paper has n authors.

The investigations about the effects of additional objectives, objective conflicts, objective reduction, and aggregation in Chapter 2 and Chapter 3 are composed from a series of papers, starting from the very first results on objective conflicts and objective reduction (Brockhoff and Zitzler, 2006b, 2007a), to the general measure of objective conflicts and the objective reduction technique of (Brockhoff and Zitzler, 2006a, 2007a),(Brockhoff et al., 2007b), and (Brockhoff and Zitzler, 2009b), to the aggregation of objectives in (Brockhoff and Zitzler, 2009a). The running time analyses of Chapter 2 occurred in (Brockhoff et al., 2007a) and (Brockhoff et al., 2009) respectively.

Chapter 4 about the hypervolume indicator has been mainly made up from three contributions: the proposed weighted hypervolume in (Zitzler et al., 2007), the running time analysis in (Brockhoff et al., 2008) and the incorporation of objective reduction and objective aggregation into hypervolume-based algorithms in (Brockhoff and Zitzler, 2007b) and (Brockhoff and Zitzler, 2009a).

Dimo Brockhoff

Acknowledgments

First of all, I would like to thank my supervisor Eckart Zitzler for supporting me within the last four years and also for teaching me scientific research in general. I am quite sure that without his challenging questions and his general sense of "hot topics", I would not have been able to finish this thesis in such a short time.

I would also like to thank my co-referee Kaisa Miettinen for her willingness to read and judge the thesis, for the challenging but fruitful comments on the manuscript and for coming to Zurich for the final exam as well as for interesting and inspiring discussions during her stay.

Many thanks go to all of my co-authors for the experience I gained throughout the different fruitful collaborations. My special thank goes to those persons who contributed in part to the content of this thesis, namely Tobias Friedrich, Nils Hebbinghaus, Christian Klein, Frank Neumann, Lothar Thiele, and Eckart Zitzler.

Thanks also go to the Swiss National Science Foundation (SNF) for their financial support under grant 112079.

Moreover, I am indebted to all my colleagues at TIK, especially to the (current and former) SOPies Johannes Bader, Stefan Bleuler, Tim Hohm, and Tamara Ulrich. Without the inspiring discussions about scientific and non-scientific topics as well as the necessary free-time activities and the daily "Zvieri", the last four years would not be as funny as they were. Special thanks go to Johannes Bader who has been always helpful with respect to graphical requests and also helped for some of the figures in this thesis. Thanks are also going to my entire family who always assisted me wherever and whenever they could.

Last but not least, I would like to thank the most important person in my life, Anne, simply for everything—since I am not able to express in words how much I love you...

1

Introduction

1.1 Motivation

Many problems in practice are multiobjective in nature with examples ranging from nurse scheduling to aircraft construction (Qiu, 1997; Paechter et al., 1998; Coello Coello and Hernández Aguirre, 2002; Fleming et al., 2005; Hughes, 2007; Sülflow et al., 2007). Tackling a multiobjective optimization problem involves the simultaneous consideration of multiple objectives that are, in general, in conflict with each other as for example the objectives price and quality while buying a certain product or the objectives risk and profit if the personal stock portfolio is rearranged. In such a multiobjective optimization scenario, solving a multiobjective problem corresponds to finding or approximating the set of so-called *Pareto-optimal solutions*, i.e., solutions for which no other solution is better in all objectives. Furthermore, a multiobjective optimization scenario always involves a (human) decision maker that states which solution is the most preferable one among the Pareto-optimal solutions or within an obtained *set* of solutions.

Depending on how the decision maker is involved, three main approaches to multiobjective optimization can be identified. In *a priori* methods, the decision maker is involved before the search and a usually single-objective optimization aims at finding the best solution according to the decision maker's preferences. *A posteriori* approaches provide the decision maker with a *set* of solutions and postpone the decision about which solution is the best one after the search whereas in *interactive* approaches, the decision maker is involved during the search and specifies her preferences depending on the information gain while the solutions approach the set of Pareto-optimal solutions. One way to tackle multiobjective optimization problems is to use algorithms from the classical field of *multicriteria decision making* (MCDM). They usually aim at finding a single solution that fits the decision maker's preferences best. Many approaches in this field are known and we refer to text books such as (Miettinen, 1999) or (Ehrgott, 2005) for an extensive overview. On the other hand, *evolutionary multiobjective optimization* (EMO) techniques (Deb, 2001; Coello Coello et al., 2007) mainly aim at finding a good approximation of the *set* of Pareto-optimal solutions in one run and postpone the decision making step after the search (Deb, 2001; Coello Coello et al., 2007) although recently, also interactive methods have become of interest (Jaszkiewicz and Branke, 2008).

The most often used approach in EMO is to employ evolutionary algorithms (EAs)[1] for the search of good trade-offs between the objectives. Evolutionary algorithms mimic evolutionary processes observed in nature and they belong to the class of randomized search heuristics. A set of solutions, the *population*, is evolved by successively applying mutation, recombination, and selection on the population's individuals yielding better and better solutions in an iterative process. Since an EA evolves a population of solutions, it allows to compute a set of solutions before a decision maker has to articulate her preferences. After the search, the decision maker can then use the obtained solutions to decide which solution, according to her preferences, is to be implemented. Deciding on a solution when information about the trade-offs between objectives is known is usually much easier than specifying the preferences before the optimization process—a necessary step when tackling a multiobjective problem by the weighted aggregation method or other classical scalarization techniques (cf. Miettinen, 1999). We will give a more detailed introduction into the class of *multiobjective evolutionary algorithms* (MOEAs) later on in this chapter.

Although multiobjective evolutionary algorithms have been successfully applied to multiobjective problems, experimental studies have shown that state-of-the-art MOEAs have difficulties if the number of objectives is high, i.e., larger than 4 (Khare et al., 2003; Purshouse and Fleming, 2003b; Wagner et al., 2007). Unfortunately, such *many-objective problems* with 5 or more objectives occur frequently in practice (Qiu, 1997; Paechter et al., 1998; Coello Coello and Hernández Aguirre, 2002; Fleming et al., 2005; Hughes, 2007; Sülflow et al., 2007) and it is often even easier to specify as many objectives as possible before the search instead of including the decision maker in an early stage of the optimization process where most of the time no knowledge about the problem is available and a decision which objectives are the most important ones is not possible[2]. MCDM approaches

[1] Also other bio-inspired approaches such as particle swarm optimization fall into the category of evolutionary multiobjective optimization. Within this thesis, however, we focus on evolutionary algorithm based approaches.

[2] When designing a new product, for example, it is often easier to consider all possible

can handle many-objective problems but on the other hand have the drawback that only single solutions rather than a set of solutions are created. Since MOEAs have been shown to be able to efficiently produce a set of solutions that show the trade-offs among the objectives for problems with 2 or 3 objectives, the question arises how existing multiobjective evolutionary algorithms can be enhanced to obtain also a set of "good" solutions if the number of objectives is high. Work in terms of evolutionary many-objective optimization is limited indicating that even the basic questions are not solved yet, see for example the recent review of Ishibuchi et al. (2008). This is the main reason why several studies presented in this thesis consider a low number of objectives although the aim is to better understand and improve evolutionary algorithms for many-objective optimization[3].

One recent trend to tackle multiobjective problems with more than 2 or 3 objectives with MOEAs employs so-called indicator-based MOEAs—in particular *hypervolume indicator* based MOEAs became very popular in recent years. Instead of optimizing the objective functions with respect to the Pareto dominance relation only, these algorithms employ the hypervolume indicator for guiding the search towards the set of Pareto-optimal points. The hypervolume indicator is a set quality measure introduced for performance assessment by Zitzler and Thiele (1998a) that assigns a set of solutions a real value which has the property that whenever a solution set is better than another one in terms of Pareto dominance, the hypervolume indicator value of the former is higher than the one of the latter. Fortunately, this property allows to guide the search towards better solutions in multiobjective and especially many-objective problems and hypervolume-based MOEAs have been shown to be better suited to multiobjective problems than other state-of-the-art MOEAs if the number of objectives increases (Wagner et al., 2007). However, the applicability of hypervolume indicator based MOEAs is limited by the fact that the exact hypervolume indicator calculation is exponential in the number of objectives (Beume and Rudolph, 2006; Bringmann and Friedrich, 2008)—one reason why also hypervolume-based MOEAs were not applicable to multiobjective problems with more than 5 objectives in the beginning of this thesis project[4].

objective functions that possibly have an influence on the product instead of deciding early in the development process which objectives to omit. The reason is that at this early stage of the so-called *design space exploration*, almost no information is available about whether, and if how much, a certain objective function has an impact on the set of non-dominated solutions generated by an optimization method.

[3] Especially in the area of hypervolume-based search, discussed below, even the basic principles behind the search process have not been fully understood for problems with a few objectives which improved recently starting with the work of Auger et al. (2009c).

[4] The thesis project started in July 2005 when the author began his doctoral thesis at ETH Zurich. In the following, we will always refer to this time around July 2005 if a formulation like "in the beginning of this thesis project" is used.

In the light of this discussion, several questions arise that affect both decision making and search in many-objective optimization. Examples are

- Why have MOEAs difficulties with many-objective problems? In particular, what happens if objectives are added to a problem formulation and how does the addition or omission of objectives affects the search with multiobjective evolutionary algorithms?
- Which objectives are the most important? Are there some that are redundant?
- Can the set of considered objectives be reduced without or only slightly changing the problem to make both decision making and search easier?
- How can current techniques be enhanced to be able to tackle problems with many objectives efficiently?

Section 1.4 presents the open questions with many-objective problems in more detail. Before, we introduce the basic notations and terms for multiobjective optimization (Sec. 1.2) and present a brief overview of the multiobjective evolutionary algorithms field in Sec. 1.3. Section 1.5 presents the key contributions of this thesis and Sec. 1.6 gives a brief outline of the remaining chapters.

1.2 Multiobjective Optimization

Throughout this thesis, we consider a *multiobjective optimization problem* as the problem to optimize a set of k objective functions $\mathcal{F} = \{f_1, \dots, f_k\}$, i.e. without loss of generality minimizing[5] the vector function

$$\begin{aligned} f : X &\rightarrow \mathbb{R}^k \\ \vec{x} &\mapsto f(\vec{x}) = (f_1(\vec{x}), \dots, f_k(\vec{x})) \end{aligned}$$

that maps solutions from the *decision space* X into the *objective space* $\mathbb{R}^k$. Note that the type of decision space is not defined here, i.e., both continuous (e.g., $X = \mathbb{R}^n$) and discrete search spaces (e.g., $X = \{0,1\}^n$) are considered. We assume that the weak Pareto dominance relation is the underlying preference structure according to which the optimization is to be carried out. To this end, we define a solution $\vec{x} \in X$ to *weakly dominate* another solution $\vec{y} \in X$ ($\vec{x} \preceq_{\mathcal{F}} \vec{y}$) if and only if $\vec{x}$ is not worse than $\vec{y}$ in all objectives in $\mathcal{F}$. Throughout the thesis, we consider the notation $\vec{x} \preceq_{\mathcal{F}'} \vec{y}$

[5] Where it is necessary, we will also consider an equivalent maximization of the objectives but, in general, assume minimization.

in order to indicate that the weak Pareto dominance relation is used with respect to a particular objective set $\mathcal{F}'$, often $\mathcal{F}' \subseteq \mathcal{F} := \{f_1, f_2, \dots, f_k\}$:

$$\vec{x} \preceq_{\mathcal{F}'} \vec{y} :\Longleftrightarrow \forall f_i \in \mathcal{F}' : f_i(\vec{x}) \leq f_i(\vec{y})$$

which coincides with the usual weak Pareto dominance relation $\preceq_{\mathcal{F}}$ if $\mathcal{F}' = \mathcal{F}$. For better readability, we will sometimes only list the indices of the objective functions instead of the function names themselves, e.g., $\preceq_{\{1,2\}}$ instead of $\preceq_{\{f_1,f_2\}}$. In addition, we will use the following standard terms:

(i) $\vec{x}$ *dominates* $\vec{y}$ if $\vec{x} \preceq_{\mathcal{F}'} \vec{y}$ and $\vec{y} \not\preceq_{\mathcal{F}'} \vec{x}$

(ii) $\vec{x}$ and $\vec{y}$ are *comparable* if either $\vec{x} \preceq_{\mathcal{F}'} \vec{y}$ or $\vec{y} \preceq_{\mathcal{F}'} \vec{x}$

(iii) $\vec{x}$ and $\vec{y}$ are *incomparable* if neither $\vec{x} \preceq_{\mathcal{F}'} \vec{y}$ nor $\vec{y} \preceq_{\mathcal{F}'} \vec{x}$

(iv) $\vec{x}$ and $\vec{y}$ are *indifferent* if both $\vec{x} \preceq_{\mathcal{F}'} \vec{y}$ and $\vec{y} \preceq_{\mathcal{F}'} \vec{x}$

(v) the *Pareto(-optimal) set* contains all so-called *Pareto-optimal solutions* $\vec{x}$ that either weakly dominate or are incomparable to any other solution $\vec{y} \in X$

(vi) the *Pareto(-optimal) front* is the image of the Pareto set in the objective space.

Furthermore, we will use the set-notation for relations, i.e.,

$$\preceq_{\mathcal{F}'} = \{(\vec{x}, \vec{y}) \in A \times A \,|\, \forall f_i \in \mathcal{F}' : f_i(\vec{x}) \leq f_i(\vec{y})\}$$

where $A \subseteq X$ is a particular set of solutions under consideration; it will be clear from the context which set A is meant. We also generalize the Pareto dominance relation to *solution sets*, i.e., we say a solution set $A \subseteq X$ is weakly dominating a solution set $B \subseteq X$ with respect to $\mathcal{F}$ ($A \preceq_{\mathcal{F}} B$) if

$$\forall \vec{b} \in B : \exists \vec{a} \in A : \vec{a} \preceq_{\mathcal{F}} \vec{b} \ .$$

Accordingly, A is *dominating* B with respect to $\mathcal{F}$ if

$$\forall \vec{b} \in B : \exists \vec{a} \in A : \vec{a} \preceq_{\mathcal{F}} \vec{b} \wedge \vec{b} \not\preceq_{\mathcal{F}} \vec{a} \ .$$

Given a multiobjective optimization problem and the corresponding weak dominance relation, we assume that the Pareto set is sought. In other words, we would like to find all minimal elements of the relation $\preceq_{\mathcal{F}}$ for a given set of k objectives. However, if the problem itself is hard to solve and/or the Pareto set is too large to enumerate (e.g., if it is a continuous set), we are satisfied with an approximation of the Pareto set that is close to the Pareto set and as diverse as possible, cf. (Deb, 2001). Similarly, we denote a solution set the objective vectors of which are an approximation of the Pareto front as *Pareto front approximation*. Note that the Pareto set is always well-defined for finite search spaces X but also in the case of continuous optimization, the existence of a non-empty Pareto set can be proven if some assumptions on the problem are given, see for example (Henig, 1982) or (Miettinen, 1999, p. 35). We do not go into the details here and assume a non-empty Pareto set throughout the thesis.

As to decision making, we assume that the decision maker has to choose her favorite solution in the end, i.e., the solution that fits her preferences best. To this end, information gained during or after the search process can be used to articulate preferences, e.g., in terms of aspiration levels or (p)reference points, that themselves might steer the search towards interesting regions of the search space in an interactive manner.

1.3 Multiobjective Evolutionary Algorithms

Evolutionary Algorithms mimic evolutionary processes and belong to the class of problem-independent randomized search heuristics. Similar to their single-objective counterparts, cf. for example (Eiben and Smith, 2003), *multiobjective evolutionary algorithms* maintain a set of solutions, the so-called *population*. According to their biological counterparts, the solutions in the population are called *individuals* which usually undergo mating selection, recombination/crossover, mutation, and environmental selection in an iterative loop of *generations*. The top left plot in Fig. 1 illustrates this cycle of successive selection and variation of the individuals.

The main advantage of multiobjective evolutionary algorithms is the generation of a *set* of solutions in a single run. In contrast, other approaches to solve multiobjective optimization problems, such as the multiobjective proximal bundle method (Miettinen, 1999), normal boundary intersection (Das and Dennis, 1998), or other scalarization approaches (Miettinen, 1999; Ehrgott, 2005), compute only one solution at a time. According to Oliveto et al. (2007b), there are two more reasons why MOEAs—and evolutionary algorithms in general—are used: “often there are not enough resources in terms of money, time, or knowledge [available] to construct a problem-depending algorithm” and sometimes, “the function that needs to be optimized may not be known”, e.g., if the objective function values can only be obtained by a simulation of a given system. The last scenario is therefore also known under the term of *black-box optimization*[6].

[6] Note that the statement on the limited time to develop problem-dependent algorithms cannot be directly transferred to problem types that are not of combinatorial nature; however, also for continuous optimization problems, it is better if an algorithm is problem-independent and, e.g., not tailored towards quadratical functions only. Fur-

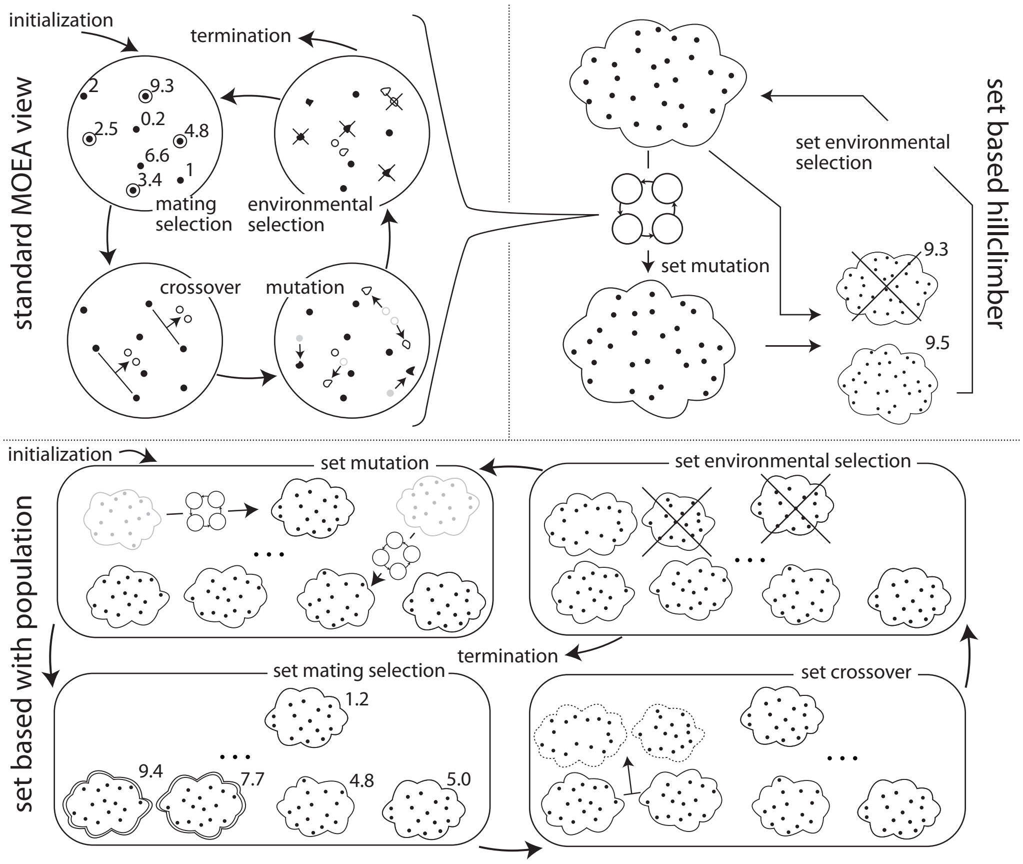


Figure 1: Comparison of different views of multiobjective evolutionary algorithms: (top left) general view with recombination, mutation, and selection on single solutions; (top right) the same algorithm as a hillclimber on solution sets, the cycle of solution-based recombination, mutation, and selection can be seen as a complex mutation operator on solution sets; (bottom) a full set-based algorithm with set-based mutation, recombination, and selection.

Although the first evolutionary algorithms date back to the 1960s, it took some time until the first evolutionary algorithm has been proposed for solving multiobjective problems in the mid 1980s by Schaffer (1985). Since then, many different MOEAs have been proposed. After a first generation of Pareto dominance based MOEAs without elitism such as the Nondominated Sorting Genetic Algorithm (NSGA) (Srinivas and Deb, 1994), the Niched-Pareto Genetic Algorithm (NPGA) (Horn et al., 1994), and the Multi-Objective Genetic Algorithm (MOGA) (Fonseca and Fleming, 1993), the second generation of MOEAs entirely focus on elitism, a theoretical requirement that is needed to provide convergence to the Pareto front, see for example (Rudolph, 2001) or (Deb, 2001) for details. The most popular

thermore, we would like to mention that black-box optimization problems can also be tackled by some classical derivative-free MCDM methods with the drawback that they again mainly compute a single solution at a time.

MOEAs of this second generation are the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) by Deb et al. (2002) and the Strength Pareto Evolutionary Algorithm 2 (SPEA2) by Zitzler et al. (2002).

The main working principles of those MOEAs are depicted in the upper left plot of Fig. 1. After an initialization step where the population is usually randomly drawn from the decision space, a generational loop produces new solutions by evolutionary principles until a certain stopping criterion terminates the algorithm—a certain number of generations or a predefined period of time are the main stopping criteria used in practice. The population evolves over the generations for most MOEAs in the same manner: first, a mating selection step identifies the solutions that undergo recombination and mutation to produce new solutions. Recombination, sometimes also called crossover, takes two or more solutions (or *parents*) and produces one or more new solutions that preferable lie in the search space region between[7] the parent solutions. Mutation, on the other hand, slightly changes a solution where small changes should have a higher probability than large changes. Where the mating selection in multiobjective evolutionary algorithms most often is performed randomly, the main difference in the algorithms is in the environmental selection step where it is decided how the population of the next generation is generated from the old individuals and the produced offspring.

In general, we distinguish between two main principles in the environmental selection step. We denote an algorithm that takes both the μ parents and the λ offspring into account and chooses the best μ ones for survival according to an optimization criterion[8] as an *algorithm with plus selection*; such an EA is referred to as $(\mu + \lambda)$-EA. If only the λ offspring are taken into account, one uses the term *comma selection* ((μ, λ)-EA). Both NSGA-II and SPEA2 are therefore plus-strategies which are also called elitist. For details on the algorithms, we refer to the original publications (Deb et al., 2002; Zitzler et al., 2002). Extensive introductions to MOEAs can be found for example in (Deb, 2001) or (Coello Coello et al., 2007).

Recently, a new idea came up by Zitzler and Künzli (2004) and others to incorporate user preferences explicitly into search. Instead of using the Pareto dominance directly, the authors suggested to optimize a certain *indicator function*. Indicator-based MOEAs can, therefore, be seen as a third generation of MOEAs with the Indicator-Based Evolutionary Algorithm (IBEA) (Zitzler and Künzli, 2004), the $\mathcal{S}$-metric Selection Evolutionary Multiobjective Algorithm (SMS-EMOA) (Beume et al., 2007), and

[7] The meaning of the term “between” highly depends on the search space and the used recombination operator.

[8] This optimization criterion is often called the *individual’s fitness* in the evolutionary computation field. An example for such an optimization criterion, that is also used in the last part of the thesis, is the individuals’ hypervolume indicator loss, cf. Sec. 4.1.2.

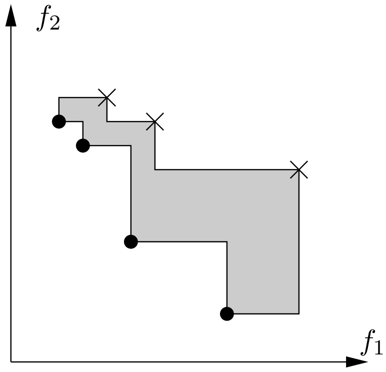


Figure 2: Illustration of the hypervolume indicator for a solution set $A \subseteq X$ (dots) and a reference set $R \in \mathbb{R}^2$ (crosses).

the multiobjective version of the Covariance Matrix Adaptation Evolution Strategy (MO-CMA-ES) (Igel et al., 2007) as the most popular algorithms in this category.

The most popular indicator function is the *hypervolume indicator*, first proposed by Zitzler and Thiele (1998a) as the 'size of the [objective] space covered'. The hypervolume indicator, also known under the name $\mathcal{S}$-metric, has the property that whenever a set of solutions is dominating another set the indicator value of the former set is higher than the one of the latter. Therefore, the objective vectors of a solution set that maximizes the hypervolume indicator cover the Pareto front entirely (Fleischer, 2003).

Several equivalent definitions of the hypervolume indicator have been proposed, cf. Sec. 4.1.1. Here, we define the hypervolume indicator in its simplest form of (Emmerich et al., 2005; Bader and Zitzler, 2008): the hypervolume indicator $I_H(A, R)$ of a solution set A is the Lebesgue measure of the image of all solutions that are weakly dominated by the solution set A and at the same time weakly dominate a certain reference set $R \subset \mathbb{R}^k$:

$$I_H(A, R) = \lambda \left(\{ \vec{z} \in \mathbb{R}^k \mid \exists \vec{a} \in A : \exists \vec{r} \in R : f(\vec{a}) \leq \vec{z} \leq \vec{r} \} \right) \tag{1.1}$$

where $\lambda(\cdot)$ is the Lebesgue measure. Figure 2 illustrates the idea of the hypervolume indicator for a 2-dimensional objective space.

Recently, the concept of set-based MOEAs has been proposed, the basic principle of which says that in order to solve a multiobjective optimization problem, i.e., in order to find a set of solutions, a MOEA should rather optimize sets of solutions instead of single solutions within one population (Zitzler et al., 2008, 2009; Bader et al., 2009). In the light of this discussion, standard MOEAs can be interpreted as hillclimbers for set problems, cf. the upper right plot in Fig. 1: the variation and selection operators that are performed on single solutions can be seen as a complicated mutation operator on the entire set, i.e., the application of solution based recombination, mutation, and selection yields a new (mutated) set of solutions. A general set based algorithm would therefore use set based recombination, set based mutation and set based selection as depicted in the bottom of Fig. 1. Bader et al. (2009) discuss these different views in more detail and also highlight the link to parallel MOEAs. Throughout this thesis, however, we only investigate algorithms that can be seen as hillclimbers according to Bader et al. (2009)—with only a few proposed MOEAs that do not fit into this scheme.

Furthermore, we want to mention various simple MOEAs that have been proposed mainly to be theoretically analyzed or to show certain general properties of MOEAs. The most often used are the global Simple Evolutionary Multiobjective Optimizer (SEMO) (Giel, 2003; Laumanns et al., 2004b,a), the Fair Evolutionary Multiobjective Optimizer (FEMO), (Laumanns et al., 2004b,a), the Restricted Evolutionary Multiobjective Optimizer (REMO) (Kumar and Banerjee, 2006), and the Simple Indicator-Based Evolutionary Algorithm (SIBEA) (Zitzler et al., 2007). Whenever we use one of these algorithms for a theoretical analyses in this thesis, we will give a detailed description of the algorithms.

To conclude, many multiobjective evolutionary algorithms are known from the literature and evolutionary multiobjective optimization has gained more and more interest in the last decade. In the meantime, also a biannual conference on evolutionary multiobjective optimization has been organized (Zitzler et al., 2001; Fonseca et al., 2003; Coello Coello et al., 2005; Obayashi et al., 2007) where especially the conference in 2007 pointed out the increasing interest in many-objective problems which are tackled in this thesis.

1.4 Research Questions

Especially from a theoretical point-of-view, many research questions have been open in the field of multiobjective evolutionary algorithms for many-objective problems in the beginning of this thesis project. We group them into three main categories regarding (i) the general effects of adding, omitting, and aggregating objectives, (ii) decision making in many-objective scenarios and (iii) many-objective search.

Effects of Adding, Omitting, and Aggregating Objectives

It is known from several empirical studies that current multiobjective evolutionary algorithms such as NSGA-II and SPEA2 have difficulties to provide solutions close to the Pareto front if the number of objectives is high (Khare et al., 2003; Purshouse and Fleming, 2003b; Wagner et al., 2007). Developing new algorithms that are efficient and find good Pareto front approximations also if the number of objectives is high is therefore one major goal with many-objective optimization. However, this is only the second step after one understands why and when algorithms like NSGA-II and SPEA2 have problems with many-objective problems.

To this end, it is crucial to investigate the properties of many-objective problems in general if the number of objectives changes. The main question here is how the dominance relation and, depending on this, the behavior of a MOEA changes if objectives are added to or omitted from a problem formulation.

At first sight, it seems to be unquestioned that the addition of objectives in general adds further difficulties to a problem which was also stated by various authors (Fonseca and Fleming, 1995; Horn, 1997; Deb, 2001; Coello Coello et al., 2007; Fleming et al., 2005). However, a few publications are known, where the opposite is shown for specific problems—either in experimental studies such as (Knowles et al., 2001) or (Jensen, 2004) or in theoretical analyses (Scharnow et al., 2004; Neumann and Wegener, 2006). The question that arises with respect to these inconsistent statements is whether the change in difficulty depends on the problem itself or whether one and the same problem can become harder and easier to solve, depending on the type of objective that is added. Furthermore, it needs to be investigated why and when additional objectives can help or when they add further difficulties to a problem.

In recent years, several publications have been published that are dealing with the stated questions of how the Pareto dominance relation changes if the set of objectives is altered, see for example (Gal and Leberling, 1977; Gal and Hanne, 1999; Fliege, 2007; Mäkelä and Nikulin, 2008; Malinowska, 2008). However, all of them investigate the influence of additional objectives on the Pareto set solely and within a scenario where the objective functions are known explicitly. How the addition of objectives effects multiobjective evolutionary algorithms in a black-box-scenario, where the objective functions are *not* given explicitly but can only be sampled, still remained open in the beginning of this thesis project.

Decision Making with Many Objectives

A decision maker who has to evaluate a set of solutions, e.g., as found by a MOEA, is basically overstrained if many objectives have to be taken into account. The difficulty of visualizing high-dimensional data and the huge amount of information are two reasons for that.

Automatically providing a compact representation of the Pareto front approximation under consideration while the dominance relation among the solutions is preserved is one aspect of many-objective optimization scenarios that will have a high impact in practice but which received only little attention so far. Other questions that a decision maker is interested in in practice need to be tackled as well. Amongst others they include the questions which objectives are conflicting with each other; which objectives are the most important ones, or which objectives are necessary and which ones are redundant.

From the above questions, mainly the question of conflicting objectives has been studied before this thesis project. However, it turns out that for three formally stated conflict definitions (Deb, 2001; Purshouse and Fleming, 2003a; Tan et al., 2005), an example can be given for which the dominance relation does not change if an objective is omitted—although with respect to all three conflict definitions, the original set of objectives is conflicting. This observation leads to the question of whether a conflict definition can be given that accounts for changes in the Pareto dominance relation if and only if the objective set contains conflicting objectives. Furthermore, a measure of how much the Pareto dominance relation is changed, i.e., a definition of a *degree of conflict* would complete this study and might help a decision maker to know how accurate a set of solutions is with respect to all objectives if only a subset of them is considered.

Search Algorithms for Many-Objective Problems

The high number of objectives in many-objective problems causes problems not only for decision making but also for the search itself. Since state-of-the-art MOEAs like NSGA-II and SPEA2 have difficulties to optimize the objective functions with respect to the Pareto dominance relation due to many incomparable solutions, several studies proposed the usage of the hypervolume indicator as a selection criterion (Huband et al., 2003; Beume et al., 2007; Igel et al., 2007). Within these so-called hypervolume-based MOEAs, the hypervolume indicator induces a search direction towards the Pareto front where the incomparabilities in the dominance relation do not allow for an efficient optimization.

However, also hypervolume indicator based MOEAs have a drawback if the number of objectives is high: the best known algorithms to exactly compute the hypervolume indicator are exponential in the number of objectives (Beume and Rudolph, 2006). In the meantime, the $\#\mathcal{P}$-hardness of the hypervolume computation has been shown (Bringmann and Friedrich, 2008) which indicates that we cannot expect to find significantly faster (exact) algorithms in the future. Hence, to apply hypervolume-based MOEAs to problems with many objectives, the computation time needs to be reduced in practice by applying new techniques. Two different approaches seemed to be reasonable in the beginning of this thesis project. On the one hand, Monte Carlo sampling could be used to estimate the hypervolume indicator values; this approach has been shown to be successful in the meantime in several studies, e.g., (Bader et al., 2008; Bader and Zitzler, 2009; Auger et al., 2009a). On the other hand, a decrease in the running time can also be achieved by reducing the number of objectives automatically before the hypervolume indicator is applied. With respect to the second approach, the question remains whether the reduced objective set still allows for an efficient search direction.

Regarding objective reduction approaches applied during search, not many studies are known. Besides the classical scalarization techniques that transform a many-objective problem into a single-objective one (see Miettinen, 1999; Ehrgott, 2005, for an overview), that have for example been used by Hughes (2003) in a multiobjective optimizer, only one objective reduction approach was known when this thesis project started: Deb and Saxena (2005) proposed to use principal component analysis (PCA, (see Jolliffe, 2002)) to reduce the objectives in an iterative MOEA procedure (published as Deb and Saxena, 2006). The same authors also published variations of this approach later on that allow for a better detection of redundant information that can be omitted in a later MOEA run (Saxena and Deb, 2007, 2008). However, the approaches of Deb and Saxena do not aim at preserving the Pareto dominance relation and are not able to report an error measure if objectives are omitted.

One other aspect of hypervolume-based search remained open in the beginning of this thesis project. Using the hypervolume indicator to guide the search introduces a certain bias, i.e., certain solution sets are preferred against others due to their higher hypervolume indicator value although the Pareto dominance relation might state their incomparability. In the light of this fact, two main questions can be identified. First, one should investigate the bias of the hypervolume indicator. In particular, the question arises which (finite) set of μ points maximizes the hypervolume indicator—in other words, what is the optimization goal in hypervolume-based search. Second, the question remains open how user preferences can be incorporated into hypervolume-based search algorithms if the inherent bias of the hypervolume is not desired by the decision maker.

Many different approaches of how user preferences can be modeled and how they can be incorporated into multiobjective search algorithms are known—Coello Coello (2000), Rachmawati and Srinivasan (2006), and Jaszkiewicz and Branke (2008) provide surveys of these approaches with an emphasis on the incorporation into MOEAs. As the authors of the surveys mention, the known methods are not applicable if many objectives are to be optimized. Combining them with the hypervolume indicator and the above mentioned ideas to make hypervolume-based algorithms efficient enough for many-objective optimization would solve this problem. The approach of including arbitrary user preferences into the search could furthermore be used within an interactive search scenario (Miettinen, 1999) in which a decision maker articulates her preferences, e.g., towards desired reference points, during search.

1.5 Contributions

The main aim of this thesis is to provide both theoretical foundations about many-objective optimization problems and new approaches to tackle them in terms of hypervolume-based search. In the course of this thesis, several contributions to the field of many-objective optimization have been made—ranging from a general investigation about what happens if objectives are added to or omitted from the problem formulation to the question how preferences can be incorporated into the search.

Effects of Adding, Omitting, and Aggregating Objectives

Besides a general framework of objective conflicts and the definition of redundant and minimum objective sets, the question how the Pareto dominance relation is changing if objectives are added, removed, or aggregated has been investigated in detail. Both theoretical and experimental investigations gave insights into differences between problems with only one or a few objectives and those with many.

Rigorous running time analyses of simple MOEAs have been performed to theoretically investigate the effect of additional objectives on the optimization process. In particular, it has been shown that the addition of objectives can either increase or decrease the expected running time of a simple MOEA depending on the added objective. Furthermore, we have shown that even for two equally difficult single-objective problems multiobjective optimization can be beneficial, i.e., solving the two problems simultaneously in a biobjective scenario is faster than optimizing them in separated single-objective optimization runs.

Decision Making with Many Objectives

The main contribution with respect to assisting decision makers in many-objective optimization scenarios is the development of a general objective reduction framework to automatically reduce the number of objectives while the Pareto dominance relation is preserved or only slightly changed.

To this end, the δ-error as a general error measure has been proposed to quantify changes in the dominance structure and that is based on the general investigations on the effects of additional objectives mentioned above.

Using this error measure, two problem formulations are proposed. On the one hand, we ask for an objective subset of given size resulting in the smallest δ-error (`k-EMOSS` problem). On the other hand, we ask for the smallest objective set yielding a predefined δ-error (δ-`MOSS` problem). The complexity of both problems is analyzed and their $\mathcal{NP}$-hardness is proven. Both exact and heuristic objective reduction algorithms tackling these problems have been proposed and analyzed. In addition, the approach of omitting objectives has been generalized to objective aggregation for which algorithms have been proposed as well.

The usefulness of these new objective reduction techniques has been shown in several applications. Various well-known test problems and a radar waveform problem, proposed by Hughes (2007), have been used to show the benefits of the new approach with respect to automatically deriving relationships among objective sets and providing a reduced representation of the high-dimensional outcomes of MOEAs in general.

Search Algorithms for Many-Objective Problems

The contributions to the development of efficient evolutionary algorithms for many-objective problems especially with respect to hypervolume-based algorithms are threefold.

First, a rigorous running time analysis of a hypervolume indicator based MOEA has shown for the first time that the concept behind hypervolume-based selection in MOEAs is working from a theoretical perspective. Second, the hypervolume indicator has been generalized to a weighted case to incorporate user preferences into the search within a many-objective scenario. Different weights can be assigned to certain regions of the objective space indicating the importance of these regions for the decision maker. Preliminary experimental results show that a MOEA optimizing this weighted hypervolume is attracted by regions with high weights and therefore can be guided by the decision maker's preferences[9]. Third, the incorporation of the proposed objective reduction techniques into a hypervolume-based algorithm showed the usefulness of the approach. Not only could the running time be reduced but also solutions of higher quality can be obtained for problems with a high number of objectives.

[9] In the meantime, Auger et al. (2009a) showed that also in a many-objective scenario, the proposed weighted hypervolume indicator can guide the search effectively if combined with Monte Carlo sampling.

1.6 Overview

The thesis is organized as follows. In the next chapter, we start with general investigations on the effect of additional objectives, both from the perspective of the underlying dominance relation and with respect to the running time of multiobjective evolutionary algorithms. Chapter 3 investigates the problem of objective reduction by omitting objectives in detail. Based on a generalized error measure, two different problem definitions are introduced, their complexity is analyzed, and both exact and heuristic algorithms are presented and applied to various problems. The chapter also considers the more general problem of objective reduction by aggregating objectives and presents corresponding algorithms to reduce the number of objectives. The benefit of objective reduction as a tool for deriving statements about the relationships between objectives is shown on the real world example of a radar waveform optimization problem. Chapter 4 investigates hypervolume indicator based search algorithms from three different perspectives. First, the running time of a hypervolume-based algorithm is analyzed theoretically for the first time. Second, the incorporation of user preferences into the search is tackled by proposing a weighted version of the standard hypervolume indicator. Third, the question how the objective reduction techniques from Chapter 3 can be used to speed-up hypervolume-based search for many-objective optimization is answered. In the end of this thesis, conclusions are drawn and future research directions are pointed out in Chapter 5. A list of acronyms and a list of symbols can be found in Appendix B and Appendix C respectively.

2

Effects of Adding Objectives

Several multiobjective problems in practice contain many objectives, see for example (Paechter et al., 1998; Gobbi et al., 1999; Coello Coello and Hernández Aguirre, 2002; Rudenko et al., 2002). Furthermore, it is even convenient in practice to add as many objectives as possible to a problem formulation without the need to specify preferences among them[1]. On the other hand, all existing multiobjective evolutionary algorithms in the beginning of this thesis project were known to have difficulties to produce a solution set close to the Pareto front if the number of objectives is larger than 3 (Khare et al., 2003; Wagner et al., 2007; Purshouse and Fleming, 2007). Knowles and Corne (2007) showed under the assumption of a one-to-one mapping between decision and objective space that for some problems, even a simple random search can outperform evolutionary algorithms (in this case PESA-II (Corne et al., 2001)) if the number of objectives is more than 5. Furthermore, the investigations of Purshouse and Fleming (Purshouse and Fleming, 2007) show that the behavior of a multiobjective evolutionary algorithm on a problem with few objectives cannot be generalized to a higher number of objectives.

In the light of these facts, the two questions arise which properties of many-objective problems cause the difficulties for multiobjective evolutionary algorithms and how we can deal with them to develop algorithms that are efficient if the number of objectives is high.

[1] In such a case, the idea is to learn about the problem and which objectives are interesting after a first optimization of all objectives. Later on, when information about the objectives' trade-offs is known, the decision which objectives to consider in the end is supposed to be easier than defining preferences among the objectives beforehand.

One explanation, various researchers list in favor of the assumption that, in general, the search becomes harder the more objectives are involved is the increase in the number of incomparable solutions (Fonseca and Fleming, 1995; Horn, 1997; Deb, 2001; Fleming et al., 2005; Purshouse and Fleming, 2007; Knowles and Corne, 2007; Coello Coello et al., 2007). Winkler (1985) proved that the number of incomparable solutions increases if further randomly generated objectives are added. Thereby, on the one hand the Pareto front may become larger and on the other hand the power of the dominance relation to guide the search may diminish.

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 (2004) 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. (2001) 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 have been shown to reduce optimization cost until good solutions are found (Mezura-Monets and Coello Coello, 2007) although Runarsson and Yao (2005) pointed out that this is not effective on all kinds of problems due to a wrong search bias. Furthermore, the addition of an objective has been shown to be beneficial in terms of reducing bloat in Genetic Programming, see for example (Bleuler et al., 2001), (Ekárt and Németh, 2001), and (de Jong et al., 2001). Besides these empirically oriented studies, there are theoretical results supporting the hypothesis that multiobjectivization can help: Scharnow et al. (2004) showed that the Single Source Shortest Path problem is easier to solve for simple EAs when formulated as a multiobjective problem; Neumann and Wegener (2006) 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. Recently, similar investigations have been made for finding good approximations of the vertex cover problem (Friedrich et al., 2007).

These contradicting opinions on what happens if the number of objectives is increased do not allow for a general statement and the question occurs whether the effect of an additional objective depends on the problem formulation itself or if one and the same problem can become both harder and easier depending on the type of objective that is added. This chapter investigates the effects of additional objectives from a theoretical perspective. Thereby, the goal is twofold:

- On the one hand, we provide a basic investigation on how adding and omitting objectives in a multiobjective problem scenario affects the (weak) Pareto dominance relation between the solutions (Sec. 2.1).

- On the other hand, we show how the addition of objectives affects the running time of multiobjective evolutionary algorithms by means of rigorous running time analyses (Sec. 2.2).

The results build the basis of new approaches to tackle many-objective problems in terms of objective reduction as presented in the next chapter.

2.1 The Effects of Additional Objectives on the Dominance Structure

Most multiobjective evolutionary algorithms are explicitly or implicitly using the (weak) Pareto dominance relation to guide the search. For example, if non-dominated sorting (NSGA-II), dominance count (SPEA2), or the hypervolume indicator (SMS-EMOA, MO-CMA-ES) are taken into account, all mentioned algorithms favor a set of solutions against a second one if the former dominates the latter. Understanding the effects of additional objectives on the dominance structure therefore helps to understand how dominance relation based multiobjective evolutionary algorithms are affected by additional objectives.

To illustrate how the weak Pareto dominance relation is modified when objectives are removed from or added to a problem, its representation in terms of a *relation graph* is useful. The relation graph for the weak Pareto dominance relation is given by the tuple $(A, \preceq_{\mathcal{F}'})$ for a solution set A and an objective set $\mathcal{F}'$. It contains for each solution a corresponding node and for each solution pair $\vec{x}, \vec{y} \in A \subseteq X$ an edge from the node associated with $\vec{x}$ to the node associated with $\vec{y}$ if and only if $\vec{x}$ weakly dominates $\vec{y}$ with respect to $\mathcal{F}'$.

Example 1. *Consider the multiobjective scenario depicted in Fig. 3 by a parallel coordinates plot*[2]*. There are four objectives f_1, f_2, f_3, and f_4, and four solutions $\vec{a}$ (solid line), $\vec{b}$ (dashed), $\vec{c}$ (dotted), and $\vec{d}$ (dashed-dotted) which are pairwisely incomparable with respect to the objective set $\mathcal{F} = \{f_1, f_2, f_3, f_4\}$. The relation graphs for all possible relations $\preceq_{\mathcal{F}' \subseteq \mathcal{F}}$ that are associated with specific objective subsets are shown in Fig. 4. As the solutions are pairwisely incomparable, the relation graph of $\preceq_{\{f_1, f_2, f_3, f_4\}}$ contains only the reflexive edges (Fig. 4(o)).*

Now, how does adding an objective affect the overall relation graph? Starting with a single-objective problem, the weak Pareto dominance rela-

[2] cf. (Purshouse and Fleming, 2003a)

	f_1	f_2	f_3	f_4
$\vec{a}$	5.0	4.5	2.0	1.0
$\vec{b}$	2.5	1.0	1.0	7.0
$\vec{c}$	2.0	3.0	4.0	8.0
$\vec{d}$	1.0	4.0	3.0	6.0

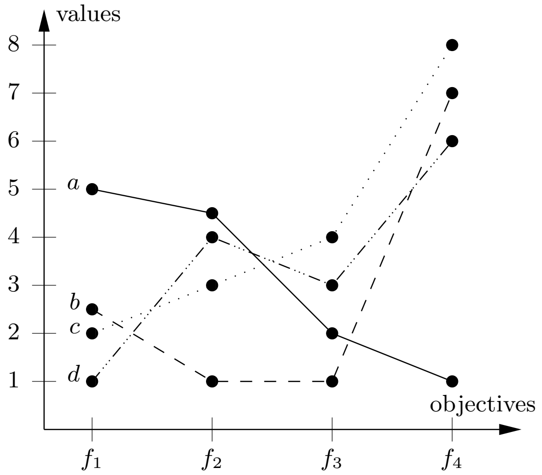
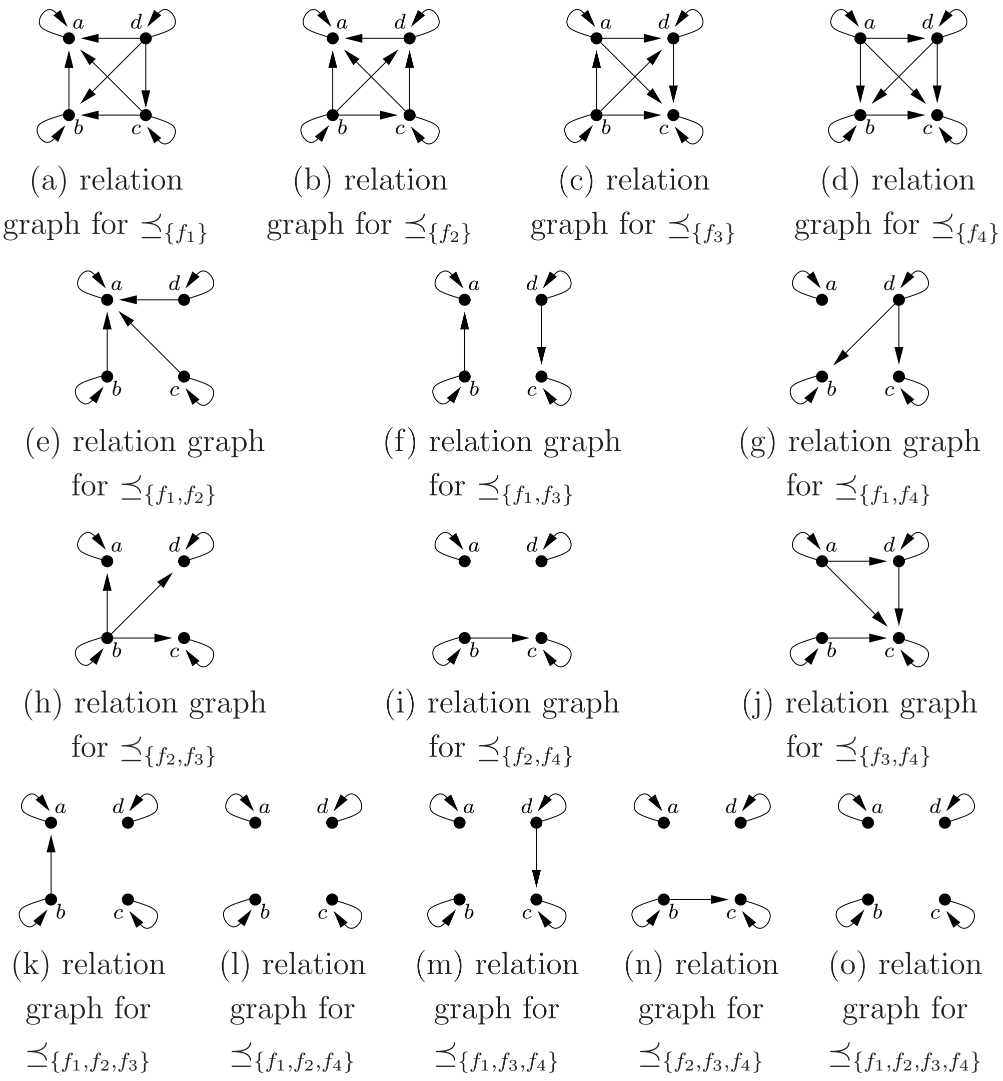


Figure 3: Objective function values (left) and parallel coordinates plot (right) for the given example with four solutions and four objectives; all objectives have to be minimized.

tion $\preceq_{\{f_i\}}$ always forms a total preorder[3], i.e., all solution pairs are comparable, cf. Fig. 4(a-d). With an additional objective, the relation between any two solutions $\vec{x}, \vec{y} \in X$ can be changed in two ways: (i) $\vec{x}$ and $\vec{y}$ have been comparable, but not indifferent, and now become incomparable because $\vec{x}$ is better regarding the first objective and $\vec{y}$ regarding the second (or vice versa), or (ii) $\vec{x}$ and $\vec{y}$ have been indifferent, but now one solution dominates the other one because it is better regarding the additional objective. The same holds if an objective is added to a multiobjective problem. Regarding the relation graph, that means that an additional objective either leaves the edges between two nodes unchanged or removes exactly one edge; overall, adding objectives can only remove edges from the relation graph. Contrariwise, if one or several objectives are omitted, edges are added to the relation graph: incomparable solutions may become comparable, and a solution dominated by another one may become indifferent to it.

Example 2. *Consider the solution pair $\vec{a}, \vec{b}$ in Fig. 3. When taking only objective f_1 into account as a single-objective minimization problem, solution $\vec{b}$ is preferred to solution $\vec{a}$, i.e., $\vec{b}$ weakly dominates $\vec{a}$, see Fig. 4(a). If objective f_2 is added, $\vec{b}$ still weakly dominates $\vec{a}$ since solution $\vec{b}$ has smaller objective values than $\vec{a}$ in both objectives f_1 and f_2 (Fig. 4(b) and (e)). In the case of adding objective f_4, the two solutions $\vec{a}$ and $\vec{b}$ become incomparable due to the fact that $\vec{a}$ weakly dominates $\vec{b}$ with respect to f_4. The edge between $\vec{a}$ and $\vec{b}$ in the corresponding relation graph disappears, see Fig. 4(l).*

[3] A relation that is reflexive, transitive and total is called total preorder; if it is also antisymmetric, it is called a total order. Note, that the weak Pareto dominance relations for single objectives are usually only total preorders and not total orders, since solutions with the same objective value can exist, i.e., the antisymmetry of $\preceq_{\{f_i\}}$ cannot be guaranteed. For the definition of preorders and the background of order theory in general, we refer to extensive text books such as (Schröder, 2003) or (Harzheim, 2005).



(a) relation graph for $\preceq_{\{f_1\}}$

(b) relation graph for $\preceq_{\{f_2\}}$

(c) relation graph for $\preceq_{\{f_3\}}$

(d) relation graph for $\preceq_{\{f_4\}}$

(e) relation graph for $\preceq_{\{f_1,f_2\}}$

(f) relation graph for $\preceq_{\{f_1,f_3\}}$

(g) relation graph for $\preceq_{\{f_1,f_4\}}$

(h) relation graph for $\preceq_{\{f_2,f_3\}}$

(i) relation graph for $\preceq_{\{f_2,f_4\}}$

(j) relation graph for $\preceq_{\{f_3,f_4\}}$

(k) relation graph for $\preceq_{\{f_1,f_2,f_3\}}$

(l) relation graph for $\preceq_{\{f_1,f_2,f_4\}}$

(m) relation graph for $\preceq_{\{f_1,f_3,f_4\}}$

(n) relation graph for $\preceq_{\{f_2,f_3,f_4\}}$

(o) relation graph for $\preceq_{\{f_1,f_2,f_3,f_4\}}$

Figure 4: Relation graphs for the solutions depicted in Fig. 3.

Since a solution $\vec{x}$ weakly dominates another solution $\vec{y}$ with respect to an objective set if and only if $\vec{x}$ weakly dominates $\vec{y}$ with respect to every single objective, an edge can only be contained in the relation graph for $\preceq_{\mathcal{F}}$ if for every subset $\mathcal{F}' \subseteq \mathcal{F}$ the corresponding relation graph contains the edge. This can be formalized in the following theorem.

Theorem 1. *If* $\mathcal{F} = \{f_1, \ldots, f_k\}$ *is a set of* k *objective functions then* $\preceq_{\mathcal{F}} = \bigcap_{1 \leq i \leq k} \preceq_{\{f_i\}}$.

Proof. Let $\vec{x}, \vec{y} \in X$. Then $\vec{x} \preceq_{\mathcal{F}} \vec{y} \iff \forall i \in \{1, \ldots, k\} : f_i(\vec{x}) \leq f_i(\vec{y}) \iff \forall i \in \{1, \ldots, k\} : \vec{x} \preceq_{\{f_i\}} \vec{y} \iff (\vec{x}, \vec{y}) \in \bigcap_{1 \leq i \leq k} \preceq_{\{f_i\}}$. □

Contrary to these results about adding objectives, Handl et al. (2008) recently investigated how the *decomposition* of a single objective into different objective functions changes the dominance structure. Here, a single objective function is said to be decomposed into k new objectives if for each solution in the decision space, the sum of the new objective values equals the original single-objective function value. The observations by Handl et al. (2008) point out that with the decomposition of an objective into several ones, the Pareto dominance relation can change in less different ways than when adding a new objective: whenever a solution dominates another solution with respect to the decomposed objectives (without being indifferent), the same relation already holds for the original objective (the "gradient cannot be reversed"); in other words, the only possible change is that indifferent solutions can become incomparable. The possibility of introducing a direction by decomposing the single objective is not given (a "gradient cannot be introduced").

2.2 The Effect of Additional Objectives on Multiobjective Evolutionary Algorithms

After investigating the influences of adding or omitting objectives on the dominance structure, the question remains, how these changes in the problem definition affect the running time of evolutionary algorithms.

Rigorous running time analyses of evolutionary algorithms on single-objective problems have been carried out extensively in the last decade (Jansen and Wegener, 2001; He and Yao, 2002; Droste et al., 2002; Beyer et al., 2002; Giel and Wegener, 2003, and many more) since the early convergence results, e.g., by Rudolph (1994). For a detailed overview of recent theoretical studies on single-objective evolutionary algorithms, we refer to the review of Oliveto et al. (2007b). Also investigations of the expected running time of multiobjective evolutionary algorithms have been performed, amongst others (Giel, 2003; Laumanns et al., 2004b,a; Horoba and Neumann, 2008; Friedrich et al., 2008). Some of these running time analyses have shown that a decomposition of a single-objective problem into two or more objectives is beneficial for simple MOEAs. Scharnow et al. (2004), for example, have shown that the single source shortest path problem is easier to solve in a multiobjective formulation than in the single-objective version. The same holds for the minimum spanning tree (Neumann and Wegener, 2006) or the vertex cover problem (Friedrich et al., 2007). A more general investigation about what can happen if a single-objective problem is decomposed into two objectives has been carried out by Handl et al. (2008) recently. On the basis of four different problems, the authors show in this study that the decomposition of a single objective into two objectives can both reduce and increase the expected running time of simple evolutionary algorithms—dependent on the decomposition.

However, the question of how a multiobjective evolutionary algorithm is affected if a *new* objective is added and the single objective is kept in the new formulation remains open. Furthermore, no examples are known whether one and the same problem can become harder and easier depending on the chosen objective. It is also not obvious whether two equally difficult problems can be simultaneously solved in a shorter time if they are combined to one problem.

All three questions will be addressed in the following by means of rigorous running time analyses and supplementary experimental results. Section 2.2.2 presents a problem for which simple evolutionary algorithms can become both slower and faster if different objectives are added to the problem. In contrast to (Scharnow et al., 2004), (Neumann and Wegener, 2006), (Friedrich et al., 2007), and (Handl et al., 2008), where the original objective is replaced by two other objectives, we here consider the case that the original objective remains in the objective set. The algorithms considered in the analyses are presented in Sec. 2.2.1. Section 2.2.3 shows that two equally difficult single-objective problems can be solved faster if they are combined to a multiobjective problem and Sec. 2.2.4 experimentally underpins the theoretical results, shown before, for the case of biobjective problems that become 3-objective problems by adding a third objective.

The main observation behind both the running time analyses and the experimental studies is that problems may contain so-called *plateaus* (Jansen and Wegener, 2001). A plateau is a part of the search space where the problem does not indicate any search direction. More precisely, the definition of a plateau involves a neighborhood function: whenever in a set of neighbored solutions, the objective functions do not indicate a search direction—because all solutions are either pairwisely incomparable or indifferent—the solution set is called a plateau.

As we will show in the reminder of this chapter, an additional objective can remove or introduce those plateaus. Since an evolutionary algorithm, be it a single- or a multiobjective one, performs a random walk on these plateaus as it was shown for some of the well-known combinatorial optimization problems (Giel and Wegener, 2003; Neumann and Wegener, 2004; Witt, 2005)[4], the removal or introduction of plateaus by adding objectives can change the running time behavior of evolutionary algorithms 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 evolutionary algorithm. By *good* and *deceptive* search direction we mean that the Pareto dominance relation is giving

[4] Although this fact was only reported for combinatorial optimization problems, also continuous MOEAs might perform a random walk on plateaus. Furthermore, there is no obvious reason why non-combinatorial problems cannot contain plateaus as well.

Algorithm 1 (1+1)EA

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

the evolutionary algorithm the correct or wrong information about where to expect better solutions: the introduced search direction is called "correct" or "good" if following the indicated direction will reduce the distance towards the Pareto set (in terms of a distance measure such as Hamming distance); if following the indicated direction will drive the MOEA away from the Pareto set, the search direction is called deceptive.

2.2.1 Global SEMO and Other Simple Algorithms

Many multiobjective evolutionary algorithms are known from the literature, cf. Sec. 1.3. However, for theoretical running time analyses, only very simple algorithms have been considered so far due to the complex analyses necessary for practical but more complicated algorithms. Also here, we consider only simple algorithms that are well-known from various theoretical studies but at the same time contain the basic concepts of other more complicated Pareto dominance based algorithms which allows us to rigorously analyze their running time behavior and to study the changes if additional objectives are considered.

As to the decision space, we consider the set of binary strings of length n, i.e., $X = \{0,1\}^n$. The objective space will either be spanned by one or by more discrete objectives, i.e., the objective space is $\mathbb{N}^k$ with varying k. Concerning the algorithms, we therefore examine both single-objective ($k = 1$) and multiobjective ($k > 1$) evolutionary algorithms. Note, that we—in contrast to the previous section—consider maximization problems here to comply with the original publications (Brockhoff et al., 2007a, 2009).

For single-objective optimization problems, our analyses are based on the (1+1)EA (Algorithm 1) which has been considered in many theoretical investigations on simple test functions, see, e.g., (Droste et al., 2002), as well as on some of the best-known combinatorial optimization problems (amongst others Giel and Wegener, 2003; Neumann and Wegener, 2004; Witt, 2005; Oliveto et al., 2007a; Neumann, 2008). 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 $1/n$. Note that for the theoretical running time analyses to follow, we assume that the algorithm is never stopped; instead, we are interested in the number of objective function calls until an optimal solution is found for the first time.

Algorithm 2 Global SEMO

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

To this end, we define the number of constructed solutions until an optimal one has been created for the first time as the *running time* or *optimization time* of such a single-objective algorithm. Since the algorithms we investigate are of stochastic nature, the running time is a discrete random variable. Often, and also in this study, the expected running or optimization time is sought, i.e., the expectation of the number of objective function evaluations until the first optimal solution is found.

In the studies to follow, we compare the single-objective (1+1)EA with its multiobjective counterpart called Global SEMO (Global Simple Evolutionary Multiobjective Optimizer; Algorithm 2) (Laumanns et al., 2002b; Giel, 2003) which has been investigated in the context of different multiobjective problems, e.g., spanning tree problems (Neumann, 2004; Neumann and Wegener, 2006), vertex cover (Friedrich et al., 2007), or minimum cut and multicut problems (Neumann et al., 2008; Neumann and Reichel, 2008). Global SEMO starts with an initial population P that consists of one single individual that is chosen uniformly at random in the decision space. In each generation, an individual $\vec{x}$ is chosen uniformly at random from P to produce one child $\vec{x}'$ by mutation. In the mutation step, each bit of $\vec{x}$ is flipped with probability $1/n$ to produce the offspring $\vec{x}'$. After that, $\vec{x}'$ is added to the population if it is not dominated by any individual in P. If $\vec{x}'$ is added to P all individuals of P that are dominated by $\vec{x}'$ or have the same objective vector as $\vec{x}'$ are removed from P. As for the (1+1)EA, Global SEMO never stops and we are interested in the number of objective function computations until for each objective vector in the Pareto front a solution is found that maps to this vector. Global SEMO is detailed in Algorithm 2.

