Offline and Online Objective Reduction in Evolutionary Multiobjective Optimization Based on Objective Conflicts

Dimo Brockhoff and Eckart Zitzler
{brockhoff,zitzler}@tik.ee.ethz.ch

TIK-Report No. 269
Institut für Technische Informatik und Kommunikationsnetze
ETH Zürich
Gloriastrasse 35, CH–8092 Zürich, Schweiz

April 2007

Abstract. In recent years, multiobjective problems with many objectives, i.e., more than three, have gained interest. Since the consideration of many objectives cause obvious problems in terms of visualization, decision making and computational cost, the question arises whether objectives can be omitted to avoid or at least diminish the mentioned problems. To answer the question how an objective reduction can help in tackling problems with many objectives, we both theoretically and experimentally investigate how an addition or omission of objectives affects the problem characteristics. Furthermore, we propose a general definition of conflict between objective sets which provides the basis for a relation-based objective reduction method. Exact and heuristic algorithms to reduce the number of objectives under consideration are developed. How a reduction of the objective set can be utilized both offline, i.e., in the decision making step and online, i.e., within the search is