With respect to multiobjective problems, we define the *running time* or *optimization time* of a multiobjective evolutionary algorithm as the number of constructed solutions until for each point of the Pareto front a solution that maps to this point in objective space has been included into the population. The expectation of this random variable is again called *expected running time* or *expected optimization time*.

In addition to the above introduced algorithms (1+1)EA and Global SEMO, we consider variants that use a different mutation operator. This asymmetric mutation operator has been proposed by Jansen and Sudholt (2005) in order to have a simple mutation operator that preserves the number of bits that are set to one. Jansen and Sudholt (2005) argue that in many combinatorial optimization problems that are tackled by evolutionary algorithms, the number of bits that are set to one is either very high or very low for optimal solutions[5]. The normal mutation operator that flips each bit with probability $1/n$ always tends to produce offspring with an equal number of ones and zeros—the probability to generate an offspring with the same number of ones is low, especially if the parent solution has only a few bits set to one. This observation leads to the assumption that the new asymmetric mutation operator might reduce the running time of evolutionary algorithms on some combinatorial optimization problems—that the usage of asymmetric mutation can drastically decrease the running time of simple evolutionary algorithms on simple test functions has been proven theoretically in the work of Jansen and Sudholt (2005).

To define the asymmetric mutation operator, let us denote the number of ones in a given bitstring $\vec{x}$ by $|\vec{x}|_1$ and the number of zeros in this string by $|\vec{x}|_0$. The asymmetric mutation operator of Jansen and Sudholt (2005) flips each bit of the parent solution $\vec{x}$ dependent on its value. If the bit is set to 1, it is flipped to 0 with a probability that is anti-proportional to the number of one bits in the bitstring of $\vec{x}$, more precisely, with a probability of $1/(2|\vec{x}|_1)$. Contrariwise, the same holds for a flip from a 0 to a 1 while zeros and ones are interchanged. Algorithm 3 shows the pseudo code of this asymmetric operator. We denote the algorithms that differ from the (1+1)EA and Global SEMO by using the mutation operator given in Algorithm 3 by (1+1)EA$_{asy}$ and Global SEMO$_{asy}$ respectively.

Algorithm 3 Asymmetric Mutation Operator

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

[5] One example, given in the work of Jansen and Sudholt (2005) is the minimum spanning tree problem in an edge selection representation where all minimum spanning trees have exactly $n-1$ of the often $\Theta(n^2)$ bits set to 1.

2.2.2 Adding Objectives to a Plateau

The basis for all running time analyses to follow has been given already in Sec. 2.1: the addition of an objective can only make

(i) comparable solutions incomparable and

(ii) an indifferent relation between solutions a comparable one.

Of course, both cases can occur simultaneously if an objective is added. Surprisingly, in both cases, a problem can become easier or harder to solve as is shown analytically in the following. Generally speaking, case (i) turns a region, where the weak Pareto dominance relation indicates a direction, into a plateau of incomparable solutions, whereas case (ii) turns a plateau of indifferent solutions into a region where a direction is given by the weak Pareto dominance relation. 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 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 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 relations between solutions will lead to the Pareto front or behave deceptively, the computational effort to identify the Pareto optima may decrease or increase.

The running time analyses presented in this section mainly investigate the second type of change in the dominance relation on the basis of a simple plateau function. The problem formulation $\textsc{Plateau}_1 : \{0,1\}^n \to \mathbb{N}$ contains a set of $n-1$ search points that form a plateau, i.e., these search points have the same objective value of $n+1$. We denote by $SP_1 := \{1^i 0^{n-i}, 1 \leq i < n\}$ this set of search points and define $\textsc{Plateau}_1$ as

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

Note, that this function is similar to the function $\textsc{spc}_n$ already investigated by Jansen and Wegener (2001). The only difference is that the all-zero string 0^n belongs to the plateau of indifferent solutions in $\textsc{spc}_n$ but not in $\textsc{Plateau}_1$. The differences in the absolute objective function values between the two problem definitions do not influence the dominance structure between the solutions.

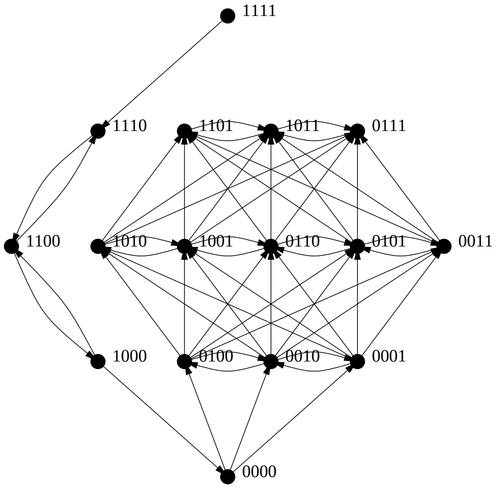


Figure 5: Relation graph for the objective function $\textsc{Plateau}_1 : \{0,1\}^4 \to \mathbb{N}$. Reflexive and transitive edges are omitted for clarity.

The relation graph of $\textsc{Plateau}_1$ for $n = 4$ is shown in Fig. 5. 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 1^n is large. Nevertheless, the structure of the plateau admits a fair random walk. The following theorem shows an expected optimization time of $\Theta(n^3)$ for the (1+1)EA.

Theorem 2. *The expected running time of the (1+1)EA on* $\textsc{Plateau}_1$ *is* $\Theta(n^3)$.

Proof. As the relative structure of $\textsc{Plateau}_1$ and $\textsc{spc}_n$ (as defined in (Jansen and Wegener, 2001)) are identical besides the inclusion of 0^n in the plateau or not, we can reuse all ideas used in the proof of Jansen and Wegener (2001) for the expected running time $O(n^3)$ of the (1+1)EA on $\textsc{spc}_n$. Therefore, also on $\textsc{Plateau}_1$ the expected running time of the (1+1)EA can be bounded by $O(n^3)$.

We will now prove a lower bound of $\Omega(n^3)$. In the initialization step of the (1+1)EA, a solution $\vec{x} \in \{0,1\}^n$ is produced that fulfills $|\vec{x}|_1 \leq \frac{2}{3}n$ with probability $1 - o(1)$ which can be shown by applying Chernoff bounds, cf. (Motwani and Raghavan, 1995). As long as the current solution is not in SP_1 and not equal to 1^n, the value $|\vec{x}|_1$ is non-increasing. Thus, the first individual $\vec{x}$ chosen by the (1+1)EA that is in the set SP_1 has the property $|\vec{x}|_1 \leq \frac{2}{3}n$ 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 bits of $\vec{x}$ that are 0's or the last bits of $\vec{x}$ that are 1's can be flipped. The probability to flip 4 or more of these bits 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 that is 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 $\vec{x}$ 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_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 (Doerr et al., 2006). 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 (Giel and Wegener, 2003; Doerr et al., 2006; Neumann, 2008) point out that some of the well-known combinatorial optimization problems such as maximum matching or Eulerian cycle have natural objective functions introducing plateaus of a similar structure as in Plateau$_1$. This observation shows that the investigation of the simple Plateau$_1$ function and how the addition of objectives influences the running time of simple evolutionary algorithms play a key role for understanding the affects of additional objective on other combinatorial optimization problems. A better understanding of the effects of additional objectives to a problem may also lead to more efficient search heuristics if objectives are added.

In the following, we investigate the effect of adding two of the simplest non-trivial objective functions to the problem Plateau$_1$ and consider the behavior of Global SEMO on these functions. More precisely, we consider the maximization of the biobjective problems

$$\textsc{Plom}(\vec{x}) := (\textsc{Plateau}_1(\vec{x}),\ |\vec{x}|_1)$$

$$\textsc{Plzm}(\vec{x}) := (\textsc{Plateau}_1(\vec{x}),\ |\vec{x}|_0)$$

and show that Global SEMO is faster (cf. Theorem 3) on Plom and exponentially slower on Plzm (cf. Theorem 5) than the (1+1)EA on Plateau$_1$. Note that according to Knowles et al. (2001), two main properties have to be fulfilled if additional objectives or the decomposition into several new objectives should be beneficial in terms of multiobjectivization for discrete problems: (i) the single-objective optimum should be included in the new Pareto set[6] and (ii) the Pareto front of the new problem should be as small

[6] This property is always fulfilled if objectives are only added and the original ones are kept.

as possible. Otherwise, the optimization algorithm cannot find the optimum of the original problem in reasonable time when optimizing the reformulated problem. Both the inclusion of the single-objective optimum within the new Pareto front and the small size of the resulting front is given for the two problems Plom and Plzm[7]. We would also like to mention that the focus in the following lies on the investigation of the search behavior on the plateau SP_1. The fact that the problems Plom and Plzm only have one or two Pareto-optimal solutions is not a restriction. Problems with larger Pareto sets can be constructed for which the same differences in running time can be shown if different objectives are added. As mentioned above, problems where the addition of objectives can reduce the running time of evolutionary algorithms are known in practice (Jensen, 2004; Greiner et al., 2007) and the reason for the study to follow is to understand the reasons for such a change in running time.

We now consider how Global SEMO optimizes problem Plom. The first observation is that all $\vec{x} \in SP_1$ are comparable in Plom while they are indifferent in Plateau$_1$. The second objective $|\vec{x}|_1$ of Plom gives the Global SEMO the “right direction” to move on the former plateau $(n+1,\cdot)$ up to the only Pareto optimum 1^n. Furthermore, all solution pairs that are not in SP_1 are either indifferent or incomparable which, in addition, does not indicate the “wrong” direction for the algorithm. This can be seen nicely in the relation graph of Plom in Fig. 6. Both observations are the reason why Global SEMO is optimizing Plom significantly faster than the (1+1)EA optimizes Plateau$_1$ which is proven in the following theorem.

Theorem 3. *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 maps to at most $n+3$ different values. If the initial random search point $\vec{x} \in \{0,1\}^n$ 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 (Motwani and Raghavan, 1995) 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 trade off between the number

[7] Note further that for both problems Plom and Plzm, the second objective changes the relation between solutions in comparison to the single-objective problem Plateau$_1$ and therefore changes the runtime behavior of evolutionary algorithms that take the weak Pareto dominance relation into account. This property of the additional objectives is not reflected by previous definitions of conflict between objectives as, e.g., in (Deb, 2001) and Tan et al. (2005), but is covered by the definition of conflict proposed later on. See Chapter 3 for details on the different conflict definitions.

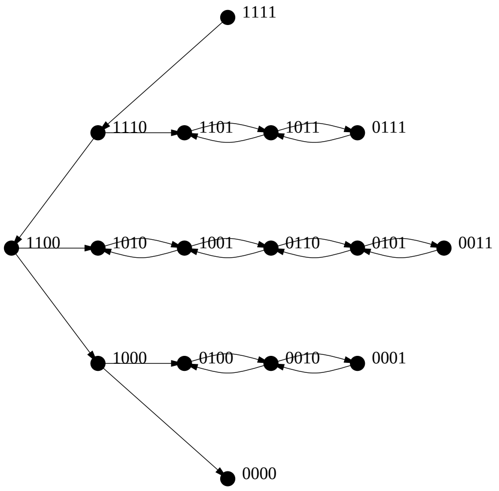


Figure 6: Relation graph for the biobjective problem PLOM : $\{0,1\}^4 \rightarrow \mathbb{N}^2$. Reflexive and transitive edges are omitted for clarity.

of ones and zeros due to the incomparability of the solutions outside SP_1. 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 $\vec{x}$ with $|\vec{x}|_1 = k$ to an $\vec{x}'$ with $|\vec{x}'|_1 = k+1$ 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 even if the path SP_1 is never reached.

For the proof of the lower bound we consider a slight modification of 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 $\vec{x}$ with $\text{PLATEAU}_1(x) > n$ because a new search point outside SP_1 is either incomparable or indifferent to all solutions in the population. We will show that with probability $1 - o(1)$ the modified model is not different from Global SEMO in the phase we are analyzing later on. 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 $\vec{x}$ having $|\vec{x}|_1 = i$ that is newly generated in our modified model is uniformly distributed in $\{\vec{x} \in \{0,1\}^n, |\vec{x}|_1 = i\}$. For every such element $\vec{x}$ the probability that it is mutated to an element $\vec{x}'$ with $|\vec{x}'|_1 = i$ and $\text{PLATEAU}_1(\vec{x}') \geq n$ is exactly $1/\binom{n}{|\vec{x}'|_1}$. Thus, the probability that every produced $\vec{x}' \in \{0,1\}^n$ with $3 \leq |\vec{x}'|_1 \leq n-3$ in $\Theta(n^2 \log n)$ steps fulfills $\text{PLATEAU}_1(\vec{x}') < n$ is at least

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

In other words, with probability $1 - o(1)$ our model behaves in $\Theta(n^2 \log n)$ steps exactly like Global SEMO and produces no solution $\vec{x}'$ for which

$\textsc{Plateau}_1(\vec{x}') > n$ as long as every element $\vec{x}$ of the current population fulfills $3 \leq |\vec{x}|_1 \leq n-3$. For the lower bound proof it is enough to restrict ourselves to this fraction of cases.

Let $a_{\min} := \min_{\vec{x}\in P} \min\{|\vec{x}|_0, |\vec{x}|_1\}$ be the minimal number of ones respectively zeros of an individual in the current population P. Until the first individual $\vec{x} \in \{0,1\}^n$ with $\textsc{Plateau}_1(\vec{x}) > n$ is produced, the value $a_{\min}$ is decreasing and the population size is increasing. More informally, the population spreads over the plateau $X \setminus \{SP_1 \cup \{1^n\}\}$ of incomparable and indifferent solutions. After the initialization, $a_{\min} \geq \frac{n}{3}$ holds with high probability using Chernoff bounds. We regard the phase where $a_{\min}$ is in the range between $\frac{n}{3}$ and $\frac{n}{4}$ and show that the population size after this phase is of order $\Theta(n)$ with probability at least 1/2.

To this end, 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 where $\frac{n}{4} \leq a_{\min} \leq \frac{n}{3}$ is bounded above 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 $\frac{n}{3}$) ones respectively zeros has to be flipped. The probability for this extra flip is at most $\frac{n}{3}/n = \frac{1}{3}$. 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 where $\frac{n}{4} \leq a_{\min} \leq \frac{n}{3}$ 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 $\frac{n}{4}$ ones respectively zeros. 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 $2\,n^{1/4}$ ones respectively zeros left in every element of the current population of size $\Theta(n)$.

For every $\vec{x}$ in the current population, we define $a(\vec{x}) := \min\{|\vec{x}|_0, |\vec{x}|_1\}$ such that $a_{\min} = \min_{\vec{x}\in P}\{a(\vec{x})\}$. Now we consider the time to reduce $a_{\min}$ from $n^{1/4}$ to 3. The probability to produce from a solution $\vec{y}$ with $a(\vec{y}) > a_{\min} + n^{1/4}$ an improving $\vec{z}$ 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 $\vec{z}$ with $|\vec{z}|_1 > |\vec{y}|_1$ by flipping k of the remaining 0-bits (respectively the remaining 1-bits) of a solution $\vec{y}$. The probability to flip k of these bits in a single mutation step of a solution $\vec{y}$ with $a(\vec{y}) \leq a_{\min} + n^{1/4}$ is upper bounded by $\left(\frac{a_{min}+n^{1/4}}{n}\right)^k = O(n^{-3k/4})$. Since the probability that a $\vec{y}$ with $a_{\min} \leq |\vec{y}|_1 \leq a_{\min} + n^{1/4}$ 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 $\vec{y}$ 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 $\vec{z}$ with $a(\vec{z}) < a_{\min}$ 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 1-steps under the condition that an $\vec{x}$ with $a(\vec{x}) = a_{\min}$ has been chosen for mutation is

$\frac{n}{a_{min}}$. Thus, the expected time to reduce the value $a_{\min}$ from $n^{1/4}$ to 3 is of 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 search point $\vec{x} \in \{0,1\}^n$ with $\textsc{Plateau}_1(\vec{x}) > n$ is determined by Global SEMO is $\Omega(n^2 \log n)$, which completes the proof. □

Using the asymmetric mutation operator, the function $\textsc{Plateau}_1$ becomes much harder. Jansen and Sudholt (2005) have shown that the probability that (1+1)EA$_{asy}$ optimizes $\textsc{Plateau}_1$ in $2^{O(n^{1/4})}$ steps is bounded above by $2^{-\Omega(n^{1/4})}$. In contrast to this, the search gets easier if the objective $|\vec{x}|_1$ is added, i.e., if $\textsc{Plom}$ is optimized by Global SEMO, as it is shown in the next theorem.

Theorem 4. *The expected optimization time of Global SEMO$_{asy}$ on* $\textsc{Plom}$ *is* $\Theta(n^2)$

Proof. First assume that the population contains an element $\vec{x} \in \{1^i 0^{n-i}, 1 \leq i \leq n\}$. For such an element $\vec{x}$, Global SEMO$_{asy}$ behaves on $\textsc{Plom}$ like the (1+1)EA$_{asy}$ on $|x|_1$. According to (Jansen and Sudholt, 2005), (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 $\vec{x} \notin \{1^i 0^{n-i}, 1 \leq i \leq n\}$. 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 $\vec{x}$ 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 $\vec{x}$ is

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

Since

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

we can bound this probability by $p(\vec{x}) \geq \frac{1-e^{-1/2}}{2}$ where we denote Euler's number by $e \approx 2.718\ldots$ throughout this work. As two elements $\vec{x}, \vec{y} \in (\{0,1\}^n \setminus \{1^i 0^{n-i}, 1 \leq i \leq n\})$ with $|\vec{x}|_0 \neq |\vec{y}|_0$ 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 $\Omega(1/n)$.

The proof of the lower bound largely follows the proof of Theorem 3. Again, we show that the population size is linear after $a_{\min}$ first leaves the interval $[\frac{n}{4}, \frac{n}{3}]$ 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 ones (respectively zeros) has to be flipped. The probability for this flip is at most $\frac{n}{3}\frac{1}{2(n/4)} = \frac{2}{3}$, 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 with high probability of size $|P| = \Theta(n)$. Also with high probability, we have $a_{\min} \geq \frac{n}{5}$. Now consider the probability p_{ij} to produce in the next mutation step from $\vec{x}$ with $|\vec{x}|_0 = i$ a solution $\vec{x}'$ with $|\vec{x}'|_0 = j$ where $j < i$. Jansen and Sudholt (2005) have shown that $p_{ij} \leq 2^{j-i}$. Let $\vec{x} \in P$ be the solution with the smallest number of zeros. Denote by $D = |\vec{x}|_0$ the Hamming distance of $\vec{x}$ to 1^n. Consider a solution $\vec{y} \in P$ with $|\vec{y}|_0 = |\vec{x}|_0 + k$, where $k \in [0, |\vec{x}|_1]$. A mutation step from $\vec{y}$ to $\vec{x}$ 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 zeros 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 of the runtime (in n) in comparison to the analyses on PLOM comes from the solutions $\vec{x}$ from the former plateau SP_1. These search points are comparable in PLZM as in PLOM, but this time, the second objective $|\vec{x}|_0$ 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", i.e., that the choice of an additional objective can influence the runtime of an evolutionary algorithm drastically.

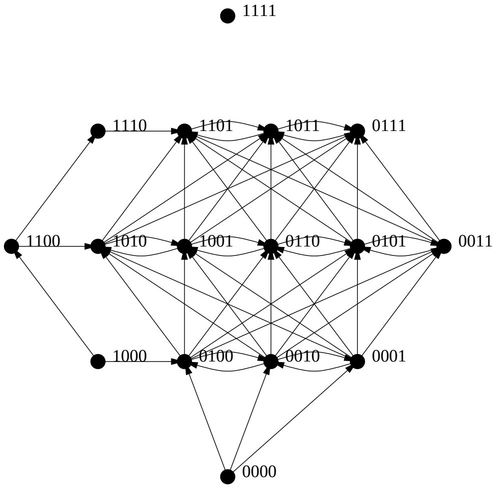


Figure 7: Relation graph for the biobjective problem PLZM : $\{0,1\}^4 \rightarrow \mathbb{N}^2$. Reflexive and transitive edges are omitted for clarity.

Theorem 5. *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} make up the entire Pareto front of PLZM, cf. the relation graph of PLZM in Fig. 7. We show that the claimed lower bound holds for obtaining the search point 1^n which is the optimal search point for PLATEAU$_1$. Considering the running time until the remaining two Pareto-optimal solutions are found is therefore not necessary to prove the theorem.

The initial search point consists with probability $1 - e^{-\Omega(n)}$ of at most $2n/3$ ones using Chernoff bounds. Accepted steps increasing the number of ones have to produce a solution of $SP_1 \cup \{1^n\}$. The probability to reach 1^n directly from a search point $\vec{x} \notin SP_1$ 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 ones can only be increased by producing the search point 1^n directly. As each individual in the population has at most $3n/4$ ones with probability $1 - e^{-\Omega(n)}$, the probability of obtaining 1^n is upper bounded by $2^{-n/4}$ for both algorithms. Hence, the overall time to achieve the search point 1^n is $e^{\Omega(n)}$ with probability $1 - e^{-\Omega(n)}$. □

In summary, we have seen that one and the same problem can become harder or easier to solve for an evolutionary algorithm if a second objective is added. Rigorous running time analyses showed that the differences of the expected running times can be exponential in the number of bits in the representation. Section 2.2.4 will investigate the effect of additional objectives also experimentally showing that the effects are not only asymptotic results but can be seen in practice. Before, the next section will present a further running time analysis which shows what can happen if two equally difficult single objectives are combined to a biobjective problem.

2.2.3 Coping With Two Plateaus

In Sec. 2.2.2, the added objectives were easy to solve individually for the (1+1)EA. The main reason for the smaller running time of PLOM as compared to PLATEAU$_1$ is that both functions have the same global optimum. One can also argue that not the PLATEAU$_1$ function is becoming easier to solve by adding a second objective but that adding the PLATEAU$_1$ function made the easy functions $|\vec{x}|_0$ and $|\vec{x}|_1$ harder to solve. Therefore, the question arises whether combining two objectives to a biobjective problem may result in a faster optimization process than optimizing the different objective functions separately. In the following, we show that the combination of two equally complex problems yields an easier problem if both functions are optimized as a biobjective problem.

We know from Theorem 2 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

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

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

$$\textsc{Plateaus}(\vec{x}) = (\textsc{Plateau}_1(\vec{x}), \textsc{Plateau}_2(\vec{x}))$$

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. Figure 8 shows the relation graphs of PLATEAU$_1$, PLATEAU$_2$, and PLATEAUS exemplary for $n = 4$ decision variables.

Theorem 6. *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 constitute the Pareto front of PLATEAUS as the solutions 1^n and 0^n 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

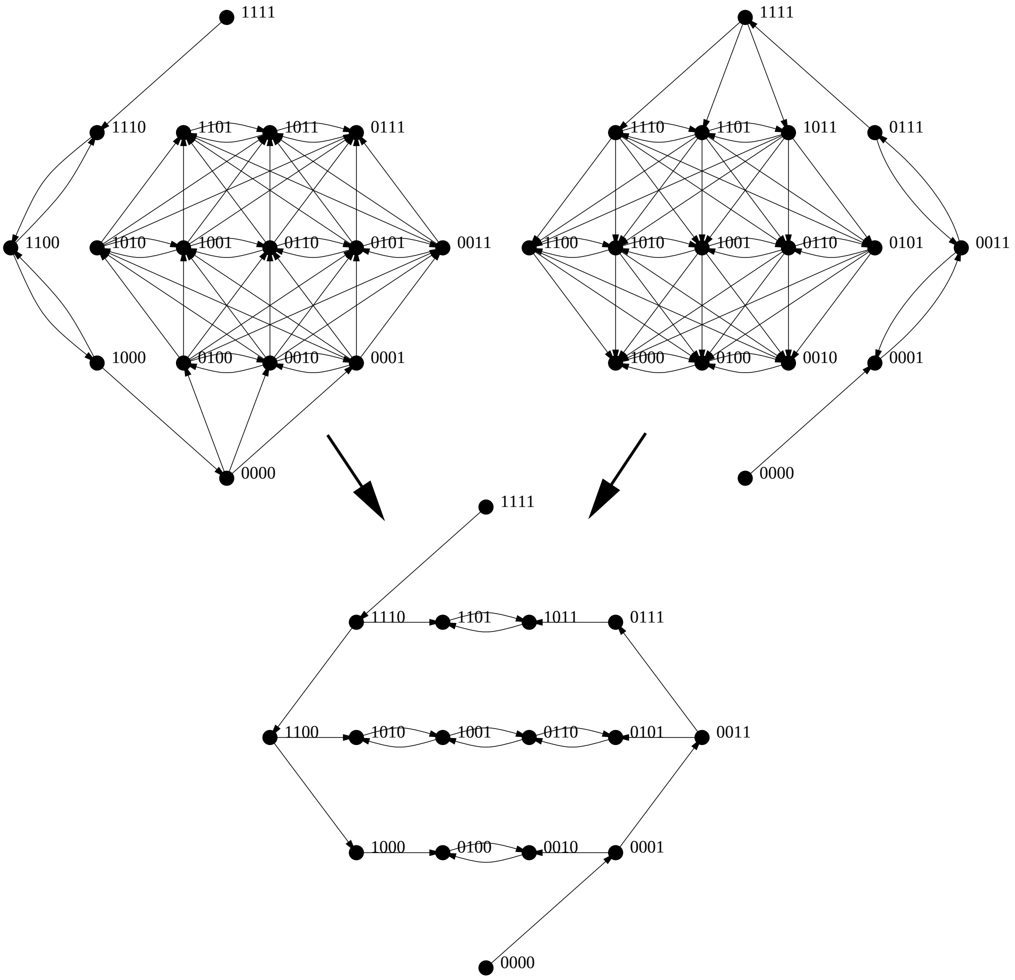


Figure 8: Relation graphs for the functions Plateau$_1$ (upper left) and Plateau$_2$ (upper right) as well as the combination of them (bottom, problem (Plateau$_1$,Plateau$_2$)). Reflexive and transitive edges are omitted for clarity.

the considered problem which shows that the search points 1^n and 0^n are the only Pareto-optimal ones.

The population size of Global SEMO on Plateaus is always bounded by $O(n)$ as each objective function attains at most $n+3$ different values. In a first step, we consider the number of generations 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 $\vec{x}$ with the largest Plateau$_1$-value in the population. This is determined by the number of zeros in $\vec{x}$. Assume that $|\vec{x}|_0 = k$ holds. Then, the probability to produce from $\vec{x}$ a solution $\vec{x}'$ with a higher number of zeros is at least $(n-k)/(en)$. The probability of choosing $\vec{x}$ 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 $\vec{x}$ be the search point with objective vector $(n+1, k)$ in the population. Flipping the bit x_{k+1} in $\vec{x}$ leads to a solution $\vec{x}'$ with objective vector $(n+1, k+1)$ that dominates and therefore replaces $\vec{x}$. 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 analogue to the lower bound proof of Theorem 3, since the functions PLATEAUS and PLOM are the same on the set $\{0,1\}^n \setminus (\{0^i 1^{n-i}, 0 \le i \le n\} \cup \{1^i 0^{n-i}, 0 \le i \le n\})$. □

As we already mentioned before, Jansen and Sudholt (2005) 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 7. *The expected optimization time of Global SEMO$_{asy}$ on the problem* PLATEAUS *is* $\Theta(n^2)$.

Proof. As in the proof of Theorem 6, we first bound the expected number of generations until the population includes search points with objective vectors $(n+1, \cdot)$ and $(\cdot, n+1)$ and assume that the Pareto-optimal solutions with objective vectors $(n+2, n)$ and $(n, n+2)$ respectively have not been obtained before. For obtaining $(n+1, \cdot)$, consider the search point $\vec{x}$ with the largest PLATEAU$_1$ value. Assume that it has $|\vec{x}|_0 = k$ zeros. The probability to obtain from $\vec{x}$ a solution with more zeros, that therefore is incomparable to $\vec{x}$ or even dominates $\vec{x}$, can be bounded by $(1-e^{-1/2})/2$, as shown in the proof of Theorem 4. 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 $O(n^2)$ generations. 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 $(n+1,k)$, the corresponding bitstring consists of k ones followed by $n-k$ zeros. This search point can be improved by flipping the $(k+1)$th zero to one. The probability for this to happen is

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

which can be bounded by

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

Hence, after an expected number of $O(n^2)$ generations the population will contain the search point with objective vector $(n+2,n)$. 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 4, since the functions PLATEAUS and PLOM are the same on the set $\{0,1\}^n \setminus (\{0^i1^{n-i}, 0 \leq i \leq n\} \cup \{1^i0^{n-i}, 0 \leq i \leq n\})$. □

As we have seen, the combination of two equally difficult single-objective problems can be solved faster if they are combined to a new biobjective problem. This result theoretically supports the usage of additional objectives in practice as it was proposed by Knowles et al. (2001) and Jensen (2004). However, the formulation of the new objectives is not obvious and might need lots of problem knowledge to avoid an increase in the running time. Furthermore, it is not obvious how the theoretical results can be generalized to practically relevant algorithms such as the NSGA-II or SPEA2 since they use an additional secondary selection criterion that has not been investigated theoretically yet. Further research in this direction is therefore required to fully answer the question what makes many-objective problems difficult for those algorithms.

2.2.4 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: (i) can the asymptotical results also be observed for small instances, (ii) 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 $\textsc{Plateau}_1$, and (iii) 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 biobjective problem with the same plateau SP_1 that was considered above whereas Sec. 2.2.4.2 shows that an addition of objectives can increase or decrease the running time of Global SEMO also for other kinds of plateaus. The 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 an objective that introduces a direction on the corresponding search space region, the new direction increases or decreases the running time depending on whether the introduced direction is deceptive or not.

2.2.4.1 Similar Plateaus With More Objectives

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

$$\textsc{LeadingOnes}(\vec{x}) = \sum_{i=1}^{n} \prod_{j=1}^{i} x_j$$

and

$$\textsc{TrailingZeros}(\vec{x}) = \sum_{i=1}^{n} \prod_{j=i}^{n} (1 - x_j)$$

which were first investigated in (Laumanns et al., 2004a) as the problem $\textsc{Lotz}(\vec{x}) = (\textsc{LeadingOnes}(\vec{x}), \textsc{TrailingZeros}(\vec{x}))$. Here, we consider the slightly changed functions

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

and

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

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 $f_1(\vec{x}) = f_2(\vec{x})$ holds, if $\vec{x} \in SP_1$. In this case, we have to cope with the same plateau as given by the function Plateau$_1$. The search point 1^n is also here the only (Pareto-)optimal search point with the objective vector $(n+2, n+2)$. Again, the focus lies on the plateau and not on the fact that there is only a single Pareto-optimal solution as argued above. The only difference between the biobjective 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 Sec. 2.2.2. Right before finding the Pareto-optimal point 1^n, 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.

Figure 9 shows the box plots of the running times of 31 independent Global SEMO runs on all three problems for different bitstring lengths $(n \in \{5, 10, 15, 20, 25, 30\})$[8]. The non-parametric Kruskal-Wallis test with the extension to multiple comparisons[9] has been performed for every bitstring length 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. 9. Note that the runs were aborted if no Pareto-optimal search point has been found in the first 100,000 generations to keep the time for the experiments at a manageable level. This and the large variance of the single runs explain the unexpected decrease of the me-

[8] 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 "+".

[9] As implemented in the PISA performance assessment toolbox (Bleuler et al., 2003) and described in (Conover, 1999) on pages 288ff.

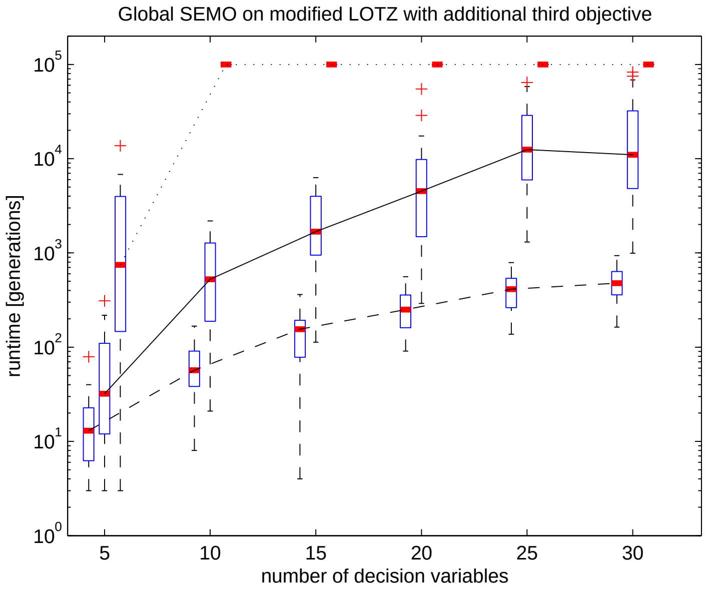


Figure 9: Comparison of the running times for Global SEMO if a third objective is added to the modified LOTZ problem: original problem (f_1, f_2) (solid line), $(f_1, f_2, |x|_1)$ (dashed), and $(f_1, f_2, |x|_0)$ (dotted). Note, that the runs are aborted if no Pareto-optimal point has been found in the first 100,000 generations. For clarity, the three boxplots corresponding to a specific number of decision variables have been slightly shifted horizontally.

dian between the original problem with 25 and the one with 30 decision variables.

2.2.4.2 Different Kinds of Plateaus

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

Plateaus of Indifferent Solutions

The basis biobjective problem we use for the investigation of plateaus of indifferent solutions is the original LOTZ of (Laumanns et al., 2004a). All solutions with the same number of leading ones and trailing zeros 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 ones nor to the trailing zeros of a solution $\vec{x}$ as the middle block $\vec{x}_M$. In addition, $|\vec{x}|$ denotes the length of the bitstring $\vec{x}$. Adding objectives that take into account only the bits in the middle blocks base function g (Laumanns et al., 2004a):

$$\max \quad g_1(\vec{x}) = \textsc{LeadingOnes}(\vec{x})$$
$$\max \quad g_2(\vec{x}) = \textsc{TrailingZeros}(\vec{x})$$

slower with

$$\min \quad g_{(i)}(\vec{x}) = |\vec{x}_M| - \textsc{LeadingZeros}(\vec{x}_M) - \textsc{TrailingOnes}(\vec{x}_M)$$

faster with

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

Figure 10: Definitions of the functions illustrating the changes of running time with respect to making indifferent solutions comparable (problems $(g_1, g_2, g_{(\cdot)})$).

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 3-objective problem will be higher or lower than for the original biobjective problem.

Minimizing the objective $g_{(i)}$ in addition to Lotz $= (g_1, g_2)$ as defined in Fig. 10 increases the running time of Global SEMO due to its deceptive behavior: on the one hand, the number of leading ones and trailing zeros has to be maximized to reach the Pareto front; on the other hand, the additional objective $g_{(i)}$ rewards a higher number of leading zeros in the middle block as well as a higher number of trailing ones. 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 $g_{(ii)} = |\vec{x}_M|_1$ (see Fig. 10) 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. Figure 11 shows the corresponding box plots. The non-parametric

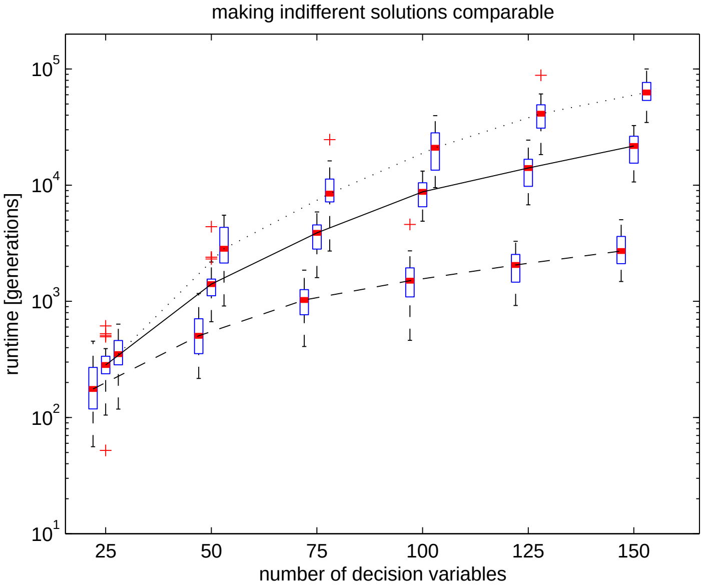


Figure 11: Boxplots of the running times until the first Pareto-optimal point is reached for the problems (g_1, g_2) (solid line), $(g_1, g_2, g_{(i)})$ (dotted), and $(g_1, g_2, g_{(ii)})$ (dashed). Note, that the runs are aborted if no Pareto-optimal point has been found in the first 100,000 generations. For clarity, the boxplots corresponding to each number of decision variables have been slightly shifted horizontally.

Kruskal-Wallis test for multiple comparisons described in (Conover, 1999) 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.

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 a multiobjective evolutionary algorithm in both ways. To this end, the problem (h_1, h_2) as defined in Fig. 12 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. Figure 13 illustrates the objective space of 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 base function h:

$$\max \quad h_1(\vec{x}) = \begin{cases} \textsc{LeadingOnes}(\vec{x}) & \text{iff } 0 \equiv (n - |\vec{x}_M|) \bmod 4 \\ & \text{or } 1 \equiv (n - |\vec{x}_M|) \bmod 4 \\ -\textsc{LeadingOnes}(\vec{x}) - n \cdot \left\lfloor \frac{\lfloor (n-|\vec{x}_M|)/2 \rfloor}{2} \right\rfloor & \text{else} \end{cases}$$

$$\max \quad h_2(\vec{x}) = \begin{cases} \textsc{TrailingZeros}(\vec{x}) & \text{iff } 0 \equiv (n - |\vec{x}_M|) \bmod 4 \\ & \text{or } 1 \equiv (n - |\vec{x}_M|) \bmod 4 \\ -\textsc{TrailingZeros}(\vec{x}) - n \cdot \left\lfloor \frac{\lfloor (n-|\vec{x}_M|)/2 \rfloor}{2} \right\rfloor & \text{else} \end{cases}$$

slower with

$$\min \quad h_{(i)}(\vec{x}) = \tfrac{n}{2} - |\tfrac{n}{2} - |\vec{x}_M||$$

faster with

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

Figure 12: Definitions of the functions illustrating the changes of running time with respect to making comparable solutions incomparable (problems $(h_1, h_2, h_{(\cdot)})$).

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 front are made incomparable, Global SEMO needs more time to find the Pareto front than for the original problem (h_1, h_2). The objectives $h_{(i)}$ and $h_{(ii)}$ defined in Fig. 12 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 biobjective problem.

Figure 14 shows the boxplots of 31 independent Global SEMO runs for different numbers of decision variables ($n \in \{25, 50, 75, 100, 125, 150\}$) on all three problems. 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. 14 indicate that Global SEMO has a higher average running time on $(h_1, h_2, h_{(i)})$ and a lower average running time on $(h_1, h_2, h_{(ii)})$ as the original problem (h_1, h_2) 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.

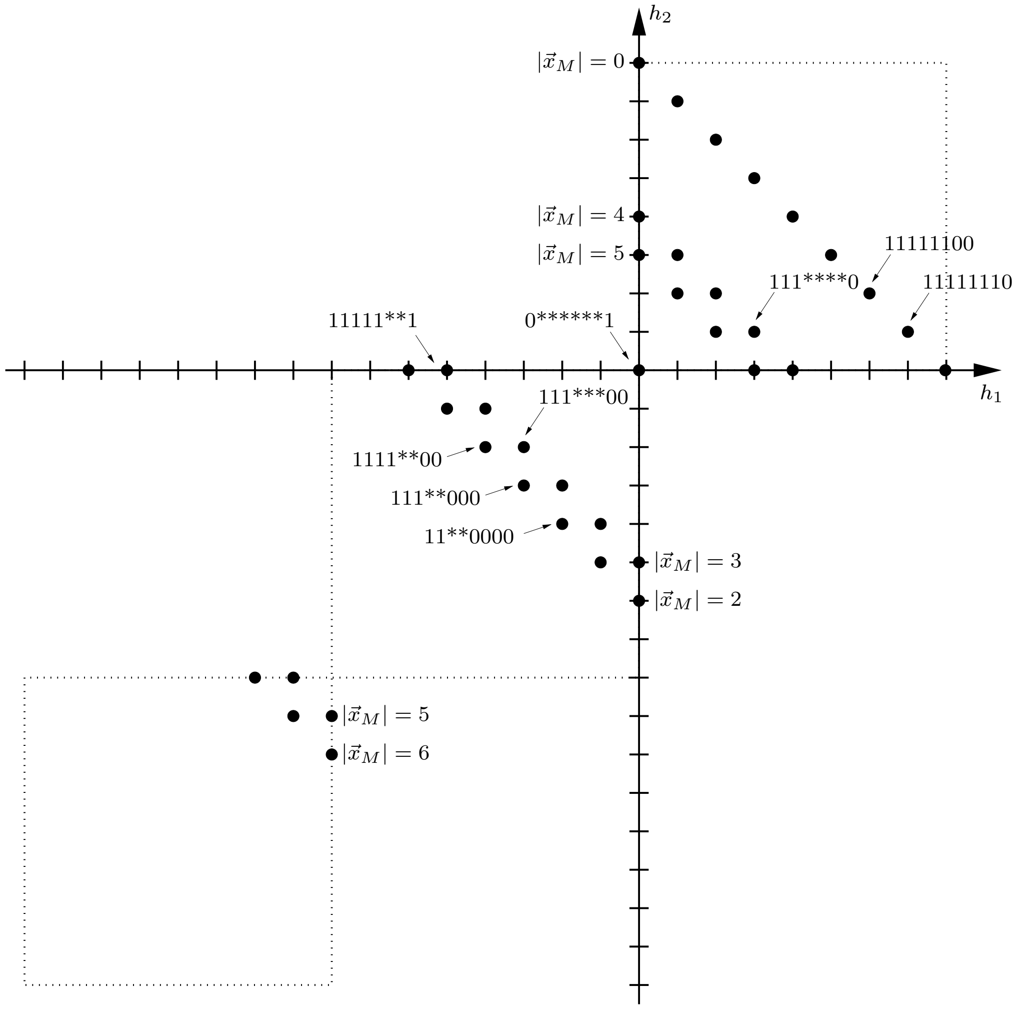


Figure 13: Illustration of the objective space for the modified LOTZ problem (h_1, h_2) and $n = 8$ decision variables. For some objective vectors, the corresponding solutions in decision space are indicated where a "*" denotes either a "1" or a "0" on the corresponding bit string position.

2.3 Summary

In this chapter, we have investigated the question of how additional objectives or the omission of them affect (i) the structure, i.e., the dominance relation, of a given optimization problem and (ii) the search behavior of evolutionary algorithms applied to this problem.

It turned out that the omission of objectives can only introduce comparabilities whereas addition of objectives, on the contrary, can change the dominance structure only in the other direction: dominance relations between solution pairs can only be removed from the dominance relation—making comparable solution pairs incomparable or removing an indifference relationship. The addition of objectives can, therefore, only introduce a direction to plateaus of indifferent solutions or remove a direction in a cer-

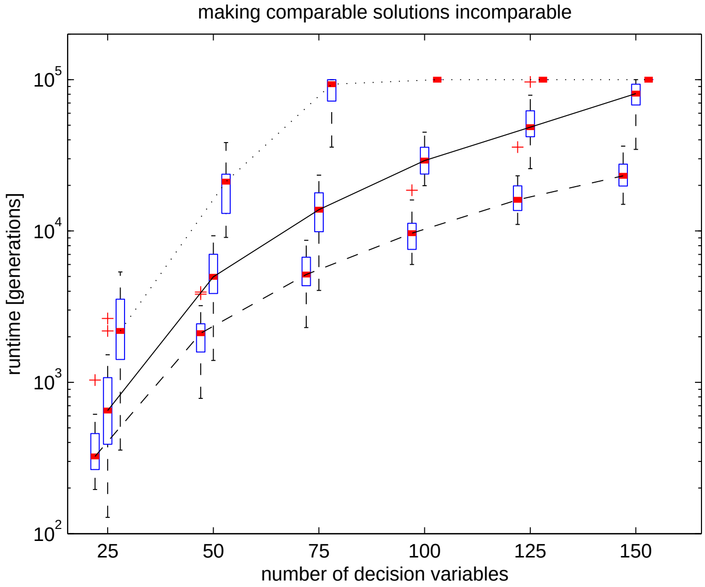


Figure 14: Boxplots of the running times until the first Pareto-optimal point is reached for the problems (h_1, h_2) (solid line), $(h_1, h_2, h_{(i)})$ (dotted), and $(h_1, h_2, h_{(ii)})$ (dashed). Note, that the runs are aborted if no Pareto-optimal point has been found in the first 100,000 generations. For clarity, the three boxplots corresponding to a specific number of decision variables have been slightly shifted horizontally.

tain search space region by making the solutions incomparable. Rigorous running time analyses have shown that these changes can vary the running time of simple evolutionary algorithms drastically in both ways: 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. We have also shown that the simultaneous optimization of two equally difficult problems in terms of multiobjective optimization can be faster than solving the single problems individually resulting in a first theoretical proof that multiobjectivization can help to reduce the running time of evolutionary algorithms when further incomparabilities between solutions are added by an additional (helper-)objective.

It is obvious that the considered problems are artificial, however, plateaus of indifferent and incomparable solutions do exist in practically relevant combinatorial problems. Also the considered algorithms are simplified versions of practically relevant ones. Nevertheless, global SEMO uses algorithmic concepts such as the mutation operator or the selection scheme that are also used in other algorithms in practice. Thus, the presented results have different implications—also for algorithms and problems that differ from the investigated ones. 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.

In addition to the work presented in this chapter, several theoretical studies investigating the effects of additional objectives have been published recently. Malinowska (2002) and later on Fliege (2007) and Mäkelä and Nikulin (2008) investigate how additional objectives change the set of Pareto-optimal solutions. These studies, however, do not investigate the effect on optimization algorithms; only Gal and Hanne (1999) showed that even the addition or omission of essential objectives that do not change the Pareto set can change the outcome of some multiobjective optimization techniques. As already discussed above, Handl et al. (2008) investigate the effects of multiobjectivization, i.e., the decomposition of a single-objective optimization problem into a multiobjective one. All mentioned studies enhance the understanding of the difference between single-objective and multiobjective problems—one important aspect in the fields of evolutionary multiobjective optimization and multicriteria decision making that might even become more interesting in the future in combination with the newly proposed set-based view of multiobjective optimization (Zitzler et al., 2009; Bader et al., 2009). Whether the observed effects of additional objectives can also be seen in practical applications, for example if the number of objectives is larger than two or if the search space is continuous, remains future work.

3

Objective Reduction

In the last decade, there has been a growing interest in applying evolutionary algorithms to multiobjective optimization problems, mainly to approximate the set of Pareto-optimal solutions. However, most of the publications in this area deal with problems where only a few, i.e., between two and four, objectives are involved, while studies with many objectives are rare, cf. (Coello Coello et al., 2007). The reason is that a large number of optimization criteria leads to further difficulties with respect to decision making, visualization, and computation; for instance, it has been shown empirically that state-of-the-art MOEAs such as NSGA-II and SPEA2 do not scale well with an increasing number of objectives (Khare et al., 2003; Purshouse and Fleming, 2003b; Wagner et al., 2007), i.e., the algorithms have more and more difficulties to find solution sets that are close to the Pareto set if the number of objectives increases. 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. The 2007 conference on evolutionary multi-criterion optimization (Obayashi et al., 2007) revealed that there is a need to handle such many-objective scenarios, and the challenge is to develop concepts and methods to tackle the aforementioned difficulties.

An interesting research question that arises in this context is whether actually all optimization criteria are necessary and some of the objectives may be omitted without—or with only slightly—changing the problem characteristics. The motivation behind this question lies in the observation that additional objectives cause problems mainly when they are competing with existing ones; a set of non-conflicting criteria can be represented by a single objective. Methods for automated objective reduction can be beneficial for both decision making and search. On the one hand, the decision maker would have to consider fewer objective values per solution, it would be easier to visualize the solutions, and the number of nondominated solutions is likely to decrease as shown by Winkler (1985), resulting in a further reduction of the information which has be taken into account. On the other hand, search algorithms may work more efficiently and consume less computational resources, if the number of objectives is decreased adaptively. Similar issues emerge with computationally expensive objective functions, e.g., when extensive simulations need to be carried out in order to determine the objective function values. These statements only conflict with the results in the previous chapter at first sight; we argue here that additional objectives can be helpful in some cases (as pointed out in the previous chapter) but on the other hand, they should be automatically omitted if their omission does not affect the induced weak Pareto dominance relation but increases the running time of evolutionary algorithms due to their high dependency on the number of objectives, e.g, for hypervolume-based algorithms, cf. Chapter 4.

The issue of objective reduction has gained only little attention in the literature so far, and existing methods are either restricted to particular function classes or do not take the underlying dominance structure into account. In this chapter and based on the investigations in the previous one, we propose a methodology for objective reduction that allows both to consider black-box optimization criteria and to maintain and control the dominance structure. The key contributions are:

- Formal notions of objective conflicts, a degree of conflict, redundant objectives, and minimum objective sets;

- A definition of different types of objective reduction problems including the aggregation of objectives and the design of corresponding exact and greedy algorithms; running time analyses thereof are also presented;

- A systematic study of the efficacy of the proposed approach on various benchmark problems and a real-world application;

In the following, we will first review related work (Sec. 3.1) before presenting the theoretical foundations (Sec. 3.2 and Sec. 3.3) and the corresponding objective reduction algorithms (Sec. 3.4, Sec. 3.5, and Sec. 3.6). The application of the objective reduction methods in decision making is demonstrated in Sec. 3.7.

3.1 Related Work

Although the need for approaches to handle many-objective problems is apparent in terms of real-world applications, not many studies are known that deal with the reduction of the objective set to reduce the complexity of a problem.

To the best of my knowledge, the first publication in the field of MOEAs that pointed out the possibility of omitting objectives is the one by Purshouse and Fleming (2003a). The paper discusses in detail various relationships between single objectives such as conflict, harmony, and independence together with their effect on evolutionary multiobjective optimization. In a short paragraph, the authors mention that (traditional) dimensionality reduction techniques could be used to simplify both decision making and search, but they do not propose a concrete approach for a reduction of the objective set.

Dimensionality reduction is a well-known problem in many areas like statistics and data mining, and various methods to extract and select *features*[1] are known. One can distinguish between two distinct approaches: *feature extraction* and *feature selection*. The task in feature extraction is to determine a (small) set of *arbitrary* features, while the task in feature selection is to find the smallest subset of the *given* features, representing the given data best. Translated to the multiobjective optimization field, one can ask either for a set of arbitrary objectives or for a subset of given objectives which describes the original problem best. Several approaches to solve feature extraction and feature selection problems are known from the literature. Some of the most popular feature extraction methods are Principal Component Analyses (PCA) (Jolliffe, 2002), Independent Component Analysis (ICA) (Hyvärinen et al., 2001), or Self-Organizing Maps (Kohonen, 2001). Clustering and biclustering techniques (Jain et al., 1999) are examples for feature selection methods (Langley, 1994; Liu and Motoda, 2008). Also evolutionary algorithms have been used for feature extraction and selection tasks, see for example (Vafaie and de Jong, 1993). We refer to survey articles on dimensionality reduction such as (Fodor, 2002) for a more detailed overview.

Although there are many dimensionality reduction approaches available, the common dimensionality reduction techniques cannot be used directly as an objective reduction technique in evolutionary multiobjective optimization as Purshouse and Fleming (2003a) already pointed out. The reason is that the Pareto dominance relation is not taken into account—in other words: it cannot be ensured that the Pareto dominance relation is maintained while the number of objectives is reduced.

[1] Usually, the variables under consideration are called features.

An objective reduction approach which preserves the dominance structure has been proposed by Gal and Leberling (1977) for the case when the objective functions are explicitly given as linear combinations of the (real) decision variables, i.e., the Pareto set is determined within the problem formulation. Hence, this approach, as well as a generalization to a broader definition of redundancy among objectives by Agrell (1997), is restricted to the narrow class of linear problems and inapplicable to general black-box scenarios. Also newer results on objective reduction in the MCDM field (Malinowska, 2002, 2006; Malinowska and Torres, 2008) are based on the assumption of explicit objective functions and therefore not applicable in a black-box scenario. Furthermore, it was pointed out by Gal and Hanne (1999, 2006) that these objective reduction approaches cause troubles with search algorithms as they are based on a conflict definition that considers the effect of an objective omission on the Pareto set only. Changing a given problem to a problem with less objectives but the same Pareto set might be of high interest if the entire Pareto set can be found by optimizing the smaller objective set only. However, in practice this is not always given as the omission of redundant objectives might introduce further difficulties with respect to the adjustment of parameters of the search algorithms to the new problem.

The question of finding new objectives that reformulate the original problem with fewer objectives (feature extraction) has been already addressed in the context of coevolution (de Jong and Bucci, 2006, 2008). This approach, however, is restricted to certain types of objective functions that arise in the context of coevolution, namely so called tests, i.e., objective functions that are either 1 or 0.

The only studies that tackled objective reduction in an evolutionary multiobjective optimization setting have been proposed by Deb and Saxena. In (Deb and Saxena, 2005, 2006), the authors propose a method that is based on principal component analysis to decrease the number of objectives during search. The method aims at computing a set of “the most important conflicting objectives” by omitting redundant ones, i.e., those that are less influential with respect to the principal components. It was incorporated into the algorithm NSGA-II and used to shrink the objective set iteratively in the course of multiple optimization runs. Furthermore, it was tested on and primarily invented for problems where the Pareto front has a lower dimension than the problem formulation itself. Since the approach of Deb and Saxena (2005, 2006) considers the correlation between objectives as an indicator for the conflict between them, it cannot guarantee that the (weak) Pareto dominance relation, and therefore the Pareto set, is preserved. In addition, no quantitative measure can be specified by how much the dominance relation changes when objectives are omitted. The same holds for a recently published extension that is based on two non-linear dimensionality reduction techniques (Saxena and Deb, 2007) and that was applied to problems where constraints are considered as objectives (Saxena and Deb, 2008).

In the following, we propose an approach for objective reduction that both is suited to black-box optimization problems and allows to maintain and control the dominance structure. To this end, we first investigate objective conflicts in general and give a new definition of conflicting objective *sets* in particular. Furthermore, we introduce the terms of redundant and minimum objective sets that build the basis of the objective reduction method to follow.

3.2 Objective Conflicts

3.2.1 Conflicting, Redundant, and Minimum Objective Sets

In Chapter 2, we have seen what happens if objectives are added to a problem formulation. Based on the general investigations on the Pareto dominance relation and especially on Theorem 1, it is easy to see under which circumstances objectives can be omitted without changing the problem structure: whenever the underlying relation graph remains the same. We will use the notion of conflicting and non-conflicting objective sets to capture this observation.

Definition 1. *Two objective sets* $\mathcal{F}_1, \mathcal{F}_2$ *are called* conflicting *if the induced weak Pareto dominance relations differ, i.e.,* $\preceq_{\mathcal{F}_1} \neq \preceq_{\mathcal{F}_2}$ *and* non-conflicting *otherwise* ($\preceq_{\mathcal{F}_1} = \preceq_{\mathcal{F}_2}$)*.*

This definition is somehow a general form of the ideas of non-essential objectives in the field of multicriteria decision making: A *non-essential* or *redundant* objective of a multiobjective optimization problem is, according to Gal and Leberling (1977), an objective the omission of which does not change the Pareto set. Although algorithms for identifying the set of non-essential objectives are known (Gal and Leberling, 1977; Agrell, 1997; Malinowska, 2006; Malinowska and Torres, 2008), the approaches are only applicable if the objective functions are linear[2], i.e., of the type $f_i(\vec{x}) = \sum_{j=1}^{n} \alpha_j x_j$ for $\alpha_j \in \mathbb{R}$. Furthermore, newer results identified that the limitation of the definition of non-essential objectives to the Pareto set instead of considering all solutions might introduce difficulties for some MCDM techniques (Gal and Hanne, 1999, 2006): for example, when omitting non-essential objectives, the solutions found by scalarization techniques like the weighted

[2] Although Agrell (1997) considers also more general problems, his algorithm for finding the non-essential objectives is only applicable in the linear case.

sum differ from solutions found if all objectives are optimized. Moreover, heuristic methods such as MOEAs do not guarantee to find Pareto-optimal solutions and objective reduction methods that are solely based on considering the Pareto set neglect the impact of the Pareto dominance relation during the search.

The definition of non-conflicting objectives as in Def. 1 avoids these problems by investigating the entire dominance relation: whenever an objective subset $\mathcal{F}' \subseteq \mathcal{F}$ is non-conflicting with the entire objective set $\mathcal{F}$, an omission of the objectives in $\mathcal{F} \setminus \mathcal{F}'$ will preserve the weak dominance relation, otherwise, the weak dominance relation will change. Similar to the studies about non-essential objectives in the MCDM field, a few definitions of conflict exist in the MOEA literature as well: Deb (2001) and Tan et al. (2005) define conflict also only depending on the Pareto front or the presence or absence of comparable solutions, while Purshouse and Fleming (2003a) define conflict as a property of objective pairs. As the following example shows, the three mentioned definitions cannot indicate whether objectives can be omitted without affecting the dominance structure.

Example 3. *Assume that the four solutions $\vec{a}$, $\vec{b}$, $\vec{c}$, and $\vec{d}$ in Fig. 15 represent either the entire search space or the Pareto set*[3]*. Then, the original objective set $\{f_1, f_2, f_3, f_4\}$ is conflicting according to Deb (2001) as there is no single optimal solution but four Pareto-optimal ones. For the same reason of incomparable solution pairs, the objective set is also conflicting according to Tan et al. (2005). In addition, every objective pair "exhibits evidence of conflict" as defined by Purshouse and Fleming (2003a). The three conflict definitions mentioned may lead to the conclusion that all objectives are necessary. However, objective f_3 can be omitted and all solutions remain incomparable to each other with regard to the objective set $\{f_1, f_2, f_4\}$, i.e., the weak Pareto dominance relation on the search space stays unaffected, cf. Fig. 4(l) and (o) in Chapter 2. In contrast to the three above mentioned conflict definitions, Def. 1 classifies the objective sets $\{f_1, f_2, f_4\}$ and $\{f_1, f_2, f_3, f_4\}$ as non-conflicting.*

This example indicates that objective conflict appears to be rather a set-based property than a property of objective pairs which is the reason why we defined conflict between objective *sets* in Def. 1. Similarly, the question of whether objectives can be omitted while the dominance structure is preserved cannot be decided by considering relations between objective pairs only; the $\mathcal{NP}$-hardness proof of the δ-`MOSS` problem in Sec. 3.3 will support this statement. We will use the term redundancy to state whether objectives in an objective set can be omitted or not (necessary and sufficient criterion) without changing the entire dominance relation—in contrast to

[3] The objective values of the four solutions are exactly the same than in Fig. 3 in Chapter 2.

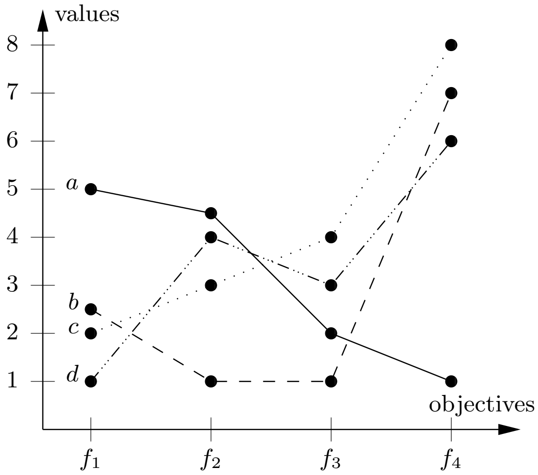


Figure 15: Parallel coordinates plot for the given example with four solutions and four objectives that have to be minimized; the objective function values are exactly the same than in Fig. 3.

the definition of conflicting objectives in the MCDM field as, e.g., in (Gal and Leberling, 1977) and subsequent publications, where only the influence of objectives on the Pareto set is considered.

Definition 2. *A set $\mathcal{F}' \subseteq \mathcal{F}$ of objectives is called* redundant *if and only if there exists an objective subset $\mathcal{F}'' \subset \mathcal{F}'$ that is non-conflicting with $\mathcal{F}'$.*

The additional question of which objective set is the smallest among those ones, that are non-conflicting with the entire objective set, can be denoted as finding a minimum objective set, and will be defined as follows.

Definition 3. *An objective set $\mathcal{F}' \subseteq \mathcal{F}$ is denoted as*

- minimal with respect to $\mathcal{F}$ *iff $\mathcal{F}'$ is both not redundant and non-conflicting with $\mathcal{F}$;*
- minimum with respect to $\mathcal{F}$ *iff $\mathcal{F}'$ is the smallest minimal objective set with respect to $\mathcal{F}$.*

In the following, we often just talk about minimal and minimum objective sets if it is clear from the context that we refer to minimal/minimum objective sets with respect to the set $\mathcal{F}$ of all objectives. A minimal objective set is a subset of the original objectives that cannot be further reduced without changing the associated preorder. A minimum objective set is the smallest possible set of original objectives that preserves the original order on the search space. By definition, every minimum objective set is minimal, but not all minimal sets are at the same time minimum.

Example 4. *In the example depicted in Fig. 15, the entire objective set is redundant since the objective set $\{f_1, f_2, f_4\}$ induces the same dominance* *relation as* $\{f_1, f_2, f_3, f_4\}$*, see Fig. 4(l) and (o) in Chapter 2. The set* $\{f_1, f_2, f_4\}$ *is at the same time minimal and minimum with respect to the entire objective set because no other objective subset with three or fewer objectives induces the same dominance relation as all objectives.*

Note that in general neither every minimal objective set is at the same time minimum nor a unique minimum objective set exists.

3.2.2 Measuring the Degree of Conflict

The requirement that the underlying relation graph must not change is often too strict in practice; the size of the minimum objective set may be close to the number of original objective functions. In order to achieve a more substantial reduction of the objective set, a continuous measure of conflict is helpful that allows to gradually tune the acceptable changes in the dominance relation. Before defining such a measure, we will illustrate the basic idea in the following example.

Example 5. *Let us again consider the above example which is again depicted in Fig. 16. We have seen that the omission of objective* f_3 *does not change the underlying dominance structure (Example 3). A further omission of an objective would change the dominance relation by making one (if* f_1 *is omitted), two (if* f_2 *is omitted), or even three (if* f_4 *is omitted) solution pairs comparable, cf. Fig. 4 in Chapter 2. When examining in detail what happens if, e.g.,* f_1 *is omitted together with* f_3*, we observe that as a result solution* $\vec{c}$ *is weakly dominated by solution* $\vec{b}$*. As* $\vec{b}$ *and* $\vec{c}$ *are incomparable with respect to the entire objective set, we make an error by omitting* f_1 *and* f_3 *and wrongly assuming that* $\vec{b}$ *weakly dominates* $\vec{c}$*. However, if the* f_1 *value of* $\vec{c}$ *was larger by an additional term of* $\delta = 0.5$ $\vec{b}$ *would weakly dominate* $\vec{c}$ *with respect to both the set* $\{f_2, f_4\}$ *and the entire objective set. Thus, we would make no error.*

The δ value of 0.5 in the above example can be used as a measure to quantify the difference in the dominance structure induced by $\{f_2, f_4\}$ and the entire objective set. By computing the δ values for all solution pairs, we can then determine the maximum error. Similar, we can define the average δ-error if we average the obtained δ values over all solution pairs. These ideas result in the definition of a general error measure for a certain set of solutions $A \subseteq X$ if the objective set $\mathcal{F}'$ is considered instead of the original objective set $\mathcal{F}$:

Definition 4. *Given two sets of objectives* $\mathcal{F}'$ *and* $\mathcal{F}$ *and a set of solutions* $A \subseteq X$*, we define the* maximum δ-error δ_{max} *as*

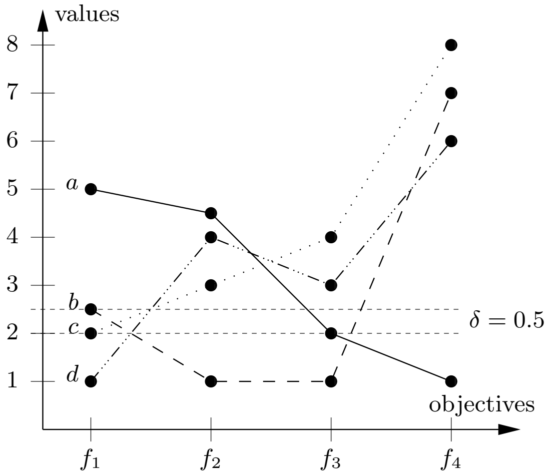


Figure 16: Illustration of δ-error if the objectives f_1 and f_3 are omitted. If instead of the entire set of objectives $\mathcal{F}$, the objective set $\mathcal{F}' = \{f_2, f_4\}$ is considered, the only solution pair where $\mathcal{F}'$ and $\mathcal{F}$ do not induce the same order is the pair $\vec{b}, \vec{c}$ (although $b \not\preceq_{\mathcal{F}} \vec{c}$ holds, $\vec{b} \preceq_{\mathcal{F}'} \vec{c}$). However, if the objective values of $\vec{b}$ would be smaller by a constant of $\delta = 0.5$, $\vec{b}$ would weakly dominate $\vec{c}$ already with respect to all objectives and the induced Pareto dominance relations $\preceq_{\mathcal{F}}$ and $\preceq_{\mathcal{F}'}$ would be the same for the solution pair $\vec{b}, \vec{c}$. In other words, no dominance would be wrongly assumed between $\vec{b}$ and $\vec{c}$ if we consider the ε-dominance relation with $\varepsilon = \delta = 0.5$.

$$\delta_{max}(A, \mathcal{F}', \mathcal{F}) = \max_{\substack{\vec{x},\vec{y} \in A \\ \vec{x} \not\preceq_{\mathcal{F}} \vec{y} \\ \vec{x} \preceq_{\mathcal{F}'} \vec{y}}} \left\{ \max_{f_i \in \mathcal{F}} \{ f_i(\vec{x}) - f_i(\vec{y}) \} \right\}$$

and the average δ-error δ_{avg} *is defined as*

$$\delta_{avg}(A, \mathcal{F}', \mathcal{F}) = \frac{1}{|A| \cdot (|A| - 1)} \sum_{\substack{\vec{x},\vec{y} \in A \\ \vec{x} \not\preceq_{\mathcal{F}} \vec{y} \\ \vec{x} \preceq_{\mathcal{F}'} \vec{y}}} \max_{f_i \in \mathcal{F}} \{ f_i(\vec{x}) - f_i(\vec{y}) \} \quad .$$

The meaning of the maximum error $\delta_{\max}$ is that whenever we wrongly assume that $\vec{x}$ weakly dominates $\vec{y}$ with respect to an objective subset $\mathcal{F}'$, we also know that $\vec{x}$ is not worse than $\vec{y}$ in all objectives by an additive term of δ.

Example 6. *Consider again the example from above, depicted in Fig. 16. When considering $\mathcal{F}' := \{f_2, f_4\}$, the induced Pareto dominance relation gives only for the solution pair $(\vec{b}, \vec{c})$ a wrong relation ($\vec{b} \preceq_{\mathcal{F}'} \vec{c}$ although $\vec{b} \not\preceq_{\mathcal{F}}$*

$\vec{c}$) which results in an error of $\delta = \max_{f_i \in \mathcal{F}}\{f_i(\vec{b}) - f_i(\vec{c})\} = 0.5$. Therefore, the maximum error for all solutions $A = \{\vec{a}, \vec{b}, \vec{c}, \vec{d}\}$ is $\delta_{max}(A, \{f_2, f_4\}, \mathcal{F}) = 0.5$; for $\mathcal{F}' := \{f_1, f_3\}$, the maximum error is $\delta_{max}(A, \{f_1, f_3\}, \mathcal{F}) = 6$ induced by the solutions $\vec{b}$ and $\vec{a}$ and their f_4 values, cf. the relation graphs of Fig. 4 in Chapter 2. However, for $\{f_2, f_4\}$ only one solution pair results in an error whereas for $\{f_1, f_3\}$, both the solution pair $(\vec{b}, \vec{a})$ (with $\delta_{max}(\{\vec{b}, \vec{a}\}, \{f_1, f_3\}, \mathcal{F}) = 6$) and the pair $(\vec{d}, \vec{c})$ (with $\delta_{max}(\{\vec{d}, \vec{c}\}, \{f_1, f_3\}, \mathcal{F}) = 1$) give an error. This results in average errors of $\delta_{avg}(A, \{f_2, f_4\}, \mathcal{F}) = 0.5/12 \approx 0.042$ and $\delta_{avg}(A, \{f_1, f_4\}, \mathcal{F}) = (6 + 1)/12 \approx 0.583$ respectively.

Note, that we refer to the maximum δ-error whenever we say δ-error without any further explanation. Furthermore, note that we always assume that all objective values have the same scale such that the small errors δ are comparable among the objectives. In addition, the above definition of δ-error assumes that an error made close to the Pareto front is of the same importance to a decision maker than the same error made far away from the Pareto front. Situations where a decision maker prefers extremal solutions with maximal objective function values are not considered here. The same holds for objective functions for which the possible objective function values are not equally distributed: the case that, e.g., solutions close to extremal values are more unlikely than ones with mid-range values, is not considered in this study. If one or several of these preferences are given by the decision maker, the definition of δ-error should be adjusted accordingly. Here, however, we assume the most general preference relation given by a decision maker to be the (weak) Pareto dominance relation and therefore define the δ-error as above—postponing a more general definition of δ-error for future work.

The given idea of determining the maximum error $\delta_{\max}$ can be used to define a continuous definition of objective conflict, namely *δ-conflicting objective sets*, which is based on the weak (additive) ε-dominance relation $\preceq^{\varepsilon}_{\mathcal{F}'}$ defined as[4]

$$\preceq^{\varepsilon}_{\mathcal{F}'} := \{(\vec{x}, \vec{y}) \mid \vec{x}, \vec{y} \in A \wedge \forall i \in \mathcal{F}' : f_i(\vec{x}) \leq f_i(\vec{y}) + \varepsilon\}$$

where $A \subseteq X$, $\mathcal{F}' \subseteq \mathcal{F}$, and $\varepsilon \in \mathbb{R}_{>0}$, cf. (Zitzler et al., 2003). Instead of using, e.g., the number of edges in which the corresponding relation graphs differ as a *degree* of conflict, we take the $\delta_{\max}$ error into account and define two objective sets as δ-non-conflicting if the corresponding ε-dominance relations with $\varepsilon = \delta$ are identical.

Definition 5. *Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two objective sets. We call $\mathcal{F}_1$* δ-non-conflicting *with $\mathcal{F}_2$ if and only if both $\left(\preceq_{\mathcal{F}_1} \subseteq \preceq^{\delta}_{\mathcal{F}_2}\right)$ and $\left(\preceq_{\mathcal{F}_2} \subseteq \preceq^{\delta}_{\mathcal{F}_1}\right)$ holds; otherwise $\mathcal{F}_1$ and $\mathcal{F}_2$ are denoted as* δ-conflicting.

[4] Note that also the multiplicative ε-dominance relation of (Zitzler et al., 2003) can be used; all the following results apply to the multiplicative ε-dominance relation as well.

Note that the relations $\preceq_{\mathcal{F}_1}$, $\preceq_{\mathcal{F}_2}$, $\preceq^{\delta}_{\mathcal{F}_1}$, and $\preceq^{\delta}_{\mathcal{F}_2}$ have not to be defined over the entire decision space X; the relation can also be restricted to a certain set of solutions $A \subseteq X$. The following theorem, the proof of which is given in Appendix A to improve readability, shows the connection between the definition of δ-conflict with respect to the ε-dominance relation and the maximum δ-error of Def. 4.

Theorem 8. *Let $\mathcal{F}_1, \mathcal{F}_2$ be two objective sets and $A \subseteq X$ a set of solutions. Then, $\mathcal{F}_1$ is $\overline{\delta}$-non-conflicting with $\mathcal{F}_2$ with respect to A for all $\overline{\delta} \geq \max\{\delta_{max}(A, \mathcal{F}_1, \mathcal{F}_2), \delta_{max}(A, \mathcal{F}_2, \mathcal{F}_1)\}$ and no $\underline{\delta} < \max\{\delta_{max}(A, \mathcal{F}_1, \mathcal{F}_2), \delta_{max}(A, \mathcal{F}_2, \mathcal{F}_1)\}$ exists such that $\mathcal{F}_1$ is $\underline{\delta}$-non-conflicting with $\mathcal{F}_2$.*

The above definition of δ-non-conflicting objectives is useful for changing a problem formulation by considering a different objective set. When replacing an objective set $\mathcal{F}_1$ by another objective set $\mathcal{F}_2$ which is δ-non-conflicting with $\mathcal{F}_1$, one can be sure after the replacement that for any $\vec{x}, \vec{y} \in X$, $\vec{x}$ either weakly dominates $\vec{y}$ with respect to both objective sets and we make no error, or $\vec{x}$ dominates $\vec{y}$ with respect to $\mathcal{F}_2$ and $\vec{x}$ weakly δ-dominates $\vec{y}$ with respect to $\mathcal{F}_1$. In other words, we make an error by considering $\mathcal{F}_2$ instead of $\mathcal{F}_1$ only if we wrongly assume that $\vec{x}$ weakly dominates $\vec{y}$, although $\vec{x}$ does not weakly dominate $\vec{y}$ with respect to $\mathcal{F}_1$. In this case, the error is bounded by δ: $\vec{y}$ is not better than $\vec{x}$ in any objective in $\mathcal{F}_1$ by an additive term of δ. As a consequence, we know that for any Pareto-optimal solution with respect to $\preceq_{\mathcal{F}_1}$ there exists a Pareto-optimal solution with respect to $\preceq_{\mathcal{F}_2}$ that weakly δ-dominates the former with respect to $\mathcal{F}_1$ (and vice versa)[5]. When replacing an objective set by a δ-non-conflicting subset of this objective set, one can guarantee that the resulting Pareto-optimal set is not worse than the original Pareto-optimal set by an additive term of δ in any omitted objective.

Based on this extended notion of conflict, one can canonically generalize the definitions of redundancy, minimal and minimum objective sets as follows.

Definition 6. *A set $\mathcal{F}' \subseteq \mathcal{F}$ of objectives is called* δ-redundant *if and only if there exists an objective subset $\mathcal{F}'' \subset \mathcal{F}'$ that is δ-non-conflicting with $\mathcal{F}'$.*

Definition 7. *Let $\delta \geq 0$. An objective set $\mathcal{F}'' \subseteq \mathcal{F}'$ is denoted as*

- δ-minimal with respect to $\mathcal{F}'$ *iff $\mathcal{F}''$ is both not δ-redundant and δ-non-conflicting with $\mathcal{F}'$;*
- δ-minimum with respect to $\mathcal{F}'$ *iff $\mathcal{F}''$ is the smallest δ-minimal objective set with respect to $\mathcal{F}'$.*

[5] Note that the definition of Pareto optimality with respect to a certain set of objectives is the same than the one given in Sec. 1.2 and we only specify it here explicitly as the underlying set of objectives changes.

A further aspect that can be of interest is to ask for the minimum error δ that is possible when restricting the size of the reduced objective set by an upper bound. This leads to the `k-EMOSS` problem that will be introduced in the following section.

Example 7. *Regarding the above example (Fig. 16), the set* $\{f_1, f_3, f_4\}$ *is* 0.5*-minimal but not* 0.5*-minimum with respect to the entire objective set, since the smaller set* $\{f_2, f_4\}$ *is* 0.5*-minimal as well. Because no objective set with one objective only induces an error smaller than or equal to* 0.5*, the set* $\{f_2, f_4\}$ *is also* 0.5*-minimum with respect to the entire objective set.*

3.3 Computing Minimum Objective Sets: Problems and Their Complexity

After we have defined what minimum objective sets are, obvious open questions are how we can compute them and how difficult in terms of complexity theory the corresponding problem is. This section will investigate the latter question before we provide objective reduction algorithms in the next section. Regarding the measure of conflict, as defined in the previous section, there are two perspectives of objective reduction: on the one hand, given an error δ, one may ask for a δ-minimum objective set; on the other hand, one can ask for a δ-minimum objective subset of predefined size `k` with the smallest possible δ-error. These problems can be formalized as follows.

Definition 8. *Given a* $\delta \in \mathbb{R}$ *and a set* $A \subseteq X$ *of* m *solutions, together with the objective values* $f_i(\vec{x}) \in \mathbb{R}$ *where* $1 \leq i \leq k$ *and* $\vec{x} \in A$*, the problem* δ-`MINIMUM OBJECTIVE SUBSET`*,* δ-`MOSS` *for short, is to compute an objective subset* $\mathcal{F}' \subseteq \mathcal{F}$ *which is* δ*-minimum with respect to* $\mathcal{F}$*.*

Definition 9. *Given a* `k` $\in \mathbb{N}$ *and a set* $A \subseteq X$ *of* m *solutions, together with the objective values* $f_i(\vec{x}) \in \mathbb{R}$ *where* $1 \leq i \leq k$ *and* $\vec{x} \in A$*, the problem* `MINIMUM OBJECTIVE SUBSET OF SIZE k WITH MINIMUM ERROR`*, or* `k-EMOSS` *for short, is to compute an objective subset* $\mathcal{F}' \subseteq \mathcal{F}$ *which has size* $|\mathcal{F}'| \leq$ `k` *and is* δ*-non-conflicting with* $\mathcal{F}$ *with the minimal possible* δ*.*

As the set A, we can imagine either the entire search space ($A = X$), which is only feasible for small search spaces, or an arbitrary sample of the search space such as a Pareto front approximation or the population of an evolutionary multiobjective optimizer ($A \subset X$). Unfortunately, both problems are considered to be hard to solve in general as the next theorem states.

Theorem 9. *Both the* δ-`MOSS` *problem and the* `k-EMOSS` *problem are* $\mathcal{NP}$*-hard.*

Before we prove Theorem 9, we state another simple objective reduction problem, the $\mathcal{NP}$-hardness of which can be shown easily by a Turing reduction[6] from the $\mathcal{NP}$-hard set cover problem. This `MINIMUM OBJECTIVE SUBSET PROBLEM` (`MOSS`) is then later used to show the $\mathcal{NP}$-hardness of the generalized δ-`MOSS` and `k-EMOSS` problems.

Definition 10. *Given a set $A \subseteq X$ of m solutions together with the weak Pareto dominance relation $\preceq_{\mathcal{F}}$ and for all objective functions $f_i \in \mathcal{F}$ $(1 \leq i \leq k)$ the single relations $\preceq_i$ where $\bigcap_{1 \leq i \leq k} \preceq_i = \preceq_{\mathcal{F}}$, the problem* `MINIMUM OBJECTIVE SUBSET`*, or* `MOSS` *for short, is to compute an index $I \subseteq \{1, \ldots, k\}$ of minimum size with $\bigcap_{i \in I} \preceq_i = \preceq_{\mathcal{F}}$.*

Theorem 10. *The problem* `MOSS` *is $\mathcal{NP}$-hard.*

Proof. First, we recapitulate the definition of the $\mathcal{NP}$-hard `SET COVER PROBLEM`, or `SCP` for short, from Garey and Johnson (1990):

> *Given a collection $C = \{C_1, \ldots, C_l\}$ of $l \in \mathbb{N}$ subsets of a finite set $S = \{1, \ldots, m\}$ $(m \in \mathbb{N})$, compute an index $I \subseteq \{1, \ldots, l\}$ of minimum size with $\bigcup_{i \in I} C_i = S$.*

A Turing transformation `SCP` $\leq_T$ `MOSS` proves the $\mathcal{NP}$-hardness of `MOSS`. Figure 17 gives an illustrative example of the transformation for a collection of $l = 3$ subsets.

Starting from the `SCP` instance consisting of the set $S = \{s_1, \ldots, s_m\}$ and the subsets C_i with $1 \leq i \leq l$, all relations $\preceq_i$ as well as $\preceq_{\mathcal{F}}$ in the `MOSS` instance are defined on the basic set $A := \{\vec{x}_1, \ldots, \vec{x}_m, \vec{x}'_1, \ldots, \vec{x}'_m\}$ where the $\vec{x}_j$ and $\vec{x}'_j$ $(1 \leq j \leq m)$ are single solutions. The relation $\preceq_{\mathcal{F}}$ will be the reflexive closure of the antichain on A, i. e., $\preceq_{\mathcal{F}}$ only contains the elements $(\vec{x}_j, \vec{x}_j)$ and $(\vec{x}'_j, \vec{x}'_j)$ for $1 \leq j \leq m$, see Fig. 17. The relations $\preceq_i$ with $1 \leq i \leq l$ are all constructed in the same way. A relation $\preceq_i$ is build up on the linear preorder[7] $[\vec{x}_1, \vec{x}'_1, \vec{x}_2, \vec{x}'_2, \ldots, \vec{x}_m, \vec{x}'_m]$ which additionally, contains the element $(\vec{x}'_j, \vec{x}_j)$ iff $s_j \notin C_i$. Furthermore, we have to compute another relation $\preceq_{k+1}$ which is the reverse linear preorder $[\vec{x}'_m, \vec{x}_m, \vec{x}'_{m-1}, \vec{x}_{m-1}, \ldots, \vec{x}'_1, \vec{x}_1]$. After this transformation, we question our `MOSS` oracle once. The resulting index $I_{\texttt{SCP}}$ for the `SCP` problem will be then $I_{\texttt{SCP}} := I_{\text{oracle}} \setminus \{l+1\}$ if the oracle produces I_{oracle} as its output.

It remains to show that the transformation yields an exact algorithm for `SCP` with polynomial running time, under the assumption that there is an exact polynomial time algorithm $\mathcal{A}$ for `MOSS`. Let us assume that $(S =$

[6] For general details on complexity theory and Turing reductions in particular, we refer to the book of Garey and Johnson (1990).

[7] We write a linear preorder of n solutions as $[\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n]$ and interpret it as the relation $\{(\vec{x}_i, \vec{x}_j) \mid 1 \leq i \leq j \leq n\}$.

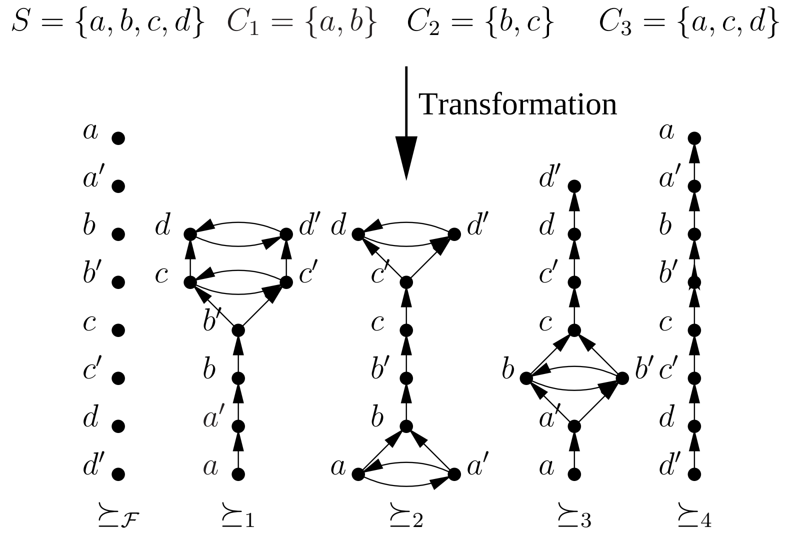

$S = \{a, b, c, d\}$ $C_1 = \{a, b\}$ $C_2 = \{b, c\}$ $C_3 = \{a, c, d\}$
Transformation
a a' b b' c c' d d'
d d' c c' b' b a' a
d d' c' c b' b a a'
d' d c' c b b' a' a
a a' b b' c c' d d'
$\succeq_{\mathcal{F}}$
$\succeq_1$
$\succeq_2$
$\succeq_3$
$\succeq_4$

Figure 17: An example for the Turing reduction from SCP to MOSS. The reflexive and transitive edges are omitted for clarity.

$\{s_1, \ldots, s_m\}, C_1, \ldots, C_l)$ is the SCP instance with $C_i = \{c_1, \ldots, c_{|C_i|}\} \subseteq S$. Via the described transformation and the hypothetical algorithm $\mathcal{A}$, we can compute the index $I_{\texttt{SCP}} := I_{\mathcal{A}} \setminus \{l + 1\}$ as the output corresponding to the SCP instance. Obviously, the computation of $I_{\texttt{SCP}}$ is possible in polynomial time using a polynomial algorithm for MOSS where the transformation needs $O(lm^2)$ time and produces a MOSS instance of size $O(lm^2)$. To complete the proof, we still have to show (i) why always $l + 1 \in I_{\mathcal{A}}$, (ii) why $I_{\mathcal{A}} \setminus \{l + 1\}$ is a correct output for our SCP instance, and (iii) why the computed index $I_{\mathcal{A}} \setminus \{l + 1\}$ is minimum.

First, we will take a look at the question (i), i. e., why $\preceq_{l+1}$ is always needed to yield $\preceq_{\mathcal{F}}$ as the intersection of some $\preceq_i$. Because in $\preceq_{\mathcal{F}}$ no pair $\vec{x}, \vec{y} \in A$ with $\vec{x} \neq \vec{y}$ is comparable, for each pair $\vec{x}, \vec{y} \in A$, $\vec{x} \neq \vec{y}$, there has to be at least one $i \in I_{\mathcal{A}}$ where $\vec{x} \not\preceq_i \vec{y}$ and at least one $j \in I_{\mathcal{A}}$ with $\vec{y} \not\preceq_j \vec{x}$. Considering the pair $\vec{x}_1, \vec{x}_1'$, for all $\preceq_i$ with $i \in \{1, \ldots, l\}$, $\vec{x}_1 \preceq_i \vec{x}_1'$ holds by construction. Only in the last objective f_{l+1}, $\vec{x}_1 \preceq_{l+1} \vec{x}_1'$ does not hold. Consequently, $\preceq_{l+1}$ is always needed, to construct $\preceq_{\mathcal{F}}$ as the intersection of single $\preceq_i$'s. Now we show (ii) why $I := I_{\mathcal{A}} \setminus \{l + 1\}$ is always a correct output for the given SCP instance. As we have seen before, $l + 1 \in I_{\mathcal{A}}$ and therefore, the intersection of the $\preceq_i$'s does not contain any pairs $(\vec{x}_\nu, \vec{x}_\mu)$, $(\vec{x}_\nu, \vec{x}_\mu')$, $(\vec{x}_\nu', \vec{x}_\mu)$, and $(\vec{x}_\nu', \vec{x}_\mu')$ with $1 \leq \nu < \mu \leq m$ and no pairs $(\vec{x}_\nu, \vec{x}_\nu')$ with $1 \leq \nu \leq m$. The construction of the relations $\preceq_i$ with $i \in \{1, \ldots, l\}$ results in the absence of pairs $(\vec{x}_\nu, \vec{x}_\mu)$, $(\vec{x}_\nu, \vec{x}_\mu')$, $(\vec{x}_\nu', \vec{x}_\mu)$, and $(\vec{x}_\nu', \vec{x}_\mu')$ with $1 \leq \mu < \nu \leq m$ in the intersection if there will be at least one $i \in I_{\mathcal{A}}$ selected with $1 \leq i \leq l$. There only remains the possibility of pairs $(\vec{x}_\nu', \vec{x}_\nu)$ with $1 \leq \nu \leq m$ in the intersection. To avoid this, for each $\nu \in \{1, \ldots, m\}$ there must be at least one $i \in \{1, \ldots, l\}$ in $I_{\mathcal{A}}$ with $\vec{x}_\nu' \not\preceq_i \vec{x}_\nu$. By construction of the Turing transformation, this can only occur if $c_\nu \in C_i$.

Thus, $\bigcup_{i \in I_{\mathcal{A}} \setminus \{l+1\}} C_i = \{1, \ldots, m\} = S$. Last, we have to show (iii) why the computed index $I_{\mathcal{A}} \setminus \{l+1\}$ is a minimum index for `SCP`. Assume that $I_{\mathcal{A}} \setminus \{l+1\}$ is not a minimum index for `SCP`, i.e., there is a smaller index J with $|J| < |I|$ and $\bigcup_{j \in J} C_j = S$. As one can easily see from the above transformation, $J \cup \{l+1\}$ would be a smaller index for `MOSS` than $I_{\mathcal{A}}$. □

We can now come back to the proof that δ-`MOSS` and `k-EMOSS` are $\mathcal{NP}$-hard problems:

Proof of Theorem 9. First, we prove the $\mathcal{NP}$-hardness of δ-`MOSS` by a Turing reduction from the above $\mathcal{NP}$-hard `MOSS` problem. Secondly, we prove the $\mathcal{NP}$-hardness of `k-EMOSS` via a Turing reduction from δ-`MOSS`.

`MOSS` $\leq_T$ δ-`MOSS`
The idea of this Turing reduction is to compute objective values for all the solutions in A of a `MOSS` instance yielding the same weak dominance relation as $\preceq_{\mathcal{F}}$ and $\preceq_i$ respectively. With those objective vectors and $\delta = 0$, the δ-`MOSS` oracle is asked once for a 0-minimum objective set. This objective set can directly be used as output for the `MOSS` problem, since the two problems with $\delta = 0$ ask for the same minimum objective set. It remains to show how the objective values are computed and that it is possible within polynomial time. Starting with the `MOSS` instance $(A, \preceq_{\mathcal{F}}, \preceq_i \text{ for all } 1 \leq i \leq k)$, the δ-`MOSS` instance is computed in time $O(k \cdot |A|^2)$ as follows. Choose $\delta = 0$. Assign the solutions' ith objective values according to a topological sorting of $\preceq_i$. As there are at most $O(|A|^2)$ edges in the relation graphs of the $\preceq_i$, the topological sorting costs $O(|A|^2)$ per objective, resulting in a runtime of $O(k \cdot |A|^2)$ in total.

δ-`MOSS` $\leq_T$ `k-EMOSS`
A δ-minimum objective set with respect to $\mathcal{F}$ is obviously of size $1 \leq l \leq k$. Asking the `k-EMOSS` oracle with the same objective values than the δ-`MOSS` instance and all possible sizes $1 \leq l \leq k$ iteratively, the smallest computed objective set which has an error of at most δ is a δ-minimum set, i.e., it can be taken as output for the δ-`MOSS` problem. The Turing transformation can be done in linear time regarding the δ-`MOSS` instance. □

Since the two problems δ-`MOSS` and `k-EMOSS` together with the restricted `MOSS` problem are $\mathcal{NP}$-hard, we cannot expect to find polynomial algorithms that solve the problems exactly unless $\mathcal{P} = \mathcal{NP}$. However, we propose an exact algorithm with exponential running time in the next section to have a reference algorithm when investigating the potential of objective reduction later on. Furthermore, the next section presents several approximation algorithms based on greedy heuristics that are fast enough in practice and, for the special case of the `MOSS` problem, ensure the best theoretically possible approximation ratio.

3.4 Computing Minimum Objective Sets: Algorithms

In the following, we propose both exact and approximation algorithms to solve the two problems δ-`MOSS` and `k-EMOSS`. Corresponding implementations of the algorithms are freely available for download at `http://www.tik.ee.ethz.ch/sop/download/supplementary/objectiveReduction/`.

3.4.1 An Exact Algorithm

In this subsection, we propose an exact algorithm, that is exponential in the number k of objectives involved but polynomial in the number $|A|$ of solutions, and that is suited to both problem formulations δ-`MOSS`, and `k-EMOSS` respectively. The practical use of this algorithm is twofold. On the one hand, this algorithm is used later to investigate the potential of the proposed objective reduction approach by computing the maximally achievable objective reduction for some test problems. On the other hand, the exact algorithm provides a basis to compare the quality of objective subsets computed by heuristic approaches that are presented later in this section.

Instead of simply considering all 2^k possible objective subsets and computing whether they are minimal with respect to the set $\mathcal{F}$ of all objectives and the entire set of solutions A, the basic idea of the exact algorithm is to consider solution pairs separately. This separate information is then combined to get all minimal objective sets for increasing sets of solution pairs. The algorithm considers all solution pairs $(\vec{x}, \vec{y})$ successively in arbitrary order. The solution pairs considered so far are stored in the set M. The set S_M contains at any time all minimal objective subsets $\mathcal{F}'$ together with the minimal δ' value such that $\mathcal{F}'$ is δ'-non-conflicting with the set $\mathcal{F}$ of all objectives when taking into account only the solution pairs in M.

The algorithm uses a subfunction $\delta_{\max}(\{\vec{x}, \vec{y}\}, \mathcal{F}_1, \mathcal{F}_2)$, to compute for two solutions $\vec{x}, \vec{y} \in A$ and two objective sets $\mathcal{F}_1, \mathcal{F}_2$ the maximum $\delta_{\max}$ error from Def. 4 such that $\mathcal{F}_1$ and $\mathcal{F}_2$ are $\delta_{\max}$-non-conflicting with respect to the solution set $\{\vec{x}, \vec{y}\}$. To guarantee, that the set S_M contains only pairs $(\mathcal{F}', \delta')$ such that $\mathcal{F}'$ is always δ'-minimal with respect to $\mathcal{F}$ with the smallest δ' possible, the union $\sqcup$ of two sets of objective subsets S_1, S_2 is done with simultaneous deletion of not δ'-minimal pairs $(\mathcal{F}', \delta')$ as follows:

$$
\begin{aligned}
S_1 \sqcup S_2 := \{(\mathcal{F}_1 \cup \mathcal{F}_2, \max\{\delta_1, \delta_2\}) \mid (\mathcal{F}_1, \delta_1) \in S_1 \wedge (\mathcal{F}_2, \delta_2) \in S_2 \\
\wedge \nexists (\mathcal{F}_1', \delta_1') \in S_1, (\mathcal{F}_2', \delta_2') \in S_2 : \\
\big(\mathcal{F}_1' \cup \mathcal{F}_2' \subset \mathcal{F}_1 \cup \mathcal{F}_2 \wedge \max\{\delta_1', \delta_2'\} \leq \max\{\delta_1, \delta_2\}\big) \\
\wedge \nexists (\mathcal{F}_1', \delta_1') \in S_1, (\mathcal{F}_2', \delta_2') \in S_2 : \\
\big(\mathcal{F}_1' \cup \mathcal{F}_2' \subseteq \mathcal{F}_1 \cup \mathcal{F}_2 \wedge \max\{\delta_1', \delta_2'\} < \max\{\delta_1, \delta_2\}\big)\big\}
\end{aligned}
$$

The full procedure is detailed in Algorithm 4. Note that the running time of Algorithm 4 is polynomial in the number $m := |A|$ of solutions but exponential in the number k of objectives. Due to the $\mathcal{NP}$-hardness of the objective reduction problems, we cannot expect to find much faster exact algorithms. Nevertheless, the exact algorithm is applicable for instances with few objectives and a moderate number of solutions as experimental results will show in Sec. 3.5.

Theorem 11. *Algorithm 4 solves both the* δ-`MOSS` *and the* `k-EMOSS` *problem exactly in time* $O(m^2 \cdot k \cdot 2^k)$.

For details and the very technical correctness proof, we refer to Appendix A. The upper bound for the running time of the exact algorithm can be derived by computing the maximum size of the set S_M. As S_M contains at most $O(2^k)$ objective subsets of size $O(k)$, the computation of $S_M \sqcup S_{\{(\vec{x},\vec{y})\}}$ in line 9 is possible in time $O(k \cdot 2^k)$. The outer loop will be finished after at most $O(m^2)$ iterations. Thus, the entire algorithm runs in time $O(m^2 \cdot k \cdot 2^k)$. Note, that the exact algorithm can be easily parallelized, as the computation of the sets $S_{\{(\vec{x},\vec{y})\}}$ are independent for different pairs $(\vec{x}, \vec{y})$. It can also be accelerated if line 9 of Algorithm 4 is tailored to either the δ-`MOSS` or the `k-EMOSS` problem by including a pair $(\mathcal{F}', \delta')$ into $S_{M \cup \{(\vec{x},\vec{y})\}}$ only if $\delta' \leq \delta$, and $|\mathcal{F}'| \leq$ `k` respectively, which is done in the implementation available for download and used for the experiments to follow. Moreover, the algorithm can be used to compute *all* minimum objective subsets if simply the entire set S_M is outputted instead of a single element $(\mathcal{F}', \delta')$.

That Algorithm 4 indeed has a running time that is exponential in the number of objectives shows the following theorem which is based on a constructed `MOSS` instance.

Theorem 12. *The worst-case running time of Algorithm 4 for the* `MOSS` *problem is* $\Omega(m^2 \cdot 2^{k/3})$.

Proof. Figure 18 shows the parallel coordinates plot of an instance for which Algorithm 4 needs time $\Omega(m^2 \cdot 2^{k/3})$ if m solutions and k objectives are involved. We assume that this instance consists of an even number of m

Algorithm 4 An exact algorithm for the problems δ-MOSS and k-EMOSS

```
Init:
    $M := \emptyset$
    $S_M := \emptyset$
for all pairs $\vec{x}, \vec{y} \in A$, $\vec{x} \neq \vec{y}$ of solutions do
    $S_{\{(\vec{x},\vec{y})\}} := \emptyset$
    for all objective pairs $f_i, f_j \in \mathcal{F}$, not necessary $i \neq j$ do
        compute $\delta_{ij} := \delta_{\max}(\{\vec{x}, \vec{y}\}, \{f_i\} \cup \{f_j\}, \mathcal{F})$
        $S_{\{(\vec{x},\vec{y})\}} := S_{\{(\vec{x},\vec{y})\}} \sqcup (\{f_i\} \cup \{f_j\}, \delta_{ij})$
    end for
    $S_M := S_M \sqcup S_{\{(\vec{x},\vec{y})\}}$
    $M := M \cup \{(\vec{x}, \vec{y})\}$
end for
Output for $\delta$-MOSS: $(\mathcal{F}', \delta')$ in $S_M$ with minimal size $|\mathcal{F}_{\min}|$ and $\delta' \leq \delta$
Output for k-EMOSS: $(\mathcal{F}', \delta')$ in $S_M$ with size $|\mathcal{F}'| \leq$ k and minimal $\delta'$
```

solutions $A := \{\vec{x}_1, \ldots, \vec{x}_m\}$ together with the relation $\preceq_{\mathcal{F}}$ and $k = 3/2 \cdot m$ relations $\preceq_i$ corresponding to the objective functions $\mathcal{F} := \{f_1, \ldots, f_{3/2 \cdot m}\}$ where only the solutions $\vec{x}_{2i-1}$ and $\vec{x}_{2i}$ for $1 \leq i \leq m/2$ are incomparable. The incomparability of such pairs is only caused by their $3i$th, $(3i+1)$th, and $(3i+2)$th objective values, i. e., we need either the objective pair f_{3i-2}, f_{3i-1} or the pair f_{3i-1}, f_{3i} to describe the incomparability, cf. the parallel coordinates plot in Fig. 18. Thus, whenever Algorithm 4 considers a new pair $\vec{x}_{2i-1}, \vec{x}_{2i}$ of incomparable solutions, the size of the set S reduplicates. Because we have $m/2 = k/3$ of those incomparable pairs, S is of size $2^{k/3}$ after the algorithm considered all of the $k/3$ incomparable pairs. This is possible after the first $k/3$ of altogether $\binom{m}{2}$ steps of the algorithm, which results in a running time of at least $(\binom{m}{2} - k/3) \cdot 2^{k/3} = \Omega(m^2 \cdot 2^{k/3})$. Note that this restricted example can be easily extended to the case where $m > k$. □

3.4.2 Heuristics

The three heuristic algorithms, we propose in this section, are better suited for large instances of the δ-MOSS problem, and k-EMOSS respectively, than the proposed exact algorithm. They are much faster but therefore do not guarantee to find a δ-minimum objective set. Nevertheless, the sizes of the objective sets, and the δ-errors respectively, are close to the sizes and errors of the δ-minimal sets found by the exact algorithm, see Sec. 3.5. In addition, for the case of 0-MOSS, the greedy algorithm proposed below has the best approximation ratio possible as we will detail in the following.

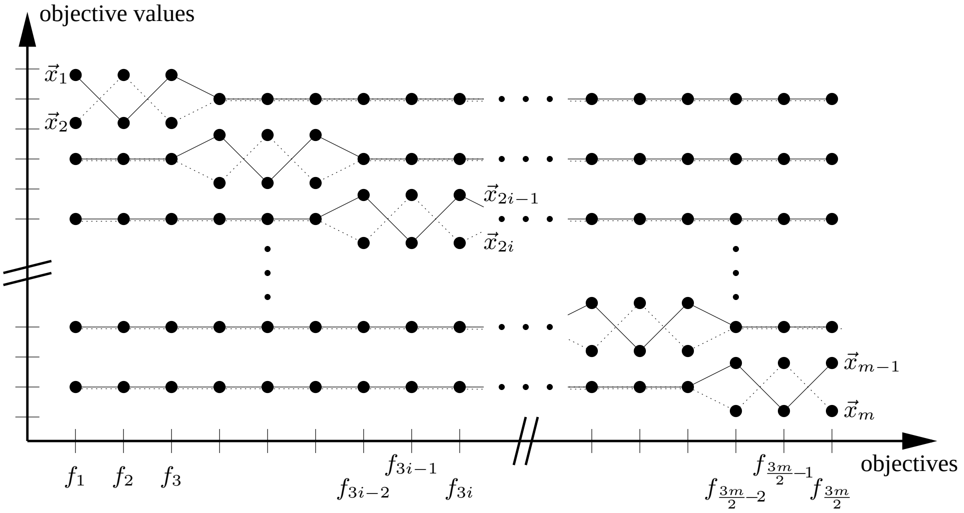


Figure 18: The parallel coordinates plot of an instance with an even number m solutions and $k = 3/2 \cdot m$ objectives for which the exact algorithm needs time $\Omega(m^2 \cdot 2^{k/3})$.

3.4.2.1 A Greedy Algorithm for δ-MOSS

The general idea of the proposed approximation algorithm for δ-MOSS is to adapt the well-known greedy algorithm for the set cover problem to compute an objective subset $\mathcal{F}'$, δ-non-conflicting with the set $\mathcal{F}$ of all objectives in a greedy way. This greedy algorithm for the set cover problem chooses at each step the set that covers most of the uncovered elements until all elements are covered (Slavík, 1996). With respect to the δ-MOSS problem, this algorithm translates into the following one.

Starting with an empty set $\mathcal{F}'$ of objectives, the algorithm chooses in each step the objective f_i the addition of which removes most of the edges in the relation graph of $\preceq_{\mathcal{F}'}$ which are not contained in the relation graph for all objectives, i.e., $\mathcal{F}$. Since we are interested in approximating the δ-MOSS problem, i.e., finding a δ-non-conflicting objective set, we do not care about remaining edges in $\preceq_{\mathcal{F}'}$ which imply an error of at most δ. This idea is formalized with the following generalization of the weak ε-dominance, the (δ_1, δ_2)-dominance relation:

Definition 11. *Let $\delta_1, \delta_2 \in \mathbb{R}$ and $\mathcal{F}_1, \mathcal{F}_2$ be two objective subsets. The (δ_1, δ_2)-dominance relation $\preceq^{\delta_1,\delta_2}_{\mathcal{F}_1,\mathcal{F}_2}$ on X is defined as $\vec{x} \preceq^{\delta_1,\delta_2}_{\mathcal{F}_1,\mathcal{F}_2} \vec{y} :\Longleftrightarrow (\forall f_1 \in \mathcal{F}_1 : f_1(\vec{x}) \leq f_1(\vec{y}) + \delta_1) \wedge (\forall f_2 \in \mathcal{F}_2 : f_2(\vec{x}) \leq f_2(\vec{y}) + \delta_2)$ for all $\vec{x}, \vec{y} \in X$.*

The (δ_1, δ_2)-dominance relation states that with respect to objective set $\mathcal{F}_1$ a solution δ_1-dominates another one and with respect to the objective set $\mathcal{F}_2$ the same solution δ_2-dominates the latter. Within the greedy algorithm for δ-MOSS, the details of which are depicted as Algorithm 5, all edges in $\preceq^{0,\delta}_{\mathcal{F}' \cup \{i\}, \mathcal{F} \setminus (\mathcal{F}' \cup \{i\})}$ are not considered; that means we do not care about so-

Algorithm 5 A Greedy Algorithm for δ-MOSS.

```
Init:
    based on the solution set $A$ and the corresponding objective vectors,
    compute the relations $\preceq_i$ for all $1 \leq i \leq k$ as well as $\preceq_{\mathcal{F}}$
    $\mathcal{F}' := \emptyset$
    $R := A \times A \setminus \preceq_{\mathcal{F}}$
while $R \neq \emptyset$ do
    $i^* = \underset{i \in \mathcal{F} \setminus \mathcal{F}'}{\operatorname{argmin}} \{ |(R \cap \preceq_i) \setminus \preceq^{0,\delta}_{\mathcal{F}' \cup \{i\}, \mathcal{F} \setminus (\mathcal{F}' \cup \{i\})} | \}$
    $R := (R \cap \preceq_{i^*}) \setminus \preceq^{0,\delta}_{\mathcal{F}' \cup \{i^*\}, \mathcal{F} \setminus (\mathcal{F}' \cup \{i^*\})}$
    $\mathcal{F}' := \mathcal{F}' \cup \{i^*\}$
end while
```

lutions inducing an error of at most δ in the objectives in $\mathcal{F} \setminus (\mathcal{F}' \cup \{i\})$, i.e., in the set of objectives that are not taken. For the proof of the polynomial running time and the correctness stated in Theorem 13, we refer to Appendix A.

Theorem 13. *Given the objective vectors $f(\vec{x}_1), \ldots, f(\vec{x}_m) \in \mathbb{R}^k$ and a $\delta \in \mathbb{R}$, Algorithm 5 always provides an objective subset $\mathcal{F}' \subseteq \mathcal{F}$, δ-non-conflicting with $\mathcal{F} := \{f_1, \ldots, f_k\}$ in time $O(\min\{k^3 \cdot m^2, k^2 \cdot m^4\})$.*

Note, that Algorithm 5 does not necessarily yield a δ-minimal objective set. However, by simply checking whether an additional omission of single objectives in the computed set $\mathcal{F}'$ leaves the dominance relation unchanged, a δ-minimal set can be guaranteed. The asymptotic running time will stay the same, since the additional check of δ-minimality can be done in time $O(k^2 \cdot m^2)$. For the case of $\delta = 0$, Algorithm 5 corresponds to a well-known greedy algorithm for the set cover problem where $\preceq^{0,0}_{\mathcal{F}' \cup \{i\}, \mathcal{F} \setminus (\mathcal{F}' \cup \{i\})} = \preceq_{\mathcal{F}}$. In this case, we can use results on the approximation ratio of polynomial algorithms for the set cover problem (Slavík, 1996; Feige, 1998) to prove that Algorithm 5 has the best possible approximation ratio of $\Theta(\log |A|)$. Since the problem MOSS is $\mathcal{NP}$-hard, we cannot expect to solve the problem exactly in polynomial time unless $\mathcal{P} = \mathcal{NP}$. However, we can aim at polynomial algorithms that approximate an optimal solution, i.e., algorithms that try to find an objective subset that is probably not optimal but where the size can be upper bounded. The term approximation ratio then refers to such an upper bound on the number of objectives in a computed objective set divided by the number of objectives in an optimal set, i.e., a minimum objective set.

Theorem 14. *Algorithm 5 with $\delta = 0$ can be seen as an approximation algorithm for the* MOSS *problem with the best possible approximation ratio of $\Theta(\log |A|)$.*

Proof. First of all, we have to comment on the fact that Algorithm 5 cannot directly be applied to the MOSS problem since the instances of δ-MOSS and MOSS differ. In case the instance size for the δ-MOSS problem is not too high, i.e., the description length of the objective vectors is not too large[8] the instance for 0-MOSS can be easily transferred to a MOSS instance of similar size by computing the corresponding relations from the given objective vectors. Then, results on the approximation ratio of a greedy algorithm for the set cover problem (SCP) can be used to prove the theorem.

To show the upper bound on the approximation ratio, we sketch the proof of a Turing reduction MOSS $\leq_T$ SCP and refer to Theorem 19 in Appendix A for the entire proof. Given an instance for MOSS, consisting of the relations $\preceq_{\mathcal{F}} \subseteq X \times X$ and $\preceq_i \subseteq X \times X$ with $\bigcap_{1 \leq i \leq k} \preceq_i = \preceq_{\mathcal{F}}$, we can compute an SCP instance as follows. The set S in the SCP instance contains an element $s_{x,y}$ for each $(\vec{x}, \vec{y}) \in \preceq_{\mathcal{F}}$. A subset C_i of S in the SCP instance contains an element $s_{x,y}$ iff $\vec{x} \not\preceq_i \vec{y}$. The output for the MOSS problem, is the index I, computed by the SCP oracle. The Turing reduction needs time linear in the instance size and produces an SCP instance of linear size. This translation shows the equivalence between MOSS and SCP: Where SCP is asking for the smallest set of C_i's such that $\cup C_i = S$, MOSS asks for the smallest number of objectives $\preceq_i$ such that $\cap \preceq_i = \preceq_{\mathcal{F}}$. Since Algorithm 5 uses this equivalence and then acts like the greedy algorithm for SCP, the upper bound $O(\log m)$ for the approximation ratio of the greedy algorithm for SCP, where m is the number of elements in the ground set of SCP which corresponds to the number of solutions in the MOSS instance, is directly translated to Algorithm 5.

For proving that Algorithm 5 has an approximation ratio of $\Omega(\log |A|)$, we use conclusions made for SCP. The work of Feige (1998) showed that there is no $\varepsilon > 0$ such that an approximation algorithm can solve SCP with approximation ratio $(1 - \varepsilon) \ln m$, unless $\mathcal{NP} \subset TIME(m^{O(\log \log m)})$. With the Turing transformation from SCP to MOSS in the proof of Theorem 10, Feige's lower bound for SCP yields a lower bound of $\Omega(\log 2m) = \Omega(\log m) = \Omega(\log |A|)$ for MOSS. This is due to the fact that in the transformation from SCP to MOSS the size m of the set S is transformed into the set A of size $2m$. Assuming, that there is a polynomial approximation algorithm for MOSS with an approximation ratio of $o(\log m)$, we get a contradiction to Feige's results, because we can transform each SCP instance in polynomial time into a MOSS instance with A of size $2m$ and solve SCP via the $o(\log m)$ algorithm for MOSS. □

[8] By description length, we mean the length of a binary representation of the objective values which might be high in comparison to the number of objectives and solutions involved if we allow for a high precision or exponentially large values (in m and k) of the objectives. If we do not restrict the description length here, the complexity results do not hold, e.g., if the instance size is already exponential in the number of solutions.

Algorithm 6 A greedy algorithm for `k-EMOSS`

```
Init:
    $\mathcal{F}' := \emptyset$
while $|\mathcal{F}'| < \mathtt{k}$ do
    $\mathcal{F}' := \mathcal{F}' \cup \underset{i \in \mathcal{F} \setminus \mathcal{F}'}{\operatorname{argmin}} \{\delta_{\max}(A, \mathcal{F}' \cup \{i\}, \mathcal{F})\}$
end while
```

3.4.2.2 A Greedy Algorithm for `k-EMOSS`

A simple greedy heuristic to approximate the `k-EMOSS` problem is to choose the `k` objectives iteratively. Starting with an empty set $\mathcal{F}'$ of objectives in each of the `k` steps, the algorithm chooses the next objective f_i to be included into $\mathcal{F}'$ as the objective yielding the smallest δ such that $\mathcal{F}' \cup \{f_i\}$ is δ-non-conflicting with the entire objective set, see Algorithm 6. Algorithm 6 obviously computes always an objective subset of size `k` which is δ-non-conflicting with the entire objective set but does not guarantee to find the set with minimal δ.

Theorem 15. *Algorithm 6 needs time $O(m^2 \cdot k^3)$ to compute an objective subset of size* `k`.

Proof. The greedy algorithm needs time $O(m^2 \cdot k^3)$ altogether since at most k loops with k calls of the $\delta_{\min}$ subfunction are needed. One call of the $\delta_{\min}$ function needs time $\Theta(m^2 \cdot k)$ and all other operations need time $O(1)$ each. □

3.4.2.3 A Second Greedy Algorithm for `k-EMOSS` based on Omission of Objectives

In this section, we present a second greedy algorithm for the `k-EMOSS` problem, allowing a kind of hierarchical clustering of the objective set yielding a visualization of the computed δ-errors in a tree, as depicted in Fig. 19. Instead of constructing a δ-non-conflicting objective set by *adding* objectives as in Algorithm 6, Algorithm 7 *removes* objectives greedily until the resulting subset has the desired number of `k` objectives. At each step, the objective pair f_i, f_j with the smallest δ-error between f_i and f_j is selected and the objective that maximizes the error between f_i and f_j is omitted. Algorithm 7 provides the details. If the algorithm is run with $\mathtt{k} = 1$, each of its steps can be visualized as an inner node in a tree, cf. Fig. 19, which can support the decision maker with useful information on the measure of conflict between objective pairs. Starting with the set of all objectives at the leafs, each iteration of the algorithm corresponds to an inner node where one objective is omitted; the later an objective is omitted, the closer the corresponding node is to the root (Fig. 19).

Algorithm 7 A second greedy algorithm for `k-EMOSS`, based on omitting objectives.

```
1:  Init:
2:      $\mathcal{F}' := \mathcal{F}$
3:  while $|\mathcal{F}'| > \mathtt{k}$ do
4:      $(f_r, f_s) := \underset{f_i, f_j \in \mathcal{F}'}{\operatorname{argmin}} \{\delta_{\max}(A, \{f_i\}, \{f_j\})\}$
5:      if $\delta_{\max}(A, \{f_r\}, \{f_s\}) < \delta_{\max}(A, \{f_s\}, \{f_r\})$ then
6:          $\mathcal{F}' := \mathcal{F}' \setminus \{f_s\}$
7:      else
8:          $\mathcal{F}' := \mathcal{F}' \setminus \{f_r\}$
9:      end if
10: end while
```

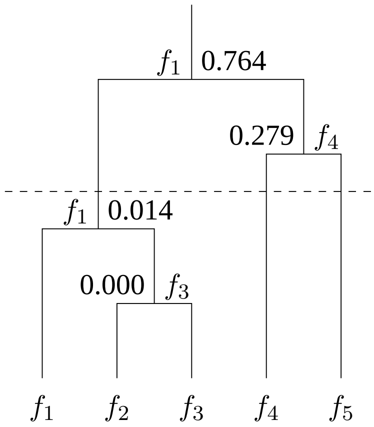


Figure 19: Example of a tree computed by Algorithm 7. The course of the algorithm can be identified by a horizontal line that moves from the leaves to the root. Each inner node thereby indicates the omission of an objective and the corresponding δ-error after this omission. Note that the objective noted at each inner node is the only objective from the corresponding subtree that is *kept*. The dashed line, e.g., corresponds to the situation after the objectives f_2 and f_3 have been omitted, i.e., when the objective set is reduced to $\{f_1, f_4, f_5\}$ and the error is $\delta = 0.014$.

Theorem 16. *Algorithm 7 needs time* $O((k - \mathtt{k}) \cdot k^2 \cdot m^2) = O(k^3 \cdot m^2)$ *to compute an objective subset of size* $\mathtt{k}$.

Proof. The computation of the minimal δ-error in the $\delta_{\max}(A, \{f_i\}, \{f_j\})$ function costs $O(m^2)$ for each objective pair f_i, f_j since for all $O(m^2)$ possible pairs of solutions the resulting δ-error regarding the two objectives f_i, f_j can be computed in constant time. This $\delta_{\max}$ computation has to be computed for at most $O(k^2)$ objective pairs per iteration of the while loop. The if-statement can be executed in constant time because the computation of the maxima can be done before within line 4 without increasing the runtime asymptotically. At most $k - \mathtt{k} = O(k)$ iterations of the while loop result in the overall running time stated. □

3.5 Investigating Minimum Objective Sets

Regarding the proposed objective reduction algorithms, three main questions arise. First, what is the usefulness of these algorithms regarding concrete problems, in particular how much can the objective set be reduced? Secondly, how good are the objective sets computed by the greedy methods in comparison with the exact algorithm? And third, how does the objective reduction approach proposed here compare to the one of (Deb and Saxena, 2006)? This section provides first experimental results for these questions, whereas Sec. 3.7 shows in more detail how the algorithms can be employed in decision making on the basis of a real-world application. How the algorithms can be employed during search is part of the last chapter of this thesis.

The validation of the algorithms regarding the three questions above is done in two different scenarios. On the one hand, the indicator-based evolutionary algorithm IBEA, proposed in (Zitzler and Künzli, 2004), is used to generate Pareto front approximations for various test problems which are used as inputs for the objective reduction algorithms. Altogether four different test problems are considered: the three problems DTLZ2, DTLZ5, DTLZ7 (Deb et al., 2005), and the 0-1-knapsack problem with instances of 100, 250 and 500 items, denoted as KP100, KP250, and KP500 (Laumanns et al., 2004a). The population size μ of IBEA varies with the number k of objectives, i.e., $\mu = 100$ for $k = 5$, $\mu = 200$ for $k = 15$, and $\mu = 300$ for $k = 25$. For simplicity, only one IBEA run per problem instance is performed. Other parameters are chosen according to the standard settings of the PISA package presented in (Bleuler et al., 2003). On the other hand, we consider a random scenario where the objective values for a set of solutions are generated at random using a uniform distribution over the interval $[0, 1] \subset \mathbb{R}$. This corresponds to randomly chosen solutions of a problem with objectives, the induced total preorders of which are chosen uniform randomly from the set of all total preorders.

The reason for choosing the mentioned problems is to provide a wide range of problems with different characteristics that allows to both test the objective reduction algorithms extensively and to provide statements as general as possible. Note also that the problems have been chosen from the set of all available problems in the PISA package before the objective reduction algorithms have been applied such that the choice of the problems is not tailored towards the algorithms.

3.5.1 Investigating δ-Minimum Objective Sets

To show the potentials of our objective reduction approach, we use the exact Algorithm 4 to compute 0-minimum objective sets in the random scenario

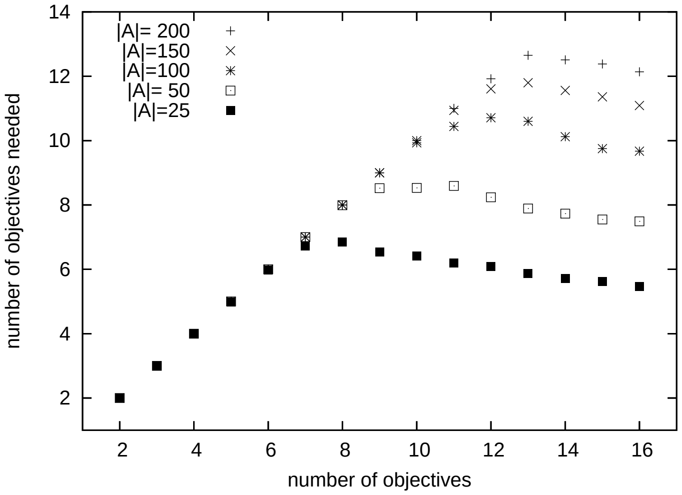


Figure 20: Size of the computed minimum sets for different number k of randomly chosen objectives and the number $|A|$ of solutions.

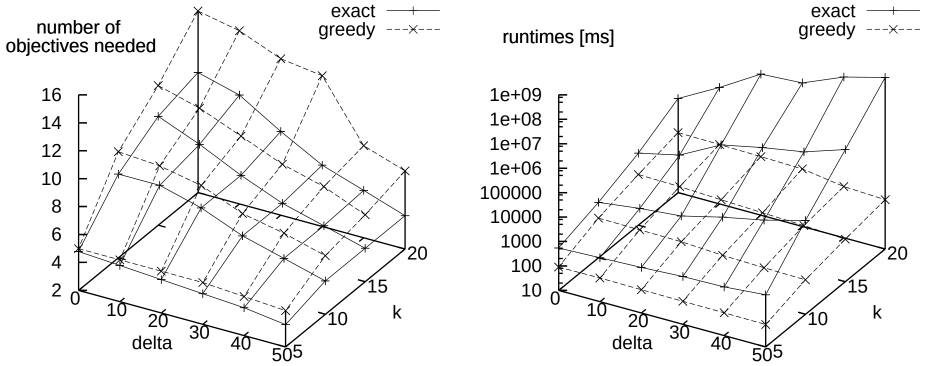


Figure 21: Comparison of the exact and the greedy algorithm for δ-MOSS on the 0-1-knapsack problem.

and δ-minimum sets for the entire search space of the 0-1-knapsack problem with 7 items[9]. The sizes of the 0-minimum objective sets in the random scenario, averaged over 100 independent random samples, are shown in Fig. 20; the sizes of the δ-minimum sets, averaged over 5 knapsack instances, can be found in the left part of Fig. 21.

Regarding the random scenario, the resulting sizes of the minimum objective subsets behave similar for all tested solution set sizes $|A|$: with increasing number of objectives, the size of the computed minimum set increases up to a specific point, depending on the number of solutions, and further decreases with more objectives. The larger the search space, i.e., the more solutions we generate, the less objectives can be omitted. With

[9] With more items, the entire search space of size $2^{\#\text{items}}$ would be too large to handle with the exact algorithm due to its running time.

200 solutions and 16 objectives, however, about 25% of the objectives can be omitted without changing the underlying dominance structure. Another observation is that the potential of the objective reduction not only relies on the number of objectives but also on the number of solutions: the smaller the set A of solutions, the more objectives can be omitted.

The investigation of the 0-1-knapsack problem indicates a similar behavior of the δ-minimum objective sets: the more objectives are used in the problem formulation, the more objectives can be omitted. Furthermore, by increasing the allowed δ-error, even more objectives can be omitted. For example, the δ-minimum sets contain only 4.4 objectives in average for the 20-objective knapsack problem if we allow an error of $\delta = 50$; instead, 10.6 objectives are needed to preserve the dominance relation with no error. However, the interpretation of the influence of δ on the results is difficult as the range of δ highly depends on the problem instances as we will see later.

As the running times, depicted in the right-hand plot of Fig. 21 for the knapsack instances and in Fig. 22 for the random scenario, indicate, the exact algorithm is not applicable for larger instances of practical size. Therefore, the greedy algorithms with their smaller running times have been developed to cope with problem instances with hundreds of solutions and a few tens of objectives in reasonable time. The following section shows that the greedy algorithm for the δ-`MOSS` problems yields comparable results to the exact algorithm. Because of that, only the greedy algorithms are used in the reminder of this thesis.

3.5.2 Investigating Approximate Objective Reduction

Before we investigate approximate objective reduction by applying the greedy algorithms, we briefly compare the exact Algorithm 4 with the greedy Algorithm 5 on δ-`MOSS`. To this end, we both use the random objective problem and the 0-1-knapsack problem with 7 items as described above and the results of which are shown in Fig. 22 and Fig. 21 respectively.

For both problems, the comparison shows the same two aspects. First, the objective sets computed with the greedy algorithm are not too large in comparison to the minimum sets computed with the exact algorithm. Nevertheless, the difference between the sizes of the objective sets computed by the two algorithms increase with more objectives. Second, the greedy algorithm is—as expected—much faster than the exact algorithm. The runtime is a large advantage of the greedy algorithm, especially for larger values of δ because the heuristic's runtime decreases with larger δ, cf. the right hand plot in Fig. 21.

Within the scenario of given Pareto front approximations for the DTLZ and knapsack instances with various numbers of objectives, we further inves-

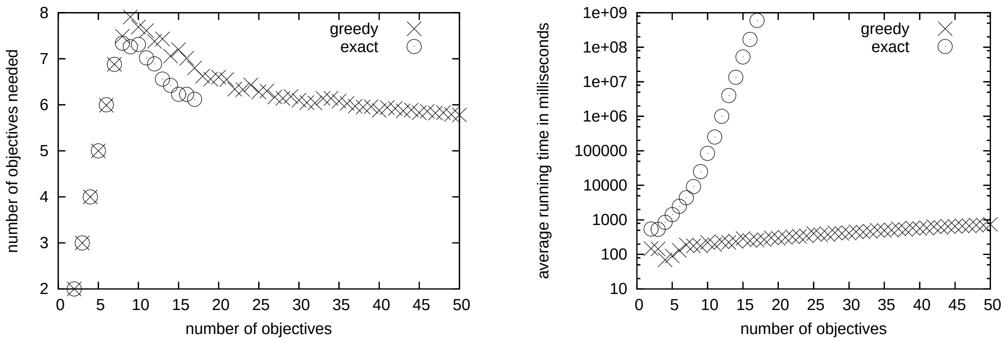


Figure 22: Comparison between the exact and the greedy algorithm for the 0-`MOSS` problem on sets with 32 solutions and random objective values. The left plot shows the size of the computed objective sets averaged over 100 runs for different number of objectives. In the right plot, the average running times of both algorithms are shown for 100 runs on each number of objectives.

tigate the ability of the greedy objective reduction methods to approximate the generalized δ-`MOSS` and `k-EMOSS` problems. To be able to compare the results for the different test problems and the varying number of objectives, we choose the δ and `k` values relatively. On the one hand, the error δ is chosen relatively to the spread of the IBEA population after 100 generations, i.e., the difference between the largest and highest objective value in the IBEA population corresponds to an error of $\delta = 1$. On the other hand, the size `k` of the objective sets is denoted relatively to the number $k \in \{5, 15, 25\}$ of objectives in the problem formulation. We choose four different δ values for the δ-`MOSS` problem $(0\%, 10\%, 20\%, 40\%)$ and three different values for `k` $(30\%, 60\%, 90\%)$. Table 1 shows the results.

With $\delta = 0$, the results for the test problems are similar to those for the random problem. Although an objective reduction is possible while preserving the preorder on the solutions, further objectives can be omitted if we allow changes of the dominance structure within the dimensionality reduction. For example, the knapsack instance with 500 items and 25 objectives does not allow an omission of objectives while preserving the dominance relation on the 300 solutions. Permitting an error of 20%, 8 objectives can be omitted, while even 12 objectives can be omitted if an error of 40% is allowed. However, the influence of a greater error δ on the resulting objective set size depends significantly on the problems. For example, only small errors yield fundamentally smaller objective sets for the DTLZ7 instances, while even a large error produces no further reduction for all DTLZ2 and DTLZ5 instances. By examining the `k-EMOSS` problem for the 18 instances in Table 1, we see similar results in a different manner. The smaller the chosen size `k` of the resulting objective sets, the larger the error in the corresponding dominance structure.

Overall, one can see that the problem instances with a higher number of objectives can be reduced further than problem instances with only 5

Table 1: Sizes (for δ-MOSS) and relative errors (for k-EMOSS) of objective subsets for different problems, computed with the Algorithm 5, and 6 respectively. For δ-MOSS, the δ value is chosen relatively to the maximum spread of the IBEA population after 100 generations; in the case of k-EMOSS, the specified size k of the output subset is denoted relatively to the problem's number of objectives.

	#objectives	#solutions	δ-MOSS				k-EMOSS		
			0%	10%	20%	40%	30%	60%	90%
KP100	5	100	5	5	5	5	0.926	0.516	0.486
KP100	15	200	11	10	10	9	0.818	0.348	0.000
KP100	25	300	13	13	13	11	0.597	0.000	0.000
KP250	5	100	5	5	5	4	0.859	0.697	0.280
KP250	15	200	11	11	10	9	0.762	0.342	0.000
KP250	25	300	12	12	12	11	0.575	0.000	0.000
KP500	5	100	5	5	5	4	0.748	0.504	0.237
KP500	15	200	15	15	14	10	0.643	0.435	0.278
KP500	25	300	25	23	17	13	0.472	0.320	0.138
DTLZ2	5	100	5	5	5	5	0.991	0.970	0.920
DTLZ2	15	200	13	13	13	13	0.942	0.891	0.000
DTLZ2	25	300	18	18	18	18	0.832	0.782	0.000
DTLZ5	5	100	5	5	5	5	0.952	0.906	0.896
DTLZ5	15	200	11	11	11	11	0.860	0.803	0.000
DTLZ5	25	300	13	13	13	13	0.820	0.000	0.000
DTLZ7	5	100	5	5	1	1	0.135	0.134	0.132
DTLZ7	15	200	10	1	1	1	0.078	0.070	0.000
DTLZ7	25	300	11	1	1	1	0.050	0.000	0.000

objectives. However, this might be due to the fact that the number of solutions increases only slightly with the number of objectives such that the entire dimensionality of the search space cannot be represented, e.g., by only 300 solutions in a 25-objective space. Furthermore, we would like to stress the exception of DTLZ7 in the considered set of test functions where probably the different scaling of the objectives allows for a large reduction of the objectives even if the error is quite small.

3.5.3 Comparison to a PCA-based Objective Reduction Approach

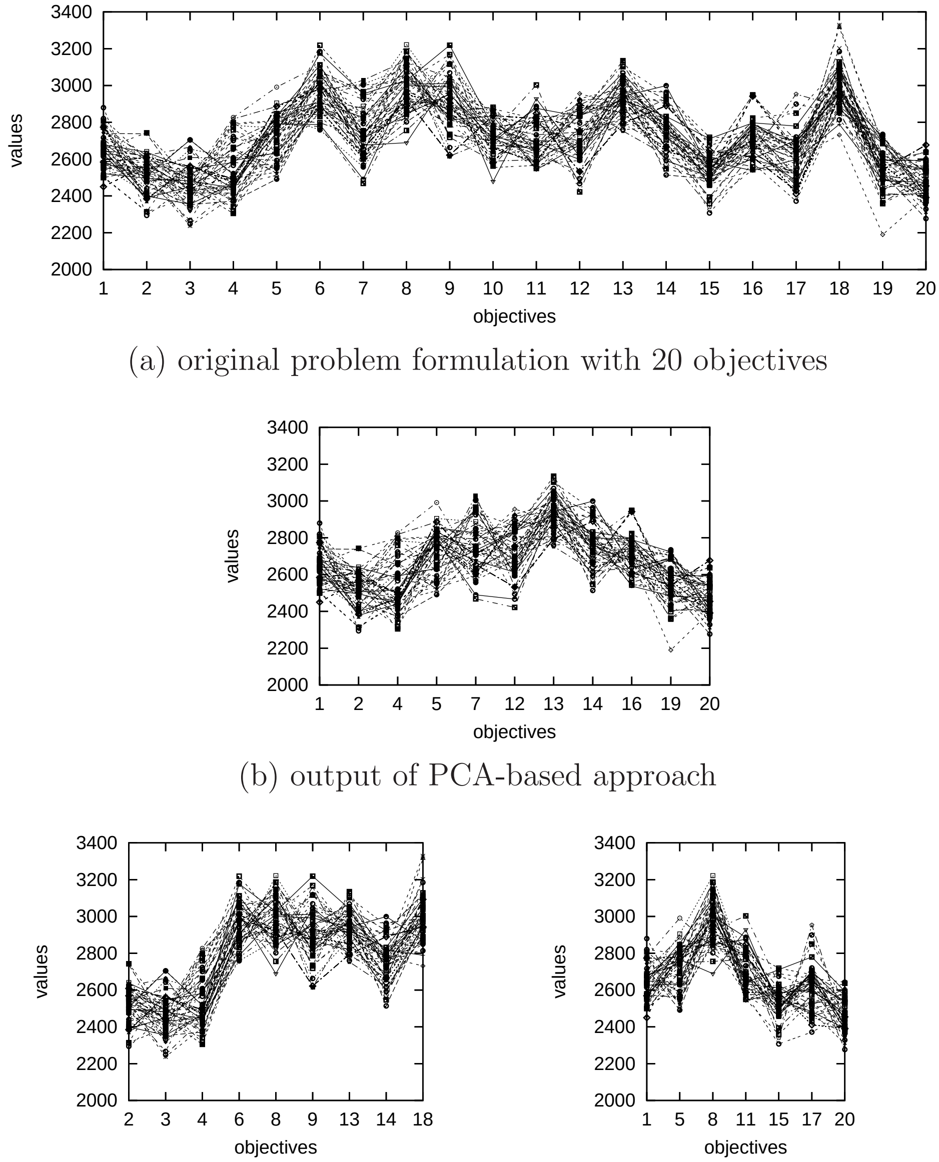
Before we come to the case of objective aggregation in the next section, we compare the proposed objective reduction approach to the method of Deb and Saxena (2005, 2006) on `k-EMOSS` for a knapsack instance with 20 objectives. We apply both methods on a Pareto front approximation for a knapsack instance with 100 items and 20 objectives, generated with an IBEA run as before (100 generations, population size 50). Since the PCA-based objective reduction method of Deb and Saxena cannot handle the `k-EMOSS` problem directly, we choose different threshold cuts (TC) such that all possible sizes of objective subsets are computed, where the TC determines the number of examined eigenvectors. Because an additional eigenvector causes either 0, 1, or 2 additional objectives in the resulting objective subset, objective subsets with 1, 5, 6, and 10 objectives cannot be generated by the PCA-based method for the considered knapsack instance. Note, that Deb and Saxena's method also performs an additional reduction of objectives using a reduced correlation matrix. Nevertheless, the method does not necessary yield, in general, δ-minimal sets, similar to our greedy algorithm.

Table 2 shows the computed objective subsets together with the absolute and relative[10] δ-errors for the objective subsets computed with the method of Deb and Saxena, the exact Algorithm 4 and the greedy Algorithm 6 for `k-EMOSS`. In addition, Table 2 presents the used TC vales for the method of Deb and Saxena and Fig. 23 provides parallel coordinates plots for the computed sets, 0-non-conflicting with the set of all objectives.

The fewer objectives are removed from the objective set, the smaller the δ-error gets for all methods similar to what we have seen before. Although the exact algorithm shows that only 7 objectives are necessary to yield no error, the other two approaches perform noticeable reductions of objectives. But since Deb and Saxena's method is not especially developed for `k-EMOSS`, the resulting objective sets cause larger errors in the dominance structure than the corresponding sets, computed with the greedy algorithm[11]. Note, that the method of Deb and Saxena yields a 0-non-conflicting subset of size 11 if one chooses the TC value as recommended in (Deb and Saxena, 2005) as 95%. This last result is especially interesting since the considered knapsack problem does not obviously have a Pareto front of lower dimension

[10] The relative error δ_{rel} is the absolute error $\delta_{\max}$ divided by the spread of the IBEA population, i.e., the same than the normalized δ-error above.

[11] The same is expected if the more recent objective reduction algorithms proposed in (Saxena and Deb, 2007) would be considered here. However, the comparison with these algorithms as well as with other more recent approaches, e.g., with the algorithms proposed in (López Jaimes et al., 2008, 2009) is not the main focus of this thesis.



(a) original problem formulation with 20 objectives

(b) output of PCA-based approach

(c) output of greedy Algorithm 6 (d) output of exact Algorithm 4

Figure 23: Visualization of the results from Table 2. The plots show the objective values for the 50 solutions computed by an IBEA run on a knapsack instance with 20 objectives. Figure (a) shows the values for the complete set of 20 objectives. The other figures show the objective subsets, 0-non-conflicting with the whole objective set, computed by the approach of Deb and Saxena (2006) (b), the greedy Algorithm 6 (c), and the exact Algorithm 4 (d).

than the original problem formulation with 20 objectives. Although Deb and Saxena applied their algorithm only to those types of problems with a lower-dimensional front and, in addition, it does not aim at preserving the dominance relation, the PCA-based approach is also applicable to more general problem scenarios and preserves the dominance structure.

Table 2: Comparison between the PCA-based approach of Deb and Saxena (2006) with the exact Algorithm 4 and the greedy Algorithm 6 for `k-EMOSS` on a Pareto front approximation of a knapsack instance with 20 objectives.

	PCA-based				`k-EMOSS` exact			`k-EMOSS` greedy		
# obj	TC	δ_{max}	δ_{rel}	objective set	δ_{max}	δ_{rel}	objective set	δ_{max}	δ_{rel}	objective set
1	-	-	-	-	552	0.9154	18	552	0.9154	18
2	0.0000-0.5410	603	1.0000	4,14*	485	0.8043	8,9	508	0.8425	6,18
3	0.5411-0.6704	546	0.9055	4,7,14	447	0.7413	6,12,15	462	0.7662	6,9,18
4	0.6705-0.7702	546	0.9055	4,14,16,19	363	0.6020	7–9,11	418	0.6932	6,9,14,18
5	-	-	-	-	289	0.4793	3,4,8,9,20	369	0.6119	4,6,9,14,18
6	-	-	-	-	129	0.2139	3–5,8,9,18	356	0.5904	2,4,6,9,14,18
7	0.7703-0.8442	466	0.7728	2,4,7,12,14,16,19	0	0.0000	1,5,8,11,15,17,20	324	0.5373	2,4,6,9,13,14,18
8	0.8443-0.9235	466	0.7728	2,4,5,7,12,14,16,19	0	0.0000	1,5,8,11,15,17,20	287	0.4760	2,4,6,8,9,13,14,18
9	0.9236-0.9472	357	0.5920	1,2,4,5,7,12,14,16,19	0	0.0000	1,5,8,11,15,17,20	0	0.0000	2–4,6,8,9,13,14,18
$\geq$11	$\geq$ 0.9473	0	0.0000	1,2,4,5,7,12–14,16,19,20	0	0.0000	1,5,8,11,15,17,20	0	0.0000	2–4,6,8,9,13,14,18

*Note, that for $0.3983 \leq TC \leq 0.5410$, the original set is $4, 7, 14$, but the final reduction using the reduced correlation matrix omits objective 7.

It is worth noting as well that in the case of 11 and more objectives, all objective reduction approaches find different objective sets that are non-conflicting with the entire objective set. Moreover, also if only a few objectives are desired in the resulting objective set, the exact and the greedy algorithm find different objective sets the δ-errors of which only differ slightly. This might indicate that there are at least two small almost equivalent objective subsets that contain most of the problem structure; however, further research is necessary here.

The parallel coordinates plots in Fig. 23 show in addition, how the objective reduction approaches reduce the amount of data, a decision maker would have to consider in an a posteriori scenario while the Pareto dominance relation among the solutions is not affected: The huge amount of objective values in the original 20-objective Pareto front approximation (Fig. 23(a)) can be reduced by more than 60% for the exact algorithm—counted in the number of the objective values shown to the decision maker (Fig. 23(d)). Also the greedy, and therefore much faster algorithm can reduce the number of objectives shown to a decision maker from 20 to 9 (Fig. 23(c)).

3.6 Objective Reduction by Aggregating Objectives

In this section, we generalize the ideas of objective reduction in the previous section by considering aggregations of several objectives and thereby make use of the fact that when aggregating objectives instead of omitting them less information is lost, i.e., the δ-error can be further decreased. In this context, the main goal can be restated as follows: find a minimum set of new objectives where each of them represents a weighted sum of a subset of the original objectives such that the dominance structure between solutions is (mainly) preserved. Clearly, this formulation also captures the omission of objectives as weights can be set to 0. Note that this idea of grouping the objectives and optimizing the weighted sums of them separately while keeping the overall nature of a multiobjective problem is not new. However, when Koski and Silvennoinen (1987) introduced this idea and applied it to several truss bar design problems, the grouping of the objectives as well as the choice of the weights have been left to the decision maker. Here, we aim at finding the weights automatically depending on a given set of solutions.

The next subsection defines the problem formally and investigates the benefit of objective aggregation in terms of the achieved δ-error and the changes in the dominance structure in general. Later on, we present a greedy algorithm to approximate the optimal solution to the problem of

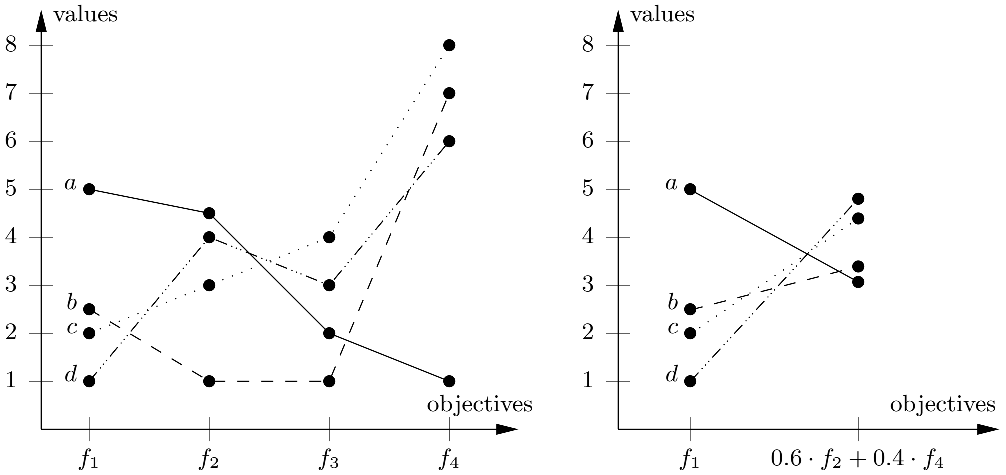


Figure 24: Example of Fig. 3: (left) original objectives; (right) original objective f_1 and aggregated objective $0.6 \cdot f_2 + 0.4 \cdot f_4$. For details, see text.

finding a minimum objective subset that preserves most of the dominance structure (Sec. 3.6.2). It works by iteratively selecting a pair of objectives that is integrated into a new objective using weighted-sum aggregation. The selection of the objective pair yielding the best δ-error can be performed exactly in an efficient way as we show in Sec. 3.6.2.2. The experimental validation in Sec. 3.6.3 indicates that especially with a large number of objectives the new method can better preserve the problem characteristics and has advantages over the proposed methods that omit objectives only.

3.6.1 Aggregation vs. Omission of Objectives

The main purpose of this section is to understand how the Pareto dominance relation changes if objectives are aggregated and what is the difference to omitting objectives. Before we start our investigation with the same example used in Chapter 2 and in the previous sections, we quickly recapitulate the result for objective omission from Sec. 2.1: the only possible changes in the dominance relation if objectives are omitted are that comparable solutions can become indifferent and incomparable solutions can become comparable or indifferent. In other words, only new comparabilities can be introduced by omitting objectives.

Example 8. *Let us consider again the example from Chapter 2, depicted in Fig. 24. We recapitulate that the omission of f_3 does not change the dominance structure but no set of two objectives induces the same Pareto dominance relation than the set of original objectives.*

We have seen that the omission of objectives introduces new comparabilities and information about the original dominance structure gets lost. As an alternative, the number of objectives can also be reduced by the aggregation of objectives and thereby keeping more of the information about the original objectives as we will see in the following. Here, we consider the simplest case of objective aggregation: the weighted sum approach. An *aggregated objective* f^{a} is thereby a linear combination of the original ones, i.e., $f_i^{\mathrm{a}} = \sum_{f_j \in \mathcal{F}} w_{i,j} \cdot f_j$, where we assume without loss of generality[12] that the non-negative weights $w_{i,j}$ $(1 \leq j \leq k)$ for each of the new objectives sum up to 1. This formalism also contains the omission of objectives as an aggregation where all weights are either 1 or 0.

Example 9. *Once again, we consider the above example in Fig. 24. As we have seen, no objective pair can preserve the weak Pareto dominance relation entirely. However, if we allow the aggregation of objectives, a set of two aggregated objectives can be found that preserves the dominance relation completely. The right hand plot of Fig. 24 shows the parallel coordinate plot for such a set* $\{f_1, 0.6 \cdot f_2 + 0.4 \cdot f_6\}$. *All four solutions are still pairwisely incomparable, i.e., the original dominance relation is preserved.*

When aggregating objectives, the only change in the dominance structure is the introduction of comparabilities—exactly as with the omission of objectives. If for a pair $\vec{a}, \vec{b} \in X$ of solutions, $\vec{a} \preceq_{\mathcal{F}} \vec{b}$ holds, i.e., $f_i(\vec{a}) \leq f_i(\vec{b})$ for all $1 \leq i \leq k$, $\vec{b}$ will stay weakly dominated with respect to any set of aggregated objectives if the weights are all positive and the objectives are coming from $\mathcal{F}$. If, however, for a pair $\vec{a} \not\preceq_{\mathcal{F}} \vec{b}$ holds, the aggregation can introduce the domination of $\vec{a}$ if for all newly introduced objectives $f_i^{\mathrm{a}}(\vec{a}) \not> f_i^{\mathrm{a}}(\vec{b})$ holds due to the choice of the weights.

In Sec. 3.3, given a set $A \subseteq X$ of solutions and a $\mathtt{k} \in \mathbb{N}$, the `k-EMOSS` problem was introduced as the problem of finding the best objective *subset* $\mathcal{F}' \subseteq \mathcal{F}$ of size $\mathtt{k}$ that minimizes the resulting (maximum) δ-error. Here, we generalize this problem to finding the best set of $\mathtt{k}$ *aggregated* objectives such that the resulting δ-error is minimized and denote it as the *Optimal Aggregation Problem*:

Definition 12. *Given a* $\mathtt{k} \in \mathbb{N}$, *a set* $A \subseteq X$ *of solutions, and a chosen* δ*-error (either* $\delta = \delta_{max}$ *or* $\delta = \delta_{avg}$*). Let* $\mathcal{F} = \{f_1, \ldots, f_k\}$ *be the set of all objectives. The* `OPTIMAL AGGREGATION PROBLEM`, *or* `OAP` *for short, with respect to* δ *is defined as follows: Find a set of weight vectors* $W = \{\vec{w} = (w_1, \ldots, w_k) \in [0,1]^k \mid \sum_{1 \leq i \leq k} w_i = 1\}$ *with* $|W| = k$ *such that the* δ*-error of the set of aggregated objective vectors*

$$\delta(A, \bigcup_{(w_1,\ldots,w_k) \in W} \{\sum_{i=1}^{k} w_i f_i\}, \mathcal{F})$$

is minimal.

[12] If $\sum_{f_j \in \mathcal{F}} w_{i,j} = 1$ is not given, we can easily ensure the property by using the weights $w_{i,j} / \sum_{f_j \in \mathcal{F}} w_{i,j}$ instead which does not change the Pareto dominance relation.

In the remainder of this thesis, we will refer to the problem of finding the optimal aggregation with respect to the maximum error as $\texttt{OAP}_{\texttt{max}}$ and denote the optimal aggregation problem with respect to the average δ-error as $\texttt{OAP}_{\texttt{avg}}$. For both problems, we propose a greedy heuristic in the following.

3.6.2 A Greedy Heuristic for Finding the Best Aggregation

As the special case of `k-EMOSS` is already $\mathcal{NP}$-hard (see Sec. 3.3), the generalized problems $\texttt{OAP}_{\texttt{max}}$ and $\texttt{OAP}_{\texttt{avg}}$ are also at least $\mathcal{NP}$-hard and therefore too complex to solve them exactly under the assumption $\mathcal{P} \neq \mathcal{NP}$. To approximate the $\texttt{OAP}_{\texttt{max}}$ and $\texttt{OAP}_{\texttt{avg}}$ problems, we, therefore, propose a greedy approximation algorithm the idea of which is to iteratively aggregate objective pairs until the desired number of objectives is reached—in other words, the algorithm resembles the approach of hierarchical clustering.

3.6.2.1 Main Procedure

Algorithm 8 shows the pseudo code of the aggregation procedure. It carries a set W of weight vectors $w = (w_1, \ldots, w_k) \in \mathbb{R}^k_{\geq 0}$ along which, at any time, is a representation of the set of aggregated objectives. Each weight vector $w = (w_1, \ldots, w_k)$ corresponds to the aggregated objective $f^a = \sum_{i=1}^k w_i f_i$. Starting with the original objectives, i.e., with k weight vectors containing exactly one 1-entry and otherwise zeros (line 2), the δ-error equals 0. Then, in each step of the while-loop, where $|W|$ denotes the number of elements in W, the objective pair the aggregation of which yields the smallest error is aggregated and the corresponding weight vectors and the δ-error are adjusted. To this end, for each objective pair, represented by the weight vectors in W, the weight α when optimally aggregating this objective pair is computed (line 7). Optimally in this case, means that the δ-error is minimized when deleting the objective pair f_q, f_r and adding a new objective $f_{\text{new}} = \alpha f_q + (1-\alpha) f_r$. In terms of the corresponding weight vectors, the new set of weight vectors W' after the aggregation of f_q and f_r can be written as the old set W without $\vec{q}$ and $\vec{r}$ but with the new $w_{\text{new}} = \alpha\vec{q} + (1-\alpha)\vec{r}$ (line 8). How the optimal weight α can be computed in the function $\texttt{aggregateOptimally}(A, \mathcal{F}', \mathcal{F})$ such that the δ-error is minimized will be explained in detail in the following.

3.6.2.2 Optimally Aggregating Two Objectives

Assume, without loss of generality, that we want to aggregate the two objectives f_1 and f_2 in a set of objectives $\mathcal{F}$ optimally, i.e., we have to

Algorithm 8 A Greedy Heuristic for the problems $\texttt{OAP}_{\texttt{max}}$ and $\texttt{OAP}_{\texttt{avg}}$.

```
Require: solution set $A \subseteq X$ with set of objectives $\mathcal{F} = \{f_1, \ldots, f_k\}$;
    number of desired objectives $\texttt{k} \geq 1$;
    type of $\delta$-error: type $\in \{\max, \text{avg}\}$
Init:
$W = \{\vec{w} \in \mathbb{R}^k_{\geq 0} \mid \sum_{i=1}^k w_i = 1 \wedge \exists i \in \{1, \ldots, k\} : w_i = 1\}$
$\delta = 0$
while $|W| > \texttt{k}$ do
    $\delta_{\text{best}} = +\infty$
    for all $\vec{q}, \vec{r} \in W, \vec{q} \neq \vec{r}$ do
        $\alpha = \texttt{aggregateOptimally}(\vec{q}, \vec{r}, A, W, \mathcal{F})$
        $W' = (W \setminus \{\vec{q}, \vec{r}\}) \cup \{\alpha\vec{q} + (1-\alpha)\vec{r}\}$
        $\delta' = \delta_{\text{type}}\left(A, \bigcup\limits_{(w'_1,\ldots,w'_k \in W')} \left\{\sum\limits_{i=1}^{k} w'_i f_i\right\}, \mathcal{F}\right)$
        if $\delta' \leq \delta_{\text{best}}$ then
            $W_{\text{best}} = W'$
            $\delta_{\text{best}} = \delta'$
        end if
    end for
    $W = W_{\text{best}}$
    $\delta = \delta_{\text{best}}$
end while
return $(W, \delta)$
```

find the weight $\alpha \in [0,1]$ such that the δ-error between the original objective set $\mathcal{F}$ and the new set $\mathcal{F}' = \mathcal{F} \setminus \{f_1, f_2\} \cup f_{\text{new}}$ is minimal where $f_{\text{new}} = \alpha f_1 + (1-\alpha) f_2$ is the new aggregated objective. In the following, we will use for the current set of objectives excluding the two objectives that have to be aggregated the term *remaining objectives* and call the objective set $\mathcal{F}'_{\text{rem}} := \mathcal{F} \setminus \{f_1, f_2\}$.

The idea behind the function $\texttt{aggregateOptimally}(A, \mathcal{F}'_{\text{rem}} \cup \{f_{\text{new}}\}, \mathcal{F})$ is to determine for each solution pair $\vec{a}, \vec{b} \in A$ a function $\Delta_{(\vec{a},\vec{b})} : [0,1] \to \mathbb{R}_{\geq 0}$ that gives for each possible weight $\alpha \in [0,1]$ the δ-error that is introduced if the objective pair is aggregated to the new objective f_{new}. Figure 27 gives some examples how this $\Delta_{(\vec{a},\vec{b})}$ function can look like. How this function $\Delta_{(\vec{a},\vec{b})}$ can be computed is explained later on in detail. Assuming we know the functin $\Delta_{(\vec{a},\vec{b})}$ for all possible $\vec{a}, \vec{b} \in X$, the best weight α_{opt} over all solution pairs and the corresponding δ-error δ_{opt} can then be computed as

$$\begin{aligned}\alpha_{\text{opt}} &= \operatorname*{argmin}_{\alpha\in[0,1]} \Delta_A(\alpha)\\ \delta_{\text{opt}} &= \min_{\alpha\in[0,1]} \Delta_A(\alpha)\end{aligned}$$

where, depending on the problem to solve,

$$\Delta_A(\alpha) = \Delta_{A,\max}(\alpha) = \max_{\vec{a},\vec{b}\in A} \Delta_{(\vec{a},\vec{b})}(\alpha)$$

for $\texttt{OAP}_{\texttt{max}}$ and

$$\Delta_A(\alpha) = \Delta_{A,\text{avg}}(\alpha) = \frac{1}{|A|(|A|-1)} \sum_{\vec{a},\vec{b}\in A} \Delta_{(\vec{a},\vec{b})}(\alpha)$$

for $\texttt{OAP}_{\texttt{avg}}$. In other words, if the error depending on the weight α is known for all solution pairs, the maximum δ-error $\Delta_{A,\max}$ is computed as the maximum over all solution pairs whereas the average δ-error $\Delta_{A,\text{avg}}$ is the δ-error averaged over all solution pairs. The optimal weight is then chosen in the best weight interval, i.e., the weight interval with the minimal δ-error, see again Fig. 27 for an illustration. We would like to mention already here, that the choice of α_{opt} is not unique—most of the time it is rather an *optimal weight interval* than a single value. We will discuss the actual choice of α_{opt} in the interval with smallest δ-error later and decide to fix the center of the optimal interval as the optimal weight for the moment.

Now, we explain how to determine the function $\Delta_{(\vec{a},\vec{b})}$. To this end, we fix $\vec{a}$ and $\vec{b}$ and distinguish between two situations:

(a) $\delta(\{\vec{a},\vec{b}\}, \mathcal{F}'_{\text{rem}}, \mathcal{F}) = 0$, i.e., even if we omit the objectives f_1 and f_2, we make no error. In this case, α can be chosen arbitrarily in $[0,1]$ and the δ-error is 0, i.e., $\forall\alpha \in [0,1] : \Delta_{(\vec{a},\vec{b})}(\alpha) = 0$.

(b) $\vec{a}$ and $\vec{b}$ are standing with respect to the remaining objectives $\mathcal{F}'_{\text{rem}}$ in a different relationship than with respect to the entire objective set $\mathcal{F}$, i.e., it depends on the choice of α which error we make. In this case (i) $\vec{a} \preceq_{\mathcal{F}'_{\text{rem}}} \vec{b}$ but $\vec{a} \not\preceq_{\mathcal{F}} \vec{b}$ and/or (ii) $\vec{b} \preceq_{\mathcal{F}'_{\text{rem}}} \vec{a}$ but $\vec{b} \not\preceq_{\mathcal{F}} \vec{a}$ can hold.

Assume first that (i) holds but not (ii). In this case, we need to choose α such that $\vec{a} \not\preceq_{\mathcal{F}'} \vec{b}$ holds with respect to the new objective set $\mathcal{F}' = \mathcal{F}'_{\text{rem}} \cup \{f_{\text{new}}\}$ to make no error, or in other words, we need to ensure that $f_{\text{new}}(\vec{a}) > f_{\text{new}}(\vec{b})$. This inequality can be rewritten as

$$\begin{aligned}&f_{\text{new}}(\vec{a}) > f_{\text{new}}(\vec{b})\\ &\qquad\Longleftrightarrow \alpha f_1(\vec{a}) + (1-\alpha) f_2(\vec{a}) > \alpha f_1(\vec{b}) + (1-\alpha) f_2(\vec{b})\\ &\qquad\Longleftrightarrow \alpha(f_1(\vec{a}) - f_1(\vec{b}) + f_2(\vec{b}) - f_2(\vec{a})) > f_2(\vec{b}) - f_2(\vec{a}) \qquad (3.1)\end{aligned}$$

yielding—depending on the precise objective values of $\vec{a}$ and $\vec{b}$—an interval $S \subseteq [0,1]$ of α, where the δ-error is zero[13]. For all other choices of α, $f_{\text{new}}(\vec{a}) \leq f_{\text{new}}(\vec{b})$ according to Equation 3.1 and together with the fact that $\vec{a} \preceq_{\mathcal{F}'_{\text{rem}}} \vec{b}$ but $\vec{a} \not\preceq_{\mathcal{F}} \vec{b}$, the relation between $\vec{a}$ and $\vec{b}$ induced by $\mathcal{F}'_{\text{rem}}$ and $\mathcal{F}'_{\text{rem}} \cup \{f_{\text{new}}\}$ is the same and by aggregating f_1 and f_2, we make an error that is the same as if we would omit f_1 and f_2 entirely. Therefore

$$\Delta^{(i)}_{(\vec{a},\vec{b})} = \begin{cases} 0 & \text{if } \alpha \in S \\ \delta_{\text{max/avg}}(\{\vec{a},\vec{b}\}, \mathcal{F}'_{\text{rem}}, \mathcal{F}) & \text{else} \end{cases}$$

where the δ-error in the else case again depends on the problem to solve. Examples of this function for different solution pairs can be seen in Fig. 27.

The case where (ii) but not (i) holds follows analogously by changing the roles of $\vec{a}$ and $\vec{b}$ and yields the similar function

$$\Delta^{(ii)}_{(\vec{a},\vec{b})} = \begin{cases} \delta_{\text{max/avg}}(\{\vec{a},\vec{b}\}, \mathcal{F}'_{\text{rem}}, \mathcal{F}) & \text{if } \alpha \in S \\ 0 & \text{else} \end{cases} .$$

What remains to investigate is the case where both (i) and (ii) hold. In this case, we make an error with any choice of α: either $\vec{a} \preceq_{\{f_{\text{new}}\}} \vec{b}$ and the error is $\Delta^{(i)}_{(\vec{a},\vec{b})}$ as in case (i) or $\vec{b} \preceq_{\{f_{\text{new}}\}} \vec{a}$ and the error is $\Delta^{(ii)}_{(\vec{a},\vec{b})}$. In case both $\vec{a} \preceq_{\{f_{\text{new}}\}} \vec{b}$ and $\vec{b} \preceq_{\{f_{\text{new}}\}} \vec{a}$ hold, i.e., if $\vec{a}$ and $\vec{b}$ are indifferent with respect to f_{new}, the resulting error is the maximum of both errors $\Delta^{(i)}_{(\vec{a},\vec{b})}$ and $\Delta^{(ii)}_{(\vec{a},\vec{b})}$. Thus,

$$\Delta_{(\vec{a},\vec{b})}(\alpha) = \max\{\Delta^{(i)}_{(\vec{a},\vec{b})}, \Delta^{(ii)}_{(\vec{a},\vec{b})}\} .$$

Example 10. *The left hand plot of Fig. 25 shows two solutions $\vec{a}$ and $\vec{b}$ with two objectives f_1, f_2, the aggregation of which to the new objective $f_{new} = \alpha f_1 + (1-\alpha) f_2$ causes an error for all choices of $\alpha \in [0,1]$. The reason why we make an error for all choices of α is that the two solutions become comparable for any choice of α although they are incomparable with respect to the original objectives. The resulting dominance relation changes from $\vec{b} \preceq_{\{f_{new}\}} \vec{a}$ to $\vec{a} \preceq_{\{f_{new}\}} \vec{b}$ when α is increased, cf. the middle plot of Fig. 25. The value α_c for which both solutions are indifferent (the intersection point α_c in the middle plot of Fig. 25), can be computed when the inequality in*

[13] Such an interval obviously exists if $f_1(\vec{a}) - f_1(\vec{b}) + f_2(\vec{b}) - f_2(\vec{a}) \neq 0$ when the inequality in (3.1) can be easily solved by dividing by $(f_1(\vec{a}) - f_1(\vec{b}) + f_2(\vec{b}) - f_2(\vec{a}))$. In case that $f_1(\vec{a}) - f_1(\vec{b}) + f_2(\vec{b}) - f_2(\vec{a}) = 0$, i.e., if $f_1(\vec{a}) - f_1(\vec{b}) = f_2(\vec{a}) - f_2(\vec{b})$, the objective value distances between $\vec{a}$ and $\vec{b}$ are the same for f_1 and f_2 and therefore either $f_{\text{new}}(\vec{a}) < f_{\text{new}}(\vec{b})$ or $f_{\text{new}}(\vec{b}) > f_{\text{new}}(\vec{a})$ with respect to the new aggregated objective f_{new} for all choices of α. Thus, the interval for α is either empty (if $f_{\text{new}}(\vec{a}) < f_{\text{new}}(\vec{b})$ for all choices of α and we always make an error) or the entire interval $[0,1]$ if $f_{\text{new}}(\vec{a}) > f_{\text{new}}(\vec{b})$ for all choices of α and we never make an error by the aggregation of f_1 and f_2.

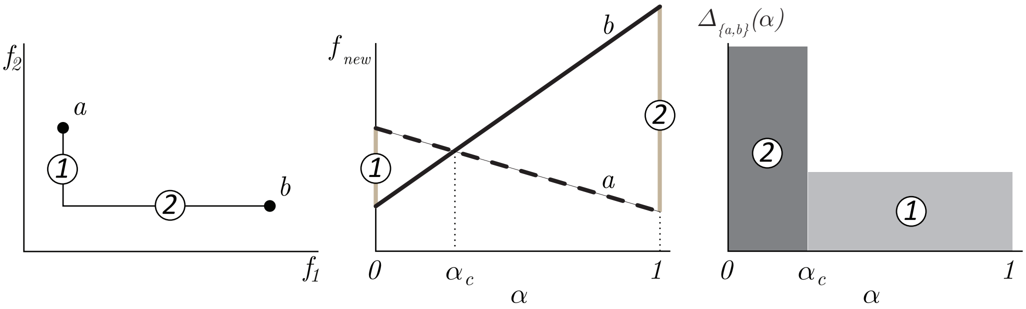


Figure 25: Illustration of the aggregation of two objectives that is optimal for two solutions $\vec{a}$ and $\vec{b}$: (left) original objectives with distances between $\vec{a}$ and $\vec{b}$ in f_1 (②) and f_2 direction (①); (middle) resulting aggregated objective values $f_{\text{new}} = \alpha f_1 + (1-\alpha) f_2$ for all choices of $0 \leq \alpha \leq 1$; (right) corresponding $\delta_{\max}$-error for all choices of α where the dark gray area corresponds to a choice of α such that $\vec{b}$ is wrongly assumed to weakly dominate $\vec{a}$ and the error equals the distance ② and the light gray area corresponds to the case where $\vec{a}$ weakly dominates $\vec{b}$ with respect to the aggregated objective and the error is ①.

Equation 3.1 is changed to an equality. The rightmost plot of Fig. 25 shows the resulting δ-errors $\Delta_{\{\vec{a},\vec{b}\}}$ for each choice of α (indicated as "①" and "②" in Fig. 25 and corresponding to the distances $f_2(\vec{a}) - f_2(\vec{b})$ (①) and $f_1(\vec{b}) - f_1(\vec{a})$ (②) in the left hand plot of Fig. 25).

Note that in the biobjective Example 10, the set of remaining objectives is empty. In the following example, the set of three objectives has to be reduced by one objective as it would be the case within a run of Algorithm 8, i.e., the set of remaining objectives always contains one objective. In addition, we consider more than one solution pair here.

Example 11. *Consider four solutions $\vec{a}, \vec{b}, \vec{c}, \vec{d} \in A$ with the objective vectors $f(\vec{a}) = (1,8,4)$, $f(\vec{b}) = (6,2,7)$, $f(\vec{c}) = (3,4,4)$, and $f(\vec{d}) = (0,7,7)$ as depicted in Fig. 26. In the following, we illustrate the progress of Algorithm 8 if the average δ-error has to be minimized and the objective set has to be reduced to* $\mathtt{k} = 2$ *objectives. After initialization, the algorithm computes the optimal choice of α for each objective pair. Figure 27 shows the computation for the objective pair (f_1, f_2) in more detail: for each solution pair $\vec{x}, \vec{y}$, the $\Delta_{(\vec{x},\vec{y})}$ function is computed as described above (left plot of Fig. 27) and the average over all these functions is taken as the overall error function $\Delta_A(\alpha)$ (right hand plot of Fig. 27). The optimal choice of α yielding a minimal δ-error has to lie in the interval indicated by the arrow in the right hand plot of Fig. 27. For the other two objective pairs (f_1, f_3) and (f_2, f_3), similar Δ_A functions can be obtained, see Fig. 28. The best intervals with an error of $\delta_{avg} = 0$ can be obtained when aggregating f_1 and f_2 with an $\alpha \in (0.5, 0.65)$ and when aggregating f_2 and f_3 with an $\alpha \in (0.6, 0.65)$ by simply choosing the intervals with the smallest δ-error.*

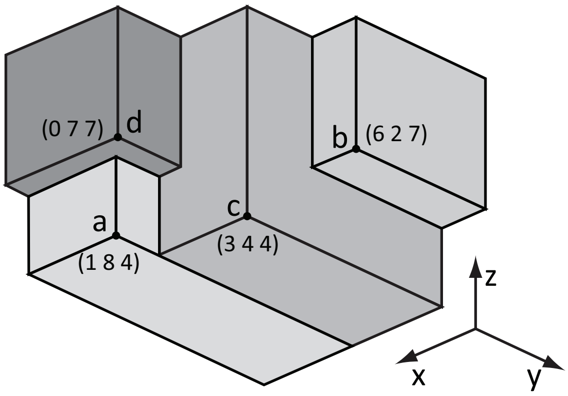


Figure 26: Illustration of the four points $\vec{a}, \vec{b}, \vec{c}, \vec{d}$ of Example 11.

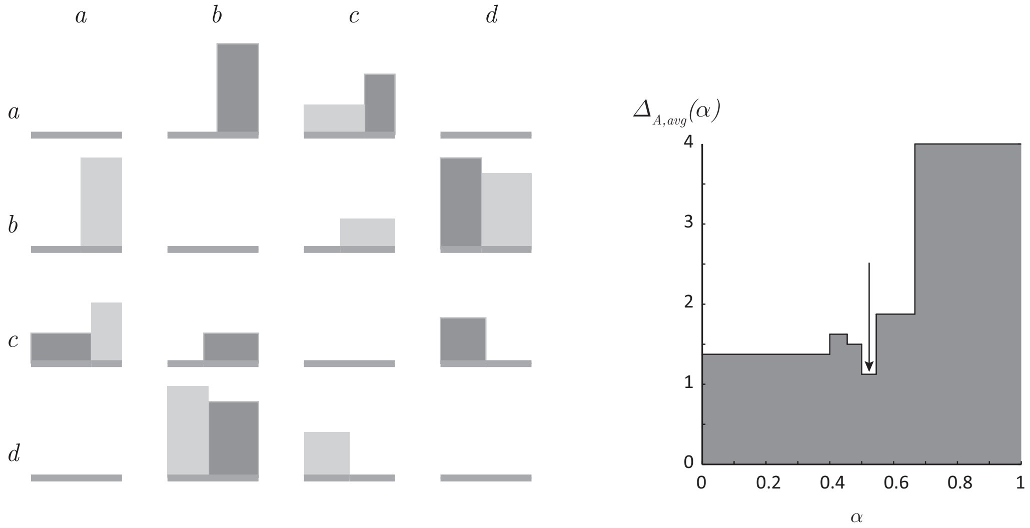


Figure 27: Computation of δ-errors dependent on the weight for all solution pairs (left) and corresponding δ-error averaged over all solution pairs (right) if the first two objectives in Example 11 are aggregated. The arrow in the right hand plot indicates the optimal weight interval.

Note that it is not specified in Algorithm 8 how the α value has to be chosen within an optimal interval. Although each choice yields the same δ-error in the current aggregation step, the choice within the optimal interval might influence the errors we make when aggregating other objectives in a future step of the algorithm. Therefore, we compare different strategies experimentally in the following.

3.6.3 Experimental Validation

After presenting the aggregation heuristic for the $\texttt{OAP}_{\texttt{max}}$ and $\texttt{OAP}_{\texttt{avg}}$ problems, three main questions remain open: (i) Is it important how to choose the weight within the optimal interval found by the aggregation heuristics?

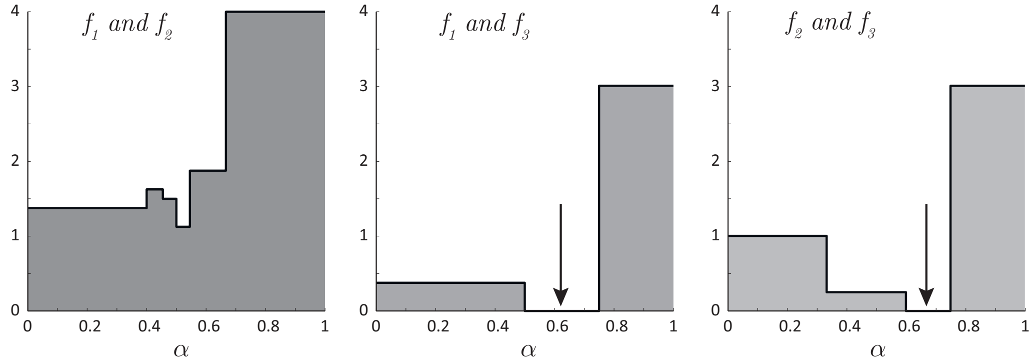


Figure 28: Illustration of the δ-errors averaged over all solutions for the objective pairs f_1, f_2 (left), f_1, f_3 (middle), and f_2, f_3 (right) of Example 11. The optimal intervals are indicated by arrows.

(ii) How much can the δ-error be reduced by the new aggregation heuristics in comparison to simply omitting objectives? and (iii) What can be gained by reducing the number of objectives during the search? While the third question will be tackled separately in the last chapter of this thesis, this section is addressing the first two experimentally. In the last section of this chapter, we apply the proposed objective reduction approaches afterwards to a real-world problem of optimizing a radar waveform to show their usefulness with respect to decision making.

The Influence of Different Weight Choices Within the Optimal Interval

As discussed above, the weight choice within the optimal interval found by the proposed aggregation heuristics might have an influence on the overall δ-error although in the current step, all choices lead to the same δ-error. To investigate the influence of different weight choices on the overall resulting error, we compare four variants of the greedy $\texttt{OAP}_{\texttt{max}}$ heuristic: If the optimal weight interval and the corresponding objective pair is found, we either choose the weight in the middle of this interval (variant “CENTER”), uniformly at random within the interval (“UNIFORM”), or as the center of the left (“LEFT”) or right (“RIGHT”) half of the interval.

Settings To perform the comparison, we created $3 \cdot 51$ random instances by choosing the objective values uniformly at random in $[0, 1]$: 51 solution sets of 50 solutions with 4 objectives, 51 sets of 100 solutions with 6 objectives, and 51 sets of 200 solutions with 8 objectives. These instances have been handed over to the greedy $\texttt{OAP}_{\texttt{max}}$ algorithm that had to reduce the objective set to 25 and 50 percent of the original objectives[14]. The resulting maximum δ-errors have been ranked for all four variants "CENTER", "UNIFORM", "LEFT", and "RIGHT" (rank 1: best, rank 4: worst) and then compared by means of the Kruskal-Wallis test with the additional Conover-Inman procedure for multiple comparisons as described in (Conover, 1999) on pages 288–290. All tests have been performed on the basis of a p-value of 0.05.

Results Without any exception, no significant difference between the algorithm variants could be observed. Apparently, the greedy property of the algorithm, i.e., not looking ahead to weight choices in future steps of the algorithm, does not allow for different behavior between the four variants. In other words, a significantly better choice in one step might be outweighed by a worse choice in one of the next steps and vice versa. Since the four variants do not show significant differences, we decided to use the "CENTER" variant exclusively in the following due to its slightly faster implementation.

Comparison Between Aggregation and Omission

To experimentally validate the proposed aggregation heuristics for the $\texttt{OAP}_{\texttt{max}}$ and $\texttt{OAP}_{\texttt{avg}}$ problems, we compare them with the `k-EMOSS` heuristic in Algorithm 6.

Settings For the comparison, Pareto front approximations of 18 test problem instances have been generated similar to the experiments in Sec. 3.5 by running the indicator-based evolutionary algorithm IBEA, proposed in (Zitzler and Künzli, 2004), for 100 generations with the standard settings of the PISA package (Bleuler et al., 2003). The only parameter that changed between the test cases was the population size which was chosen as 100 for the 5-objective problems and as 200 for the 15-objective problems. In addition to the DTLZ2, DTLZ3, DTLZ7 (Deb et al., 2005)[15], WFG3, WFG6, WFG8 (Huband et al., 2006)[16] and three instances of the 0-1-knapsack problem with 100 (KP100), 250 (KP250), and 500 (KP500) items (Laumanns et al., 2004a) with 5 and 15 objectives each, we also considered two instances of a network processor design problem called EXPO (Künzli et al., 2004) with 3 and 4 objectives. The Pareto front approximations for the EXPO instances had 43 solutions and 143 solutions respectively. Also here, the choice of the used test functions is somehow arbitrary, with the idea of providing a wide range of problems, and which is not tailored towards the algorithms.

[14] More precisely, to 1 and 2 objectives for the 4-objective instances, to 1 and 3 for the 6-objective instances, and to 2 and 4 for the 8-objective instances.

[15] The number of decision variables has been set to 250.

[16] For all WFG problems, the number of decision variables has been also fixed to 250 and the number of position variables has been chosen to 168 and the number of distance variables to 82 according to the recommendations in (Huband et al., 2006) to be able to keep the numbers constant over all numbers of objectives.

Based on these test instances, the three considered greedy algorithms for the `k-EMOSS`, the $\mathtt{OAP}_{\mathtt{max}}$, and the $\mathtt{OAP}_{\mathtt{avg}}$ problem respectively, were used to reduce the objectives to 90%, 60%, and 30% of the number of original objectives, i.e., to 1, 3, and 4 for the 5-objective problems and to 4, 9, and 13 for the 15-objective problems. The EXPO instances had to be reduced to 1 and 2 (3-objective) and to 1, 2, and 3 objectives (4-objective) respectively. Table 3 shows the resulting normalized[17] $\delta_{\max}$-errors for all three algorithms and, in addition, the average δ-error for the aggregation heuristics proposed in Sec. 3.6.2. Note that a δ-value of 0.0000 in Table 3 refers to no errors whereas a $*$ is used to indicate an error that is smaller than 0.0005 but that is not exactly 0.

Results For all considered problem instances, except when the 15-objective DTLZ2 problem is reduced to 4 objectives ($\lfloor 15 \cdot 30\% \rfloor = 4$), the aggregation heuristic optimizing the maximum error yields lower or the same errors than the heuristic that omits objectives, cf. Table 3. Therefore, we conclude that the error can, in general, be decreased for the same number of objectives when aggregation is allowed. For example, the preservation of the entire dominance relation ($\delta = 0$) can be achieved for the reduction to 60% of the original objectives for 4 of the 12 DTLZ and knapsack problem instances if aggregation is allowed whereas the `k-EMOSS` heuristic cannot find objective subsets of this sizes without making an error. Note that the only case where the omission of objectives yields a smaller error than both aggregation heuristics occurs for a reduction to 30%, i.e., a small number of objectives. This might be due to the fact that the greedy heuristic omitting objectives creates the objective subset by *adding* objectives greedily instead of reducing the number of objectives step-by-step as the greedy aggregation heuristics do.

As expected, the aggregation heuristic optimizing the average error performs better with respect to the average error and worse with respect to the maximum error in most cases compared to the heuristic optimizing the maximum error. The next section will show that a low average δ-error might be beneficial when visualizing solution sets of many-objective problems.

Regarding the absolute δ-errors and the difference between the maximum and the average δ-errors, we can conclude that the maximum δ-error is usually caused by a single or a small number of solution pairs and a high maximum value does not coincide with a large average error as can be seen in Table 3. An average error that is close to $1/|A|$ (i.e., 1/100 or 1/200 in Table 3) together with a high or medium maximum error therefore indicates that the reduced objective set induces only for a few solution pairs the wrong Pareto dominance relation among the solutions.

[17] The δ-errors have been normalized to the objective values of each instance such that the difference between the highest and lowest objective value equals 1 for every objective.

Table 3: Comparison of the aggregation heuristics with the greedy `k-EMOSS` heuristic in Algorithm 6. The table entries show for the greedy `k-EMOSS` algorithm the maximum δ-errors as in Table 1 whereas for the aggregation heuristics both the maximum (left of slash) and the average (right of slash) δ-errors are shown.

	# objectives	# solutions	greedy `k-EMOSS`			greedy aggregation maximum δ-error			greedy aggregation average δ-error		
			30%	60%	90%	30%	60%	90%	30%	60%	90%
KP100	5	100	.9263	.5164	.4860	.7829/.2659	.3781/.0072	.2576/.0003	1.000/.2649	.4911/.0060	.2576/.0003
KP100	15	200	.8180	.3483	.0000	.7705/.0209	.0000/.0000	.0000/.0000	.8977/.0130	.0000/.0000	.0000/.0000
KP250	5	100	.8588	.6967	.2797	.8588/.3149	.2758/.0013	.1189/.0001	.8156/.3049	.6300/.0014	.1189/.0000
KP250	15	200	.7622	.3421	.0000	.6921/.0200	.0000/.0000	.0000/.0000	.7807/.0090	.0000/.0000	.0000/.0000
KP500	5	100	.7484	.5041	.2370	.6829/.1924	.3638/.0117	.2079/.0028	.6735/.1934	.3773/.0074	.2370/.0018
KP500	15	200	.6425	.4350	.2775	.4333/.0297	.1991/.0012	.0866/.0000	.5691/.0175	.2608/.0008	.1800/.0000
DTLZ2	5	100	.9909	.9699	.9202	.9909/.5876	.7928/.0113	.6371/.0019	1.000/.5592	.8618/.0091	.6541/.0010
DTLZ2	15	200	.9418	.8910	.0000	.9517/.0524	.4044/ *	.0000/.0000	.9823/.0119	.4044/ *	.0000/.0000
DTLZ5	5	100	.9523	.9062	.8958	.9368/.5324	.6323/.0077	.4771/.0010	.9794/.5171	.8017/.0077	.4897/.0007
DTLZ5	15	200	.8601	.8030	.0000	.8549/.0226	.0000/.0000	.0000/.0000	.9286/.0083	.0000/.0000	0.000/.0000
DTLZ7	5	100	.1353	.1335	.1321	.1353/.1121	.1222/.0003	.0000/.0000	.1558/.1116	.1233/.0002	.0000/.0000
DTLZ7	15	200	.0778	.0700	.0000	.0748/.0001	.0000/.0000	.0000/.0000	.0778/.0001	.0000/.0000	.0000/.0000
WFG3	5	100	.6611	.0000	.0000	.6611/.2402	.0000/.0000	.0000/.0000	.6611/.2402	.0000/.0000	.0000/.0000
WFG3	15	200	.0000	.0000	.0000	.0000/.0000	.0000/.0000	.0000/.0000	.0000/.0000	.0000/.0000	.0000/.0000
WFG6	5	100	.1329	.0000	.0000	.1329/.0490	.0000/.0000	.0000/.0000	.1329/.0490	.0000/.0000	.0000/.0000
WFG6	15	200	.0000	.0000	.0000	.0000/.0000	.0000/.0000	.0000/.0000	.0000/.0000	.0000/.0000	.0000/.0000
WFG8	5	100	.5636	.0000	.0000	.5636/.1783	.0000/.0000	.0000/.0000	.5636/.1741	.0000/.0000	.0000/.0000
WFG8	15	200	.0000	.0000	.0000	.0000/.0000	.0000/.0000	.0000/.0000	.0000/.0000	.0000/.0000	.0000/.0000
EXPO	3	43	.9423	.9423	.6901	.9423/.3010	.9423/.3010	.3729/.0104	1.000/.2537	1.000/.2537	.7754/.0096
EXPO	4	143	.6665	.5049	.2202	.6665/.1684	.3248/.0213	.1197/.0017	.8048/.1414	.5484/.0050	.1281/ *

$*$ the error is here smaller than .00005 although not exactly zero

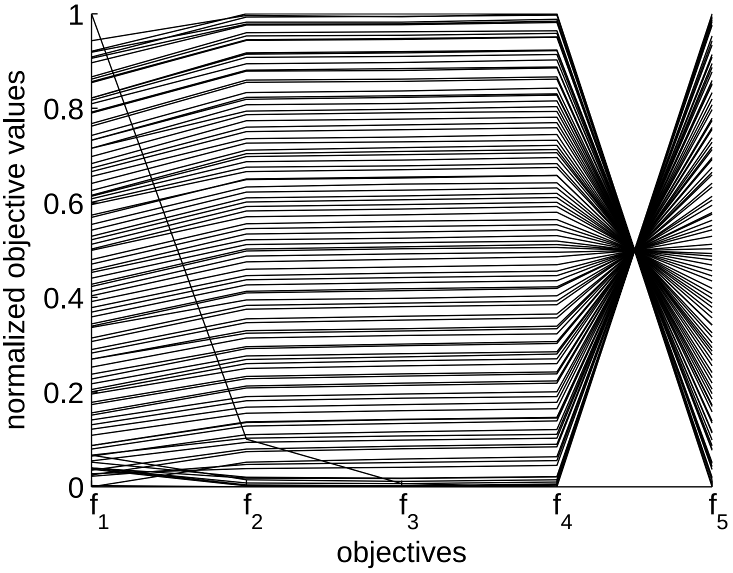


Figure 29: Normalized parallel coordinates plot of the used WFG3 instance with 5 objectives and 100 solutions.

Moreover, we would like to point out that the Pareto front approximations of the WFG test problem instances are seemingly of a certain shape since only a few objectives are necessary to describe the trade-offs between the objectives (see Table 3). Figure 29 shows as an example the used front of the WFG3 instance with 5 objectives in a normalized parallel coordinates plot. By looking at the objectives f_4 and f_5, we can observe that this objective pair induces all incomparabilities between the solutions due to their opposing nature in the found region of the search space. Therefore, the results of the WFG test problems in Table 3 might not be as representative for real-world applications as for the other problems.

3.7 Application to Decision Making

In this last section of the chapter, we provide examples how the above objective reduction algorithms and the definition of conflict can be useful in practice. In the case of offline analysis where a set of non-dominated solutions is given, the proposed approach of objective reduction can not only indicate which objectives are redundant but can also provide insights in the problem itself to make the decision making process easier. We will show these benefits exemplary for a radar waveform problem with nine objectives, recently proposed by Hughes (2007). The general question of whether objective reduction is useful during the search is the subject of the last section in the next chapter where we show experimentally that the integration of an online objective reduction can drastically improve the running time of a hypervolume-based search algorithm.

3.7.1 The Radar Waveform Optimization Problem

The real and unmodified engineering problem of radar waveform optimization, described in (Hughes, 2007), is to choose a set of waveforms for a pulsed Doppler radar allowing an unambiguous measure of both range and velocity of targets. The formalization of the radar waveform problem uses 9 objectives altogether. Hughes (2007) states various relationships between these 9 objectives due to their definitions, e.g., that the objective pairs 1 & 3, 2 & 4, 5 & 7, and 6 & 8 have a degree of correlation because they are metrics associated with the performance in range and velocity respectively[18].

Based on a set of $22,844$ non-dominated solutions, collected from multiple MOEA runs, and provided by the author of (Hughes, 2007)[19], we investigate the usefulness of the objective reduction approach proposed in this chapter in a decision making scenario, where a set of non-dominated solutions is used to learn about the problem and to get a deeper insight into the problem itself. To apply both the exact and the greedy algorithms for δ-`MOSS` and `k-EMOSS` as well as the aggregation heuristic for the $\mathtt{OAP}_{\mathtt{max}}$ and $\mathtt{OAP}_{\mathtt{avg}}$ problems, a reduction of the large set of solutions to a smaller set is necessary and performed by computing the ε-nondominated solutions out of the normalized original ones. The error ε is chosen as 6.2% yielding 107 solutions in the reduced set[20]. The computation of the smaller set of ε-nondominated solutions out of the entire set of $22,844$ solutions means that whenever we make a statement of δ-error with respect to the set of 107 solutions, we can assure that the error with respect to the set of all known solutions is at most $\varepsilon + \delta$.

3.7.2 Minimum Objective Sets

Computing all δ-minimal sets with the exact Algorithm 4 shows that for the reduced set of 107 solutions two objectives can be omitted without changing the dominance structure. With respect to the entire set of $22,844$ solutions, this means that we make only an error of at most 6.2% when omitting the correct two objectives. Nevertheless, the use of such a reduction is limited.

[18] See page 10 of (Hughes, 2007).

[19] Dr. Evan J. Hughes, Department of Aerospace, Power and Sensors, Cranfield University, Shrivenham, Swindon, Wiltshire, England, `e.j.hughes@cranfield.ac.uk`.

[20] Note, that the used error of 6.2% and the resulting solution set size of 107 is more or less arbitrary. Smaller errors, i.e., larger solution sets with up to 5000 solutions yield similar results.

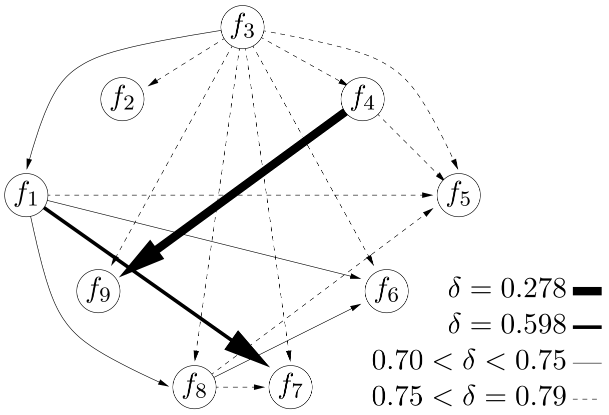


Figure 30: Radar waveform problem: visualization of minimum δ-error between objective pairs. Errors larger than 0.8 are omitted for clarity and the line width indicates the δ-error (the thicker, the smaller the error). The arrows point at the objectives which should be used.

Reducing the set of objectives from 9 to 7 still yields a huge amount of information, a decision maker has to survey, especially if more than $22,000$ solutions are to be compared.

3.7.3 Investigating Objectives Pairwisely

More useful for a decision maker would be to learn about the problem, i.e., to draw quantitative conclusions on the relationship between single objectives similar to the ones stated in (Hughes, 2007) qualitatively. The definition of δ-conflict and the corresponding algorithms to compute the δ-errors based on a set of solutions can provide such quantitative statements on objective pairs. For example, we can compute the minimum δ-error between all possible objective pairs and illustrate them as in Fig. 30. A low δ-error between an objective pair predicts that the consideration of only the one objective the arrow points at in Fig. 30 while the other objective is omitted does not change the dominance relation with an error of more than δ. Surprisingly, the smallest error occurs between objectives 4 and 9, the second smallest between objective pair 1& 7, in contrast to the prediction of Hughes (2007).

These pairwise δ-errors can, in addition, be used within the greedy Algorithm 7 for `k-EMOSS` and the greedy aggregation Algorithm 8 to obtain a tree visualization of the objective conflicts. Both the greedy Algorithm 7 and the greedy heuristic for $\mathtt{OAP}_{\mathtt{max}}$ (Algorithm 8) have been run for all possible objective set sizes ($1 \leq \mathtt{k} \leq k$) on the set of 107 non-dominated solutions. Figure 31 shows the two resulting trees together with the maximum and average δ-errors corresponding to the different levels of the tree, i.e., the computed objective sets of different size. Although both algorithms pursue the goal of creating a set of objectives that is as small as possible and

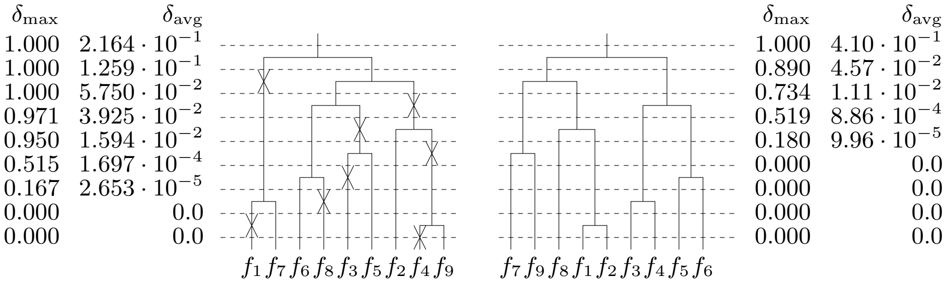

$\delta_{\max}$
δ_{avg}
1.000 $2.164 \cdot 10^{-1}$
1.000 $1.259 \cdot 10^{-1}$
1.000 $5.750 \cdot 10^{-2}$
0.971 $3.925 \cdot 10^{-2}$
0.950 $1.594 \cdot 10^{-2}$
0.515 $1.697 \cdot 10^{-4}$
0.167 $2.653 \cdot 10^{-5}$
0.000 0.0
0.000 0.0
$f_1\, f_7\, f_6\, f_8\, f_3\, f_5\, f_2\, f_4\, f_9$
$f_7\, f_9\, f_8\, f_1\, f_2\, f_3\, f_4\, f_5\, f_6$
$\delta_{\max}$
δ_{avg}
1.000 $4.10 \cdot 10^{-1}$
0.890 $4.57 \cdot 10^{-2}$
0.734 $1.11 \cdot 10^{-2}$
0.519 $8.86 \cdot 10^{-4}$
0.180 $9.96 \cdot 10^{-5}$
0.000 0.0
0.000 0.0
0.000 0.0
0.000 0.0

Figure 31: Tree Visualization for the radar waveform problem: (left) greedy k-EMOSS Algorithm 7 with objective omission; the crosses depict which objectives are omitted; (right) greedy aggregation heuristic with respect to $\delta_{\max}$-error.

at the same time preserves most of the dominance structure, the results are different. The greedy k-EMOSS heuristic identifies f_4 and f_1 as the most unimportant and f_6 as the most important objective(s) whereas the aggregation heuristic aggregates three objective pairs before the aggregation of f_7 and f_9 changes the dominance structure for the first time. However, the combination of both approaches can indicate interesting conclusions about the importance of single objectives to the overall optimization problem. In the example of Fig. 31, f_1 and f_4 are deleted first by the greedy k-EMOSS algorithm on the one hand, but also the aggregation heuristic uses these two objectives in the first two steps to reduce the number of objectives. Thus, and the fact that for both algorithms the first two reduction steps do not introduce an error in the dominance relation, one can argue that the information within the objectives f_1 and f_4 might be covered already by other objectives in the problem formulation and therefore can be identified as the least important criteria which reduces the amount of objectives a decision maker has to take into account.

Where the above examples illustrate how the proposed objective reduction approach can be used to analyze objective relations and to assist a decision maker in better understanding the problem characteristics which in the end might make it easier to decide which solutions are the most favorable ones, the following section shows that the objective reduction algorithms also might help in terms of visualization.

3.7.4 Visualizing Pareto Fronts

To show the advantages of the proposed aggregation heuristics with respect to visualization, we apply the objective omission heuristic for the k-EMOSS problem (Algorithm 6), the aggregation heuristic for the OAP_{max} and OAP_{avg} problem (Algorithm 8), and a general dimensionality reduction approach, namely PCA, to the above radar waveform optimization problem[21].

[21] The reason why Algorithm 6 is used here instead of Algorithm 7 is the fact that we would like to reduce the number of objectives to a plottable number, here k = 2, for

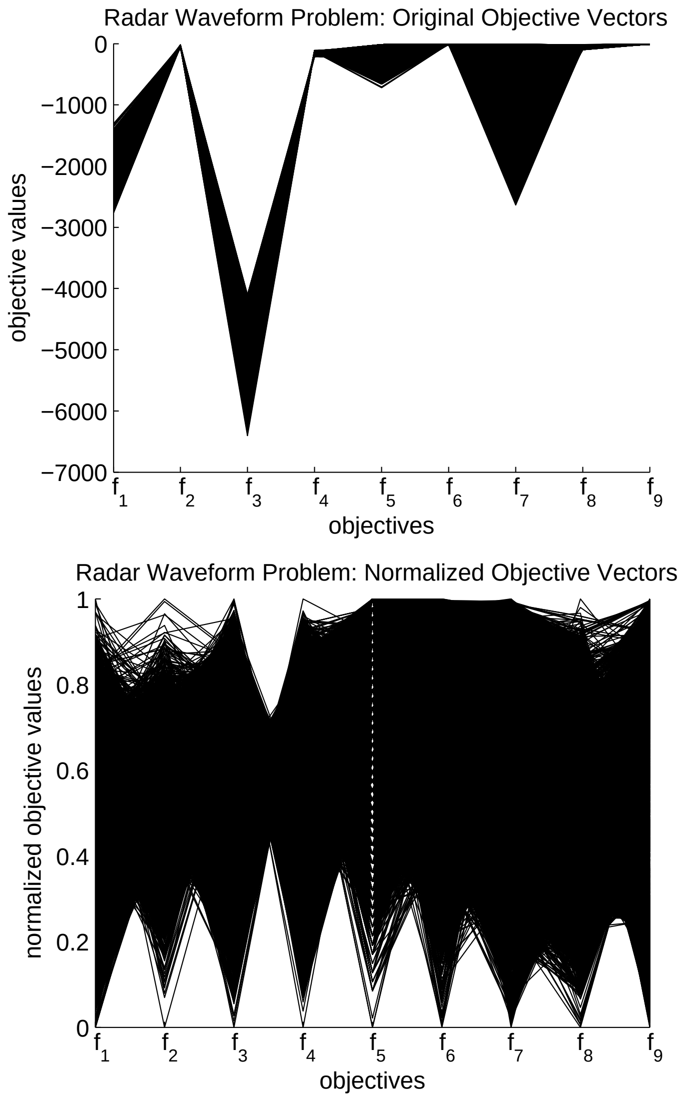


Figure 32: Parallel coordinates plots of the 22,844 solutions for the radar waveform problem: (top) original objective vectors (bottom) normalized objective vectors.

Visualizing the set of all known non-dominated solutions is a crucial task in many-objective optimization. For the radar waveform problem, the high number of both solutions and objectives makes the visualization of non-dominated solutions difficult. Even the visualization as a parallel coordinates plot does not provide a decision maker a way to easily reveal information about the problem due to the high amount of data, see Fig. 32. Here, we argue that the objective reduction techniques proposed above can help to gain a detailed understanding of the problem itself by plotting lower dimensional projections of the non-dominated solutions. To this end, we use the reduced set of 107 non-dominated solutions as the input for the

which Algorithm 6 is better suited than Algorithm 7 as it constructs the objective set by adding objectives instead of deleting some.

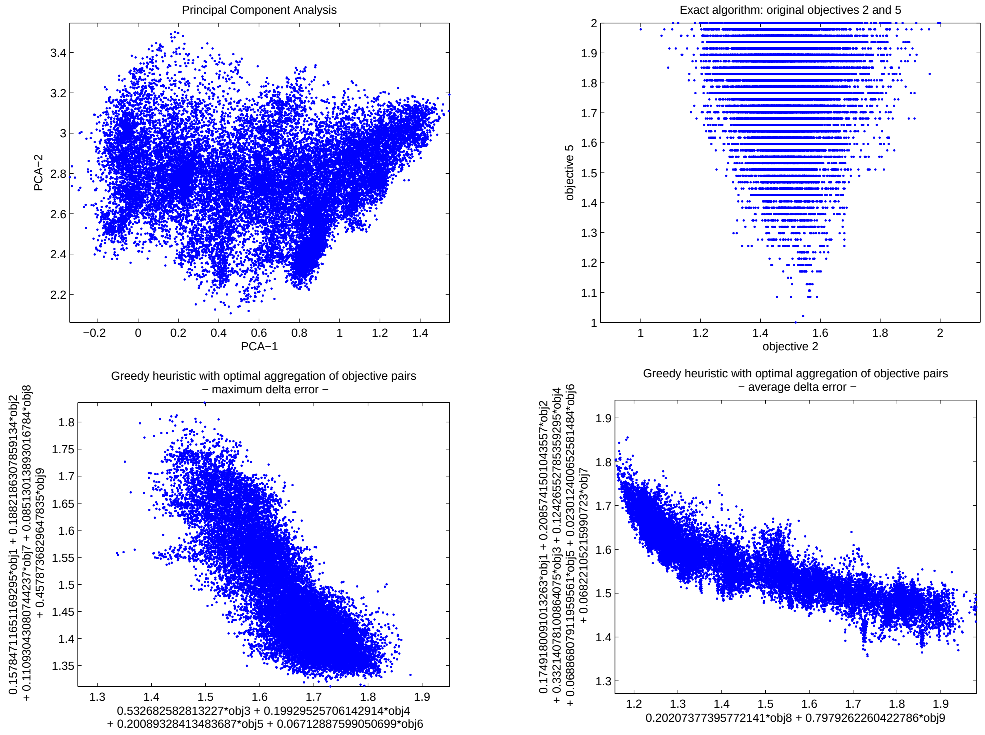


Figure 33: 2-dimensional plots of all known non-dominated solutions of the radar waveform problem where the objectives are chosen with respect to PCA (upper left), the exact `k-EMOSS` algorithm (upper right), and the greedy aggregation algorithm with maximum error (lower left) and average error (lower right).

objective reduction algorithms and reduce the number of objectives to two. Contrary, PCA is applied to the set of all 22, 844 non-dominated solutions since it is fast enough on the set of all points. The eigenvectors corresponding to the two largest eigenvalues are taken as weight vectors that aggregate the original objectives into two.

Table 4 shows the maximum and average δ-errors and the number of solution pairs that are comparable for both the reduced set of 107 solutions and the set of all 22, 844 non-dominated solutions. Figure 33 shows the 2D plots of all 22, 844 non-dominated solutions for the radar problem if the four different reduction techniques are applied.

What can be seen from the results is that the number of solution pairs that remain non-dominated increases from the PCA plot over the one with original objectives only (`k-EMOSS` heuristic) to the results of the greedy aggregation heuristic for $\texttt{OAP}_{\texttt{max}}$ and the on for $\texttt{OAP}_{\texttt{avg}}$. The reason is that general dimensionality reduction techniques such as PCA do not take into account the Pareto dominance relation between the solutions when reducing the dimensionality of data. Thus, the dominance relation is not preserved and many solutions dominate each other. In contrast, the presented objective reduction approach takes the Pareto dominance relation into account and reduces the number of objectives while the dominance relation

Table 4: Comparison of the reduction to 2 objectives for the radar waveform problem with the reduced set of 107 points (above) and the set of all 22, 844 non-dominated points (below). Shown are the number of solution pairs (total 5671 and $\approx 2.6 \cdot 10^8$ respectively) for that we wrongly assume comparability together with the maximum and average δ-errors.

		#comparable solution pairs	$\delta_{\max}$	δ_{avg}
107 solutions	Exact algorithm for `k-EMOSS`	2'835	0.9935	0.111005
	PCA	2'525	0.9797	0.088250
	Greedy heuristic for $\texttt{OAP}_{\texttt{max}}$	872	0.8898	0.022609
	Greedy heuristic for $\texttt{OAP}_{\texttt{avg}}$	585	0.9638	0.015443
all solutions	Exact algorithm for `k-EMOSS`	130'084'321	1.0000	0.087803
	PCA	134'147'513	0.9999	0.081457
	Greedy heuristic for $\texttt{OAP}_{\texttt{max}}$	49'189'796	0.9902	0.019601
	Greedy heuristic for $\texttt{OAP}_{\texttt{avg}}$	32'471'066	0.9983	0.012444

is changed as little as possible. Therefore, a decision maker looses less information about the original data set in terms of dominance relation if the objective reduction approaches are used for visualization instead of a classical dimensionality reduction technique such as PCA.

With such a reduced visualization of the set of non-dominated solutions, a decision maker might be able to decide which solutions are preferable much faster although the aggregated objectives in a plot similar to the lower plots of Fig. 33 might be more difficult to interpret. Preserving only the Pareto dominance relation during the objective reduction might also not always be sufficient; preserving the distance between solutions might be a necessary criterion as well and future research in this area is highly needed.

3.8 Summary

This chapter addressed the issue of objective reduction in many-objective optimization. Based on the effects of adding and omitting objectives on the dominance structure as investigated in the previous chapter, we proposed an objective reduction approach that is based on a general notion of objective conflict. The approach allows to identify objective sets of minimum size, while ensuring that the Pareto dominance relation is preserved or only slightly changed according to a certain, predefined error. The problems δ-`MOSS` and `k-EMOSS` formalizing this approach have been defined, their $\mathcal{NP}$-hardness has been proved, and an exact algorithm as well as several heuristics have been proposed—corresponding implementations are freely available for download at `http://www.tik.ee.ethz.ch/sop/download/supplementary/objectiveReduction/`. Furthermore, we investigated the generalization to the aggregation of objectives and proposed a greedy algorithm for this case as well. Although the proposed exact algorithm is not applicable in practice, the greedy objective reduction heuristics when applied to typical instances including solution sets with up to 300 solutions and problems with up to 20 objectives run in a few seconds (on a Dual-Core AMD Opteron linux machine with 2.6Ghz; dependent on the algorithm and the desired `k` and δ also much faster) such that they can be also used during the search as we will discuss in the next chapter.

The experimental results have demonstrated that the proposed methodology can support the decision maker by reducing the amount of data to be considered and by making quanti

That the research direction, we followed during this chapter, is interesting and that further achievements can be expected in the future is shown by other studies that have been conducted in the meantime. López Jaimes et al. (2008) for example, built upon our objective reduction approach and proposed improved algorithms for the δ-`MOSS` and `k-EMOSS` problems based on a known feature selection algorithm. López Jaimes et al. (2009) then proposed another correlation-based objective reduction algorithm and applied it in a multiobjective optimizer to reduce the number of objectives during search. Köppen and Yoshida (2007) on the other hand, proposed another idea to visualize high-dimensional objective spaces; instead of trying to reduce the δ-error as in this chapter, their method tries to preserve as many relationships between the individuals as possible when reducing the number of objectives to a lower one. Also Costa and Oliveira (2009) tackled the visualization of the outcomes of many-objective optimizers in terms of objective reduction via principal component analysis.

Although we showed the usefulness of automatically finding a compact representation of the original objectives within a decision making scenario, the proposed approach might be also helpful in other scenarios. For example, the aggregation heuristics might be helpful if an initial weighting of the objectives within the well-known weighted sum method is sought. The objective reduction approach can also be helpful in many-objective scenarios where the computation times needed for the distinct objective functions vary highly. However, the application of the proposed methods to these scenarios remains future work.

Another application area of the proposed objective reduction approaches remained untackled so far: the reduction of the number of objectives *during search* can be helpful, especially in hypervolume based search—the topic of the thesis' last chapter.

4

Hypervolume-Based Search and Objective Reduction

The hypervolume indicator, first introduced by Zitzler and Thiele (1998b, 1999), is one of the most important quality indicators in the evolutionary multiobjective optimization field. Originally proposed for performance assessment, the hypervolume indicator has also gained lots of interest within multiobjective evolutionary algorithms to guide the search (Knowles and Corne, 2003; Huband et al., 2003; Zitzler and Künzli, 2004; Nicolini, 2005; Beume et al., 2007; Igel et al., 2007; Zitzler et al., 2007; Bader et al., 2008; Bader and Zitzler, 2008; Bader et al., 2009). These hypervolume-based MOEAs have been shown to outperform classical Pareto dominance based MOEAs such as NSGA-II and SPEA2, in terms of convergence to the Pareto front especially if the number of objectives is high (Wagner et al., 2007) and nowadays seem to be the first choice when handling multiobjective optimization problems with many objectives.

The main reason for this and for the popularity of the hypervolume indicator is the fact that—in comparison to many other indicators—the hypervolume indicator is *compliant* or compatible with the Pareto dominance relation (Knowles and Corne, 2002; Zitzler et al., 2003, 2009), we also say it *refines* the Pareto dominance. The refinement property of an indicator I is fulfilled if for all sets A, B of solutions, the indicator value $I(A)$ is strictly higher than $I(B)$ whenever A is dominating B. This refinement property allows to indirectly optimize the objective functions with respect to the Pareto dominance relation while further search directions towards the Pareto set are introduced. This helps especially if many objectives are optimized since the number of incomparable solution sets increases (Winkler, 1985). Another important property of the hypervolume indicator directly follows from this refinement property: a solution set that optimizes the hypervolume indicator covers the entire Pareto front (Fleischer, 2003).

At the beginning of this thesis project, not many theoretical investigations of the hypervolume indicator were known and many properties of the hypervolume had not been understood. Although Knowles and Corne (2002) had pointed out that the choice of the hypervolume's reference point changes the order in which solution sets outperform each other, it was not understood how the choice of the reference point influences the optimal distribution of points on the Pareto front. Furthermore, the bias of the hypervolume in terms of which solutions are preferred on the Pareto front if the number of solutions is upper bounded, e.g., by the population size, had not been understood and the statements on the bias were inconsistent. Also the question why and when hypervolume-based MOEAs are beneficial had not been investigated at that time. The complexity of the hypervolume computation had not been proven and the best algorithms at that time had running times of $\Theta(n^{k-1})$ where n is the number of solutions and k the number of objectives. We refer to Bader and Zitzler (2008) for a more detailed review on algorithms for computing the hypervolume indicator.

In recent years, many important steps towards the understanding of the hypervolume as well as towards its applicability in multiobjective search have been made. For example, the open question of the problem complexity has been solved which shows that the running time of exact algorithms cannot be upper bounded by a polynomial in the number of objectives under the assumption $\mathcal{P} \neq \mathcal{NP}$ (Bringmann and Friedrich, 2008). Nevertheless, several new algorithms have been proposed in the meantime which improve the running time of the hypervolume computation in comparison to the first algorithms (While, 2005; Fonseca et al., 2006; Beume and Rudolph, 2006; While et al., 2006; Bringmann and Friedrich, 2008). The best known exact algorithm has been proposed by Beume and Rudolph (2006) when adapting an existing algorithm for Klee's measure problem the running time of which is of order $\mathcal{O}(n^{k/2})$. Moreover, Monte Carlo sampling was used to make hypervolume-based MOEAs better suited for many-objective problems (Bader et al., 2008; Bader and Zitzler, 2009; Auger et al., 2009a) such that nowadays, the running time of those algorithms is not a restriction any more in many-objective scenarios.

Furthermore, general theoretical investigations on the hypervolume indicator as a set measure have shown that hypervolume-based MOEAs do not converge in general if they only produce one solution per generation and delete the one with the smallest hypervolume contribution (Zitzler et al., 2009; Beume et al., 2009). Very recently, the choice of the reference point and the bias of the hypervolume in terms of optimal μ-distributions[1] have

[1] With the work of Fleischer (2003), it is clear that optimizing the hypervolume indi-

been investigated theoretically (Auger et al., 2009c)—basic foundations that will hopefully increase the popularity of the hypervolume further.

Several investigations on the hypervolume indicator have also been carried out throughout this thesis project. In the remainder of this chapter, three main contributions to the field of hypervolume-based MOEAs are presented. In particular, we

- provide the first rigorous running time analysis of a hypervolume-based MOEA (Sec. 4.2),
- propose a weighted hypervolume version that allows for incorporation of user preferences and validate the approach experimentally (Sec. 4.3), and we
- apply the objective reduction methods proposed in the previous chapter to a hypervolume-based MOEA to deal with many objectives although the exact hypervolume indicator computation is expensive (Sec. 4.4).

Before, the hypervolume indicator is introduced in detail in Sec. 4.1.1 and the Simple Indicator-Based Evolutionary Algorithm (SIBEA), the basic algorithm for all studies to follow, is defined in Sec. 4.1.2.

4.1 Foundations

4.1.1 The Hypervolume Indicator

Contrary to the previous sections, we consider, without loss of generalization, maximization problems here if appropriate in order to simplify the notation and to comply with the original publications (Brockhoff et al., 2008; Zitzler et al., 2007) the content of which we present below. Note also that we use the notation $\succeq$ instead of $\preceq$ here which is, for the sake of simplicity, also used between objective vectors, although it is originally defined on the decision space X. Furthermore, we restrict the investigations in this chapter to solution sets although in general the population of a MOEA is a multiset since individuals can be contained more than once. However, we assume that a restriction to sets is not crucial here but will improve the

cator will result in a set covering the entire Pareto front if no restriction on the number of points is given. However, this cannot be guaranteed in practice since the number of solutions that can be stored at the same time is bounded above, for example by the population size μ of a MOEA. The question then is which set with at most μ solutions maximizes the hypervolume; or in the terms of Auger et al. (2009c), how an optimal μ-distribution looks like.

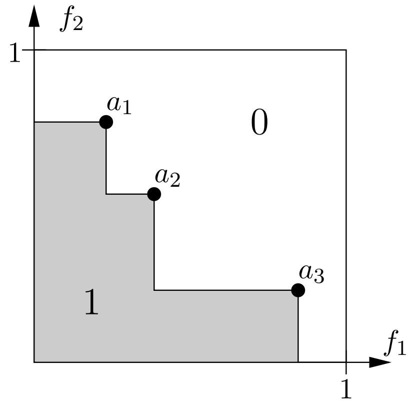


Figure 34: Illustration of the attainment function α_A for $A = \{\vec{a}_1, \vec{a}_2, \vec{a}_3\}$ in the two-dimensional case. In the gray area, the attainment function is 1 and 0 in the area above the set A in case of maximization.

clarity and readability of the following paragraphs in comparison to the formal way of using multisets throughout this chapter. We denote the set of all solution sets by 2^X and the set of all sets of objective vectors by $\Omega := 2^{\mathbb{R}^k}$.

The classical definitions of the hypervolume indicator are based on volumes of polytopes (Zitzler and Thiele, 1999) or hypercubes (Fleischer, 2003) and assume that the weak Pareto dominance is the underlying preference relation. Several other equivalent definitions have been proposed in the meantime as well as different names are used in the literature such as hypervolume metric (Van Veldhuizen, 1999), $\mathcal{S}$-metric (Zitzler, 1999), or Lebesgue measure (Laumanns et al., 1999; Fleischer, 2003). However, using the term *Lebesgue measure* for the hypervolume indicator is a little bit confusing since the Lebesgue measure is a general concept in integration theory and it is part of the definition of the hypervolume indicator, see for example Chapter 2 or the definitions in (Emmerich et al., 2005; Bader and Zitzler, 2008). Here, we give an equivalent definition based on attainment functions that allows to generalize the hypervolume indicator to a weighted version later on.

The attainment function (Grunert da Fonseca et al., 2001) gives, roughly speaking, for each objective vector in the objective space $\mathbb{R}^k$ the probability that it is weakly dominated by the outcome of a particular multiobjective optimizer. As only single sets are considered here, we can take a slightly simplified definition of the attainment function:

Definition 13 (Attainment function for an objective vector set)**.** *Given a set $A \subseteq X$ of solutions, the attainment function $\alpha_A : \mathbb{R}^k \to \{0, 1\}$ for A is defined as*

$$\alpha_A(\vec{z}) := \begin{cases} 1 & \text{if } f(A) \succeq \{\vec{z}\} \\ 0 & \text{else} \end{cases}$$

for $\vec{z} \in \mathbb{R}^k$.

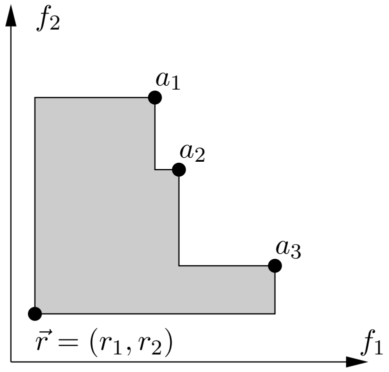


Figure 35: The hypervolume indicator of a set $\{a_1, a_2, a_3\}$ of three points and for a predefined reference point $\vec{r} = (r_1, r_2)$ when maximizing the objective functions.

This definition is illustrated for a biobjective example in Fig. 34. The concept of attainment functions can be used to give another equivalent formal definition of the hypervolume indicator. It is simply defined as the volume of the objective space enclosed by the attainment function and a predefined reference point[2], see Fig. 35 for an illustration.

Definition 14 (Hypervolume indicator)**.** *The hypervolume indicator $I_H : 2^X \to \mathbb{R}$ with reference point $\vec{r} = (r_1, \ldots, r_k)$ can be formulated via the attainment function as*

$$I_H(A) := \int_{\vec{r}}^{(\infty,\ldots,\infty)} \alpha_A(\vec{z}) d\vec{z}$$

where $A \in 2^X$ is any set of solutions and α_A is the above defined attainment function of A.

Using the hypervolume indicator for search, the optimization goal changes from finding or approximating the Pareto front to finding or approximating the set of solutions that maximizes the hypervolume indicator. Fortunately, these goals are equivalent since the hypervolume indicator has the property of being a *refinement* of the Pareto dominance relation[3].

Definition 15 (Refinement)**.** *According to Zitzler et al. (2009) we call an indicator $I : 2^X \to \mathbb{R}$ a refinement of the Pareto dominance relation if and only if*

$$\forall A, B \in 2^X : A \succeq B \ \wedge \ B \not\succeq A \Rightarrow I(A) > I(B) \ ,$$

i.e., whenever a solution set is dominating another one, the former has a higher indicator value than the latter.

[2] Most of the studies which use the hypervolume indicator define only a single reference point although a recent generalization of the hypervolume indicator defines a *set* of reference points (Bader and Zitzler, 2008). Throughout this study, we use the former definition with only one reference point for the sake of readability.

[3] Also the equivalent term of being *Pareto-compliant* has been used, e.g., in (Knowles et al., 2006).

As the weak Pareto dominance relation $\succeq$ defines only a *partial order* on Ω, the hypervolume indicator refines this relation by introducing further comparabilities between solution sets resulting in a *total order* on Ω which guarantees that any two objective vector sets are mutually comparable (Knowles et al., 2006; Zitzler et al., 2009). This especially helps to guide the search within multiobjective evolutionary algorithms if the number of objectives is high, i.e., if the weak Pareto dominance relation cannot indicate a search direction due to the high number of incomparable solution sets.

Furthermore, the refinement property of the hypervolume indicator guarantees that a set maximizing the hypervolume indicator contains for each point on the Pareto front at least one solution that is mapped to this point (Fleischer, 2003). This, however, is not the case anymore in practical applications where we restrict the size of the solution sets, e.g., by allowing only solution sets of a maximal size μ as the population of an evolutionary algorithm and if at the same time the Pareto front contains more than μ points. In this case, Fleischer (2003) showed that a set of size at most μ which maximizes the hypervolume contains only Pareto-optimal solutions. How these μ points are distributed on the Pareto front has been investigated recently by Auger et al. (2009c). Their investigations show that the distribution depends on the choice of the hypervolume's reference point as well as on the slope of the Pareto front.

4.1.2 SIBEA: the Simple Indicator-Based Evolutionary Algorithm

Several evolutionary algorithms to optimize the hypervolume have been proposed in the literature, e.g., the ESP algorithm (Huband et al., 2003), IBEA (Zitzler and Künzli, 2004), SMS-EMOA (Beume et al., 2007), or the MO-CMA-ES (Igel et al., 2007). Most of them use the same ($\mu+\lambda$)-selection scheme which will be also investigated in the remainder of this chapter. The population P of the next generation with $|P| = \mu$ is computed from the set P' of solutions that is the union of the previous population and the λ generated offsprings in the following way: after ranking of the solutions with respect to the Pareto dominance relation, e.g., by a non-dominated sorting of P' (Goldberg, 1989), the ranked solutions are, starting with the best rank, completely inserted into the new population P until the size of P is at least μ. For the first set of solutions of same rank, the inclusion of which yields a population size larger than μ, the solutions $\vec{x}$ with this rank and the smallest hypervolume contribution are successively removed from the new population where the hypervolume contribution is recalculated every time a solution is removed.

The simplest scheme of such a hypervolume-based MOEA is shown in Algorithm 9 as the Simple Indicator-Based Evolutionary Algorithm (SIBEA). We will use SIBEA as the baseline algorithm for all studies presented in this chapter. Note that the algorithm is defined to work with any unary quality indicator (see Zitzler et al., 2003) as the optimization criterion although we only use the hypervolume indicator here. The reason why the simple algorithm SIBEA is chosen here instead of a more advanced algorithm is that the studies presented in this chapter mainly investigate the selection criterion that is nearly the same in all mentioned hypervolume-based algorithms and results can be therefore carefully transfered to the more complicated algorithms if necessary.

Given a population size μ, the number N of generations and a certain indicator function I, SIBEA outputs an approximation A of the Pareto-optimal set and works as follows. In an initialization step, the population P of μ solutions is generated by independently choosing μ solutions uniformly at random from the decision space X. Then, the steps 2 (environmental selection) and 4 (mating selection and variation) are performed N times. In the environmental selection step, the population size is reduced to μ by the following procedure: first, the solutions are ranked according to dominance depth (Zitzler et al., 2004)[4] and the individuals are deleted from the population in order of their ranking. Starting with the set of solutions with the worst rank $P' \subseteq P$, the one with smallest indicator loss $d(\vec{x}) := I(P') - I(P' \setminus \{\vec{x}\})$ is deleted first. Until the population size reaches again μ, individuals are deleted by iteratively recalculating the indicator losses in the set of solutions with worst rank and deleting the solution with the smallest loss. Note that this procedure is the same than within other indicator-based algorithms such as the SMS-EMOA (Beume et al., 2007) and the MO-CMA-ES (Igel et al., 2007) where the hypervolume indicator is used. After the termination criterion is checked in step 3, μ parents are selected in SIBEA uniformly at random from the population and afterwards recombined and mutated to yield the (multi-)set of μ offspring Q'. The populations P and Q' are combined and the algorithm continues with step 2. Note that the variation operators are not specified here and will be defined later. Furthermore, note that the number of offspring is fixed to μ here although producing a more general number of $\lambda \in \mathbb{N}$ offspring in a general $(\mu + \lambda)$-strategy is also possible.

The running time analysis, we carry out in the following section, investigates a version of SIBEA that is tailored towards the optimization in discrete domain. The main differences between the simplified $(\mu{+}1)$-SIBEA described in Algorithm 10 and the general SIBEA procedure of Algorithm 9 are the specified decision space, the usage of a concrete mutation opera-

[4] Other ranking methods like nondominated sorting (Goldberg, 1989) could be also used instead.

Algorithm 9 Simple Indicator-Based Evolutionary Algorithm (SIBEA)

```
*Input:* population size $\mu$; number of generations $N$; indicator function $I$
*Output:* approximation of Pareto-optimal set $A$
*Step 1 (Initialization):*
Generate an initial (multi-)set of solutions $P$ of size $\mu$ uniformly at random from $X$; set the generation counter $m := 0$.
*Step 2 (Environmental Selection):*
Iterate the following three steps until the size of the population does no longer exceed $\mu$:
    1. Rank the population using Pareto dominance and determine the set of individuals $P' \subseteq P$ with the worst rank. Here, dominance depth (Zitzler et al., 2004) is used.
    2. For each solution $\vec{x} \in P'$ determine the loss $d(\vec{x})$ with respect to the indicator $I$ if it is removed from $P'$, i.e., $d(\vec{x}) := I(P') - I(P' \setminus \{\vec{x}\})$.
    3. Remove the solution with the smallest loss $d(\vec{x})$ from the population $P$ (ties are broken randomly).
*Step 3 (Termination):*
If $m \geq N$ then set $A := P$ and stop; otherwise set $m := m + 1$.
*Step 4 (Mating):*
Randomly select elements from $P$ to form a temporary mating pool $Q$ of size $\mu$. Apply recombination and mutation operators to the mating pool $Q$ which yields $Q'$. Set $P := P + Q'$ (multi-set union). Continue with Step 2.
```

tor, the absence of recombination, and a simplified environmental selection scheme.

As to the decision space, Algorithm 10 considers boolean vectors of length n, i.e., $X = \{0, 1\}^n$. The mutation operator is the well-known bitwise mutation operator (see for example (Droste et al., 2002)) that flips each bit independently with probability $1/n$ and that is used in the Global SEMO as well. As to the environmental selection step, the $(\mu + 1)$-SIBEA aims at optimizing the hypervolume indicator similar to other hypervolume-based MOEAs, such as SMS-EMOA, where only one new solution x' is generated per generation. Hence, the algorithm removes only the solution $\vec{x}$ with the smallest hypervolume loss $d(\vec{x})$ from $P \cup \{\vec{x}'\}$ in each generation. The ranking of the individuals with respect to dominance depth as in the original SIBEA is dropped here to make the proof simpler. However, this does not change the theoretical results, i.e., the obtained running time bound: By

Algorithm 10 $(\mu + 1)$-SIBEA

```
Parameters: population size $\mu$
Step 1 (Initialization):
Generate an initial (multi)-set of decision vectors $P$ of size $\mu$ uniformly at random from the decision space $\{0,1\}^n$.
Step 2 (Repeat):
    a) Select an element $\vec{x}$ from $P$ uniformly at random. Flip each bit of $\vec{x}$ with probability $1/n$ to obtain an offspring $\vec{x}'$. Set $P' := P \cup \{\vec{x}'\}$.
    b) For each solution $\vec{x} \in P'$ determine the hypervolume loss $d(\vec{x})$ if it is removed from $P'$, i.e., $d(\vec{x}) := I_H(P') - I_H(P' \setminus \{\vec{x}\})$.
    c) Choose an element $\vec{z} \in P'$ with smallest loss in $P'$ uniformly at random, i.e., $\vec{z} = \text{argmin}_{\vec{x} \in P} d(\vec{x})$ and set $P := P' \setminus \{\vec{z}\}$.
```

omitting the ranking of step 2.1 in Algorithm 9, only dominated points are handled differently—with the original selection scheme of Algorithm 9, always the worst point on the worst front is deleted, whereas in the simplified Algorithm 10, any dominated point is deleted with the same probability. Since the proof does only argue about the set of non-dominated points, the presented running time bound also holds for the more complex algorithm although it is not clear whether the proved bound is tight.

4.2 Running Time Analysis of a Hypervolume Indicator Based Algorithm

Many theoretical running time analyses of MOEAs have been carried out recently, see also Sec. 2.2. However, all results examine algorithms that are using some kind of dominance relation to guide the search, either the typical (weak) Pareto dominance (Giel, 2003; Laumanns et al., 2004b,a, and many others) or the ε-dominance (Horoba and Neumann, 2008). Before this thesis project started, no theoretical running time results on hypervolume indicator based MOEAs were known. Recently, Zitzler et al. (2009) provided the first result on the non-convergence behavior of a $(\mu + 1)$-strategy. An example of only 4 solutions and two objective functions shows that the (2+1)-strategy does not always converge to the solution set with the highest hypervolume value. Beume et al. (2009) investigated the non-convergence for more general continuous Pareto fronts based on the 4-solutions example. However, the same $(\mu + 1)$-strategy is used as environmental selection step in most of the available hypervolume-based MOEAs (e.g., Emmerich et al., 2005; Igel et al., 2007; Bader and Zitzler, 2008) and these algorithms perform well both on the investigated test problems and on real-world applications. This rises the questions of why these algorithms can be successfully applied to real-world problems and on which types of problems they do not converge. To bridge this gap between theory and practice, we carry out a rigorous running time analysis of the $(\mu+1)$-SIBEA showing that on the simple example function LOTZ a $(\mu+1)$-strategy can find the entire Pareto front in polynomial time. However, the analysis presented here is only a first step towards a better understanding of hypervolume-based algorithms in general[5].

In the following, we examine the biobjective maximization problem

$$\textsc{Lotz}(\vec{x}) = (\textsc{LeadingOnes}(\vec{x}), \textsc{TrailingZeros}(\vec{x}))$$

as it was defined in Sec. 2.2.4.1 and in (Laumanns et al., 2004b). The Pareto set of LOTZ consists of all $n+1$ solutions of the form 1^i0^{n-i} $(0 \leq i \leq n)$ that map to an objective vector of $(i, n-i)$. If the population size μ of a MOEA is large enough, i.e., $\mu \geq n+1$, the aim of the optimization should be to include the entire set of $n+1$ Pareto-optimal solutions in the population of the MOEA within a finite number of generations which is polynomially bounded in n. In the following, we show for the $(\mu+1)$-SIBEA that this convergence to the Pareto front is indeed given and that the entire Pareto front is contained in the algorithm's population after an expected number of $O(\mu n^2)$ generations.

Without loss of generality, we fix the reference point for computing the hypervolume to $(r_1, r_2) = (-1, -1)$. For other choices of the reference point with negative objective values, some constants in the proofs have to be adapted; however, the proof ideas and therefore the results hold for arbitrary choices of the reference point with $r_1, r_2 < 0$. With the choice of the reference point as $(-1, -1)$, every possible objective vector of LOTZ has a positive hypervolume contribution and due to the discrete Pareto front of $n+1$ solutions and the choice of $\mu \geq n+1$, the set of μ solutions maximizing the hypervolume value covers the entire Pareto front, cf. (Fleischer, 2003). Furthermore, this optimal hypervolume value, assigned to all solution sets containing the entire Pareto front, is $\Theta(n^2)$ if the reference point is set to $(-1, -1)$[6]. Before we investigate the overall running time of the

[5] A running time analysis of SIBEA on another test function has been also performed in (Brockhoff et al., 2008) but since there is still an ongoing discussion among the authors whether the proof is entirely correct we refrain from presenting the details here.

[6] The optimal hypervolume value for the LOTZ problem and a reference point of $(-1, -1)$ results from the simple summation of the stacked hypervolume boxes, given by the $n+1$ Pareto-optimal solutions: the rightmost Pareto-optimal solution has a hypervolume of $n+1$, the second Pareto-optimal point to the right dominates a hypervolume which is not dominated by the rightmost point of size n and so forth, yielding an optimal hypervolume of $\sum_{i=1}^{n+1} i = \frac{(n+1)(n+2)}{2} = \Theta(n^2)$

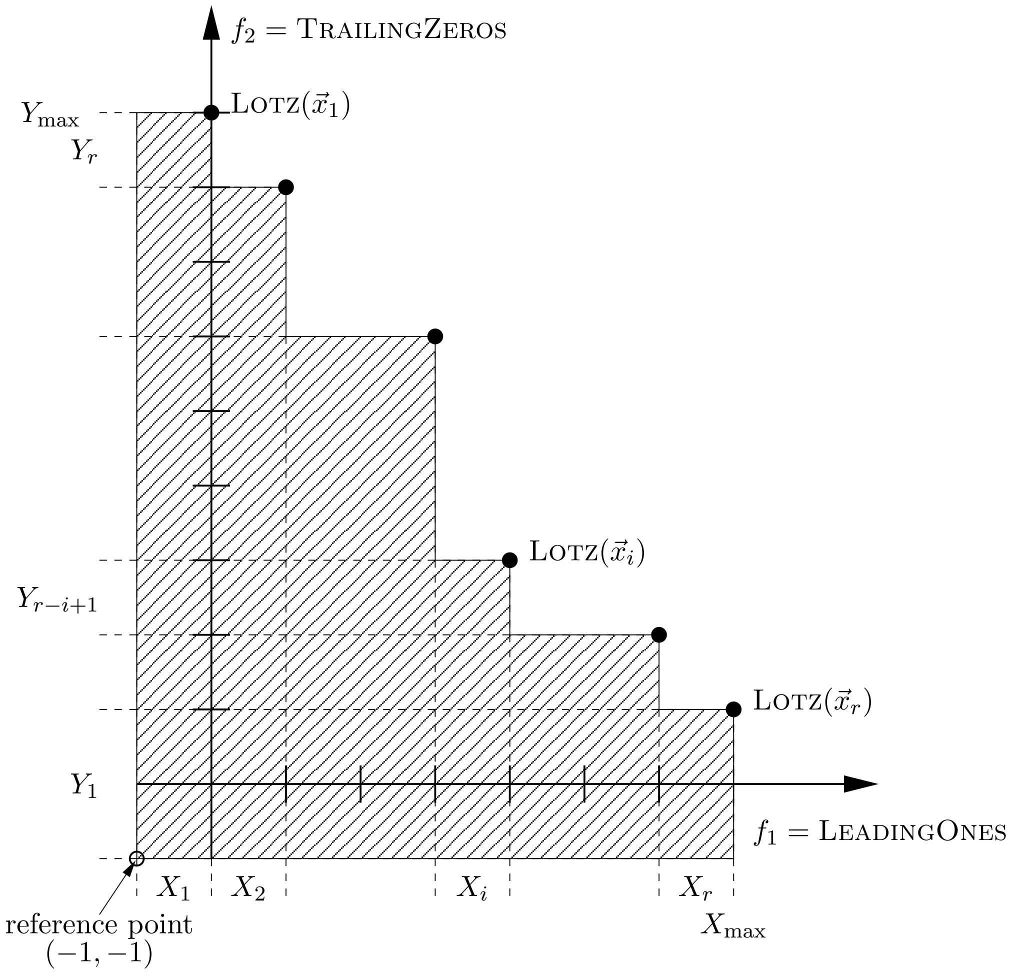


Figure 36: Illustration of the hypervolume (hatched area) and corresponding notations $X_{(\cdot)}, Y_{(\cdot)}$ in the analysis of $(\mu+1)$-SIBEA on Lotz.

$(\mu+1)$-SIBEA on Lotz in Theorem 17, we state a lemma about the running time until the first Pareto-optimal solution is found.

Lemma 1. *The expected time until the* $(\mu+1)$*-SIBEA has obtained for the first time a Pareto-optimal solution of* Lotz *is* $O(\mu n^2)$.

Proof. Throughout this proof, we consider the situation that no Pareto-optimal search point belongs to the current population P. Let $\{\vec{x}_1, \ldots, \vec{x}_r\} \subseteq P$ be the set of individuals that are not dominated by any other individual in P. Denote by H the hypervolume covered by these points. Without loss of generality, we assume that LeadingOnes($\vec{x}_i$) $\leq$ LeadingOnes($\vec{x}_{i+1}$) holds for all $1 \leq i \leq r-1$ which also implies TrailingZeros($\vec{x}_i$) $\geq$ TrailingZeros($\vec{x}_{i+1}$) for all $1 \leq i \leq r-1$, as the r individuals do not dominate each other. Let $X_1 = $ LeadingOnes($\vec{x}_1$)$+1$ and for all $2 \leq i \leq r$ let $X_i = $ LeadingOnes($\vec{x}_i$) $-$ LeadingOnes($\vec{x}_{i-1}$). Furthermore, denote by $X_{\max} = \sum_{i=1}^{r} X_i$ the maximum LeadingOnes-value with respect to the reference point $(-1,-1)$. Similar, define $Y_1 = $ TrailingZeros($\vec{x}_r$)$+1$ and

$Y_i = \textsc{TrailingZeros}(\vec{x}_{r-i}) - \textsc{TrailingZeros}(\vec{x}_{r-i+1})$ for all $2 \leq i \leq r$, and denote by $Y_{\max} = \sum_{i=1}^{r} Y_i$ the maximum TrailingZeros-value with respect to the reference point $(-1, -1)$. Figure 36 illustrates the variables defined above.

Considering one single solution $\vec{x}_i$ of the r non-dominated solutions of P, we study how the hypervolume can increase. Flipping the single bit which increases its LeadingOnes-value increases the hypervolume by at least Y_{r-i+1}; flipping the single bit which increases its TrailingZeros-value increases the hypervolume by at least X_i. Figure 37 is illustrating these two cases exemplary for two solutions. We call all these 1-bit flips applied to one of the r individuals good. Each of these $2r$ good operations happens with probability

$$\frac{1}{\mu} \cdot \frac{1}{n} \cdot (1 - 1/n)^{n-1} \geq \frac{1}{e\mu n}$$

in the next step. Note, that each good operation is accepted as it leads to a population with a larger hypervolume. The total increase of all good operations with respect to the current hypervolume H is at least

$$\max\{X_{\max}, Y_{\max}\} \geq \sqrt{X_{\max} \cdot Y_{\max}} \geq \sqrt{H}.$$

Choosing one of theses $2r$ good operations uniformly at random, the expected increase of the hypervolume is at least $\sqrt{H}/(2r)$. Hence, the expected number of good operations which is needed to increase the hypervolume by $\sqrt{H}$ is upper bounded by $2r$. Using Markov's inequality, the probability of having at least $4r$ operations to achieve this goal is upper bounded by $1/2$. Hence, with probability at least $1/2$ a phase containing $4r$ good operations is successful, i.e., increases the hypervolume by $\sqrt{H}$ with probability at least $1/2$. This implies that an expected number of 2 of these phases carrying out $4r$ such good operations each is enough to increase the hypervolume by $\sqrt{H}$.

Considering all good 1-bit flips together, the probability of carrying out one good operation in the next step of the algorithm is at least $\frac{2r}{e\mu n}$. Hence, the expected waiting time for a good operation is $O(\mu n/(2r))$ and the expected waiting time for increasing the hypervolume by at least $\sqrt{H}$ is therefore upper bounded by $O(\frac{\mu n}{2r} \cdot 2 \cdot 4r) = O(\mu n)$.

It remains to show that $O(n)$ successive increases of the hypervolume by its square-root fraction suffice to reach the maximum hypervolume value of $\Theta(n^2)$. Let $h(t)$ be the hypervolume of the current solutions after t increases by at least $\sqrt{h(t)}$. Then, $h(t+1) \geq h(t) + \sqrt{h(t)}$. We want to prove by induction that $h(t) \geq t^2/5$. The induction basis case holds trivially since

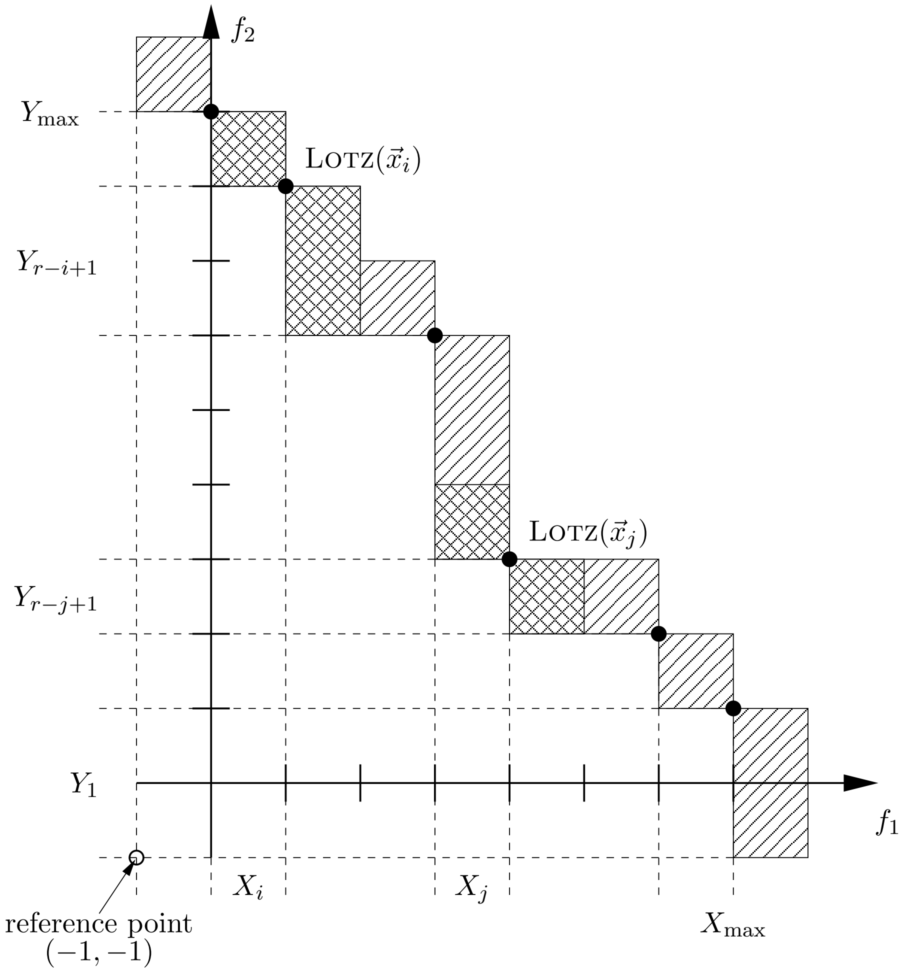


Figure 37: The hypervolume indicator progress within "good" mutation steps of $(\mu+1)$-SIBEA. The area, hatched from bottom left to top right, corresponds to the hypervolume indicator progress within all "good" mutation steps. It is the union of the single hypervolume progresses for all solutions which are detailed for two solutions x_i and x_j as the cross hatched area.

$h(0) \geq 1 \geq 0$ and $h(1) \geq 1 \geq 1/5$. In general,

$$\begin{aligned} h(t) \geq h(t-1) + \sqrt{h(t-1)} &\geq \frac{(t-1)^2}{5} + \frac{t-1}{\sqrt{5}} \\ &\geq \frac{t^2}{5} + t\left(\frac{1}{\sqrt{5}} - \frac{2}{5}\right) - \left(\frac{1}{\sqrt{5}} - \frac{1}{5}\right) \geq \frac{t^2}{5}. \end{aligned}$$

holds for all $t \geq 2$ which finishes the induction.

Therefore, the expected number of iterations for the situation where no solution of the current population is Pareto-optimal is upper bounded by $O(\mu n^2)$. □

Theorem 17. *Choosing $\mu \geq n+1$, the expected optimization time of the $(\mu+1)$-SIBEA on* Lotz *is $O(\mu n^2)$.*

Proof. Using Lemma 1, the expected time until a first Pareto-optimal solution has been obtained is $O(\mu n^2)$. Due to the hypervolume-based selection and the fact that at most $n+1$ solutions are mutually non-dominated in LOTZ (Laumanns et al., 2004b), a Pareto-optimal solution that has been found with the $(\mu+1)$-SIBEA will stay forever in the population. Increasing the number of Pareto-optimal solutions in the population increases the hypervolume indicator, i.e., the highest hypervolume value is achieved if and only if the entire Pareto front is found. Therefore, there is at least one solution in the population which has a Hamming neighbor that is Pareto-optimal and not contained in the current population—unless the whole Pareto set is already found. Hence, the expected waiting time for increasing the number of Pareto-optimal solutions in the population is $O(\mu n)$. Having reached a Pareto-optimal solution for the first time at most n additional Pareto-optimal solutions have to be produced which implies that the expected time to achieve a population including all Pareto-optimal solutions is $O(\mu n^2)$. □

Although the previous proof shows that the $(\mu+1)$-strategy finds the Pareto set of LOTZ in polynomial time, this does not imply that the same behavior can be expected on other test functions or on real-world combinatorial optimization problems. Nevertheless, the analysis showed that the technique of considering the expected hypervolume indicator progress is applicable to the running time analysis of hypervolume-based algorithms which might be also useful for the analysis of other problems and more complicated algorithms in the near future. Let us also mention that the above result will also hold if we use the additional Pareto ranking of Algorithm 9 instead of the environmental selection of $(\mu+1)$-SIBEA since the proof only argues about the set of non-dominated solutions. However, more general results are necessary in the future to argue in favor of hypervolume-based algorithms from a theoretical point-of-view.

4.3 The Weighted Hypervolume Indicator

As we already argued above, the hypervolume indicator is often used in practice mainly because of its property of being a refinement of the weak Pareto dominance relation. However, optimizing the hypervolume indicator instead of solely using the weak Pareto dominance relation to guide the search introduces a certain search bias, i.e., certain solution sets are preferred over others even if they are incomparable with respect to the weak Pareto dominance relation, which is maybe not desired from a decision maker's perspective. Auger et al. (2009c) investigate this bias of the hypervolume indicator in terms of optimal μ-distributions: how are μ solutions distributed on the Pareto front if they maximize the hypervolume indicator?

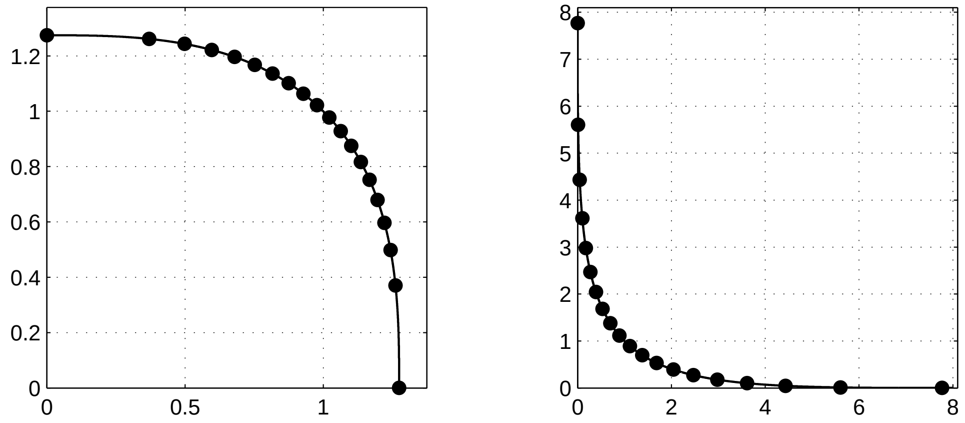


Figure 38: Illustration of the hypervolume bias towards regions of the front where the derivative is -1. The plots show the distribution of $\mu = 20$ points maximizing the hypervolume indicator for two different front shapes.

One result of their theoretical investigations for 2-objective problems is that the density of points on the Pareto front only depends on the derivative or angle of the front; in other words the hypervolume indicator favors points that are lying in Pareto front regions that have an angle of 45°, see Fig. 38 for an illustration. Since this specific property of the hypervolume indicator does not always comply with the preferences of the decision maker, the question arises, whether one can design an indicator function that is both a refinement of the weak Pareto dominance relation and that can handle specific user preferences such as a bias towards extreme solutions or a preference towards pre-defined *aspiration points*, also called *preference points*[7] (Miettinen, 1999). In this section, we introduce a general weighted hypervolume indicator that has these two properties. Furthermore, we propose three different examples of user preferences and show how they can be articulated within this weighted hypervolume approach on biobjective problem. A subsequent experimental validation shows the usefulness of this approach and indicates deficiencies and future research directions.

[7] Although the usual term in multicriteria decision making is reference point (Miettinen, 1999), we will use the term preference point throughout the text to not confuse the reader with the hypervolume indicator's reference point.

4.3.1 Introductory Example and Outline of the Proposed Approach

The basis of the weighted hypervolume indicator is the definition of the standard hypervolume indicator via the attainment function in Def. 14:

$$I_H(A) := \int_{(0,\ldots,0)}^{(1,\ldots,1)} \alpha_A(\vec{z}) d\vec{z}$$

where we set, without loss of generality, the reference point of the hypervolume to $\vec{r} = (0, \ldots, 0)$ and the upper limit of the integral to $(1, \ldots, 1)$, i.e., we assume an objective space of $Z = (0, 1)^k$. Requiring the objective values to lay between 0 and 1 instead of using $\mathbb{R}^k$ as objective space simplifies the following discussions, but does not represent a serious limitation as there exists a bijective mapping from $\mathbb{R}$ into the open interval $(0, 1) \subset \mathbb{R}$. We recapitulate that the attainment function $\alpha_A(\vec{x})$ for a given solution set $A \subseteq X$ is a binary valued function: all weakly dominated objective vectors are assigned 1, while the remaining objective vectors are assigned 0. That means all weakly dominated objective vectors have the same weight and contribute equally to the overall indicator value.

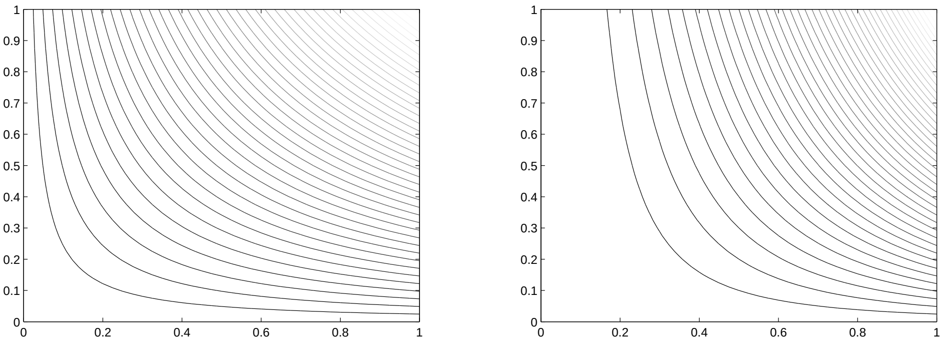
The main idea behind the approach proposed in this section is to assign different weights to different regions in the objective space Z. This can be achieved by defining a weight distribution over the objective space such that the value that a particular weakly dominated objective vector contributes to the overall indicator value can be any real value strictly greater than 0—provided the integral over the resulting function still exists. To this end, we introduce a weight distribution function $w : Z \to \mathbb{R}_{>0}$, and the hypervolume is calculated as the integral over the product of the weight distribution function and the attainment function:

$$I_H^w(A) := \int_{(0,\ldots,0)}^{(1,\ldots,1)} w(\vec{z}) \cdot \alpha_A(\vec{z}) d\vec{z}$$

As we show later, this allows to modify the standard hypervolume indicator such that (a) the new weighted hypervolume indicator is still a refinement of the weak Pareto dominance relation and (b) preference information can be flexibly introduced.

To see how different weight distribution functions affect the behavior of the corresponding modified hypervolume indicator I_H^w, it is helpful to consider equi-indicator surfaces. An equi-indicator surface $S(I, K)$ for a given indicator function I and an indicator value K is defined as the set of points $\vec{z} \in Z$ that all have the same indicator value K, i.e.,

$$S(I, K) = \{\vec{z} \in Z \,:\, I(\{\vec{z}\}) = K\} \ .$$



(a) The hypervolume indicator

(b) A biased indicator

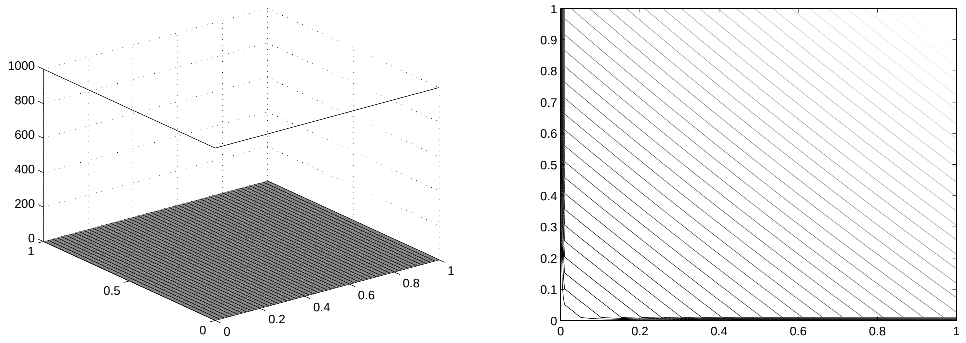
Figure 39: Equi-indicator surfaces for simple indicators in the biobjective case. The abscissa in these two-dimensional examples denotes f_1 and the ordinate f_2. Figure (a) shows (a sample of) surfaces for the hypervolume indicator I_H (weight distribution function $w((z_1, z_2)) = 1$), Figure (b) illustrates a biased, modified indicator with weight distribution function $w((z_1, z_2)) = z_1$. Points on one equipotential curve share the same indicator value.

In other words, the equi-indicator surfaces indicate the quality of single objective vectors in terms of the hypervolume indicator value assigned to them if the hypervolume indicator is considered as the optimization criterion.

If we consider a uniform weight distribution function with $w(\vec{z}) = 1$ for all $\vec{z} \in Z$, we obtain the standard hypervolume indicator I_H. In this case, the equi-indicator surfaces look for $k = 2$ objectives as depicted in Fig. 39a. Due to the convex equi-indicator surfaces, we can conclude that for single solutions, there is a bias towards the diagonal—solutions lying close to the diagonal have higher hypervolume values than solutions with the same distance to the hypervolume's reference point that lie not so close to the diagonal. That this bias towards the diagonal is, however, not given for solution sets of arbitrary size has been shown recently by Auger et al. (2009c) and is illustrated in Fig. 38.

If we change the weight distribution function to, for example, $w(\vec{z}) = z_1$ with $\vec{z} = (z_1, z_2, \ldots, z_n)$, then in the biobjective case the equi-indicator surfaces shown in Fig. 39b are obtained. Obviously, solutions with objective vectors that have large components in the direction of z_1 have higher hypervolume indicator values. Another possibility to introduce different user preferences is to impose special emphasis on the border of the objective space, see Fig. 40a. The objective vectors in the 'center' of the objective space have weight 1, while the objective vectors on the axes are assigned a substantially larger weight.[8] The corresponding equi-indicator surfaces are

[8] Since the borders have zero width, they will actually not influence the integral; therefore, dirac-type functions need to be used to make the border weights effective. In



(a) Weight distribution function with emphasis on the coordinate axes

(b) (Sample of the) equi-indicator surfaces for the corresponding indicator

Figure 40: Weight distribution function (left) and corresponding indicator (right) when stressing on coordinate axes.

shown in Fig. 40b. Here, the bias of the original hypervolume indicator for a single solution towards the diagonal is removed by putting more emphasis on the areas close to the coordinate axes. The above two examples illustrate how weight distribution functions on the objective space can be used to change the bias of the hypervolume indicator. Based on these informal observations, we will describe the underlying methodology for an arbitrary number of objectives next. Moreover, we provide three other examples of weight distribution functions for the biobjective case. Note that a generalization of these weight distribution functions to an arbitrary number of objectives is possible, see (Auger et al., 2009a). However, we restrict ourselves here to examples for biobjective problems due to the easier illustration of the concepts. Moreover, when generalizing the weight distribution functions to an arbitrary number of objectives, the high computation time of the exact hypervolume indicator calculation has to be taken into account, e.g., by sampling the weighted hypervolume indicator as in (Auger et al., 2009a), which lies outside the focus of this thesis.

4.3.2 Methodology: The Weighted-Integration Approach

The main concept of the approach proposed in this section is—as already mentioned above—to extend the basic hypervolume indicator by a weight distribution function $w : [0,1]^n \rightarrow \mathbb{R}_{>0}$ which serves to emphasize certain regions of the objective space:

this scenario, a dirac-type function can be interpreted as a weight function that has an infinite value at the axis and 0 elsewhere; however, the 2-dimensional integral over the entire objective space is a constant.

Definition 16 (Generalized Hypervolume Indicator)**.** *The generalized hypervolume indicator I_H^w with weight distribution function $w : [0,1]^n \rightarrow \mathbb{R}_{>0}$ is defined as the weighted integral*

$$I_H^w(A) := \int_{(0,\ldots,0)}^{(1,\ldots,1)} w(\vec{z}) \cdot \alpha_A(\vec{z}) d\vec{z}$$

where $A \subseteq X$ is a set of solutions.

If using this indicator as the basis for optimization algorithms or performance assessment tools, it would be important to know whether it is compliant with the concept of Pareto dominance. This property will be shown next.

Theorem 18. *Let $w : [0,1]^n \rightarrow \mathbb{R}_{>0}$ be a weight distribution function such that the corresponding generalized hypervolume indicator I_H^w is well-defined for all $A \subseteq X$. Then for any two arbitrary solution sets $A \subseteq X$ and $B \subseteq X$, it holds*

$$A \succeq B \ \wedge \ B \not\succeq A \Rightarrow I_H^w(A) > I_H^w(B),$$

i.e., also the indicator I_H^w is a refinement of the weak Pareto dominance relation according to (Zitzler et al., 2009).

Proof. If we have $A \succeq B \ \wedge \ B \not\succeq A$, then the following two conditions hold: $\forall \vec{y} \in B \ \exists \vec{x} \in A : \vec{x} \succeq \vec{y}$ and $\exists \vec{x} \in A \ \ \nexists \vec{y} \in B : \vec{y} \succeq \vec{x}$. Now we can easily see that the attainment functions of A and B satisfy $(\alpha_A(\vec{z}) = 1) \Rightarrow (\alpha_B(\vec{z}) = 1)$ as $A \succeq B$. Every point in the objective space that is weakly dominated by some element in B is also weakly dominated by some element in A. In addition, because of $B \not\succeq A$ there are some points in the objective space that are weakly dominated by points in A but not weakly dominated by points in B. Therefore, there exists a region $R \subset Z$ with $(\alpha_A(\vec{z}) = 1) \wedge (\alpha_B(\vec{z}) = 0)$ for $\vec{z} \in R$; in particular:

$$\int_{(0,\ldots,0)}^{(1,\ldots,1)} (\alpha_A(\vec{z}) - \alpha_B(\vec{z})) d\vec{z} > 0$$

Using the definition of the generalized hypervolume indicator and noting that $w(\vec{z}) > 0$, we find $I_H^w(A) > I_H^w(B)$. □

In order to simplify the definition of weight distribution functions and to avoid the use of dirac-type functions, we use a slightly different representation of the generalized hypervolume indicator where L line segments can be used to establish emphasis on zero-width regions such as axes. Every line segment l_i $(1 \leq i \leq L)$ is specified by a start point $\vec{s_i} \in Z$, an end point $\vec{e_i} \in Z$, and a corresponding weight distribution function $\overline{w}_i : [0,1] \rightarrow \mathbb{R}_{>0}$.

Using this notation, we can rewrite the generalized hypervolume indicator according to Def. 16 as follows

$$I_H^{w,\overline{w}_1,\overline{w}_2,\ldots,\overline{w}_L}(A) := \int_{(0,\ldots,0)}^{(1,\ldots,1)} w(\vec{z}) \cdot \alpha_A(\vec{z}) \cdot d\vec{z} +$$

$$\sum_{i\in\{1,2,\ldots,L\}} \int_0^1 \overline{w}_i(\vec{z}) \cdot \alpha_A(\vec{s}_i + t \cdot (\vec{e}_i - \vec{s}_i)) \cdot dt$$

Assuming that the weight distribution functions are chosen such that all integrals are well-defined, it is easy to see that the property proven in Theorem 18 is preserved.

In the following, we will discuss three examples of useful weight distribution functions that will also be used for an experimental validation of the new approach.

1. The first weight distribution function is the sum of two exponential functions in direction of the axes:

   $$w^{ext}(\vec{z}) = (e^{20\cdot z_1} + e^{20\cdot z_2})/(2 \cdot e^{20})$$

   with $L = 0$. The effect is an indicator with preference of extremal solutions. Because of the weight distribution function's steep slope near the two axes, a solution set with solutions crowded near the axes yields a larger indicator value than a population with solutions in the interior region of the objective space where the weight distribution function contribute only little to the indicator value.

2. The second weight distribution function focuses on the second objective by using an exponential function in f_2-direction:

   $$w^{asym}(\vec{z}) = e^{20\cdot z_2}/e^{20}$$

   In addition, the following line segment with a constant weight distribution function on the f_1-axis is used:

   $$\overline{w}_1^{asym}(\vec{z}) = 400, \quad \vec{s}_1 = (0,0), \quad \vec{e}_1 = (1,0)$$

   where $L = 1$. This combination results in an indicator preferring solutions with extreme f_2 values and an additional solution near the f_1 axis. The additional line segment along the f_1 axis used here instead of an additional exponential function in f_1 direction yields only a single additional solution lying near the f_1 axis instead of many solutions with large f_1 value as with the weight distribution function w^{ext} defined above.

3. Often, a decision maker has some idea which points in the search space are the most desirable ones. With the third weight distribution function, we can integrate such information into an indicator that is a refinement of the weak Pareto dominance relation by choosing a so-called preference point $(a, b) \in \mathbb{R}^2$ before the optimization. The weight distribution function defined below will then direct the search of hypervolume-based algorithms towards this point. Multiple preference points can be considered simultaneously by adding up the corresponding distinct weight distribution functions.

   The following weight distribution function is based on a ridge-like function through the preference point (a, b), parallel to the diagonal:

$$w^{pref}(\vec{z}) = \begin{cases} c+\frac{(2-((2(x-a))^2+(2(y-b))^2))}{(0.01+(2(x-a)-2(y-b))^2)} & \text{if } |z_1 - a| < 0.5 \wedge |z_2 - b| < 0.5 \\ c & \text{else} \end{cases}$$

   where $\vec{z} = (z_1, z_2)$ and $L = 0$, cf. Fig. 41. The constant $c > 0$ is mainly of theoretical interest to guarantee the Pareto compliance of the indicator. If c is chosen too big, the effect of the additional weight on the ridge does not allow to restrict the solutions to the objective space part near the preference point; if c, on the other hand, would be chosen as 0 the hypervolume indicator has no effect on the population if it is far away from the preference point. The constant c should therefore be chosen small in comparison to the values of the ridge but positive[9]; here, we use $c = 10^{-5}$.

The computation of the generalized hypervolume indicator is based on the hypercube representation of the hypervolume described in Sec. 4.1.1. It first partitions the whole unit hypercube $[0, 1]^k$ into smaller hyperrectangles based on the objective vectors contained in the set A, and then the weighted volumes of these hyperrectangles are added. To this end, the above weight distribution functions have been symbolically integrated using a commercial symbolic mathematics tool.

Note, that the proposed weight distribution functions as presented here are limited to biobjective problems. However, a generalization to a higher number of objectives has been proposed in the meantime in combination with a new Monte Carlo sampling approach to tackle the high running time of the exact hypervolume indicator calculation (Auger et al., 2009a). Although we here provide proof-of-principle results for the biobjective case only, one should keep in mind that the generalization to an arbitrary number of objectives is possible and shows qualitatively similar results.

[9] In the study of Auger et al. (2009a), the authors use a multivariate Gaussian to generalize the ridge idea to an arbitrary number of objectives and to circumvent the problem of choosing c.

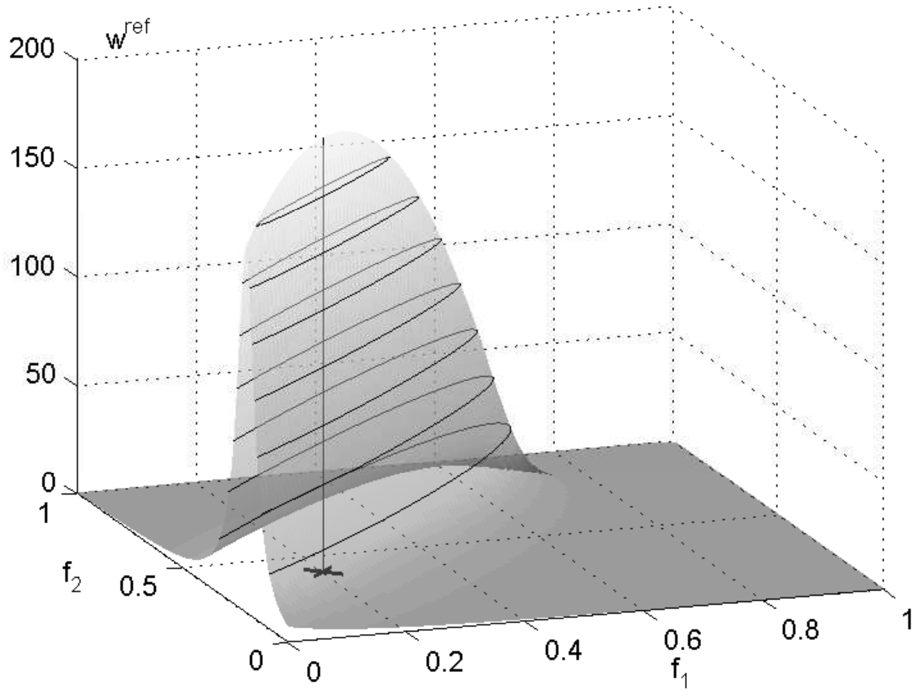
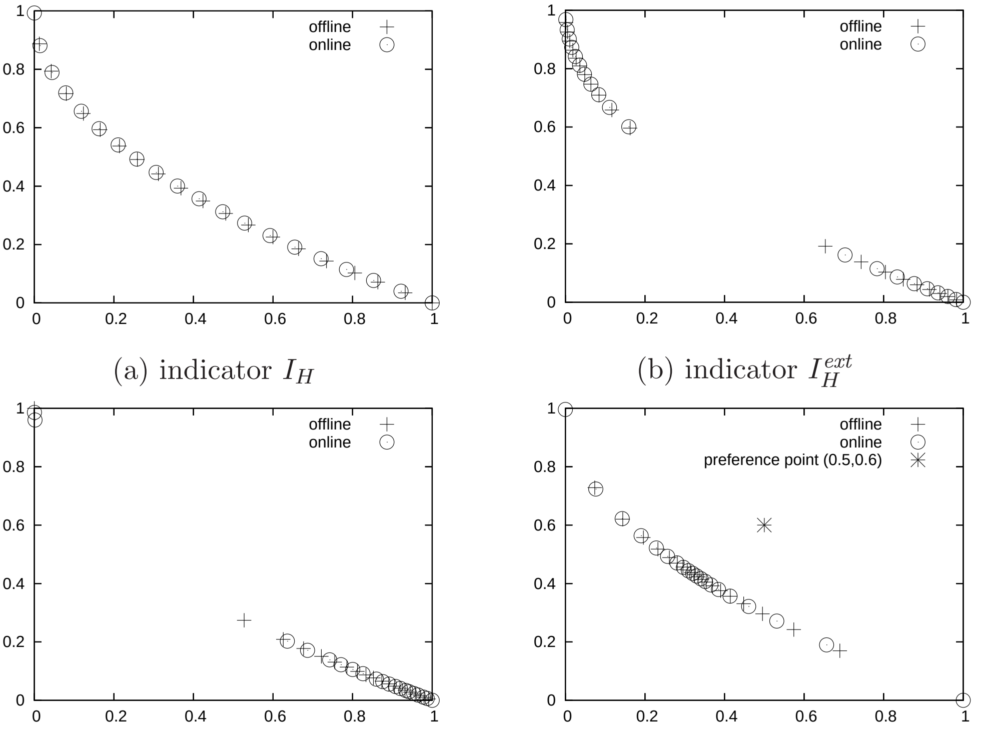


Figure 41: Illustration of the weight distribution function $w^{ref}(\vec{z})$ for the preference point $(0.2, 0.4)$ which is indicated by the cross in the middle of the ridge.

4.3.3 Proof-Of-Principle Results

We now show how the three weight distribution functions defined above influence the search process of SIBEA (Sec. 4.1.2) for three biobjective test problems.

To this end, we consider two scaling variants to obtain the maximum effect of the weighted integral: online and offline scaling. In the *online variant*, the objective function values are scaled to the interval $[0, 1]$ within each generation; to guarantee that boundary solutions contribute a nonzero hypervolume to the overall indicator value, for each axis a line segment with a constant weight distribution function is added. The *offline variant* does not scale the objective function values but the weighting distribution function. In detail, the weighted integral is only computed over and scaled to the region where the Pareto front is expected. Since any approximation set outside this region would yield an indicator value of zero, the standard hypervolume indicator value, down-scaled such that it does not interfere with the weighted integral, is added. Note that the offline variant has the drawback that the expected region of the Pareto front has to be estimated in advance which might restrict the applicability of the approach in practice. The recent generalization of the weighted hypervolume indicator approach to an arbitrary number of objectives by Auger et al. (2009a), however, does



(a) indicator I_H

(b) indicator I_H^{ext}

(c) indicator I_H^{asym}

(d) indicator I_H^{pref}

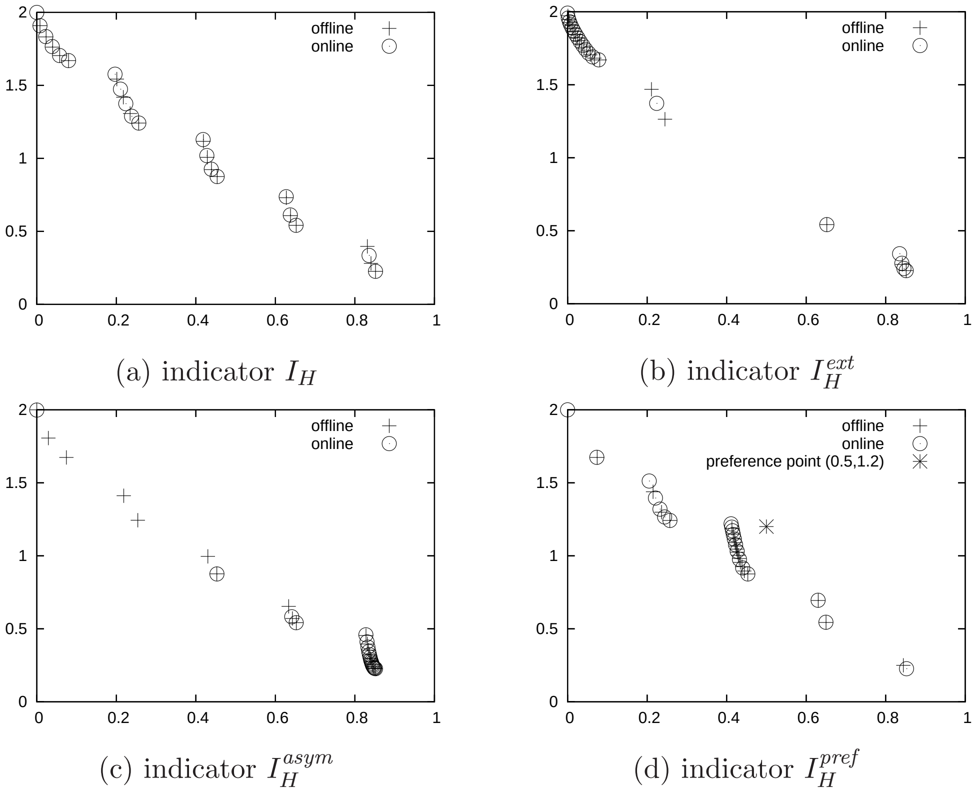
Figure 42: Pareto front approximations for the three different indicators based on weight distribution functions on the function ZDT1. For reference, the generated Pareto front approximation using the usual hypervolume indicator I_H is given in (a). The two scaling methods are plotted for comparison.

not rely on this information since no scaling is necessary anymore while the weighting distribution function is sampled.

For each of the weight distribution functions defined above, we derive two indicators, one for the online scaling method and one for offline scaling—resulting in six different indicators overall. We name the corresponding indicators I_H^{ext}, I_H^{asym}, and I_H^{pref} respectively, and distinguish between the online and the offline version. The same holds for the usual hypervolume indicator I_H, where we also distinguish between the two scaling methods.

The test functions ZDT1, ZDT3, and ZDT6 (Zitzler et al., 2000) are optimized by a SIBEA run with population size 20 for 1000 generations.[10] Note, that the ZDT functions are to be minimized. Thus, an internal transformation is performed, independent whether the online or offline scaling is enabled.

[10] The individuals are coded as real vectors with 30 (ZDT1 and ZDT3) and 10 (ZDT6 decision variables, where the SBX-20 operator is used for recombination and a polynomial distribution for mutation. The recombination and mutation probabilities were set to 1.0, according to (Deb et al., 2005).



(a) indicator I_H

(b) indicator I_H^{ext}

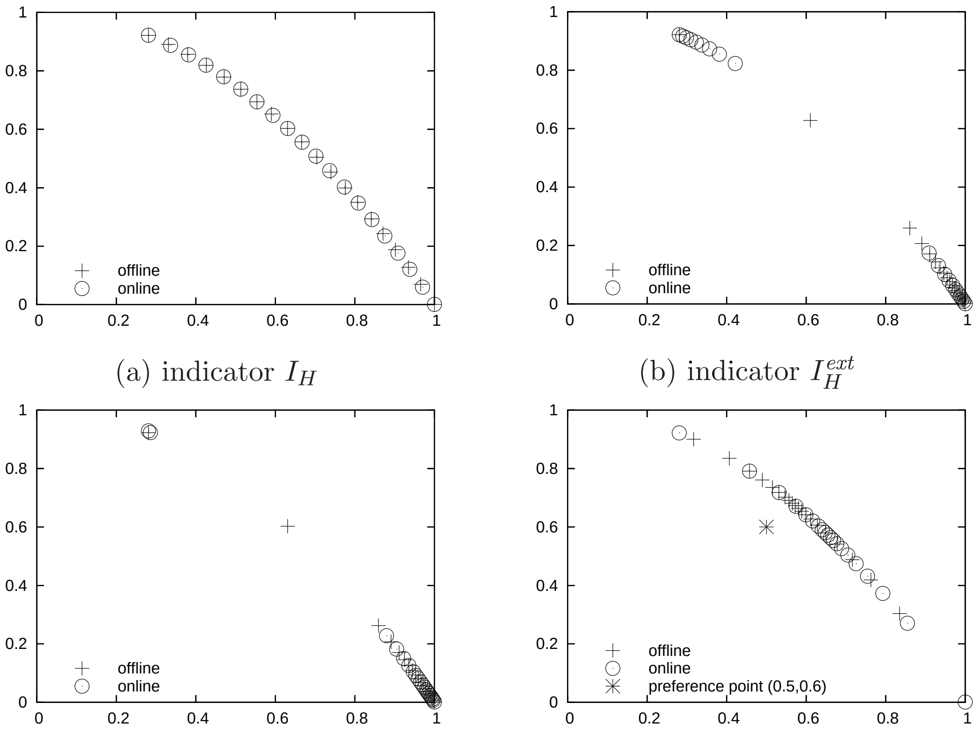
(c) indicator I_H^{asym}

(d) indicator I_H^{pref}

Figure 43: Pareto front approximations for the three different indicators and the two scaling methods on the function ZDT3. For reference, the generated Pareto front approximation using the usual hypervolume indicator I_H is given in (a).

The figures Fig. 42, Fig. 43, and Fig. 44 show the computed Pareto front approximations after 1000 generations for the three ZDT functions and the three indicators I_H^{ext}, I_H^{asym}, and I_H^{pref} with both scaling methods. The preference point for I_H^{pref} is chosen as $(0.5, 0.6)$ for ZDT1 and ZDT6 and as $(0.5, 1.2)$ for ZDT3. Due to the different front shapes, the chosen preference point lies beyond the Pareto front for ZDT6, i.e., the preference point is dominating Pareto-optimal points, or the preference point lies within the feasible region for ZDT1 and ZDT3, i.e., the preference point is dominated by some Pareto-optimal points. Furthermore, the approximation derived with the established hypervolume indicator I_H is shown as golden reference.

The experiments show two main aspects. Firstly, the behavior of the evolutionary algorithm is similar for all three problems if always the same indicator is used—independent of the front shape and the scaling method. With the indicator I_H^{ext}, the solutions accumulate near the extremal points. When using the indicator I_H^{asym}, mainly the f_2 values are minimized. Due to the additional weight on the line segment, at least one solution with large f_1 value is also kept in the population if I_H^{asym} is used. With the indicator I_H^{pref}, the population moves towards the predefined preference point



(a) indicator I_H

(b) indicator I_H^{ext}

(c) indicator I_H^{asym}

(d) indicator I_H^{pref}

Figure 44: Pareto front approximations for the three different indicators based on weight distribution functions on the function ZDT6. Plot (a) shows the generated Pareto front approximation using the usual hypervolume indicator I_H for comparison.

$(0.5, 0.6)$, and $(0.5, 1.2)$ respectively. Secondly, the weighted-integration approach seems to be feasible for designing new Pareto-compliant indicators according to specific preferences. The simple indicator-based algorithm was indeed attracted to those regions in the objective space that were particularly emphasized by means of large weight values.

When comparing the two scaling variants, online and offline, only slight differences can be observed with the test cases studied in this paper. Online scaling has the advantage that the preferences are always adapted according to the current shape of the Pareto front approximation. However, thereby the actual global indicator changes during the run and potentially cycles can occur during the optimization process—a phenomenon that emerges with several state-of-the-art algorithms such as NSGA-II and SPEA2, cf. (Laumanns et al., 2002a). Cycling is not necessarily a problem in the biobjective case, but as the number of objectives increases, it is likely that this behavior causes difficulties. The alternative is offline scaling. Here, the indicator remains fixed and can be used for comparing the outcomes of different methods. The drawback of this approach is the requirement that domain knowledge is needed: either about the location of the Pareto front or about regions of interest. This problem holds basically for all types of indicators as well as for the generalization of the approach to arbitrary objectives in (Auger et al., 2009a).

4.4 Objective Reduction for Hypervolume-Based Search

Although we have seen that hypervolume-based MOEAs can converge to the Pareto front for certain problems and that user preferences can be incorporated into the hypervolume, the shown results have been attained for biobjective problems only and it remains to tackle a general problem with most of the hypervolume-based algorithms if many objectives are considered: the best known algorithm for computing the hypervolume needs time exponentially in the number of objectives (Beume and Rudolph, 2006) and the $\#\mathcal{P}$-hardness of the hypervolume computation (Bringmann and Friedrich, 2008) underpins that no substantially faster algorithms exist under the assumption $\mathcal{P} \neq \mathcal{NP}$. To save computation time during hypervolume computation, which can be better spent in the generation of more solutions, we therefore propose to use the objective reduction techniques presented in Chapter 3 within hypervolume-based algorithms[11]. Although a reduction of the objective set during search will obviously reduce the time needed for the hypervolume computation, it is not clear whether such an objective reduction affects the quality of the found solutions. Surely, a reduction yields faster evaluations and therefore more solutions can be evaluated in the same time. But, on the other hand, an omission of objectives will cause a loss of information which might be useful during search. Hence, we propose different variants of how objective reduction strategies can be used within a hypervolume-based MOEA, namely SIBEA, and compare them in an experimental study on several test problem instances. In particular, we

- propose a general scheme to integrate objective reduction methods into a multiobjective evolutionary algorithm,
- integrate different types of objective reduction methods with objective sets of fixed, randomly and adaptively chosen sizes, and
- extensively compare the proposed algorithmic variants with and without objective reduction experimentally;

[11] Also the recently proposed idea of using Monte Carlo sampling can reduce the running time of hypervolume-based MOEAs to a reasonable one (Bader et al., 2008; Bader and Zitzler, 2009; Auger et al., 2009a).

- furthermore, also the aggregation of objectives during search is investigated and compared to the omission of objectives.

The study indicates that the (temporary) reduction of the number of objectives can improve hypervolume-based MOEAs drastically in terms of the achieved hypervolume indicator values.

4.4.1 Objective Reduction During Search

The simplest possible objective reduction method is to decide in advance which of the k objectives are considered during the search. The decision can either be driven by preferences of a human decision maker or via a dimensionality reduction technique such as PCA, applied to a set of randomly chosen solutions. The former approach has the drawback that it is often too little known about the problem such that deciding which objectives to take for optimization is difficult. Thus, we would prefer an automated technique independent of any human preference as the latter is. Nevertheless, there is still a drawback: it might be the case that in different stages of the optimization, different objectives are required to reduce the distance to the Pareto front. In terms of conflicting objectives, this would mean that an objective pair might be non-conflicting in the beginning, e.g., with respect to randomly drawn solutions, but near the Pareto front, the two objectives are conflicting implying that both objectives should be considered together to cover the entire Pareto front. Therefore, we limit the discussion to objective reduction methods applied during search.

One recently proposed approach, already discussed in the previous chapter, is based on PCA and was intended to be used for problems with many objectives the Pareto front of which, however, has a lower dimension (Deb and Saxena, 2006). The authors present a procedure to extract from a solution set the objectives which preserve most of the objective correlation. The objective reduction is integrated into the Non-dominated Sorting Genetic Algorithm NSGA-II. After NSGA-II is run, the objective reduction procedure is applied to the algorithm's outcome and the algorithm is started again while optimizing only the objectives in the computed objective set. This loop of running NSGA-II and applying the objective reduction technique afterwards is repeated until the number of objectives cannot be further reduced. Although the approach was originally intended for high-dimensional problems with low-dimensional Pareto fronts, the objective reduction procedure can also be used to reduce the number of objectives in general. A problem of the correlation-based objective reduction is the unpredictable effect on the Pareto dominance relation. The relation-based objective reduction approach, discussed in Chapter 3, however, can predict whether an error in the dominance relation occurs while the number of objectives is reduced. In the following, we present how the objective reduction approaches of Chapter 3 can be integrated into a hypervolume-based MOEA to improve the quality of its outcomes.

The basis of the incorporation of objective reduction methods into the search is again the algorithm SIBEA introduced in Sec. 4.1.2. Algorithm 11 shows again the pseudocode of SIBEA, here extended with a general objective reduction functionality. The differences to Algorithm 9 are highlighted in gray. The main difference[12] to the original SIBEA version as defined in Sec. 4.1.2 is the addition of Step 2, in which every G generations a (sub-)set of objectives is computed with a predefined objective reduction algorithm. During the algorithm itself, only these objectives are taken into account for hypervolume computation, dominance ranking and all other instructions during the next G generations. Then, another run of the objective reduction algorithm is performed based on the solutions in the current population and their objective values for *all* original objectives.

The different approaches for reducing the objective set in Step 2, defined in the following, can be grouped into three classes: either the reduced objective set has a fixed size that is defined in advance or the number of considered objectives is dynamically changing independent of the search or it changes adaptively with respect to the hypervolume indicator values obtained during search. Later on, we also consider the effect of aggregating objectives but stick to objective omission within this section.

Fixed Objective Set Size

As the simplest objective reduction method, integrated into SIBEA, we fix the number `k` of considered objectives in advance. This allows to easily adjust the computation time of the algorithms, i.e., the smaller `k`, the faster the hypervolume computation. On the other hand, it is not easy to control the quality of the computed Pareto front approximations by changing the parameter `k`. A low `k` will often yield worse results with respect to the population's hypervolume. In preliminary experiments, a reduction to 2 to 5 objectives was reasonable. When considering more than 5 objectives, the hypervolume computation becomes too time-consuming while considering at least two objectives seems to be required for optimization.

Besides a random version, where the objective sets of fixed size $\mathtt{k} = 3$ are always chosen randomly, the greedy `k-EMOSS` Algorithm 6 is used to compute an objective set with predefined size and the smallest possible δ-error. Other objective reduction algorithms can also be used but are not considered in the following experiments.

[12] Furthermore, the stopping criterion is changed to compare different algorithm variants with respect to their actual running time T (in seconds instead of generations).

Algorithm 11 SIBEA with Objective Reduction

Input: population size μ; indicator function I;

running time T in seconds; reduction frequency G in generations;

Output: approximation of Pareto-optimal set A;

Step 1 (Initialization):

Generate an initial (multi-)set of solutions P of size μ uniformly at random from X; set the current time t_0; set generation counter $m := 0$.

Step 2 (Dimensionality reduction):

If $m \equiv 0 \bmod G$: Use the objective vectors of all solutions in P to decide which objectives to consider in the following G generations while here *all* objectives are taken into account.

Step 3 (Environmental Selection):

Iterate the following three steps until the size of the population does no longer exceed μ:

1. Rank the population using Pareto dominance and determine the set of individuals $P' \subseteq P$ with the worst rank. Here, dominance depth (Zitzler et al., 2004) is used.

2. For each solution $\vec{x} \in P'$ determine the loss $d(\vec{x})$ with respect to the indicator I if it is removed from P', i.e., $d(\vec{x}) := I(P') - I(P' \setminus \{\vec{x}\})$.

3. Remove the solution with the smallest loss $d(\vec{x})$ from the population P (ties are broken randomly).

Step 4 (Termination):

If T seconds expired since t_0 then set $A := P$ and stop; otherwise set $m := m + 1$.

Step 5 (Mating):

Randomly select elements from P to form a temporary mating pool Q of size μ. Apply recombination and mutation operators to the mating pool Q which yields Q'. Set $P := P + Q'$ (multi-set union). Continue with Step 2.

Dynamically Changing the Objective Set Size

To avoid the difficult choice of the objective set size in the methods described above, the three methods, described in this section, dynamically choose the number of considered objectives. Two methods choose the objective set size randomly. More precisely, a geometrically distributed random number with $p = 0.5$ is drawn as the number `k` of considered objectives and set to the number of original objectives if the drawn `k` exceeds the number of original objectives. Thus, the expected objective set size is approximately[13] 2. We consider two different versions where the objectives are either chosen randomly or according to the greedy Algorithm 6 for `k-EMOSS`. Also here, other methods to reduce the number of objectives could be used. The third method chooses the objective set (and its size) according to a given δ-error by applying Algorithm 5 on the current population. A drawback of this reduction method is the δ-error as additional parameter since it is not clear how to choose δ in advance. While the current population is scaled to $[0, 1]^k$ before the objective vectors are used as input for Algorithm 5, preliminary experiments suggested to use δ-values not too low, e.g., $\delta = 0.8$ or $\delta = 0.9$.

Adaptively Increasing Objective Sets

Starting with one objective, we increase in this type of objective reduction the objective set size adaptively every G generations, dependent on a hypervolume improvement. If and only if the hypervolume indicator of the entire population increased within the last G generations in at most $G/10$ generations, the objective set size is increased by one. The idea behind this is that SIBEA can optimize the selected objectives as long as it can improve the population's hypervolume indicator easily. If the algorithm gets stuck, we increase the number of considered objectives to improve the hypervolume of the population and guarantee that high-dimensional Pareto fronts can be found by the algorithm. In contrast to the PCA-based objective reduction method in (Deb and Saxena, 2006) where the objective sets become only smaller while converging to the Pareto-optimal front, here the opposite happens.

We distinguish between one random version which chooses the larger objective sets always randomly and a second version which uses Algorithm 6 to compute the new objective set. Note, that the objective set is not changed unless the number of objectives is increased.

A similar idea of increasing the number of objectives during search incrementally has been already proposed by Chen and Guan (2004) where the order in which the objectives are added is independent of the problem characteristics. Here, the choice which objective to additionally optimize depends on the Pareto dominance relation between the population's individuals. Furthermore, in the last phase of the approach in (Chen and Guan, 2004), always all objectives are optimized which is not necessarily the case in the adaptive method described here.

[13] In case we would not restrict `k` to be smaller than or equal to the number of original objectives, the expected value of `k` would be exactly 2.

4.4.2 Experimental Validation of Objective Omission During Search

4.4.2.1 Setup

For an experimental comparison of the different objective reduction methods, the hypervolume indicator values, obtained in the first generation after a time limit of $T = 20$ minutes has been reached, are compared. In addition to the algorithm versions of SIBEA described in the previous section, the basic SIBEA without any objective reduction as well as NSGA-II (Deb et al., 2002) and SPEA2 (Zitzler et al., 2002) have been used in the comparison. As a baseline, random search space samples of size μ have been performed as well. Table 5 gives an overview over all algorithms used in the comparison.

For each combination of algorithm and problem, 21 runs have been performed on identical linux machines (4 cores, 64bit architecture, 2.6GHz). As test problems, we use a modified version of the DTLZ2 problem, denoted by DTLZ2$_{\text{BZ}}$ and defined in the following section, as well as the original DTLZ7 problem of Deb et al. (2005). We use $n = 200$ decision variables and problem instances with 5, 7, and 9 objectives. The implementation of SIBEA is based on the hypervolume indicator in the PISA framework (Bleuler et al., 2003) and, thus, uses only an asymptotically slow hypervolume algorithm, instead of state-of-the-art approaches like in (Beume and Rudolph, 2006). The reason is that the study at hand should justify the usage of objective reduction techniques during the search only. Using a faster algorithm would not conceptually change the results but might only make it necessary to use problems with a higher number of objectives to show similar effects. Moreover, the actual implementation in the PISA framework turned out to be faster than the asymptotically faster but more complicated algorithm of Beume and Rudolph (2006) for reasonable instances in a preliminary study. As to the implementations of NSGA-II and SPEA2, the versions from the PISA framework has been used as well. All algorithms are used with standard settings, whereas the population size is always set to $\mu = 50$ and the objective reduction frequency G equals 50.

To analyze the quality of the produced Pareto front approximations, we compute for all runs the hypervolume indicator of the first generation after the predefined time of $T = 20$ minutes elapses. Note, that the hypervolume indicator values are computed with respect to all original objectives. Note further that for some of the algorithms, e.g., SIBEA without any objective reduction on the high-dimensional problems, the time needed for the hypervolume computation in one generation was rather a day than about 20 minutes. The reference points for the hypervolume computation are chosen as $(50, \ldots, 50)$ for DTLZ2$_{\text{BZ}}$ and $(100, \ldots, 100)$ for DTLZ7 and the hyper-

Table 5: All algorithms used in the experimental comparison of different objective reduction approaches during search.

algorithm	objective set size	objective reduction method
SIBEA	complete	—
SIBEA	k = 3 fix	k-EMOSS
SIBEA	k = 4 fix	k-EMOSS
SIBEA	k = 3 fix	random
SIBEA	dynamic	0.8-MOSS
SIBEA	dynamic	0.9-MOSS
SIBEA	dynamic	random
SIBEA	dynamic	k-EMOSS
SIBEA	adaptive	random
SIBEA	adaptive	k-EMOSS
NSGA-II	complete	—
SPEA2	complete	—

volume is to be maximized[14]. The restriction of the running times to $T = 20$ minutes is arbitrary and similar results can be obtained for similar T.

For comparing the different algorithms, hypotheses have been derived from preliminary experiments which are tested on the performed 21 independent runs. As statistical test, we use the non-parametric Kruskal-Wallis test followed by the Conover-Inman procedure for multiple testing (see pages 288–290 of (Conover, 1999)) to support the hypothesis that one algorithm "systematically" produces larger hypervolume indicator values than another one by ranking all values and comparing the rank sums. The significance level has been set to $p = 0.05$. When reporting the results, we both show the ranking of the median values for different algorithms as well as the ranking obtained by the statistical tests. If no statistical difference in the median value can be detected by the Conover-Inman procedure, the two algorithms get the same ranking; otherwise, the algorithm with the higher median in the hypervolume indicator values gets the smaller rank. Smaller values and therefore higher hypervolume indicator values are always better.

[14] The reference points are chosen in a way that all possible objective vectors for all chosen numbers of objectives have a positive hypervolume contribution. This choice corresponds to selecting the reference point's objective values strictly larger than the maximally attainable objective function values for all choices of the number of objectives.

4.4.2.2 Modified DTLZ Test Problems

To compare the different objective reduction methods, we use test problems based on the DTLZ test suite of Deb et al. (2005)[15]. Many functions within the original test function suite, especially the often used DTLZ2 function, have the properties that

1. the projection of the Pareto front to fewer than k objectives collapses to one optimal point, i.e., when omitting arbitrary objectives, the search will always converge to one solution. Every multiobjective function has this property if all objectives except one are omitted. For the DTLZ function suite, however, this property even holds for every subset of objectives. The second drawback is that
2. when optimizing only a subset of fewer than k objectives, the neglected objectives are also optimized at the same time. The reason is the scaling of all objectives by a function $g(\vec{x}_M)$, indicating the distance to the real Pareto front.

We believe that real-world problems do not show these two properties and that eliminating them for the experimental study to follow will better show the usefulness of the proposed objective reduction methods in practical applications.

To eliminate the mentioned properties, we modify the original DTLZ2 function in two ways. First, we limit the range of the decision variables x_i, i.e., we cut the corners of the non-dominated fronts to circumvent property 1). In detail, we use $x_i/2 + 1/4$ instead of the decision variables x_i directly as in the original version. Figure 45 shows the changes on the resulting Pareto front exemplary for the 3-objective DTLZ2 problem together with a projection to 2 objectives.

To come up with a problem where all single objectives have to be optimized simultaneously to reach the Pareto front, i.e., avoiding drawback 2), we, secondly, introduce one scaling function $g_i(\vec{x})$ for each objective, instead of one single scaling function $g(\vec{x}_M)$. Figure 46 shows the definition of the modified DTLZ2 problem that we will call DTLZ2$_{\mathrm{BZ}}$ in the following. Figure 47 shows the formulation of the modified version of DTLZ3 that we denote by DTLZ3$_{\mathrm{BZ}}$.

Since the original DTLZ7 problem has both a Pareto front which does not collapse to a single point when projected and objective functions to which independent decision variables are associated with, we use the original version from (Deb et al., 2005) as a third test function in the following experiments. Figure 48 shows the formal definition of DTLZ7.

[15] Here, we already introduce all three test problems that will be used in the remainder of this chapter, although only two of them will be used for the comparison in this section.

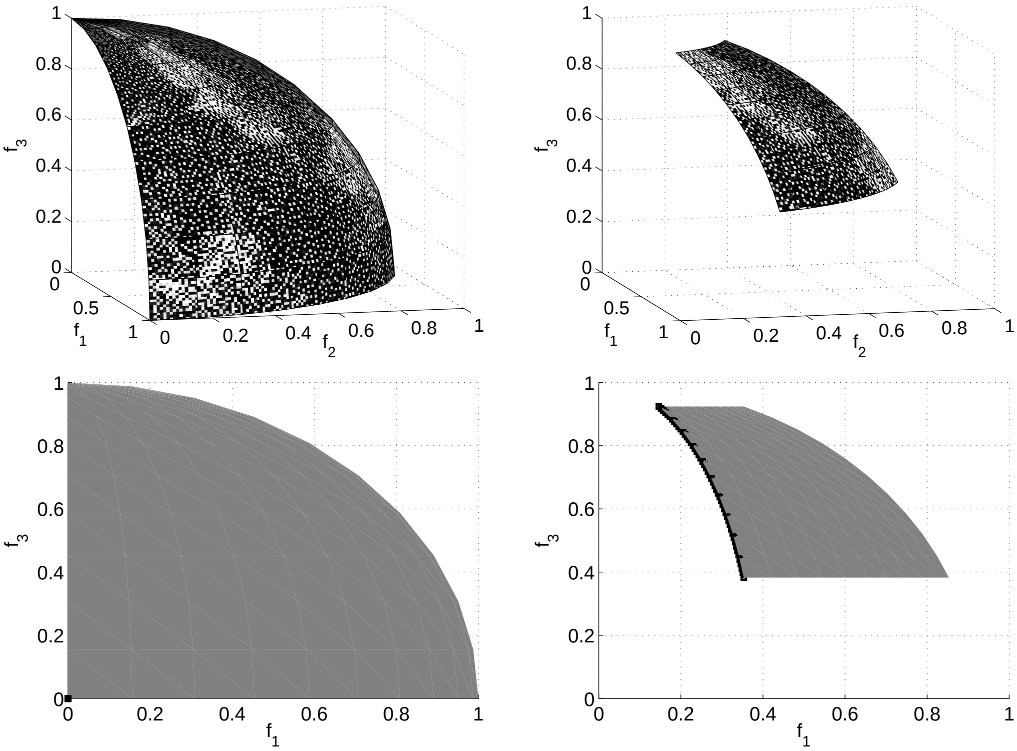


Figure 45: Visual comparison between the Pareto fronts of original DTLZ2 (left) and modified DTLZ2$_{\text{BZ}}$ (right). The first row shows the Pareto fronts for the three-dimensional problems, whereas the second row shows the same fronts projected to the f_1/f_3 plane: if objective f_2 is omitted during optimization, the front collapses to a single point, depicted in black, for DTLZ2 (left) and to a one-dimensional trade-off front (black) for the modified DTLZ2$_{\text{BZ}}$ (right). Note that the surfaces of the Pareto fronts are textured for illustration purpose.

In addition, we finally scale the objective values for the three test functions, since in general not all objectives are equally scaled in practical problems. With the scaling

$$f'_i(\vec{x}) := \begin{cases} \text{maxValue} \cdot \left(\frac{f_i(\vec{x})}{\text{maxValue}}\right)^i & \text{if } i \text{ is even} \\ \text{maxValue} \cdot \left(\frac{f_i(\vec{x})}{\text{maxValue}}\right)^{1/i} & \text{if } i \text{ is odd} \end{cases},$$

we change the ratio between the different objective values to account for those differences in practice, whereas maxValue $= 1 + ((n-k+1)/4)$ for DTLZ2$_{\text{BZ}}$, maxValue $= 1{+}125(n{-}k{+}1)$ for DTLZ3$_{\text{BZ}}$, and maxValue $= 11k$ for DTLZ7 are the maximum objective values for the original problems. Near the Pareto front, objectives with even number have larger variances than objectives with odd number. Far from the Pareto front, odd and even objectives invert their behavior.

$$
\begin{array}{ll}
\text{Min} & f_1(\mathbf{x}) = (1 + g_1(\vec{x}))\cos(\theta_1)\cdots\cos(\theta_{k-2})\cos(\theta_{k-1}),\\
\text{Min} & f_2(\mathbf{x}) = (1 + g_2(\vec{x}))\cos(\theta_1)\cdots\cos(\theta_{k-2})\sin(\theta_{k-1}),\\
\vdots & \vdots\\
\text{Min} & f_{k-1}(\mathbf{x}) = (1 + g_{k-1}(\vec{x}))\cos(\theta_1)\sin(\theta_2),\\
\text{Min} & f_k(\mathbf{x}) = (1 + g_k(\vec{x}))\sin(\theta_1),\\
\multicolumn{2}{l}{\text{where } g_i(\vec{x}) = \sum_{j=k+(i-1)\cdot\left\lfloor\frac{n-k+1}{k}\right\rfloor}^{k+i\cdot\left\lfloor\frac{n-k+1}{k}\right\rfloor-1}(x_j - 0.5)^2}\\
 & \quad\text{for } i = 1,\ldots,k-1,\\
 & g_k(\vec{x}) = \sum_{j=k+(k-1)\cdot\left\lfloor\frac{n-k+1}{k}\right\rfloor}^{n}(x_j - 0.5)^2,\\
 & \theta_i = \frac{\pi}{2}\cdot\left(\frac{x_i}{2} + \frac{1}{4}\right) \text{ for } i = 1,\ldots,k-1\\
 & 0 \le x_i \le 1, \quad \text{for } i = 1,2,\ldots,n.
\end{array}
$$

Figure 46: Definition of the modified DTLZ function DTLZ2$_{\text{BZ}}$ with n decision variables and k objectives.

$$
\begin{array}{ll}
\text{Min} & f_1(\mathbf{x}) = (1 + g_1(\vec{x}))\cos(\theta_1)\cdots\cos(\theta_{k-2})\cos(\theta_{k-1}),\\
\text{Min} & f_2(\mathbf{x}) = (1 + g_2(\vec{x}))\cos(\theta_1)\cdots\cos(\theta_{k-2})\sin(\theta_{k-1}),\\
\vdots & \vdots\\
\text{Min} & f_{k-1}(\mathbf{x}) = (1 + g_{k-1}(\vec{x}))\cos(\theta_1)\sin(\theta_2),\\
\text{Min} & f_k(\mathbf{x}) = (1 + g_k(\vec{x}))\sin(\theta_1),\\
\multicolumn{2}{l}{\text{where } g_i(\vec{x}) = 100\cdot\left[\left\lfloor\frac{n-k+1}{k}\right\rfloor + \sum_{j=k+(i-1)\cdot\left\lfloor\frac{n-k+1}{k}\right\rfloor}^{k+i\cdot\left\lfloor\frac{n-k+1}{k}\right\rfloor-1}(x_j - 0.5)^2 - \cos(20\pi(x_j - 0.5))\right]}\\
 & \quad\text{for } i = 1,\ldots,k-1,\\
 & g_k(\vec{x}) = 100\cdot\left[n - \left(k + (k-1)\cdot\left\lfloor\frac{n-k+1}{k}\right\rfloor\right)\right.\\
 & \qquad\left. + \sum_{j=k+(k-1)\cdot\left\lfloor\frac{n-k+1}{k}\right\rfloor}^{n}(x_j - 0.5)^2 - \cos(20\pi(x_j - 0.5))\right],\\
 & \theta_i = \frac{\pi}{2}\cdot\left(\frac{x_i}{2} + \frac{1}{4}\right) \text{ for } i = 1,\ldots,k-1\\
 & 0 \le x_i \le 1, \quad \text{for } i = 1,2,\ldots,n.
\end{array}
$$

Figure 47: Definition of the modified DTLZ function DTLZ3$_{\text{BZ}}$ with n decision variables and k objectives.

4.4.2.3 Results

Figures 49–54 show boxplots of the hypervolume indicators for the population obtained after the first generation that is completed after $T = 20$ minutes for all algorithms in Table 5 on the DTLZ2$_{\text{BZ}}$ and the DTLZ7 problems—only the results for all SIBEA versions with dynamic objective reduction have been omitted. The reason is the high running time of these algorithms, such that not for all test problem instances 21 runs could be performed in reasonable time. We discuss this in more detail in the follow-

$$
\begin{array}{ll}
\text{Min} & f_1(\mathbf{x}) = x_1, \\
\text{Min} & f_2(\mathbf{x}) = x_2, \\
\vdots & \vdots \\
\text{Min} & f_{k-1}(\mathbf{x}) = x_{k-1}, \\
\text{Min} & f_k(\mathbf{x}) = (1 + g(\vec{x}_M))h(f_1, f_2, \ldots, f_{k-1}, g), \\
\multicolumn{2}{l}{\text{where } g(\vec{x}_M) = 1 + \frac{9}{|\vec{x}_M|} \sum_{x_i \in \vec{x}_M} x_i,} \\
 & h(f_1, f_2, \ldots, f_{k-1}, g) = k - \sum_{i=1}^{k-1} \left[\frac{f_i}{1+g}(1 + \sin(3\pi f_i)) \right] \\
 & 0 \le x_i \le 1, \quad \text{for } i = 1, 2, \ldots, n.
\end{array}
$$

Figure 48: Definition of the non-modified DTLZ7 function with n decision variables and k objectives. As in the original definition, $\vec{x}_M$ is defined as the last $n - k + 1$ decision variables $x_k, \ldots, x_n$.

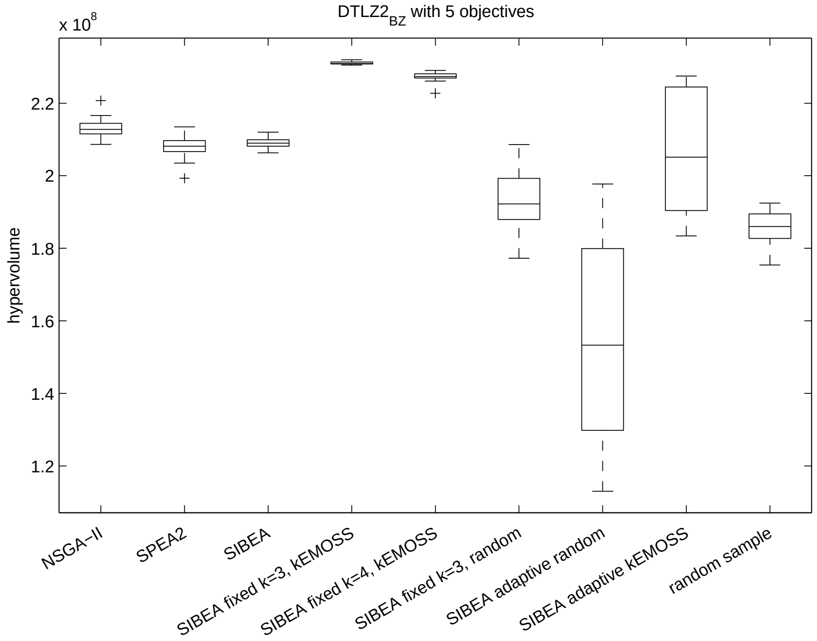


Figure 49: Boxplot comparing selected algorithms on DTLZ2$_{\text{BZ}}$ with 5 objectives. The hypervolume is upper bounded by $3.125 \cdot 10^8$.

ing paragraphs. These SIBEA versions have been also excluded from the statistic test procedure for the same reason of incomplete results due to the high running times. The results of the statistical test, however, can be found in Table 6 where the ranking of the medians of the hypervolume indicator values (in brackets) and the number of algorithms that significantly produce higher hypervolume values are shown. In both cases, a lower rank is better. In the following, we present the results of the comparison in detail.

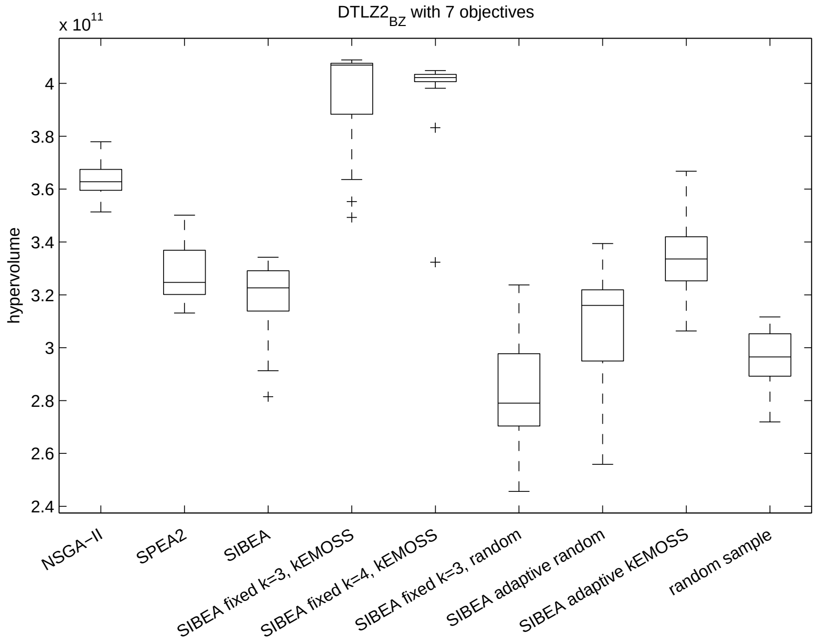


Figure 50: Boxplot comparing selected algorithms on DTLZ2$_{\mathrm{BZ}}$ with 7 objectives. The hypervolume is upper bounded by $7.8125 \cdot 10^{11}$.

SIBEA vs. NSGA-II vs. SPEA2

The experiments with SIBEA confirm, that hypervolume-based evolutionary algorithms are sensitive to the number of objectives, i.e., the running time for the hypervolume indicator computation highly depends on the number of objectives. Up to four objectives are manageable with the used population size of 50, whereas the computation time explodes to more than a day per generation for the 9-objective problems. In the allowed time interval of 20 minutes for example, SIBEA managed to reach not more than 16, 5, and 1 generations for DTLZ2$_{\mathrm{BZ}}$ with 5, 7, and 9 objectives respectively. Nevertheless, the improvement in the hypervolume indicator values in the first generations is high compared to NSGA-II or SPEA2, cf. Fig. 55 and 56. This is mainly due to the fact that NSGA-II and SPEA2 do not optimize the hypervolume indicator.

This observation underpins that the hypervolume indicator provides additional information on the search space; the information gain per objective vector evaluation is increased compared to the usage of the weak Pareto dominance relation within NSGA-II and SPEA2. If the hypervolume computation can be accelerated, one may expect that an improvement is possible also with respect to a predefined running time as it is the case for our comparison. Since the baseline hypervolume algorithm used within SIBEA is not state-of-the-art, we cannot expect that SIBEA can compete with NSGA-II and SPEA2 compared with respect to a given time interval. However, a realistic comparison of SIBEA with NSGA-II and SPEA2 was

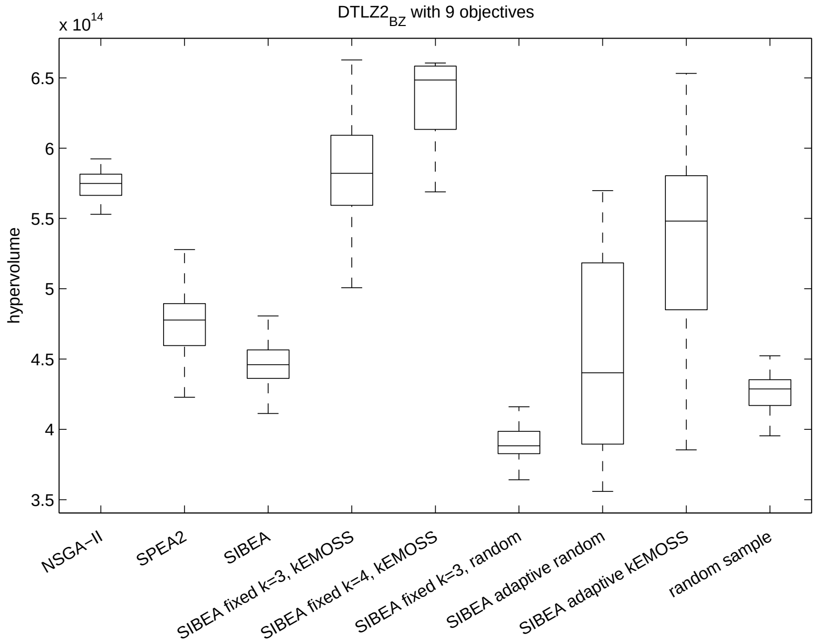


Figure 51: Boxplot comparing selected algorithms on DTLZ2$_{\mathrm{BZ}}$ with 9 objectives. The hypervolume is upper bounded by $1.954 \cdot 10^{15}$.

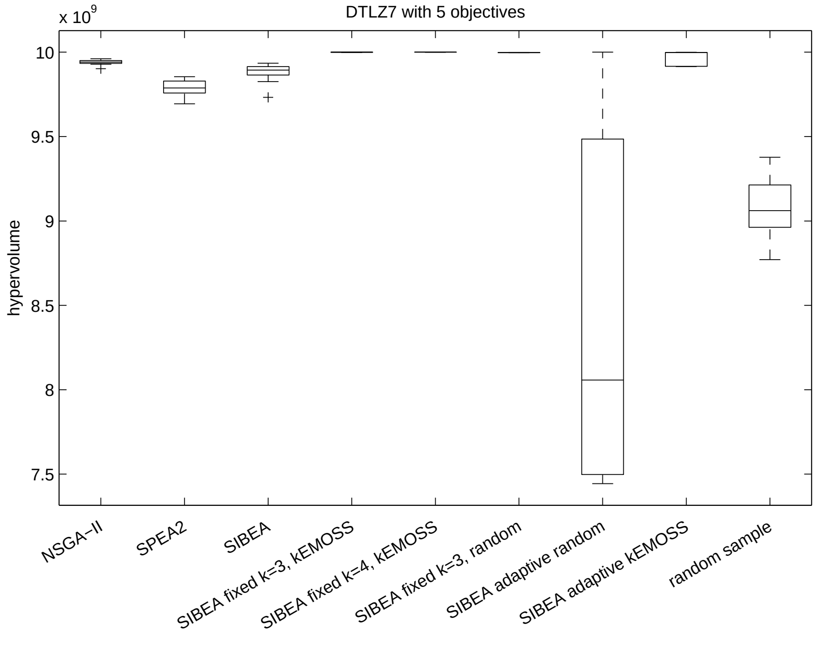


Figure 52: Boxplot comparing selected algorithms on DTLZ7 with 5 objectives. The hypervolume is upper bounded by $1 \cdot 10^{10}$.

never intended to be the focus here; in fact, we would like to emphasize the difference between the basic SIBEA without and the new versions with objective reduction strategies. That some of the objective reduction variants of SIBEA even outperformed NSGA-II and SPEA2 with respect to the Pareto-

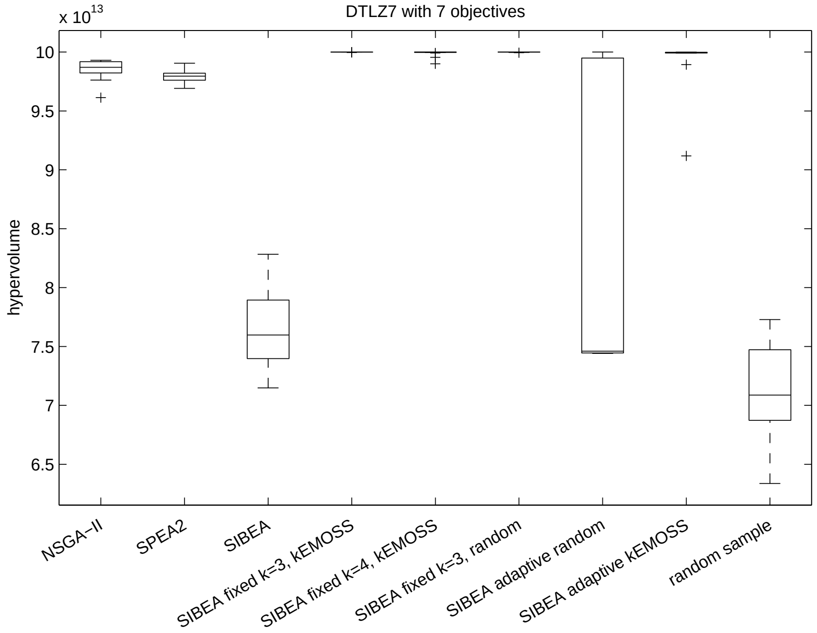


Figure 53: Boxplot comparing selected algorithms on DTLZ7 with 7 objectives. The hypervolume is upper bounded by $1 \cdot 10^{14}$.

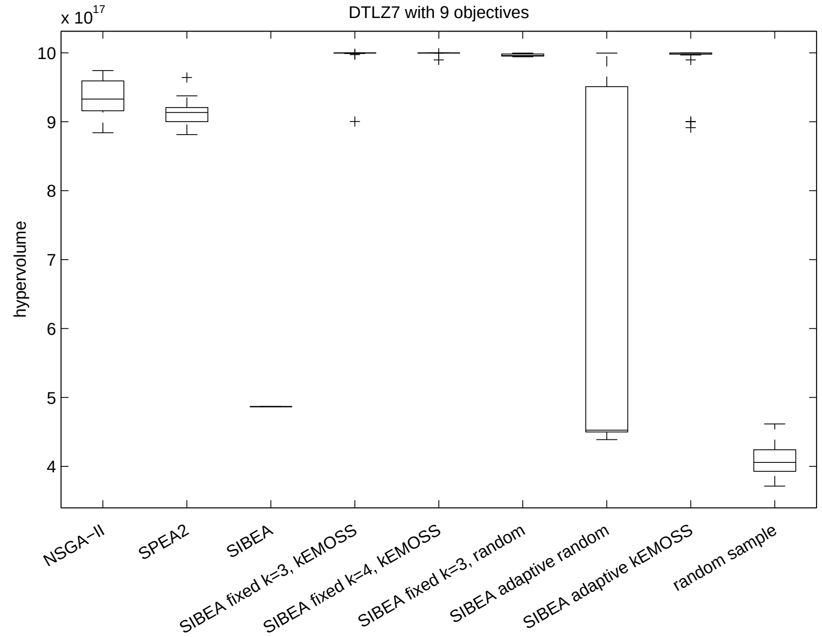


Figure 54: Boxplot comparing selected algorithms on DTLZ7 with 9 objectives. The hypervolume is upper bounded by $1 \cdot 10^{18}$. Note, that SIBEA without objective reduction needed 6435 minutes or more than 4 days to complete the first generation such that we decided to perform only one run.

Table 6: Ranking of the hypervolume indicator values for all algorithms in the comparison based on the Kruskal-Wallis test with subsequent Conover-Inman procedure. All SIBEA versions with dynamically changing objective set size are excluded (see text). The shown rank corresponds to the number of algorithms that significantly produce better hypervolume indicator values. The ranking of the medians is given in brackets. Lower values are always better.

	DTLZ2$_{\text{BZ}}$, 5 objectives	DTLZ2$_{\text{BZ}}$, 7 objectives	DTLZ2$_{\text{BZ}}$, 9 objectives	DTLZ7, 5 objectives	DTLZ7, 7 objectives	DTLZ7, 9 objectives
NSGA-II	2(3)	2(3)	1(3)	4(5)	4(5)	4(5)
SPEA2	3(5)	4(5)	4(5)	6(7)	4(6)	4(6)
SIBEA	3(4)	5(6)	5(6)	5(6)	7(7)	4(7)
SIBEA, fixed $k = 3$, k-EMOSS	0(1)	0(1)	1(2)	0(2)	0(2)	0(2)
SIBEA, fixed $k = 4$, k-EMOSS	1(2)	0(2)	0(1)	0(1)	0(3)	0(1)
SIBEA, fixed $k = 3$, random	6(7)	7(9)	8(9)	2(3)	0(1)	3(4)
SIBEA, adaptive, random	8(9)	6(7)	6(7)	5(9)	5(8)	5(8)
SIBEA, adaptive, k-EMOSS	3(6)	3(4)	3(4)	2(4)	3(4)	2(3)
random sample	7(8)	7(8)	6(8)	8(8)	8(9)	7(9)

compliant hypervolume indicator only shows again that it makes sense to consider hypervolume indicator based algorithms if the number of objectives is high—if the drawback of the high running time can be bypassed.

Remarks on the Comparison NSGA-II vs. SPEA2

Since NSGA-II and SPEA2 are not optimizing the hypervolume indicator directly, the fact that the indicator values can decrease during the search is not astonishing. While the algorithms themselves improve their population permanently, e.g., with respect to diversity, the populations of consecutive

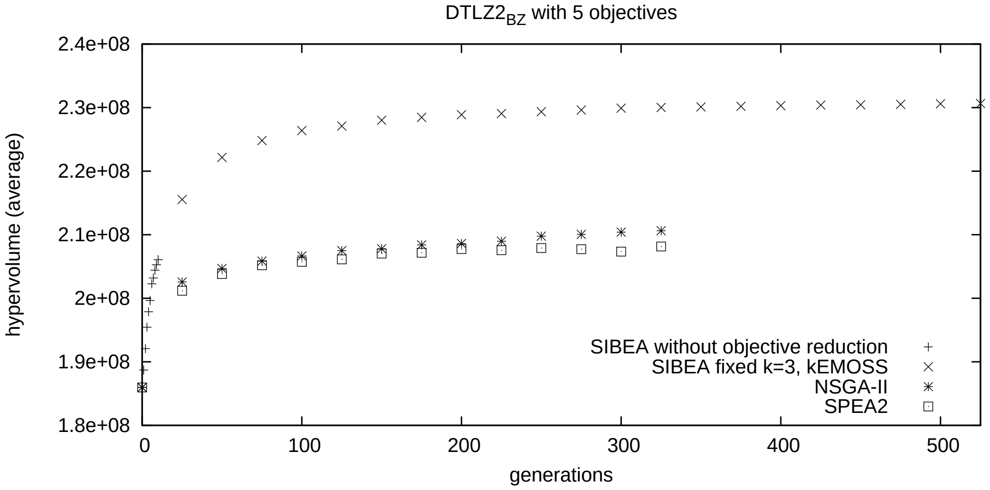


Figure 55: Course of the averaged hypervolume indicator values for selected algorithms over time on DTLZ2$_{\mathrm{BZ}}$ with 5 objectives. Note, that on the x-axis, the number of generations is plotted and not the time in seconds. Therefore, the algorithms reach different numbers of generations within the same time interval. Only the generations, all 21 runs reached, are plotted.

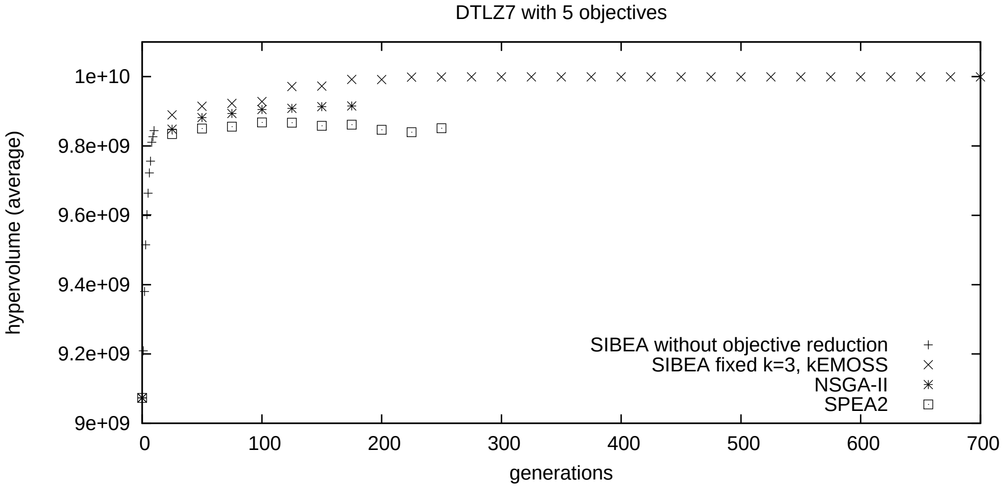


Figure 56: Course of the averaged hypervolume indicator values for selected algorithms over time on DTLZ7 with 5 objectives. As in Fig. 55, only the generations, all 21 runs reached, are plotted.

generations will often be incomparable. Therefore, the hypervolume indicator can decrease over time as can be seen in Fig. 55 and 56.

That NSGA-II obtains significantly better results than SPEA2 for all problems except DTLZ7 with 7 and 9 objectives is not surprising if we recall the conditions of the comparison: the available time is the same for all algorithms. Due to more complex computations within SPEA2, it takes more time to perform an entire step of SPEA2 than a generation lasts within NSGA-II. That means, NSGA-II is able to perform more generations than

SPEA2 within the same time interval. As NSGA-II is, e.g., able to reach generations of 1500 and more within 20 minutes on DTLZ2$_{\mathrm{BZ}}$, SPEA2 cannot process more than 860 generations within the same time. Note that in usual comparisons with respect to the number of generations, SPEA2 yields better distributions of points than NSGA-II if the number of objectives is larger than 2 whereas NSGA-II runs much faster. Here, the stopping criterion results in better results for NSGA-II than for SPEA2.

Dynamic Objective Reduction

The dynamic objective reduction strategies show mixed performance. What we can conclude is that the dynamic SIBEA versions show a high variance: with the randomly chosen objective set sizes, the dynamic strategies can at any time choose too many objectives such that the hypervolume computation will last too long to achieve competitive indicator values. This results in a ranking that is inferior in comparison to the strategies with a fixed objective set size. Thus, we advise against using the dynamic objective reduction strategies and prefer the `k-EMOSS` based strategies with fixed size and therefore predictable running time.

Objective Reduction with Fixed and Adaptively Chosen Objective Set Sizes

The objective reduction strategies with fixed `k` based on the `k-EMOSS` algorithm turn out to be the best algorithms in the comparison (always significantly better than SPEA2, NSGA-II, and SIBEA without objective reduction except for DTLZ2$_{\mathrm{BZ}}$ with 9 objectives and the comparison between NSGA-II and SIBEA with fixed $\mathtt{k} = 3$; nearly always better than the other methods). Comparing the random version and the one based on the greedy heuristic for `k-EMOSS`, the latter is always significantly better on DTLZ2$_{\mathrm{BZ}}$ where the random version cannot compete against NSGA-II and SPEA2. Between the two SIBEA variants with $\mathtt{k} = 3$ and $\mathtt{k} = 4$, no trend is visible and the statistical tests are not significant except for two examples. This indicates that both algorithms are the winner in this comparison and are recommended to be used in practice. The following more extensive comparison in the next section will, however, show that the omission algorithms with fixed `k` cannot compete against aggregation methods.

4.4.3 Aggregating Objectives During Search

If only a subset of objectives is optimized as proposed in the last section, the omitted objectives are, in general, not optimized simultaneously, i.e., the objective values of the neglected objectives can be arbitrary poor. We expect that the values of neglected objectives are kind of randomly chosen due to the fact that they are not considered within selection. In order to avoid this behavior, we suggested in an earlier work to aggregate the omitted objectives and optimize this aggregated objective simultaneously with the other objectives chosen by the objective reduction approach Brockhoff and Zitzler (2007b). Using this additional aggregated objective turned out to be inferior to the approaches presented in the previous section. The main reason was the huge impact of an additional objective on the computational effort needed for the hypervolume computation. However, the idea of aggregating objectives during hypervolume-based search to reduce the running time of the hypervolume calculation built the basis of the more advanced aggregation heuristics presented in the last chapter. Here, we show what can be gained by aggregating objectives within a hypervolume-based algorithm. To this end, we perform additional experiments by using the aggregation heuristics of Chapter 3 within SIBEA.

4.4.3.1 Basic Algorithms

In addition to SIBEA with fixed objective set size and the greedy `k-EMOSS` heuristic to reduce the number of objectives, we consider two SIBEA versions that use the aggregation heuristics for the $\texttt{OAP}_{\texttt{max}}$ and $\texttt{OAP}_{\texttt{avg}}$ problems proposed in Sec. 3.6.2. Similar as before, every $G = 50$ generations, we compute the best objective subset (for the `k-EMOSS` based $\text{SIBEA}_{\texttt{k-EMOSS}}$) or the best aggregation (versions denoted by $\text{SIBEA}_{\text{max}}$ and $\text{SIBEA}_{\text{avg}}$) on the current population and consider only the computed objectives for the next 50 generations of SIBEA. We use only the greedy `k-EMOSS` algorithm based $\text{SIBEA}_{\texttt{k-EMOSS}}$ version for the comparison with the new aggregation approach since it showed the best performance among the objective omission variants in the previous section. Due to the fact that the running time for $\text{SIBEA}_{\texttt{k-EMOSS}}$ can be controlled by fixing a certain k which is not the case for the other dynamic or random SIBEA variants, it is expected that this outperformance is the case also for similar problem instances from the DTLZ test suite that are used here.

4.4.3.2 Settings

The comparison of the three SIBEA versions is performed with the following settings. 11 runs for each combination of problem (scaled DTLZ2_{BZ}, scaled DTLZ3_{BZ}, and DTLZ7 as in Sec. 4.4.2.2 with 5, 10, and 15 objectives) and objective set size ($\texttt{k} = 2, 3, 4$) are performed where the computation time is set to $T = 15$ minutes on a 64bit AMD linux machine with 4 cores. Afterwards, the hypervolume indicator values of the populations after the time T has been reached are computed with respect to all objectives and the non-parametric Kruskal-Wallis test followed by the Conover-Inman procedure is used as in the previous comparison, i.e., with a confidence level of 0.05. The reference points of the hypervolume computation are chosen as $r_{\mathrm{DTLZ2}} = (50, \ldots, 50)$, $r_{\mathrm{DTLZ3}} = (25000, \ldots, 25000)$, and $r_{\mathrm{DTLZ7}} = (170, \ldots, 170)$ to ensure that all possible objective vectors have a positive hypervolume indicator contribution. Table 7 shows both the ranking of the median values (in brackets) and a ranking given by the outcomes of the statistical tests: for each algorithm A, the number of other algorithms that statistically outperform A is shown similar to Table 6 in the previous section. For both rankings, lower numbers are again better. Figures 57–59 shows, in addition, the box plots of the achieved hypervolume values. Most interesting here is the case of 15 objectives for all investigated problems where the statistical tests do not support the hypothesis of differences in the median values, as can be seen in Table 7.

To also compare the runs with different numbers of aggregated objectives against each other, we decided to run the algorithms again for 11 runs with different random seeds and compare all algorithms for all objective subset sizes against each other for each of the three problems and each number of original objectives. The same statistical Conover-Inman test after the mandatory Kruskal-Wallis test has been performed with a confidence level of 0.05; test problem instances and reference points are the same as before. Table 8 shows the rankings of the median of the hypervolume indicator values (again in brackets) and the number of algorithms that produce significantly higher hypervolume values as before—now by comparing all 9 different algorithms for each number of original objectives.

4.4.3.3 Results

The results of the statistical tests support the main conclusion that the aggregation has some advantages over the omission of objectives especially if the objective set is reduced to only two objectives. However, the advantage diminishes when more objectives are involved during the search. Except for the DTLZ3$_{\mathrm{BZ}}$ problem with 10 objectives, the omission heuristic always performs better with respect to the median values than the aggregation heuristics if the objective set is reduced to 4 objectives. In addition, we can observe from the box plots in Fig. 57–59 that the omission heuristic becomes better with increasing $\mathtt{k}$ whereas both aggregation heuristics become better when the size of the reduced objective set is decreased. One explanation for that is the high running time of the aggregation heuristics: the running time of 15 minutes is mainly used for deriving the aggregation in every 50th generation. For example on the DTLZ3$_{\mathrm{BZ}}$ problem with 15 objectives, most of the aggregation runs are performing 200 to 350 generations only whereas almost all SIBEA$_{\mathtt{k\text{-}EMOSS}}$ runs (except the ones running with four objectives) are able to run for 1000 generations or more.

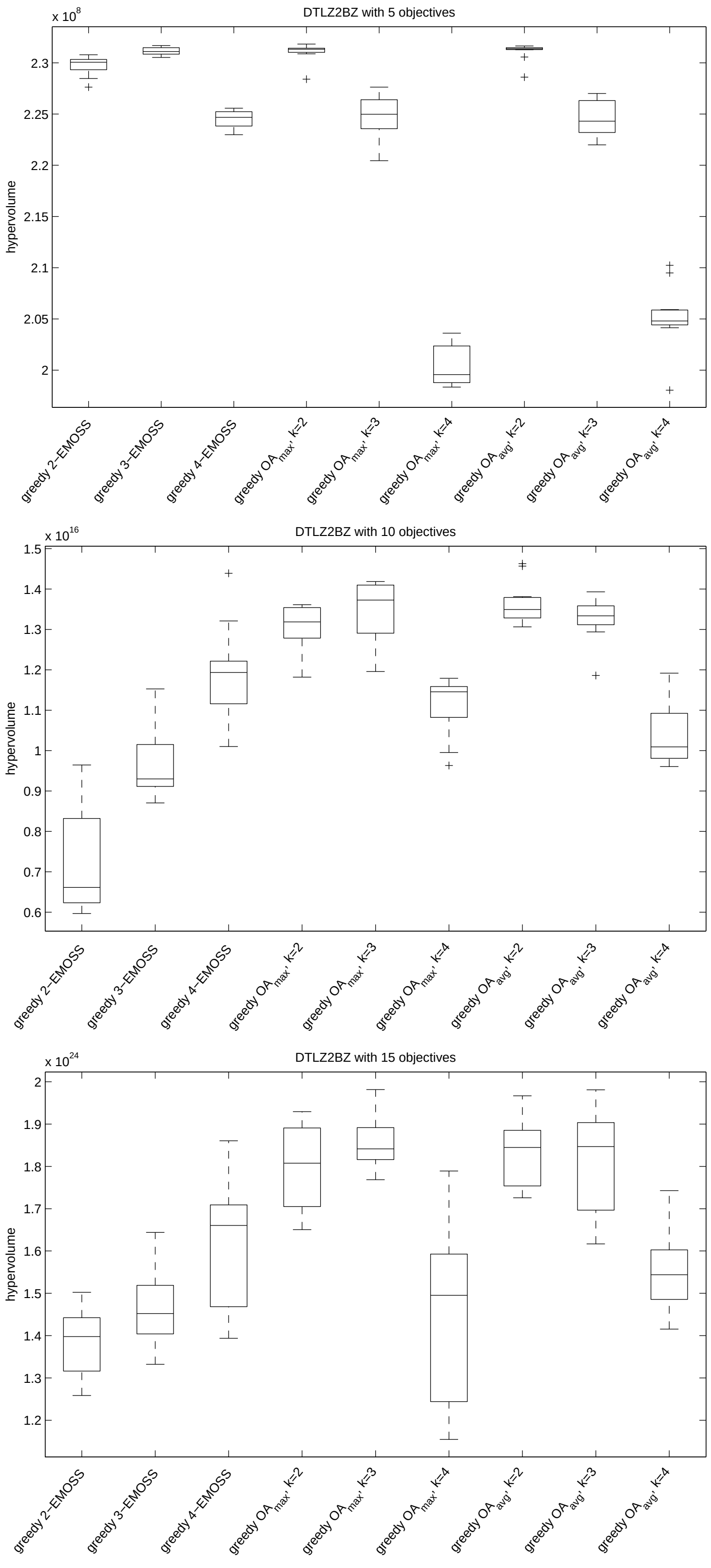


Figure 57: Boxplots of hypervolume indicator values for the SIBEA variants with objective omission and objective aggregation heuristics on the DTLZ2$_{\mathrm{BZ}}$ problem: (top) 5 objectives; (middle) 10 objectives; (bottom) 15 objectives.

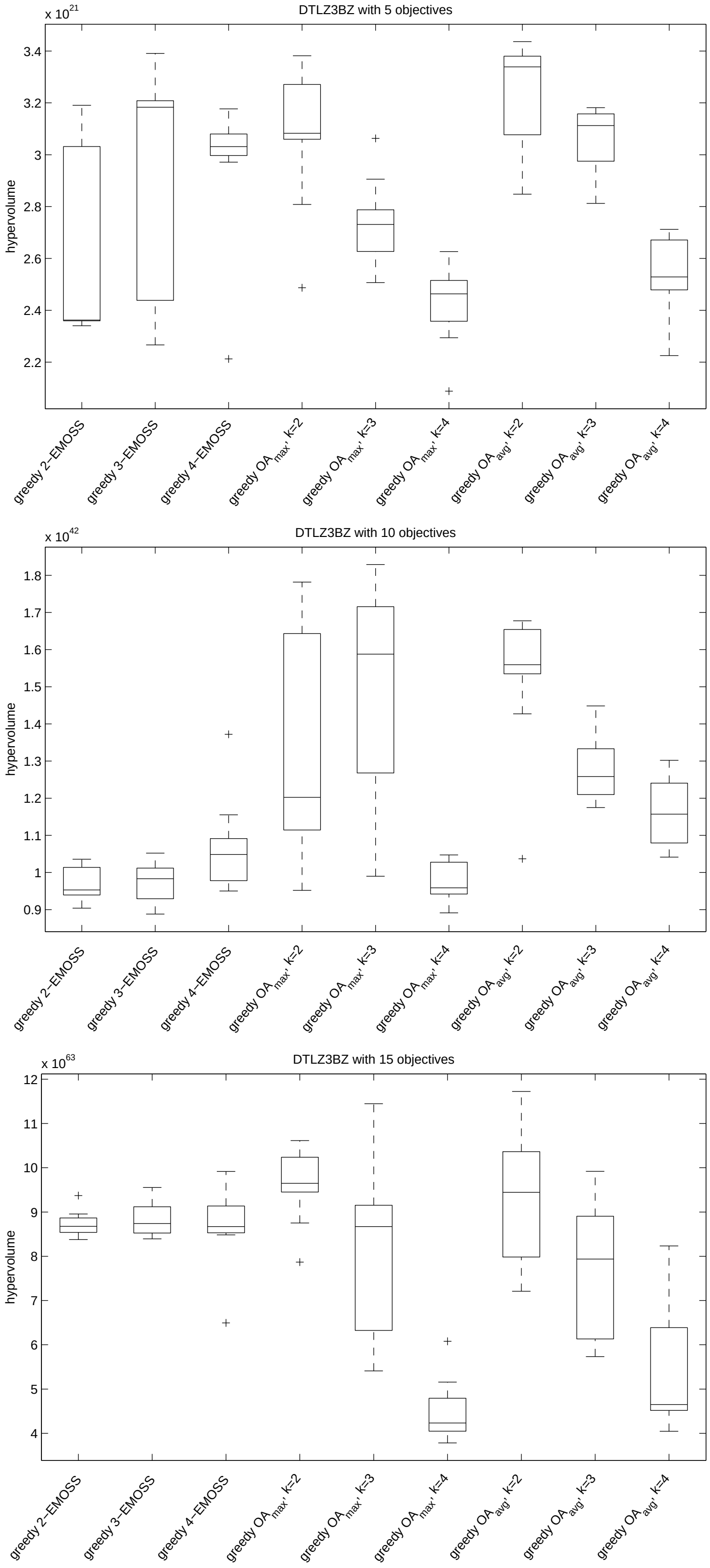


Figure 58: Boxplots of hypervolume indicator values for the SIBEA variants with objective omission and objective aggregation heuristics on the DTLZ3_{BZ} problem: (top) 5 objectives; (middle) 10 objectives; (bottom) 15 objectives.

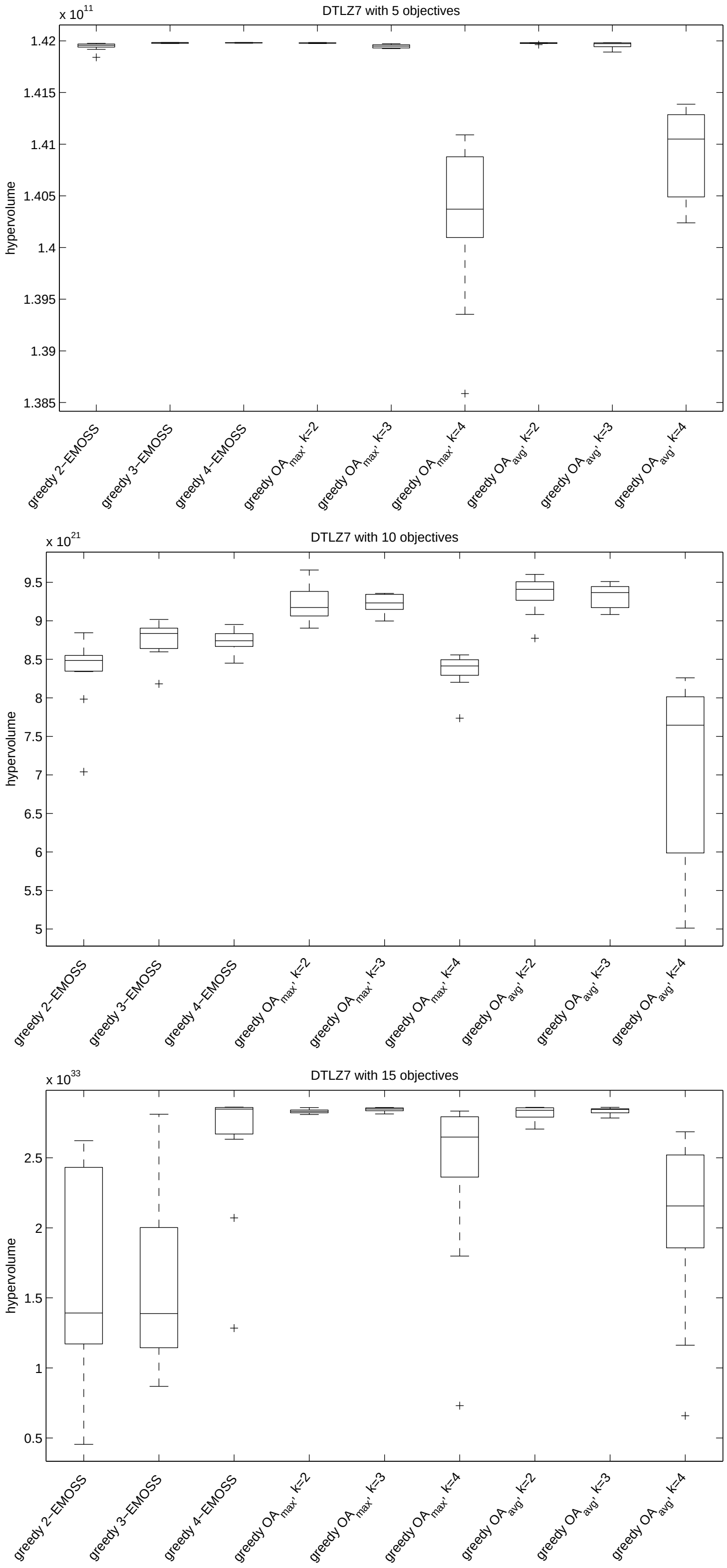


Figure 59: Boxplots of hypervolume indicator values for the SIBEA variants with objective omission and objective aggregation heuristics on the DTLZ7 problem: (top) 5 objectives; (middle) 10 objectives; (bottom) 15 objectives.

Table 7: Ranking of the hypervolume indicator values for the SIBEA versions with greedy `k-EMOSS` heuristic ($I_{H,\texttt{k-EMOSS}}$) and the aggregation heuristics with maximum ($I_{H,\max}$) and average δ-error ($I_{H,\text{avg}}$) based on the Kruskal-Wallis test with subsequent Conover-Inman procedure when the three algorithms are compared for each pair of original objective number and the number of aggregated objectives (confidence level: 0.05). The rank corresponds to the number of algorithms that significantly produce better hypervolume indicator values. The ranking of the medians is given in brackets. Lower values are always better.

problem	# original objectives	# aggregated objectives	$I_{H,\texttt{k-EMOSS}}$	$I_{H,\max}$	$I_{H,\text{avg}}$
scaled DTLZ2$_{\text{BZ}}$	5	2	2(3)	0(2)	0(1)
scaled DTLZ2$_{\text{BZ}}$	5	3	0(1)	1(2)	1(3)
scaled DTLZ2$_{\text{BZ}}$	5	4	0(1)	2(3)	1(2)
scaled DTLZ2$_{\text{BZ}}$	10	2	2(3)	1(2)	0(1)
scaled DTLZ2$_{\text{BZ}}$	10	3	2(3)	0(1)	0(2)
scaled DTLZ2$_{\text{BZ}}$	10	4	0(1)	1(2)	2(3)
scaled DTLZ2$_{\text{BZ}}$	15	2	2(3)	0(2)	0(1)
scaled DTLZ2$_{\text{BZ}}$	15	3	2(3)	0(2)	0(1)
scaled DTLZ2$_{\text{BZ}}$	15	4	0(1)	0(3)	0(2)
scaled DTLZ3$_{\text{BZ}}$	5	2	2(3)	0(2)	0(1)
scaled DTLZ3$_{\text{BZ}}$	5	3	0(1)	2(3)	0(2)
scaled DTLZ3$_{\text{BZ}}$	5	4	0(1)	1(3)	1(2)
scaled DTLZ3$_{\text{BZ}}$	10	2	2(3)	0(2)	0(1)
scaled DTLZ3$_{\text{BZ}}$	10	3	2(3)	0(1)	1(2)
scaled DTLZ3$_{\text{BZ}}$	10	4	1(2)	2(3)	0(1)
scaled DTLZ3$_{\text{BZ}}$	15	2	1(3)	0(1)	0(2)
scaled DTLZ3$_{\text{BZ}}$	15	3	0(1)	0(2)	0(3)
scaled DTLZ3$_{\text{BZ}}$	15	4	0(1)	2(3)	1(2)
DTLZ7	5	2	2(3)	0(2)	0(1)
DTLZ7	5	3	0(1)	2(3)	1(2)
DTLZ7	5	4	0(1)	2(3)	1(2)
DTLZ7	10	2	2(3)	0(2)	0(1)
DTLZ7	10	3	2(3)	1(2)	0(1)
DTLZ7	10	4	0(1)	1(2)	2(3)
DTLZ7	15	2	2(3)	0(2)	0(1)
DTLZ7	15	3	2(3)	0(1)	0(2)
DTLZ7	15	4	0(1)	0(2)	2(3)

Table 8: Ranking of the hypervolume indicator values for the SIBEA versions with greedy `k-EMOSS` heuristic ($I_{H,\texttt{k-EMOSS}}$) and the aggregation heuristics with maximum ($I_{H,\max}$) and average δ-error ($I_{H,\text{avg}}$) over all number of aggregated objectives for each problem instance based on the Kruskal-Wallis test with subsequent Conover-Inman procedure. The rank corresponds to the number of algorithms that significantly produce better hypervolume indicator values and the ranking of the medians are given in brackets.

	reduced to $\texttt{k} = 2$ objectives			reduced to $\texttt{k} = 3$ objectives			reduced to $\texttt{k} = 4$ objectives		
problem instance	$I_{H,\texttt{k-EMOSS}}$	$I_{H,\max}$	$I_{H,\text{avg}}$	$I_{H,\texttt{k-EMOSS}}$	$I_{H,\max}$	$I_{H,\text{avg}}$	$I_{H,\texttt{k-EMOSS}}$	$I_{H,\max}$	$I_{H,\text{avg}}$
scaled DTLZ2$_{\text{BZ}}$, 5 objectives	3(4)	0(1)	0(2)	0(3)	4(5)	4(6)	5(7)	8(9)	7(8)
scaled DTLZ2$_{\text{BZ}}$, 10 objectives	8(9)	1(4)	0(2)	6(8)	0(1)	1(3)	4(5)	4(6)	5(7)
scaled DTLZ2$_{\text{BZ}}$, 15 objectives	7(9)	0(3)	1(4)	7(8)	0(1)	0(2)	4(6)	4(5)	4(7)
scaled DTLZ3$_{\text{BZ}}$, 5 objectives	7(9)	0(3)	0(1)	3(6)	1(4)	0(2)	3(5)	5(8)	3(7)
scaled DTLZ3$_{\text{BZ}}$, 10 objectives	5(8)	0(3)	0(2)	1(3)	5(7)	0(1)	5(7)	4(6)	6(9)
scaled DTLZ3$_{\text{BZ}}$, 15 objectives	1(6)	0(1)	0(2)	0(4)	1(5)	1(7)	0(3)	7(9)	7(8)
DTLZ7, 5 objectives	5(7)	0(3)	0(4)	0(1)	5(6)	2(5)	0(2)	7(9)	7(8)
DTLZ7, 10 objectives	6(8)	1(3)	0(1)	4(5)	1(4)	0(2)	4(6)	6(7)	8(9)
DTLZ7, 15 objectives	5(8)	0(4)	0(3)	2(7)	0(2)	0(1)	1(5)	7(6)	7(9)

When comparing the algorithms over all possible numbers of desired objectives (Table 8), it turns out that, except for the DTLZ2_{BZ} with 15 objectives, no other algorithm produces significantly higher hypervolume values than $\text{SIBEA}_{\text{avg}}$ that reduces the number of objectives to $\mathtt{k} = 2$. However, with respect to the median hypervolume indicator values, this aggregation heuristic is assigned only twice the best rank and 4 times the second best. Nevertheless, we can also conclude in this comparison that the aggregation heuristics are often performing better and overall not worse than the greedy omission heuristic.

Although the experiments are extensive and the outperformance of the hypervolume-based algorithms with objective reduction and aggregation are producing significantly better results in the tests, one has to keep in mind that the results might highly depend on the selected test functions and that further experiments are needed to argue in favor of those algorithms in practice. However, other studies on objective reduction during search have been published recently, e.g., (López Jaimes et al., 2008, 2009) which indicate that objective reduction, including the approach presented here, is also helpful on other problems such as the 0-1-knapsack problem and can be used within other algorithms.

4.5 Summary

Indicator-based evolutionary algorithms have been shown to be very successful for dealing with multiobjective optimization in the past. Of particular interest is the hypervolume indicator—an indicator that preserves the Pareto dominance relation among solution sets if applied as an optimization criterion. This chapter contributed to a better understanding of the hypervolume indicator in terms of the first theoretical running time analysis of a hypervolume-based MOEA. Moreover, we increased the relevance and usability of the hypervolume indicator for performance assessment and search by presenting a generalized version of the hypervolume indicator that can incorporate different user preferences, such as preference points, to guide the search. Finally, we applied the objective reduction techniques proposed in the previous chapter within a hypervolume-based algorithm to decrease the running time needed for the hypervolume computation.

These three contributions can only be seen as a first step towards a general understanding of the hypervolume indicator, its properties and its application to many-objective problems. Several tasks can be identified for future work which have in part already been tackled recently.

Further running time analyses of hypervolume-based MOEAs are necessary to identify certain problem properties that allow for an efficient optimization with hypervolume-based MOEAs. A problem instance where one can show that hypervolume-based search is disadvantageous compared to dominance relation based approaches has also not been presented so far. Presenting an according running time analyses for this could also help to understand why and when hypervolume-based MOEAs are successfully applied to real-world problems.

With respect to the weighted hypervolume, most issues that needed to be obviously tackled have been considered in recent studies. First of all, the weight distribution for articulating directions towards preference points needed to be generalized to an arbitrary number of objectives. Second, the efficient computation of the weighted hypervolume indicator is an issue, especially if it is hard to obtain a function for the integral in closed form or if the number of objectives is large. Tackling both problems at the same time, Auger et al. (2009a) proposed an approach to sample the weighted hypervolume indicator efficiently for an arbitrary number of objectives and weight distribution functions that allow to include the decision maker's preference in terms of preference points and the stress of extremes.

In terms of objective reduction within hypervolume-based search, open questions for future research can be identified as well. Except in (López Jaimes et al., 2008) it has not been investigated how classical dimensionality reduction techniques, i.e., feature selection and feature extraction methods, compare to the specialized objective reduction approaches presented in this thesis. In the light of this discussion, more efficient objective reduction algorithms need to be developed in order to increase the applicability of objective reduction to real-world many-objective problems further.

5

Conclusions

5.1 Key Results

Multiobjective problems occur frequently in practice and multiobjective evolutionary algorithms (MOEAs) have been applied successfully in many application areas to approximate the set of Pareto-optimal solutions. However, if the number of objectives is high, most state-of-the-art MOEAs have difficulties to find good approximations of the Pareto front. This thesis made a first step towards understanding what causes these difficulties for MOEAs in terms of investigating what happens with the Pareto dominance relation if additional objectives are considered. We also proposed an objective reduction approach to deal with many-objective problems with respect to both decision making and search. Furthermore, we investigated hypervolume-based multiobjective evolutionary algorithms—a special type of MOEAs that turn out to be better suited for many-objective problems in practice than other well-established algorithms such as NSGA-II or SPEA2. In particular, we presented the following key results.

Effects of Additional Objectives

Adding objectives to a multiobjective optimization problem has been known to change the behavior of multiobjective evolutionary algorithms in practice. In this thesis, we investigated the effects of additional objectives on the Pareto dominance relation from a theoretical perspective. These results built the basis for rigorous running time analyses of a multiobjective evolutionary algorithm answering two previously open questions. On the one hand, we showed that one and the same problem can be made both harder and easier—depending on the objective that is added and in contrast to previous belief that the effect of additional objectives is caused by properties of the problem itself. On the other hand, we showed theoretically that there are problems where multiobjective search is highly beneficial, i.e., that the combination of two equally complex problems to a biobjective problem is easier to solve than solving the two problems independently. Note here that, although the investigated problems only have two or three objectives to keep the proofs readable, the results could be generalized to problems with many objectives.

Objective Reduction

In the third chapter, we proposed a general framework for objective reduction based on the effects of adding and omitting objectives on the dominance structure. To this end, we proposed a generalized measure for changes in the dominance structure, the so-called δ-error. Based on this measure, we gave a general definition of objective conflicts and defined the objective reduction problems δ-`MOSS` and `k-EMOSS`. For both problems, we proved the $\mathcal{NP}$-hardness and proposed exact and greedy algorithms. Furthermore, we generalized the approach to aggregating objectives and compared the algorithms experimentally to show the potentials of this approach for several test problems. The proposed objective reduction algorithms have also been applied to a radar waveform optimization problem showing that the approach can reveal underlying problem characteristics that might help to learn about a problem in practice in order to specify or simplify a first rough problem formulation.

Overall, it turned out that when objective reduction is applied during search, objective reduction algorithms that guarantee an upper bound on the resulting objective set size are favorable over strategies that dynamically adjust the resulting number of objectives. Due to the high computation time of the exact algorithms, we advise to use the greedy heuristics the computed objective set sizes and resulting errors of which are comparable to the exact algorithm whereas their computation time is still feasible for reasonable input instances.

Hypervolume Indicator Based Search

With respect to hypervolume-based optimization, the contributions of this thesis were threefold. First, we provided the first running time analysis of a hypervolume-based MOEA. Results on a simple test problem showed that a simple MOEA using a popular steady-state selection scheme based on the hypervolume indicator can find the Pareto set in polynomial time although the selection scheme does not allow for a convergence to the Pareto set in general. Second, we generalized the hypervolume indicator to a weighted version. This allows a user to articulate her preferences within a Pareto-compliant indicator. Three weight distribution functions showed the applicability of the approach exemplary for biobjective problems. Third, we applied the proposed objective reduction techniques within a hypervolume-based search algorithm. Automatically reducing the number of objectives during search has been shown to be one way to circumvent the high running time of the hypervolume computation. The proposed MOEA with objective reduction has been shown to be efficient in experiments for problems with up to 15 objectives.

5.2 Open Questions and Future Work

Regarding many-objective optimization in terms of the investigated topics objective reduction and hypervolume-based search, several questions remained open after this thesis project. However, some of them are tackled in current research or have been already solved in the meantime.

Objective Reduction

Although the objective reduction approach has been shown to be beneficial in terms of search and reducing the amount of data that has to be considered after the search, reducing the number of objectives might also help in other scenarios. For example, the aggregation heuristics might be helpful if an initial weighting of the objectives within the well-known weighted sum method is sought. In this scenario, the aggregation heuristics could be applied to a randomly chosen set of solutions before the search process to automatically find a set of weights. Since the weighted sum method is known to be unable to find Pareto-optimal solutions in regions where the Pareto front is non-convex, such an approach should use multiobjectivization in a later stage of the search process to find a good approximation of the Pareto front. The decision how the objective function should be decomposed could also be done here via the objective reduction approach. The objective reduction approach in general can also be helpful in many-objective scenarios where the evaluation times for the single objective functions vary highly. In both application areas, practical studies would be highly interesting.

The usage of state-of-the-art machine learning techniques for feature selection and feature extraction in the same manner than in (López Jaimes et al., 2008) is another interesting area of future research which is expected to make the reduction of objectives even more applicable to real-world problems with many objectives.

As to the objective reduction approach in general, several generalizations could be the basis of future work. Examples include the consideration of other error measures, e.g., with respect to the multiplicative ε-dominance or the investigation of the effects of objective reduction to arbitrary dominance structures, e.g., induced by specific unary indicators. Other questions, especially in terms of decision making, remain open. Of high practical value for a human decision maker would be, for example, if one could automatically detect which objectives are the most important ones or which objectives are equivalent and interchangeable.

Hypervolume Indicator Based Search

In contrast to the small number of researchers working in the field of objective reduction, many publications are dealing with the hypervolume indicator. Current research in hypervolume-based search is therefore characterized by a high dynamic and some questions that have been open for years have been recently solved or are going to be solved in the near future. Some examples are the generalization of the weighted hypervolume to more than 2 objectives (Auger et al., 2009a), the incorporation of Monte Carlo sampling (Bader and Zitzler, 2009), or the computational complexity of the hypervolume indicator (Bringmann and Friedrich, 2008, 2009b,a). Also the general question of how a finite set of μ solutions that is maximizing the hypervolume indicator is distributed on the Pareto front has been tackled recently (Auger et al., 2009c,b).

However, some interesting questions remain open. The most fundamental one is whether unary quality indicators that are a refinement of the Pareto dominance relation exist that cannot be defined via the hypervolume indicator and its weighted version. Related to that is the question whether there are other unary quality indicators that are Pareto-compliant and easier to compute than the hypervolume indicator.

Although many rigorous running time analyses of multiobjective evolutionary algorithms have been performed in recent years (Neumann and Wegener, 2006; Friedrich et al., 2007; Neumann and Reichel, 2008; Friedrich et al., 2008), also here, further studies, especially with respect to hypervolume-based algorithms, are necessary. First of all, the question whether the second result in (Brockhoff et al., 2008) on a large Pareto front is fully correct needs to be answered. Further running time analyses of hypervolume-based MOEAs should then try to identify certain problem properties that allow for an efficient optimization with hypervolume-based MOEAs. For example, a problem instance for which one can show that hypervolume-based search is disadvantageous compared to algorithms that only rely on the Pareto dominance relation has not been presented so far. Presenting an according running time analyses for this—or showing that this scenario cannot happen—could also help to understand why and when hypervolume-based MOEAs can be successfully applied to real-world problems. Also the new idea of finding combinatorial optimization problems for which (hypervolume-based) MOEAs are fully polynomial-time randomized approximation schemes (FPRAS) needs to be further developed. Formulated in the terms of hypervolume-based algorithms, the question is here whether multiobjective evolutionary algorithms can efficiently find solution sets that yield good approximations of the optimal hypervolume indicator value as given by the optimal μ-distribution defined in (Auger et al., 2009c).

We can conclude that the field of many-objective optimization has still many interesting questions to offer for future research. Especially the theoretical investigation of hypervolume indicator based search is still in its infancies and, from my point-of-view, will yield fruitful results in the future.

References

Agrell, P. J. 1997. On Redundancy in Multi Criteria Decision Making. European Journal of Operational Research, 98(3):571–586.

Auger, A., Bader, J., Brockhoff, D., and Zitzler, E. 2009a. Articulating User Preferences in Many-Objective Problems by Sampling the Weighted Hypervolume. In: Raidl, G. et al. (Editors), Genetic and Evolutionary Computation Conference (GECCO 2009). ACM, pages 555–562.

Auger, A., Bader, J., Brockhoff, D., and Zitzler, E. 2009b. Investigating and Exploiting the Bias of the Weighted Hypervolume to Articulate User Preferences. In: Raidl, G. et al. (Editors), Genetic and Evolutionary Computation Conference (GECCO 2009). ACM, pages 563–570.

Auger, A., Bader, J., Brockhoff, D., and Zitzler, E. 2009c. Theory of the Hypervolume Indicator: Optimal μ-Distributions and the Choice of the Reference Point. In: Foundations of Genetic Algorithms (FOGA 2009). ACM, pages 87–102.

Bader, J., Brockhoff, D., Welten, S., and Zitzler, E. 2009. On Using Populations of Sets in Multiobjective Optimization. In: Ehrgott, M. et al. (Editors), Conference on Evolutionary Multi-Criterion Optimization (EMO 2009). Springer, 5467:140–154.

Bader, J., Deb, K., and Zitzler, E. 2008. Faster Hypervolume-based Search using Monte Carlo Sampling. In: Conference on Multiple Criteria Decision Making (MCDM 2008). Springer. to appear.

Bader, J. and Zitzler, E. 2008. HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. TIK Report 286, Computer Engineering and Networks Laboratory (TIK), ETH Zurich.

Bader, J. and Zitzler, E. 2009. A Hypervolume-Based Multiobjective Optimizer for High-Dimensional Objective Spaces. In: Conference on Multiple Objective and Goal Programming (MOPGP 2008). Springer. to appear.

Beume, N., Naujoks, B., and Emmerich, M. 2007. SMS-EMOA: Multiobjective Selection Based on Dominated Hypervolume. European Journal on Operational Research, 181:1653–1669.

Beume, N., Naujoks, B., Preuss, M., and amd T. Wagner, G. R. 2009. Effects of 1-Greedy $\mathcal{S}$-Metric-Selection on Innumerably Large Pareto Fronts. In: Ehrgott, M. et al. (Editors), Conference on Evolutionary Multi-Criterion Optimization (EMO 2009). Springer, 5467:21–35.

Beume, N. and Rudolph, G. 2006. Faster S-Metric Calculation by Considering Dominated Hypervolume as Klee's Measure Problem. Technical Report CI-216/06, Sonderforschungsbereich 531 Computational Intelligence, Universität Dortmund.

Beyer, H.-G., Schwefel, H.-P., and Wegener, I. 2002. How to Analyse Evolutionary Algorithms. Theoretical Computer Science, 287:101 – 130.

Bleuler, S., Brack, M., Thiele, L., and Zitzler, E. 2001. Multiobjective Genetic Programming: Reducing Bloat by Using SPEA2. In: Congress on Evolutionary Computation (CEC 2001). IEEE Press, pages 536–543.

Bleuler, S., Laumanns, M., Thiele, L., and Zitzler, E. 2003. PISA—A Platform and Programming Language Independent Interface for Search Algorithms. In: Fonseca, C. M. et al. (Editors), Conference on Evolutionary Multi-Criterion Optimization (EMO 2003), Berlin. Springer, 2632:494–508.

Bringmann, K. and Friedrich, T. 2008. Approximating the Volume of Unions and Intersections of High-Dimensional Geometric Objects. In: Hong, S. H., Nagamochi, H., and Fukunaga, T. (Editors), International Symposium on Algorithms and Computation (ISAAC 2008). Springer, 5369:436–447.

Bringmann, K. and Friedrich, T. 2009a. Approximating the Least Hypervolume Contributor: NP-hard in General, But Fast in Practice. In: Ehrgott, M. et al. (Editors), Conference on Evolutionary Multi-Criterion Optimization (EMO 2009). Springer, 5467:6–20.

Bringmann, K. and Friedrich, T. 2009b. Don't Be Greedy When Calculating Hypervolume Contributions. In: Foundations of Genetic Algorithms (FOGA 2009). ACM, pages 103–112.

Brockhoff, D., Friedrich, T., Hebbinghaus, N., Klein, C., Neumann, F., and Zitzler, E. 2007a. Do Additional Objectives Make a Problem Harder? In: Thierens, D. et al. (Editors), Genetic and Evolutionary Computation Conference (GECCO 2007). ACM, pages 765–772.

Brockhoff, D., Friedrich, T., Hebbinghaus, N., Klein, C., Neumann, F., and Zitzler, E. 2009. On the Effects of Adding Objectives to Plateau Functions. IEEE Transactions on Evolutionary Computation, 13(3):591–603.

Brockhoff, D., Friedrich, T., and Neumann, F. 2008. Analyzing Hypervolume Indicator Based Algorithms. In: Rudolph, G. et al. (Editors), Conference on Parallel Problem Solving From Nature (PPSN X). Springer, 5199:651–660.

Brockhoff, D., Saxena, D. K., Deb, K., and Zitzler, E. 2007b. On Handling a Large Number of Objectives A Posteriori and During Optimization. In: Knowles, J., Corne, D., and Deb, K. (Editors), Multiobjective Problem Solving from Nature: From Concepts to Applications, pages 377–403. Springer.

Brockhoff, D. and Zitzler, E. 2006a. Are All Objectives Necessary? On Dimensionality Reduction in Evolutionary Multiobjective Optimization. In: Runarsson, T. P. et al. (Editors), Conference on Parallel Problem Solving from Nature (PPSN IX). Springer, 4193:533–542.

Brockhoff, D. and Zitzler, E. 2006b. On Objective Conflicts and Objective Reduction in Multiple Criteria Optimization. TIK Report 243, Computer Engineering and Networks Laboratory (TIK), ETH Zurich.

Brockhoff, D. and Zitzler, E. 2007a. Dimensionality Reduction in Multiobjective Optimization: The Minimum Objective Subset Problem. In: Waldmann, K. H. and Stocker, U. M. (Editors), Operations Research Proceedings 2006. Springer, pages 423–429.

Brockhoff, D. and Zitzler, E. 2007b. Improving Hypervolume-based Multiobjective Evolutionary Algorithms by Using Objective Reduction Methods. In: Congress on Evolutionary Computation (CEC 2007). IEEE Press, pages 2086–2093.

Brockhoff, D. and Zitzler, E. 2009a. Automated Aggregation and Omission of Objectives to Handle Many-Objective Problems. In: Conference on Multiple Objective and Goal Programming (MOPGP 2008). Springer. to appear.

Brockhoff, D. and Zitzler, E. 2009b. Objective Reduction in Evolutionary Multiobjective Optimization: Theory and Applications. Evolutionary Computation, 17(2):135–166.

Chen, Q. and Guan, S.-U. 2004. Incremental Multiple Objective Genetic Algorithms. IEEE Transactions on Systems, Man and Cybernetics, 34(3):1325–1334.

Coello Coello, C. A. 2000. Handling Preferences in Evolutionary Multiobjective Optimization: A Survey. In: Congress on Evolutionary Computation (CEC 2000). IEEE Press, pages 30–37.

Coello Coello, C. A. and Hernández Aguirre, A. 2002. Design of Combinational Logic Circuits through an Evolutionary Multiobjective Optimization Approach. Artificial Intelligence for Engineering, Design, Analysis and Manufacture, 16(1):39–53.

Coello Coello, C. A., Hernández Aguirre, A., and Zitzler, E. (Editors) 2005. Evolutionary Multi-Criterion Optimization (EMO 2005), volume 3242 of LNCS. Springer.

Coello Coello, C. A., Lamont, G. B., and Van Veldhuizen, D. A. 2007. Evolutionary Algorithms for Solving Multi-Objective Problems. Springer, Berlin, Germany.

Conover, W. J. 1999. Practical Nonparametric Statistics. John Wiley, 3 edition.

Corne, D. W., Jerram, N. R., Knowles, J. D., and Oates, M. J. 2001. PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization. In: Genetic and Evolutionary Computation Conference (GECCO 2001). Morgan Kaufmann Publishers, pages 283–290.

Costa, L. and Oliveira, P. 2009. Multiobjective Optimization: Redundant and Informative Objectives. In: Congress on Evolutionary Computation (CEC 2009). IEEE Press, pages 2008–2015.

Das, I. and Dennis, J. E. 1998. Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. SIAM Journal on Optimization, 8(3):631–657.

de Jong, E. D. and Bucci, A. 2006. DECA: Dimension Extracting Coevolutionary Algorithm. In: Genetic and Evolutionary Computation Conference (GECCO 2006).

de Jong, E. D. and Bucci, A. 2008. Objective Set Compression. In: Knowles, J., Corne, D., and Deb, K. (Editors), Multiobjective Problem Solving from Nature: From Concepts to Applications, pages 347–376. Springer.

de Jong, E. D., Watson, R. A., and Pollack, J. B. 2001. Reducing Bloat and Promoting Diversity using Multi-Objective Methods. In: Spector, L. et al. (Editors), Genetic and Evolutionary Computation Conference (GECCO 2001). Morgan Kaufmann Publishers, pages 11–18.

Deb, K. 2001. Multi-Objective Optimization Using Evolutionary Algorithms. Wiley, Chichester, UK.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. 2002. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2):182–197.

Deb, K. and Saxena, D. 2005. On Finding Pareto-Optimal Solutions Through Dimensionality Reduction for Certain Large-Dimensional Multi-Objective Optimization Problems. Kangal report no. 2005011, Kanpur Genetic Algorithms Laboratory (KanGAL).

Deb, K. and Saxena, D. 2006. Searching For Pareto-Optimal Solutions Through Dimensionality Reduction for Certain Large-Dimensional Multi-Objective Optimization Problems. In: Congress on Evolutionary Computation (CEC 2006). IEEE Press, pages 3352–3360.

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. 2005. Scalable Test Problems for Evolutionary Multi-Objective Optimization. In: Abraham, A., Jain, R., and Goldberg, R. (Editors), Evolutionary Multiobjective Optimization: Theoretical Advances and Applications, chapter 6, pages 105–145. Springer.

Doerr, B., Hebbinghaus, N., and Neumann, F. 2006. Speeding Up Evolutionary Algorithms Through Restricted Mutation Operators. In: Runarsson, T. P. et al. (Editors), Conference on Parallel Problem Solving from Nature (PPSN IX). Springer, 4193:978–987.

Droste, S., Jansen, T., and Wegener, I. 2002. On the Analysis of the (1+1) Evolutionary Algorithm. Theoretical Computer Science, 276:51–81.

Ehrgott, M. 2005. Multicriteria Optimization. Springer, Berlin, Germany, 2nd edition.

Eiben, A. E. and Smith, J. E. 2003. Introduction to Evolutionary Computing. Natural Computing Series. Springer, Berlin, Germany, 1. edition.

Ekárt, A. and Németh, S. Z. 2001. Selection Based on the Pareto Nondomination Criterion for Controlling Code Growth in Genetic Programming. Genetic Programming and Evolvable Machines, 2:61–73.

Emmerich, M., Beume, N., and Naujoks, B. 2005. An EMO Algorithm Using the Hypervolume Measure as Selection Criterion. In: Conference on Evolutionary Multi-Criterion Optimization (EMO 2005). Springer, 3410:62–76.

Feige, U. 1998. A Threshold of ln n for Approximating Set Cover. Journal of the ACM, 45(4):634–652.

Fleischer, M. 2003. The measure of Pareto optima. Applications to multiobjective metaheuristics. In: Fonseca, C. M. et al. (Editors), Conference on Evolutionary Multi-Criterion Optimization (EMO 2003). Springer, 2632:519–533.

Fleming, P. J., Purshouse, R. C., and Lygoe, R. J. 2005. Many-Objective Optimization: An Engineering Design Perspective. In: Coello Coello, C. A. et al. (Editors), Conference on Evolutionary Multi-Criterion Optimization (EMO 2005). Springer, 3410:14–32.

Fliege, J. 2007. The Effects of Adding Objectives to an Optimisation Problem on the Solution Set. Operations Research Letters, 35:782–790.

Fodor, I. K. 2002. A Survey of Dimension Reduction Techniques. Technical Report UCRL-ID-148494, U.S. Department of Energy.

Fonseca, C., Fleming, P., Zitzler, E., Deb, K., and Thiele, L. (Editors) 2003. Evolutionary Multi-Criterion Optimization (EMO 2003), volume 2632 of LNCS. Springer.

Fonseca, C. M. and Fleming, P. J. 1993. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In: Forrest, S. (Editor), Conference on Genetic Algorithms. Morgan Kaufmann, pages 416–423.

Fonseca, C. M. and Fleming, P. J. 1995. An Overview of Evolutionary Algorithms in Multiobjective Optimization. Evolutionary Computation, 3(1):1–16.

Fonseca, C. M., Paquete, L., and López-Ibáñez, M. 2006. An Improved Dimension-Sweep Algorithm for the Hypervolume Indicator. In: Congress on Evolutionary Computation (CEC 2006). IEEE Press, pages 1157–1163.

Friedrich, T., He, J., Hebbinghaus, N., Neumann, F., and Witt, C. 2007. Approximating Covering Problems by Randomized Search Heuristics Using Multi-Objective Models. In: Genetic and Evolutionary Computation Conference (GECCO 2007). ACM, pages 797–804.

Friedrich, T., Horoba, C., and Neumann, F. 2008. Runtime Analyses for Using Fairness in Evolutionary Multi-Objective Optimization. In: Rudolph, G. et al. (Editors), Conference on Parallel Problem Solving from Nature (PPSN X). Springer, 5199:671–680.

Gal, T. and Hanne, T. 1999. Consequences of Dropping Nonessential Objectives for the Application of MCDM Methods. European Journal of Operational Research, 119:373–378.

Gal, T. and Hanne, T. 2006. Nonessential Objectives Within Network Approaches for MCDM. European Journal of Operational Research, 168:584592.

Gal, T. and Leberling, H. 1977. Redundant Objective Functions in Linear Vector Maximum Problems and Their Determination. European Journal of Operational Research, 1(3):176–184.

Garey, M. R. and Johnson, D. S. 1990. Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA.

Giel, O. 2003. Expected Runtimes of a Simple Multi-objective Evolutionary Algorithm. In: Congress on Evolutionary Computation (CEC 2003). pages 1918–1925.

Giel, O. and Wegener, I. 2003. Evolutionary Algorithms and the Maximum Matching Problem. In: Symposium on Theoretical Aspects of Computer Science (STACS 2003). Springer, 2607:415–426.

Gobbi, M., Mastinu, G., and Doniselli, C. 1999. Optimising a Car Chassis. Vehicle System Dynamics, 32(2–3):149–170.

Goldberg, D. E. 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, Massachusetts.

Greiner, D., Emperador, J. M., Winter, G., and Galván, B. 2007. Improving Computational Mechanics Optimium Design Using Helper Objectives: An Application in Frame Bar Structures. In: Conference on Evolutionary Multi-Criterion Optimization (EMO 2007). Springer, 4403:575–589.

Grunert da Fonseca, V., Fonseca, C. M., and Hall, A. O. 2001. Inferential Performance Assessment of Stochastic Optimisers and the Attainment Function. In: Zitzler, E. et al. (Editors), Conference on Evolutionary Multi-Criterion Optimization (EMO 2001). Springer, 1993:213–225.

Handl, J., Lovell, S. C., and Knowles, J. 2008. Multiobjectivization by Decomposition of Scalar Cost Functions. In: Rudolph, G. et al. (Editors), Conference on Parallel Problem Solving From Nature (PPSN X). Springer, 5199:31–40.

Harzheim, E. 2005. Ordered Sets (Advances in Mathematics). Springer.

He, J. and Yao, X. 2002. From an Individual to a Population: an Analysis of the First Hitting Time of Population-based Evolutionary Algorithms. IEEE Transactions on Evolutionary Computation, 6(5):495–511.

Henig, M. I. 1982. Existence and Characterization of Efficient Decision With Respect to Cones. Mathematical Programming, 23:111–116.

Horn, J. 1997. Multicriterion Decision Making. In: Bäck, T., Fogel, D. B., and Michalewicz, Z. (Editors), Handbook of Evolutionary Computation. IOP Publishing and Oxford Univertity Press, Bristol, UK.

Horn, J., Nafpliotis, N., and Goldberg, D. E. 1994. A Niched Pareto Genetic Algorithm for Multiobjective Optimization. In: Congress on Evolutionary Computation (CEC 1994). IEEE Press, pages 82–87.

Horoba, C. and Neumann, F. 2008. Benefits and Drawbacks for the Use of Epsilon-dominance in Evolutionary Multi-objective Optimization. In: Conference on Genetic and Evolutionary Computation (GECCO 2008). ACM, pages 641–648.

Huband, S., Hingston, P., Barone, L., and While, L. 2006. A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit. IEEE Transactions on Evolutionary Computation, 10(5):477–506.

Huband, S., Hingston, P., White, L., and Barone, L. 2003. An Evolution Strategy with Probabilistic Mutation for Multi-Objective Optimisation. In: Congress on Evolutionary Computation (CEC 2003). IEEE Press, 3:2284–2291.

Hughes, E. J. 2003. Multiple Single Objective Pareto Sampling. In: Congress on Evolutionary Computation (CEC 2003). IEEE Press, pages 2678–2684.

Hughes, E. J. 2007. Radar Waveform Optimization as a Many-Objective Application Benchmark. In: Conference on Evolutionary Multi-Criterion Optimization (EMO 2007). Springer, 4403:700–714.

Hyvärinen, A., Karhunen, J., and Oja, E. 2001. Independent Component Analysis. John Wiley & Sons.

Igel, C., Hansen, N., and Roth, S. 2007. Covariance Matrix Adaptation for Multi-objective Optimization. Evolutionary Computation, 15(1):1–28.

Ishibuchi, H., Tsukamoto, N., and Nojima, Y. 2008. Evolutionary Many-Objective Optimization: A Short Review. In: Congress on Evolutionary Computation (CEC 2008). IEEE Press, pages 2424–2431.

Jain, A. K., Murty, M. N., and Flynn, P. J. 1999. Data Clustering: A Review. ACM Computing Surveys, 31(3):264–323.

Jansen, T. and Sudholt, D. 2005. Design and Analysis of an Asymmetric Mutation Operator. In: Congress on Evolutionary Computation (CEC 2005). IEEE Press, pages 190–197.

Jansen, T. and Wegener, I. 2001. Evolutionary Algorithms—How to Cope With Plateaus of Constant Fitness and When to Reject Strings of the Same Fitness. IEEE Transactions on Evolutionary Computation, 5(6):589–599.

Jaszkiewicz, A. and Branke, J. 2008. Interactive Multiobjective Evolutionary Algorithms. In: Branke, J., Deb, K., Miettinen, K., and Slowinski, R. (Editors), Multiobjective Optimization: Interactive and Evolutionary Approaches, pages 179–193. Springer.

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

Jolliffe, I. T. 2002. Principal Component Analysis. Springer.

Khare, V. R., Yao, X., and Deb, K. 2003. Performance Scaling of Multi-objective Evolutionary Algorithms. In: Conference on Evolutionary Multi-Criterion Optimization (EMO 2003). Springer, 2632:376–390.

Knowles, J. and Corne, D. 2002. On Metrics for Comparing Non-Dominated Sets. In: Congress on Evolutionary Computation (CEC 2002). IEEE Press, pages 711–716.

Knowles, J. and Corne, D. 2003. Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors. IEEE Transactions on Evolutionary Computation, 7(2):100–116.

Knowles, J. and Corne, D. 2007. Quantifying the Effects of Objective Space Dimension in Evolutionary Multiobjective Optimization. In: Conference on Evolutionary Multi-Criterion Optimization (EMO 2007). pages 757–771.

Knowles, J., Thiele, L., and Zitzler, E. 2006. A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers. TIK Report 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich.

Knowles, J. D., Watson, R. A., and Corne, D. W. 2001. Reducing Local Optima in Single-Objective Problems by Multi-objectivization. In: Zitzler, E. et al. (Editors), Conference on Evolutionary Multi-Criterion Optimization (EMO 2001). Springer, 1993:269–283.

Kohonen, T. 2001. Self-Organizing Maps, volume 30 of Springer Series in Information Sciences. Springer, 3rd edition.

Köppen, M. and Yoshida, K. 2007. Visualization of Pareto-Sets in Evolutionary Multi-Objective Optimization. In: Conference on Hybrid Intelligent Systems. IEEE Press, pages 156–161.

Koski, J. and Silvennoinen, R. 1987. Norm Methods and Partial Weighting in Multicriterion Optimization of Structures. International Journal for Numerical Methods in Engineering, 24:1101–1121.

Kumar, R. and Banerjee, N. 2006. Analysis of a Multiobjective Evolutionary Algorithm on the 0-1 Knapsack Problem. Theoretical Computer Science, 358(1):104–120.

Künzli, S., Bleuler, S., Thiele, L., and Zitzler, E. 2004. A Computer Engineering Benchmark Application for Multiobjective Optimizers. In: Coello Coello, C. A. and Lamont, G. (Editors), Applications of Multi-Objective Evolutionary Algorithms, pages 269–294. World Scientific Publishing.

Langley, P. 1994. Selection of Relevant Features in Machine Learning. In: Proceedings of the AAAI Fall Symposium on Relevance. pages 140–144.

Laumanns, M., Rudolph, G., and Schwefel, H.-P. 1999. Approximating the Pareto Set: Concepts, Diversity Issues, and Performance Assessment. Technical Report CI-7299, University of Dortmund.

Laumanns, M., Thiele, L., Deb, K., and Zitzler, E. 2002a. Combining Convergence and Diversity in Evolutionary Multiobjective Optimization. Evolutionary Computation, 10(3):263–282.

Laumanns, M., Thiele, L., and Zitzler, E. 2004a. Running Time Analysis of Evolutionary Algorithms on a Simplified Multiobjective Knapsack Problem. Natural Computing, 3(1):37–51.

Laumanns, M., Thiele, L., and Zitzler, E. 2004b. Running Time Analysis of Multiobjective Evolutionary Algorithms on Pseudo-Boolean Functions. IEEE Transactions on Evolutionary Computation, 8(2):170–182.

Laumanns, M., Thiele, L., Zitzler, E., Welzl, E., and Deb, K. 2002b. Running Time Analysis of Multi-Objective Evolutionary Algorithms on a Simple Discrete Optimization Problem. In: Conference on Parallel Problem Solving From Nature (PPSN VII). Springer, 2439:44–53.

Liu, H. and Motoda, H. (Editors) 2008. Computational Methods of Feature Selection. CRC Press.

López Jaimes, A., Coello Coello, C. A., and Chakraborty, D. 2008. Objective Reduction Using a Feature Selection Technique. In: Conference on Genetic and Evolutionary Computation (GECCO 2008). ACM, pages 673–680.

López Jaimes, A., Coello Coello, C. A., and Urías Barrientos, J. E. 2009. Online Objective Reduction to Deal with Many-Objective Problems. In: Ehrgott, M. et al. (Editors), Conference on Evolutionary Multi-Criterion Optimization (EMO 2009). Springer, 5467:423–437.

Mäkelä, M. M. and Nikulin, Y. 2008. Properties of Efficient Solution Sets Under Addition of Objectives. Operations Research Letters, 36:718–721.

Malinowska, A. B. 2002. Changes of the Set of Efficient Solutions by Extending the Number of Objectives and its Evaluation. Control and Cybernetics, 31(4).

Malinowska, A. B. 2006. Nonessential Objective Functions in Linear Multiobjective Optimization Problems. Control and Cybernetics, 35(4):873–880.

Malinowska, A. B. 2008. Weakly and Properly Nonessential Objectives in Multiobjective Optimization Problems. Operations Research Letters, 36:647–650.

Malinowska, A. B. and Torres, D. F. M. 2008. Computational Approach to Essential and Nonessential Objective Functions in Linear Multicriteria Optimization. Journal of Optimization Theory and Applications, 139(2):577–590.

Mezura-Monets, E. and Coello Coello, C. A. 2007. Constrained Optimization via Multiobjective Evolutionary Algorithms. In: Knowles, J., Corne, D., and Deb, K. (Editors), Multiobjective Problem Solving from Nature: From Concepts to Applications, pages 53–75. Springer.

Miettinen, K. 1999. Nonlinear Multiobjective Optimization. Kluwer, Boston, MA, USA.

Motwani, R. and Raghavan, P. 1995. Randomized Algorithms. Cambridge University Press.

Neumann, F. 2004. Expected Runtimes of a Simple Evolutionary Algorithm for the Multi-Objective Minimum Spanning Tree Problem. In: Yao, X. et al. (Editors), Conference on Parallel Problem Solving from Nature (PPSN VIII). Springer, 3242:80–89.

Neumann, F. 2008. Expected Runtimes of Evolutionary Algorithms for the Eulerian Cycle Problem. Computer & Operations Research, 35:2750–2759.

Neumann, F. and Reichel, J. 2008. Approximating Minimum Multicuts by Evolutionary Multi-objective Algorithms. In: Rudolph, G. et al. (Editors), Conference on Parallel Problem Solving From Nature (PPSN X). Springer, 5199:72–81.

Neumann, F., Reichel, J., and Skutella, M. 2008. Computing Minimum Cuts by Randomized Search Heuristics. In: Genetic and Evolutionary Computation Conference (GECCO 2008). ACM, pages 779–786.

Neumann, F. and Wegener, I. 2004. Randomized Local Search, Evolutionary Algorithms, and the Minimum Spanning Tree Problem. In: Genetic and Evolutionary Computation Conference (GECCO 2004). Springer, 3102:713–724.

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

Nicolini, M. 2005. A Two-Level Evolutionary Approach to Multi-criterion Optimization of Water Supply Systems. In: Conference on Evolutionary Multi-Criterion Optimization (EMO 2005). Springer, 3410:736–751.

Obayashi, S., K. Deb, K., Poloni, C., Hiroyasu, T., and Murata, T. (Editors) 2007. Conference on Evolutionary Multi-Criterion Optimization (EMO 2007), volume 4403 of LNCS. Springer.

Oliveto, P. S., He, J., and Yao, X. 2007a. Evolutionary Algorithms and the Vertex Cover Problem. In: Congress on Evolutionary Computation (CEC 2007). IEEE Press, pages 1870–1877.

Oliveto, P. S., He, J., and Yao, X. 2007b. Time Complexity of Evolutionary Algorithms for Combinatorial Optimization: A Decade of Results. International Journal of Automation and Computing, 4(3):281–293.

Paechter, B., Rankin, R. C., Cumming, A., and Fogarty, T. C. 1998. Timetabling the Classes of an Entire University with an Evolutionary Algorithm. In: Conference on Parallel Problem Solving From Nature (PPSN V). Springer, pages 865–874.

Purshouse, R. C. and Fleming, P. J. 2003a. Conflict, Harmony, and Independence: Relationships in Evolutionary Multi-criterion Optimisation. In: Conference on Evolutionary Multi-Criterion Optimization (EMO 2003). Springer, 2632:16–30.

Purshouse, R. C. and Fleming, P. J. 2003b. Evolutionary Many-Objective Optimisation: an Exploratory Analysis. In: Congress on Evolutionary Computation (CEC 2003). IEEE Press, pages 2066–2073.

Purshouse, R. C. and Fleming, P. J. 2007. On the Evolutionary Optimization of Many Conflicting Objectives. IEEE Transactions on Evolutionary Computation, 11(6):770–784.

Qiu, M. 1997. Prioritizing and Scheduling Road Projects by Genetic Algorithm. Mathematics and Computers in Simulation, 43:569–574.

Rachmawati, L. and Srinivasan, D. 2006. Preference Incorporation in Multi-objective Evolutionary Algorithms: A Survey. In: Congress on Evolutionary Computation (CEC 2006). IEEE Press, pages 962–968.

Rudenko, O., Schoenauer, M., Bosio, T., and Fontana, R. 2002. A Multiobjective Evolutionary Algorithm for Car Front End Design. In: Evolution Artificielle Conference. Springer, pages 117–135.

Rudolph, G. 1994. Convergence Analysis of Canonical Genetic Algorithms. IEEE Transactions on Neural Networks, 5(1):96–101.

Rudolph, G. 2001. Some Theoretical Properties of Evolutionary Algorithms under Partially Ordered Fitness Values. In: Evolutionary Algorithms Workshop (EAW-2001). pages 9–22.

Runarsson, T. R. and Yao, X. 2005. Search Biases in Constrained Evolutionary Optimization. IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews, 35(2):233–243.

Saxena, D. K. and Deb, K. 2007. Non-linear Dimensionality Reduction Procedures for Certain Large-Dimensional Multi-objective Optimization Problems: Employing Correntropy and a Novel Maximum Variance Unfolding. In: Conference on Evolutionary Multi-Criterion Optimization (EMO 2007). Springer, 4403:772–787.

Saxena, D. K. and Deb, K. 2008. Dimensionality Reduction of Objectives and Constraints in Multi-objective Optimization Problems: A System Design Perspective. In: Congress on Evolutionary Computation (CEC 2008). IEEE Press, pages 3204–3211.

Schaffer, J. D. 1985. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In: Grefenstette, J. J. (Editor), Conference on Genetic Algorithms and Their Applications. pages 93–100.

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

Schröder, B. S. W. 2003. Ordered Sets. Birkhäuser, Boston, MA, USA.

Slavík, P. 1996. A Tight Analysis of the Greedy Algorithm for Set Cover. In: Symposium on Theory of Computing (STOC 1996). ACM Press, pages 435–441.

Srinivas, N. and Deb, K. 1994. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. Evolutionary Computation, 2(3):221–248.

Sülflow, A., Drechsler, N., and Drechsler, R. 2007. Robust Multi-Objective Optimization in High Dimensional Spaces. In: Conference on Evolutionary Multi-Criterion Optimization (EMO 2007). Springer, 4403:715–726.

Tan, K. C., Khor, E. F., and Lee, T. H. 2005. Multiobjective Evolutionary Algorithms and Applications. Springer, London, UK.

Vafaie, H. and de Jong, K. 1993. Robust Feature Selection Algorithms. In: Tools with Artificial Intelligence (TAI '93). pages 356–363.

Van Veldhuizen, D. A. 1999. Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations. PhD thesis, Graduate School of Engineering, Air Force Institute of Technology, Air University.

Wagner, T., Beume, N., and Naujoks, B. 2007. Pareto-, Aggregation-, and Indicator-based Methods in Many-objective Optimization. In: Obayashi, S. et al. (Editors), Conference on Evolutionary Multi-Criterion Optimization (EMO 2007). Springer, 4403:742–756.

While, L. 2005. A New Analysis of the LebMeasure Algorithm for Calculating Hypervolume. In: Conference on Evolutionary Multi-Criterion Optimization (EMO 2005). Springer, 3410:326–340.

While, L., Hingston, P., Barone, L., and Huband, S. 2006. A Faster Algorithm for Calculating Hypervolume. IEEE Transactions on Evolutionary Computation, 10(1):29–38.

Winkler, P. 1985. Random Orders. Order, 1(1985):317–331.

Witt, C. 2005. Worst-Case and Average-Case Approximations by Simple Randomized Search Heuristics. In: Symposium on Theoretical Aspects of Computer Science (STACS 2005). Springer, 3404:44–56.

Zitzler, E. 1999. Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications. PhD thesis, ETH Zurich, Switzerland.

Zitzler, E., Brockhoff, D., and Thiele, L. 2007. The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration. In: Obayashi, S. et al. (Editors), Conference on Evolutionary Multi-Criterion Optimization (EMO 2007). Springer, 4403:862–876.

Zitzler, E., Deb, K., and Thiele, L. 2000. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Evolutionary Computation, 8(2):173–195.

Zitzler, E., Deb, K., Thiele, L., Coello Coello, C. A., and Corne, D. (Editors) 2001. Evolutionary Multi-Criterion Optimization (EMO 2001), volume 1993 of LNCS. Springer.

Zitzler, E. and Künzli, S. 2004. Indicator-Based Selection in Multiobjective Search. In: Yao, X. et al. (Editors), Conference on Parallel Problem Solving from Nature (PPSN VIII). Springer, 3242:832–842.

Zitzler, E., Laumanns, M., and Bleuler, S. 2004. A Tutorial on Evolutionary Multiobjective Optimization. In: Gandibleux, X. et al. (Editors), Metaheuristics for Multiobjective Optimisation. Springer, 535.

Zitzler, E., Laumanns, M., and Thiele, L. 2002. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In: Giannakoglou, K. et al. (Editors), Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001). International Center for Numerical Methods in Engineering (CIMNE), pages 95–100.

Zitzler, E. and Thiele, L. 1998a. An Evolutionary Approach for Multiobjective Optimization: The Strength Pareto Approach. TIK Report 43, Computer Engineering and Networks Laboratory (TIK), ETH Zurich.

Zitzler, E. and Thiele, L. 1998b. Multiobjective Optimization Using Evolutionary Algorithms - A Comparative Case Study. In: Conference on Parallel Problem Solving from Nature (PPSN V). pages 292–301.

Zitzler, E. and Thiele, L. 1999. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Transactions on Evolutionary Computation, 3(4):257–271.

Zitzler, E., Thiele, L., and Bader, J. 2008. SPAM: Set Preference Algorithm for Multiobjective Optimization. In: Rudolph, G. et al. (Editors), Conference on Parallel Problem Solving From Nature (PPSN X). Springer, 5199:847–858.

Zitzler, E., Thiele, L., and Bader, J. 2009. On Set-Based Multiobjective Optimization. IEEE Transactions on Evolutionary Computation. to appear.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., and Grunert da Fonseca, V. 2003. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. IEEE Transactions on Evolutionary Computation, 7(2):117–132.

A

Complementary Proofs

In the following, we provide the proofs that have been skipped for clarity in the previous chapters. Note that all theorems in this appendix are in the same order they appeared in the previous chapters; also the numbering is the same as before.

A.1 Relation Between δ-Conflict and $\delta_{\max}$

Theorem 8. *Let $\mathcal{F}_1, \mathcal{F}_2$ be two objective sets and $A \subseteq X$ a set of solutions. Then, $\mathcal{F}_1$ is $\overline{\delta}$-non-conflicting with $\mathcal{F}_2$ with respect to A for all $\overline{\delta} \geq \max\{\delta_{max}(A, \mathcal{F}_1, \mathcal{F}_2), \delta_{max}(A, \mathcal{F}_2, \mathcal{F}_1)\}$ and no $\underline{\delta} < \max\{\delta_{max}(A, \mathcal{F}_1, \mathcal{F}_2), \delta_{max}(A, \mathcal{F}_2, \mathcal{F}_1)\}$ exists such that $\mathcal{F}_1$ is $\underline{\delta}$-non-conflicting with $\mathcal{F}_2$.*

Proof. According to the definition of $\delta_{\max}(A, \mathcal{F}', \mathcal{F})$ for two objective sets $\mathcal{F}, \mathcal{F}'$, we can state that

$$\begin{aligned}&[\vec{x} \preceq_{\mathcal{F}_1} \vec{y} \Rightarrow \forall i \in \mathcal{F}_2 : f_i(\vec{x}) \leq f_i(\vec{y}) + \delta_{\max}(A, \mathcal{F}_1, \mathcal{F}_2)]\\&\qquad \wedge [\vec{x} \preceq_{\mathcal{F}_2} \vec{y} \Rightarrow \forall i \in \mathcal{F}_1 : f_i(\vec{x}) \leq f_i(\vec{y}) + \delta_{\max}(A, \mathcal{F}_2, \mathcal{F}_1)]\end{aligned}$$

holds for all solutions $\vec{x}, \vec{y} \in A$ and implies

$$\begin{aligned}
&\forall \overline{\delta} \geq \max\{\delta_{\max}(A, \mathcal{F}_1, \mathcal{F}_2), \delta_{\max}(A, \mathcal{F}_2, \mathcal{F}_1)\} : \\
&\qquad\qquad \forall \vec{x}, \vec{y} \in A : \left[\vec{x} \preceq_{\mathcal{F}_1} \vec{y} \Rightarrow \forall i \in \mathcal{F}_2 : f_i(\vec{x}) \leq f_i(\vec{y}) + \overline{\delta}\right] \\
&\qquad\qquad\qquad\qquad \wedge \left[\vec{x} \preceq_{\mathcal{F}_2} \vec{y} \Rightarrow \forall i \in \mathcal{F}_1 : f_i(\vec{x}) \leq f_i(\vec{y}) + \overline{\delta}\right] \\
&\quad \Longleftrightarrow \forall \overline{\delta} \geq \max\{\delta_{\max}(A, \mathcal{F}_1, \mathcal{F}_2), \delta_{\max}(A, \mathcal{F}_2, \mathcal{F}_1)\} : \\
&\qquad\qquad \forall \vec{x}, \vec{y} \in A : [\vec{x} \preceq_{\mathcal{F}_1} \vec{y} \Rightarrow \vec{x} \preceq^{\overline{\delta}}_{\mathcal{F}_2} \vec{y}] \wedge [\vec{x} \preceq_{\mathcal{F}_2} \vec{y} \Rightarrow \vec{x} \preceq^{\overline{\delta}}_{\mathcal{F}_1} \vec{y}] \\
&\quad \Longleftrightarrow \forall \overline{\delta} \geq \max\{\delta_{\max}(A, \mathcal{F}_1, \mathcal{F}_2), \delta_{\max}(A, \mathcal{F}_2, \mathcal{F}_1)\} : \preceq_{\mathcal{F}_1} \subseteq \preceq^{\overline{\delta}}_{\mathcal{F}_2} \wedge \preceq_{\mathcal{F}_2} \subseteq \preceq^{\overline{\delta}}_{\mathcal{F}_1} \\
&\quad \Longleftrightarrow \mathcal{F}_1 \ \overline{\delta}\text{-non-conflicting with } \mathcal{F}_2 \text{ for all } \overline{\delta} \geq \max\{\delta', \delta''\}
\end{aligned}$$

As a result of the above implication and the definition of $\delta_{\max}$, it is clear that $\mathcal{F}_1$ is $\underline{\delta}$-conflicting with $\mathcal{F}_2$ for any $\underline{\delta} < \max\{\delta_{\max}(A, \mathcal{F}_1, \mathcal{F}_2), \delta_{\max}(A, \mathcal{F}_2, \mathcal{F}_1)\}$. □

A.2 Turing Reduction From MOSS to SCP

In order to prove the approximation ratio of $\Theta(\log |A|)$ for the greedy δ-MOSS algorithm in Chapter 3, we used the following result:

Theorem 19. *The* MOSS *problem is Turing reducible to* SCP*.*

Proof. Given an instance for MOSS, consisting of the relations $\preceq_{\mathcal{F}} \subseteq A \times A$ and $\preceq_i \subseteq A \times A$ with $\bigcap_{1 \leq i \leq k} \preceq_i = \preceq_{\mathcal{F}}$, a polynomial time algorithm $\mathcal{A}$ can compute an SCP instance as follows. The set S in the SCP instance contains one element $s_{\vec{x},\vec{y}}$ for each $(\vec{x}, \vec{y}) \notin \preceq_{\mathcal{F}}$. A subset C_i of S in the SCP instance contains an element $s_{\vec{x},\vec{y}}$ iff $\neg\,(\vec{x} \preceq_i \vec{y})$. The algorithm $\mathcal{A}$ can then use a hypothetical polynomial time bounded exact algorithm for SCP, to compute the index I as an output for the MOSS problem.

The index I, computed by the SCP algorithm, is always a correct output for the MOSS problem. To see that, we show $\forall 1 \leq i \leq k : C_i \subseteq S$, first. Let $s_{\vec{x},\vec{y}} \in C_i$ for any $\vec{x}, \vec{y} \in A$ and any $1 \leq i \leq k$. By definition, $\neg\,(\vec{x} \preceq_i \vec{y})$, i. e., $\neg(f_i(\vec{x}) \leq f_i(\vec{y})) \Longleftrightarrow f_i(\vec{x}) > f_i(\vec{y})$ holds. But then $\neg(\vec{x} \preceq_{\mathcal{F}} \vec{y})$, thus, $s_{\vec{x},\vec{y}} \in S$ by definition.

Now, we are able to show that I is always a correct output for the MOSS problem. We only have to use the rules of deMorgan and the fact that

$C_i \subseteq S$ holds for all $1 \leq i \leq k$.

$$\begin{aligned}\bigcup_{i\in I} C_i = S \iff & \forall s_{\vec{x},\vec{y}} \in S : \exists i \in I : s_{\vec{x},\vec{y}} \in C_i \\ \iff & \forall \vec{x},\vec{y} \in A : [(\exists i \in I : s_{\vec{x},\vec{y}} \in C_i) \Leftrightarrow s_{\vec{x},\vec{y}} \in S] \\ \iff & \forall \vec{x},\vec{y} \in A : [(\exists i \in I : \neg\,(\vec{x} \preceq_i \vec{y})) \Leftrightarrow \neg\,(\vec{x} \preceq_{\mathcal{F}} \vec{y})] \\ \iff & \forall \vec{x},\vec{y} \in A : [(\forall i \in I : \vec{x} \preceq_i \vec{y}) \Leftrightarrow \vec{x} \preceq_{\mathcal{F}} \vec{y}] \\ \iff & \preceq_{\mathcal{F}}^{C} \iff \bigcap_{i\in I} \preceq_i = \preceq_{\mathcal{F}}\end{aligned}$$

By construction, it is clear that a minimum I is always a minimum index for MOSS. □

A.3 Correctness and Running Time Proof for the Exact Algorithm

Theorem 11. *Algorithm 4 solves both the δ-*MOSS *and the* k-EMOSS *problem exactly in time $O(m^2 \cdot k \cdot 2^k)$.*

Proof. To prove the correctness of Algorithm 4, we use Lemma 2 which we state below. It states that Algorithm 4 computes for each considered set M of solution pairs a set of pairs $(\mathcal{F}', \delta')$ of an objective subset $\mathcal{F}' \subseteq \mathcal{F}$ with the corresponding correct δ' value (i, ii) that are minimal (iii, iv). Moreover, the algorithm computes solely minimal pairs (v, vi). With Lemma 2, the correctness of Algorithm 4 follows directly from the lines 13 and 14.

The upper bound on the running time of Algorithm 4 results from the size of the set S_M. For all of the $O(m^2)$ solution pairs, the set $S_{\{(\vec{x},\vec{y})\}}$ can be computed in time $O(k^3) = o(k \cdot 2^k)$, but the computation time for $S_M \sqcup S_{\{(\vec{x},\vec{y})\}}$ can be exponential in k. As S_M contains at most $O(2^k)$ objective subsets of size $O(k)$, the computation of $S_M \sqcup S_{\{(\vec{x},\vec{y})\}}$ in line 10 is possible in time $O(k \cdot 2^k)$ and, therefore, the entire algorithm runs in time $O(m^2 \cdot k \cdot 2^k)$. □

For the following Lemma, we use a new short notation for δ-errors regarding a set M of solution pairs.

Definition 17. *Let $\mathcal{F}' \subseteq \mathcal{F}$ and $M \subseteq A \times A$. Then*

$$\delta(\mathcal{F}', M) := \max_{(\vec{x},\vec{y})\in M} \{\delta_{max}(\{\vec{x},\vec{y}\}, \mathcal{F}', \mathcal{F})\} \quad .$$

Lemma 2. *Given an instance of the δ-*`MOSS` *or the* `k-EMOSS` *problem. Let $\mathcal{F}_1 \subseteq \mathcal{F}$, $\mathcal{F}_1 \neq \emptyset$, be an arbitrary objective set and*

$$M := \{(\vec{x},\vec{y}) \in A \times A \,|\, (\vec{x},\vec{y}) \textit{ considered in Algorithm 4 so far}\} \ .$$

Then there exists always a $(\mathcal{F}_2 \subseteq \mathcal{F}_1, \delta_2) \in S_M$, such that the following six statements hold.

(i) $\delta(\mathcal{F}_2, M) = \delta_2$

(ii) $\delta(\mathcal{F}_1, M) = \delta_2$

(iii) $\nexists (\mathcal{F}_3, \delta_3) \in S_M : \mathcal{F}_3 \subset \mathcal{F}_1 \wedge \delta_3 \leq \delta_2$

(iv) $\nexists (\mathcal{F}_3, \delta_3) \in S_M : \mathcal{F}_3 \subseteq \mathcal{F}_1 \wedge \delta_3 < \delta_2$

(v) $\nexists (\mathcal{F}_3, \delta_3) \in S_M : \mathcal{F}_3 \supset \mathcal{F}_1 \wedge \delta_3 \geq \delta_2$

(vi) $\nexists (\mathcal{F}_3, \delta_3) \in S_M : \mathcal{F}_3 \supseteq \mathcal{F}_1 \wedge \delta_3 > \delta_2$

Proof. The statements (iii)-(vi) hold for any M due to the definition of the $\sqcup$-union in line 10. We, therefore, prove only (i) and (ii) by mathematical induction on $|M|$.

Induction basis: Let $|M| = 1$, i.e., $M := \{(\vec{x},\vec{y})\}$.

(a) $\vec{x}$ and $\vec{y}$ are indifferent: Thus, $\forall i \in \mathcal{F} : f_i(\vec{x}) = f_i(\vec{y})$ and $\forall \mathcal{F}' \subseteq \mathcal{F}, \mathcal{F}' \neq \emptyset : \delta(\mathcal{F}', \{(\vec{x},\vec{y})\}) = 0$. By definition of $\sqcup$, Algorithm 4 computes $S_{\{(\vec{x},\vec{y})\}} = \{(\{i\}, 0) \,|\, 1 \leq i \leq k\}$ correctly according to (i) and (ii).

(b) Without loss of generality $\vec{x} \preceq_{\mathcal{F}} \vec{y} \wedge \neg(\vec{y} \preceq_{\mathcal{F}} \vec{x})$: We can divide $\mathcal{F}$ into two disjoint sets $\mathcal{F}_=, \mathcal{F}_<$ with $\mathcal{F}_= \cup \mathcal{F}_< = \mathcal{F}$, $\mathcal{F}_< \neq \emptyset$, $\forall i \in \mathcal{F}_= : \vec{x} \preceq_i \vec{y} \wedge \vec{y} \preceq_i \vec{x}$, and $\forall i \in \mathcal{F}_< : \vec{x} \preceq_i \vec{y} \wedge \neg(\vec{y} \preceq_i \vec{x})$, i.e., $\forall i \in \mathcal{F}_= : f_i(\vec{x}) = f_i(\vec{y})$ and $\forall i \in \mathcal{F}_< : f_i(\vec{x}) < f_i(\vec{y})$. Furthermore, $\forall i \in \mathcal{F}_< : \delta(\{i\}, \{(\vec{x},\vec{y})\}) = 0$ and $\forall i \in \mathcal{F}_= : \delta(\{i\}, \{(\vec{x},\vec{y})\}) = \delta > 0$ with $\delta := \max_{j \in \mathcal{F}_<}\{f_j(\vec{y}) - f_j(\vec{x})\}$ independent of the choice of i. Therefore, $S_{\{(\vec{x},\vec{y})\}}$ contains all pairs $(\{i\}, \delta_i)$ with $1 \leq i \leq k$ and $\delta_i := \begin{cases} 0 & \text{if } i \in \mathcal{F}_< \\ \delta & \text{if } i \in \mathcal{F}_= \end{cases}$. (i) and (ii) hold, because for any $\mathcal{F}' \subseteq \mathcal{F}$, $\mathcal{F}' \neq \emptyset$, $\delta' := \delta(\mathcal{F}', \{(\vec{x},\vec{y})\})$ is either 0 or δ, depending on $\mathcal{F}' \subseteq \mathcal{F}_=$ ($\Rightarrow \delta' = \delta > 0$) or $\mathcal{F}' \not\subseteq \mathcal{F}_=$ ($\Rightarrow \delta' = 0$).

(c) $\vec{x}$ and $\vec{y}$ are incomparable: We can divide $\mathcal{F}$ into three well-defined disjoint sets $\mathcal{F}_<$, $\mathcal{F}_>$, and $\mathcal{F}_=$ with $\mathcal{F}_< \cup \mathcal{F}_> \cup \mathcal{F}_= = \mathcal{F}$, $\mathcal{F}_< \neq \emptyset$, $\mathcal{F}_> \neq \emptyset$, $\forall i \in \mathcal{F}_< : f_i(\vec{x}) < f_i(\vec{y})$, $\forall i \in \mathcal{F}_> : f_i(\vec{x}) > f_i(\vec{y})$, and $\forall i \in \mathcal{F}_= : f_i(\vec{x}) = f_i(\vec{y})$. For all singletons $\{i\}$ with $1 \leq i \leq k$,

$\delta_i := \delta(\{i\}, \{(\vec{x}, \vec{y})\}) > 0$ holds, i.e., $(\{i\}, \delta_i) \in S_{\{(\vec{x},\vec{y})\}}$ for all $i \in \mathcal{F}$ and

$$\delta_i := \begin{cases} \delta_< := \max_{j \in \mathcal{F}_>}\{f_j(\vec{x}) - f_j(\vec{y})\} & \text{if } i \in \mathcal{F}_< \\ \delta_> := \max_{j \in \mathcal{F}_<}\{f_j(\vec{y}) - f_j(\vec{x})\} & \text{if } i \in \mathcal{F}_> \\ \delta_= := \max_{j \in \mathcal{F}\setminus\{i\}}\{|f_j(\vec{x}) - f_j(\vec{y})|\} & \text{if } i \in \mathcal{F}_= \end{cases} .$$

In addition, $S_{\{(\vec{x},\vec{y})\}}$ contains only those pairs $(\{i, j\}, 0)$ with $i \in \mathcal{F}_< \wedge j \in \mathcal{F}_>$. Other pairs $(\{i, j\}, \delta)$ with $i \neq j \wedge \delta > 0$ are not in $S_{\{(\vec{x},\vec{y})\}}$ due to the $\sqcup$-union in line 7.

Now, let $\mathcal{F}' \subseteq \mathcal{F}$. Then $\mathcal{F}'_<, \mathcal{F}'_>, \mathcal{F}'_= \subseteq \mathcal{F}'$ can be defined similarly to $\mathcal{F}_>$, $\mathcal{F}_>$, and $\mathcal{F}_=$ for $\mathcal{F}$. The statement (i) holds due to the $\sqcup$-union and (ii) holds since $\delta(\mathcal{F}', \{(\vec{x}, \vec{y})\})$ can only take a value $\delta' \in \{0, \delta_<, \delta_>, \delta_=\}$ and a pair $(\mathcal{F}_2 \subseteq \mathcal{F}', \delta')$ exists in $S_{\{(\vec{x},\vec{y})\}}$:

1. $\delta(\mathcal{F}', \{(\vec{x}, \vec{y})\}) = 0$ if $\mathcal{F}'_> \neq \emptyset \wedge \mathcal{F}'_< \neq \emptyset$. But then, $i \in \mathcal{F}'_>$ and $j \in \mathcal{F}'_<$ exist and $(\{i, j\}, 0) \in S_{\{(\vec{x},\vec{y})\}}$.
2. Without loss of generality $\delta(\mathcal{F}', \{(\vec{x}, \vec{y})\}) = \delta_<$ if $\mathcal{F}'_> = \emptyset \wedge \mathcal{F}'_< \neq \emptyset$. Then there exists an $i \in \mathcal{F}'_<$ and $(\{i\}, \delta_<) \in S_{\{(\vec{x},\vec{y})\}}$
3. $\delta(\mathcal{F}', \{(\vec{x}, \vec{y})\}) = \delta_=$ if $\mathcal{F}'_> = \emptyset \wedge \mathcal{F}'_< = \emptyset$. Then $\mathcal{F}' \subseteq \mathcal{F}_=$ and there exists at least one $i \in \mathcal{F}'_=$ such that $(\{i\}, \delta_=) \in S_{\{(\vec{x},\vec{y})\}}$.

Induction step: Let $\mathcal{F}_1 \subseteq \mathcal{F}$ be an arbitrary objective set with error $\delta(\mathcal{F}_1, M \cup \{(\vec{x}, \vec{y})\})$. Assume that (i)-(vi) holds for M and $\{(\vec{x}, \vec{y})\}$. Thus, $\exists(\mathcal{F}_M, \delta_M) \in S_M$ with $\mathcal{F}_M \subseteq \mathcal{F}_1$ and (i)-(vi) and $\exists(\mathcal{F}_{\vec{x}\vec{y}}, \delta_{\vec{x}\vec{y}}) \in S_{\{(\vec{x},\vec{y})\}}$ with $\mathcal{F}_{\vec{x}\vec{y}} \subseteq \mathcal{F}_1$ and (i)-(vi). To show that an $(\mathcal{F}_2 \subseteq \mathcal{F}_1, \delta_2)$ exists in $S_{M \cup \{(\vec{x},\vec{y})\}} := S_M \sqcup S_{\{(\vec{x},\vec{y})\}}$ that fulfills (i) and (ii), we define $\mathcal{F}_2 := \mathcal{F}_M \cup \mathcal{F}_{\vec{x}\vec{y}} \subseteq \mathcal{F}_1$ and $\delta_2 := \max\{\delta_M, \delta_{\vec{x}\vec{y}}\}$. Because of $\delta(\mathcal{F}_M, M) = \delta(\mathcal{F}_1, M)$, $\delta(\mathcal{F}_M, M) = \delta(\mathcal{G}, M)$ holds for any $\mathcal{F}_M \subseteq \mathcal{G} \subseteq \mathcal{F}_1$ and because of $\delta(\mathcal{F}_{\vec{x}\vec{y}}, \{(\vec{x}, \vec{y})\}) = \delta(\mathcal{F}_1, \{(\vec{x}, \vec{y})\})$, $\delta(\mathcal{F}_{\vec{x}\vec{y}}, \{(\vec{x}, \vec{y})\}) = \delta(\mathcal{H}, \{(\vec{x}, \vec{y})\})$ holds for any $\mathcal{F}_{\vec{x}\vec{y}} \subseteq \mathcal{H} \subseteq \mathcal{F}_1$. Together with $\mathcal{F}_M \cup \mathcal{F}_{\vec{x}\vec{y}} \subseteq \mathcal{F}_1$, this yields $\delta(\mathcal{F}_M \cup \mathcal{F}_{\vec{x}\vec{y}}, M) = \delta(\mathcal{F}_1, M)$ as well as $\delta(\mathcal{F}_M \cup \mathcal{F}_{\vec{x}\vec{y}}, \{(\vec{x}, \vec{y})\}) = \delta(\mathcal{F}_1, \{(\vec{x}, \vec{y})\})$. This follows (i) and (ii):

$$\begin{aligned} \delta_2 &= \max\{\delta(\mathcal{F}_M \cup \mathcal{F}_{\vec{x}\vec{y}}, M), \delta(\mathcal{F}_M \cup \mathcal{F}_{\vec{x}\vec{y}}, \{(\vec{x}, \vec{y})\})\} \\ &= \delta(\mathcal{F}_M \cup \mathcal{F}_{\vec{x}\vec{y}}, M \cup \{(\vec{x}, \vec{y})\}) && \text{(i)} \\ &= \max\{\delta(\mathcal{F}_1, M), \delta(\mathcal{F}_1, \{(\vec{x}, \vec{y})\})\} = \delta(\mathcal{F}_1, M \cup \{(\vec{x}, \vec{y})\}) && \text{(ii)} \end{aligned}$$

□

A.4 Correctness and Running Time Proof of the Greedy Algorithm on δ-MOSS

Theorem 13. *Given the objective vectors $f(\vec{x}_1), \dots, f(\vec{x}_m) \in \mathbb{R}^k$ and a $\delta \in \mathbb{R}$, Algorithm 5 always provides an objective subset $\mathcal{F}' \subseteq \mathcal{F}$, δ-non-conflicting with $\mathcal{F} := \{f_1, \dots, f_k\}$ in time $O(\min\{k^3 \cdot m^2, k^2 \cdot m^4\})$.*

Proof. We prove the theorem with the help of the Lemmata 3 and 4 we provide below. If we show that the invariant

$$\forall(\vec{x}, \vec{y}) \in \overline{R} := (A \times A) \setminus R : \quad \vec{x} \preceq_{\mathcal{F}'} \vec{y} \Longleftrightarrow \vec{x} \preceq^{0,\delta}_{\mathcal{F}',\mathcal{F}} \vec{y} \tag{I}$$

holds during each step of Algorithm 5, the theorem is proved, due to Lemma 4 and the fact that $\vec{x} \preceq_{\mathcal{F}'} \vec{y} \Longleftrightarrow \vec{x} \preceq^{0,\delta}_{\mathcal{F}',\mathcal{F}\setminus\mathcal{F}'} \vec{y}$ holds for all $(\vec{x}, \vec{y}) \in A \times A$ if Algorithm 5 terminates, i.e., if $R = \emptyset$. We proof the invariant with induction over $|\overline{R}|$.

Induction basis: When the algorithm starts, $R = (A \times A) \setminus \preceq_{\mathcal{F}}$, i.e., $\overline{R} = \preceq_{\mathcal{F}}$. For each $(\vec{x}, \vec{y}) \in \overline{R} = \preceq_{\mathcal{F}}$ with $\vec{x} \preceq_{\mathcal{F}'} \vec{y}$, i.e., $\vec{x} \preceq_{\emptyset} \vec{y}$ with $\preceq_{\emptyset} := A \times A$, $\vec{x} \preceq_{\mathcal{F}} \vec{y}$ holds and therefore $\vec{x} \preceq^{0,\delta}_{\mathcal{F}',\mathcal{F}\setminus\mathcal{F}'} \vec{y}$. The other direction $\vec{x} \preceq^{0,\delta}_{\mathcal{F}',\mathcal{F}\setminus\mathcal{F}'} \vec{y} \Rightarrow \vec{x} \preceq_{\mathcal{F}'} \vec{y}$ always holds trivially. Thus, the invariant is correct for the smallest possible $|\overline{R}|$, after the initialization of the algorithm.

Induction step: Now let $|\mathcal{F}'| > 0$. Then, the invariant can only become false, if we change R (and with it $\overline{R}$) in line 7 of Algorithm 5. Note, first, that R becomes only smaller by-and-by, i.e., $\overline{R}$ contains more and more pairs $(\vec{x}, \vec{y}) \in A \times A$. Such a pair $(\vec{x}, \vec{y})$, already contained in $\overline{R}$, stays in $\overline{R}$ forever and fulfills the implication in the invariant (I) for every $\mathcal{F}'' \supseteq \mathcal{F}'$ if the pair fulfills it for at least one $\mathcal{F}' \subseteq \mathcal{F}$. If a function f_i is inserted into $\mathcal{F}'$ to gain $\mathcal{F}'' \supseteq \mathcal{F}'$, two possibilities for a pair $(\vec{x}, \vec{y}) \in \overline{R}$ exist. First, if $\vec{x} \npreceq_{\mathcal{F}'} \vec{y}$, then $\vec{x} \npreceq_{\mathcal{F}''} \vec{y}$ for any $\mathcal{F}'' \supseteq \mathcal{F}'$ and also $\vec{x} \npreceq^{0,\delta}_{\mathcal{F}'',\mathcal{F}\setminus\mathcal{F}''} \vec{y}$. Second, if $\vec{x} \preceq_{\mathcal{F}'} \vec{y}$, then $\vec{x} \preceq^{0,\delta}_{\mathcal{F}',\mathcal{F}\setminus\mathcal{F}'} \vec{y}$ by induction hypothesis. Thus, $\vec{x} \preceq^{\delta}_{\mathcal{F}\setminus\mathcal{F}'} \vec{y}$ and $\vec{x} \preceq^{\delta}_{\mathcal{F}\setminus\mathcal{F}''} \vec{y}$ for any $\mathcal{F}'' \supseteq \mathcal{F}'$. If $\vec{x} \preceq_{\mathcal{F}''} \vec{y}$ for any $\mathcal{F}'' \supseteq \mathcal{F}'$, then $\vec{x} \preceq^{0,\delta}_{\mathcal{F}'',\mathcal{F}\setminus\mathcal{F}''} \vec{y}$ and if $\vec{x} \npreceq_{\mathcal{F}''} \vec{y}$ for any $\mathcal{F}'' \supseteq \mathcal{F}'$ then $\vec{x} \npreceq^{0,\delta}_{\mathcal{F}'',\mathcal{F}\setminus\mathcal{F}''} \vec{y}$. Thus, a pair $(\vec{x}, \vec{y}) \in \overline{R}$ will always fulfill the implication in (I) for any $\mathcal{F}'' \supseteq \mathcal{F}'$ if it fulfills it for $\mathcal{F}'$. Beyond, a pair $(\vec{x}, \vec{y}) \in A \times A$ will only be included in $\overline{R}$ during the update of R in line 7 if (i) $(\vec{x}, \vec{y}) \notin (R \cap \preceq_{i^*})$ or if (ii) $(\vec{x}, \vec{y}) \in \preceq^{0,\delta}_{\mathcal{F}'\cup\{i^*\},\mathcal{F}\setminus(\mathcal{F}'\cup\{i^*\})}$. In case (i), the invariant stays true because for all new pairs $(\vec{x}, \vec{y})$ in $\overline{R}$, $(\vec{x}, \vec{y}) \in R \wedge (\vec{x}, \vec{y}) \notin \preceq_{i^*}$ holds. Thus, $(\vec{x}, \vec{y}) \notin \bigcap_{i\in(\mathcal{F}'\cup\{i^*\})} \preceq_i = \preceq_{\mathcal{F}'}$ and, therefore, $(\vec{x}, \vec{y}) \notin \preceq^{0,\delta}_{\mathcal{F}'\cup\{i^*\},\mathcal{F}\setminus(\mathcal{F}'\cup\{i^*\})}$ as well. In the case (ii), $(\vec{x}, \vec{y}) \in \preceq^{0,\delta}_{\mathcal{F}'\cup\{i^*\},\mathcal{F}\setminus(\mathcal{F}'\cup\{i^*\})}$ and trivially $(\vec{x}, \vec{y}) \in \preceq_{\mathcal{F}'\cup\{i^*\}}$, i.e., the invariant remains true, too.

The running time of Algorithm 5 results mainly from the computation of the relations in line 6. The initialization needs time $O(k \cdot m^2)$ altogether.

As the relation $\preceq^{0,\delta}_{\mathcal{F}'\cup\{i^*\},\mathcal{F}\setminus(\mathcal{F}'\cup\{i^*\})}$ is known from line 6, the calculation of the new R in line 7 needs time $O(m^2)$; line 8 needs only constant time. The computation of the relations $\preceq^{0,\delta}_{\mathcal{F}'\cup\{i\},\mathcal{F}\setminus(\mathcal{F}'\cup\{i\})}$ in line 6 needs time $O(k\cdot m^2)$ for each i, thus, line 6 needs time $O(k^2\cdot m^2)$ altogether. Hence, the computation time for each while loop cycle lasts time $O(k^2\cdot m^2)$. Because in each loop cycle, $|\mathcal{F}'|$ increases by one, there are at most k cycles before Algorithm 5 terminates. On the other hand, Algorithm 5 terminates if $R=\emptyset$, i.e., after at most $|A\times A| = O(m^2)$ cycles of the while loop, if in each cycle the value $|R|$ decreases by at least one—what is true due to Theorem 1. The total running time of Algorithm 5 is, therefore, $O(\min\{k,m^2\}\cdot k^2\cdot m^2) = O(\min\{k^3\cdot m^2, k^2\cdot m^4\})$. □

Lemma 3. *Let $\mathcal{F}'\subseteq\mathcal{F}$ and $\delta\geq 0$. Then $\mathcal{F}'$ is δ-non-conflicting with $\mathcal{F}$ if and only if $\preceq_{\mathcal{F}'}\subseteq\preceq^{\delta}_{\mathcal{F}}$.*

Proof. Let $\mathcal{F}'\subseteq\mathcal{F}$. Then for all $\delta\geq 0$ the relation $\preceq_{\mathcal{F}}$ is always a subset of or equal to $\preceq^{\delta}_{\mathcal{F}'}$, because $\vec{x}\preceq_{\mathcal{F}}\vec{y}$ implies that $f_i(\vec{x})\leq f_i(\vec{y})$ for all $f_i\in\mathcal{F}'$ and also $f_i(\vec{x})\leq f_i(\vec{y})+\delta$ holds for all $f_i\in\mathcal{F}'$, i.e., $\vec{x}\preceq^{\delta}_{\mathcal{F}'}\vec{y}$ for all $\vec{x},\vec{y}\in X$. Thus, $\mathcal{F}'$ is δ-non-conflicting with $\mathcal{F}$ iff $\preceq_{\mathcal{F}'}\subseteq\preceq^{\delta}_{\mathcal{F}}\wedge\preceq_{\mathcal{F}}\subseteq\preceq^{\delta}_{\mathcal{F}'}$, i.e., iff $\preceq_{\mathcal{F}'}\subseteq\preceq^{\delta}_{\mathcal{F}}$. □

Lemma 4. *Let $\mathcal{F}'\subseteq\mathcal{F}$ and $\delta>0$. Then*

$$\Big(\forall\vec{x},\vec{y}\in A:\vec{x}\preceq_{\mathcal{F}'}\vec{y}\Longleftrightarrow\vec{x}\preceq^{0,\delta}_{\mathcal{F}',\mathcal{F}\setminus\mathcal{F}'}\vec{y}\Big)$$
$$\Longrightarrow \mathcal{F}' \textit{ is } \delta\textit{-non-conflicting with } \mathcal{F} \textit{ with respect to } A.$$

Proof. Let $\mathcal{F}'\subseteq\mathcal{F}$, $\delta>0$, and $\Big(\forall\vec{x},\vec{y}\in A:\vec{x}\preceq_{\mathcal{F}'}\vec{y}\Longleftrightarrow\vec{x}\preceq^{0,\delta}_{\mathcal{F}',\mathcal{F}\setminus\mathcal{F}'}\vec{y}\Big)$, denoted by $(*)$. We observe the following two statements:

- Let $\delta_1,\delta_2,\delta_1',\delta_2'\in\mathbb{R}$ with $\delta_1\leq\delta_1'$ and $\delta_2\leq\delta_2'$, and $\mathcal{F}_1,\mathcal{F}_2,\mathcal{F}_1',\mathcal{F}_2'$ be objective sets with $\mathcal{F}_1'\subseteq\mathcal{F}_1$ and $\mathcal{F}_2'\subseteq\mathcal{F}_2$. Then both $\preceq^{\delta_1,\delta_2}_{\mathcal{F}_1,\mathcal{F}_2}\subseteq\preceq^{\delta_1',\delta_2'}_{\mathcal{F}_1,\mathcal{F}_2}$ and $\preceq^{\delta_1,\delta_2}_{\mathcal{F}_1,\mathcal{F}_2}\subseteq\preceq^{\delta_1,\delta_2}_{\mathcal{F}_1',\mathcal{F}_2'}$ holds.
- Furthermore, $\preceq^{\delta_1,\delta_2}_{\mathcal{F}_1,\mathcal{F}_2}=\preceq^{\delta_1}_{\mathcal{F}_1}\cap\preceq^{\delta_2}_{\mathcal{F}_2}$ and $\preceq^{\delta,\delta}_{\mathcal{F}_1,\mathcal{F}_2}=\preceq^{\delta}_{\mathcal{F}_1\cup\mathcal{F}_2}$.

With these observations, $\preceq_{\mathcal{F}'}\overset{(*)}{=}\preceq^{0,\delta}_{\mathcal{F}',\mathcal{F}\setminus\mathcal{F}'}=\left(\preceq^{0}_{\mathcal{F}'}\cap\preceq^{\delta}_{\mathcal{F}\setminus\mathcal{F}'}\right)\subseteq\preceq^{\delta}_{\mathcal{F}'}\cap\preceq^{\delta}_{\mathcal{F}\setminus\mathcal{F}'}$ $=\preceq^{\delta}_{\mathcal{F}}$, i.e., $\mathcal{F}'$ is δ-nonconflicting with $\mathcal{F}$ according to Lemma 3. □

B

List of Acronyms

δ-MOSS δ-Minimum Objective Subset Problem

DTLZ Deb, Thiele, Laumanns, and Zitzler's test problem

DTLZ$_{\mathbf{BZ}}$ Deb, Thiele, Laumanns, and Zitzler's test problem, modified by Brockhoff and Zitzler

EA Evolutionary algorithm

EMO Evolutionary multiobjective optimization

ESP Evolution Strategy with Probabilistic Mutation

FEMO Fair Evolutionary Multiobjective Optimizer

IBEA Indicator-Based Evolutionary Algorithm

ICA Independent Component Analysis

k-EMOSS Minimum Objective Subset of Size k With Minimum Error

KP Knapsack problem

LOTZ Leading Ones Trailing Zeros

MOGA Multi-Objective Genetic Algorithm

MCDM Multicriteria decision making

MO-CMA-ES Multiobjective Covariance Matrix Adaptation Evolution Strategy

MOEA Multiobjective evolutionary algorithm

MOSS Minimum Objective Subset Problem

NPGA Niched-Pareto Genetic Algorithm

NSGA Nondominated Sorting Genetic Algorithm

NSGA-II Nondominated Sorting Genetic Algorithm II

OAP Optimal Aggregation Problem

PCA Principal Component Analysis

PESA-II Pareto Envelope based Selection Algorithm II

PISA A Platform and Programming Language Independent Interface for Search Algorithms

REMO Restricted Evolutionary Multiobjective Optimizer

SBX Simulated Binary Crossover

SCP Set Cover Problem

SEMO Simple Evolutionary Multiobjective Optimizer

SIBEA Simple Indicator-Based Evolutionary Algorithm

SMS-EMOA $\mathcal{S}$-metric Selection Evolutionary Multiobjective Algorithm

SPEA2 Strength Pareto Evolutionary Algorithm 2

TC Threshold cut

WFG Walking fish group test problem

ZDT Zitzler, Deb, and Thiele's test problem

C

List of Symbols

2^A	power set of set A
$\|A\|$	number of elements in set A
$\|\vec{x}\|_1$	number of one bits in bitstring $\vec{x}$
$\|\vec{x}\|_0$	number of zero bits in bitstring $\vec{x}$
$\mathcal{O}(g(n))$	set of all functions $f(n) : \mathbb{N} \rightarrow \mathbb{R}$ that are asymptotically not larger than $g(n)$, i.e., $\mathcal{O}(g(n)) = \left\{ f(n) \,\middle\vert\, \exists c \in \mathbb{R} : \lim_{n\rightarrow\infty} \frac{f(n)}{g(n)} \leq c \right\}$
$\Omega(g(n))$	set of all functions $f(n) : \mathbb{N} \rightarrow \mathbb{R}$ that are asymptotically not smaller than $g(n)$, i.e., $\Omega(g(n)) = \{f(n) \mid g(n) \in \mathcal{O}(f(n))\}$
$\Theta(g(n))$	set of all functions $f(n) : \mathbb{N} \rightarrow \mathbb{R}$ that are asymptotically growing as fast as $g(n)$, i.e., $\Theta(g(n)) = \{f(n) \mid f(n) \in \mathcal{O}(g(n)) \wedge f(n) \in \Omega(g(n))\}$
$o(g(n))$	set of all functions $f(n) : \mathbb{N} \rightarrow \mathbb{R}$ that are asymptotically strictly smaller than $g(n)$, i.e. $o(g(n)) = \left\{ f(n) \,\middle\vert\, \lim_{n\rightarrow\infty} \frac{f(n)}{g(n)} = 0 \right\}$
$\omega(g(n))$	set of all functions $f(n) : \mathbb{N} \rightarrow \mathbb{R}$ that are asymptotically strictly larger than $g(n)$, i.e., $\omega(g(n)) = \{f(n) \mid g(n) \in o(f(n))\}$
1^n	a bit string consisting of n ones
0^n	a bit string consisting of n zeros
$\emptyset$	empty set
$\sqcup$	union with simultaneous deletion of not δ-minimal pairs
$\#$	number of

$\lfloor \cdot \rfloor$	floor function
$\preceq_{\mathcal{F}}$	weak Pareto dominance relation induced by the objective functions in $\mathcal{F}$
$\preceq^{\varepsilon}_{\mathcal{F}}$	additive weak ε-dominance relation induced by objective functions in $\mathcal{F}$
$\preceq^{\varepsilon_1,\varepsilon_2}_{\mathcal{F}_1,\mathcal{F}_2}$	generalization of additive weak ε-dominance to two objective sets $\mathcal{F}_1$ and $\mathcal{F}_2$
$\preceq_i$	short form for $\preceq_{\{f_i\}}$
$\leq_T$	Turing reduction
A	set of solutions
α, α_i	weights in weighted sum approach
$\alpha_A(\vec{z})$	attainment function
$d(\vec{x})$	hypervolume loss of solution $\vec{x}$
δ	δ-error in terms of ε-dominance relation
$\delta_{\max}$	maximal δ-error
δ_{avg}	average δ-error
Δ	error function during aggregation
e	Euler's number $e \approx 2.71828$
f	objective function
f^a, f_{new}	aggregated objective
I_H	hypervolume indicator
I_H^w	weighted hypervolume indicator
$\mathcal{F}$	set of objective functions, $\mathcal{F} = \{f_1, \ldots, f_k\}$
$\mathcal{F}', \mathcal{F}'', \mathcal{F}_1, \mathcal{F}_2$	set of objective functions, often a subset of $\mathcal{F}$
G	number of generations
$g_1, g_2, g_{(i)}, g_{(ii)}$	objective functions
g, g_i	distance functions in DTLZ test problems
$h_1, h_2, h_{(i)}, h_{(ii)}$	objective functions
I	indicator function
k	number of objectives in a problem
$\mathtt{k}$	desired number of objectives in an objective subset
λ	number of offspring
m	number of solutions or objective vectors
μ	number of parents, population size
n	number of bits, input length
$\mathbb{N}$	set of all natural numbers
$\mathcal{NP}$	class of all problems for which a polynomial nondeterministic Turing machine exists

$\mathcal{P}$	class of all problems for which a polynomial deterministic Turing machine exists
P	population
$\#\mathcal{P}$	class of all problems of the form "compute the number of accepting paths of a nondeterministic Turing machine"
$\vec{r}$	reference point of hypervolume indicator
$\mathbb{R}$	set of all real numbers
$\mathbb{R}_{>0}$	set of all positive real numbers
$\mathbb{R}_{\geq 0}$	set of all positive real numbers including 0
SP_1	short path $SP_1 = \{1^i 0^{n-i}, 1 \leq i < n\}$
SP_2	short path $SP_2 = \{0^{i1^{n-i}}, 1 \leq i < n\}$
T	time
W	set of weight vectors
$w(\vec{z})$	weight distribution function
X	decision space
$\vec{x}_M$	middle block of bitstring $\vec{x}$, i.e., all entries that are not leading ones or trailing zeros
Z	objective space

D

Curriculum Vitae

Personal Information

Dimo Brockhoff
Born December 5, 1979 in Bochum, Germany
Citizen of Germany

Education

2005–2009 doctoral student at Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland

2000–2005 diploma studies in computer science (Dipl.-Inform.) at the University of Dortmund, Germany

1999–2000 alternative civilian service (Zivildienst), Marienhospital, Witten

1990–1999 secondary school, Abitur at Albert-Martmöller-Gymnasium in Witten, Germany

1986–1990 primary school in Witten, Germany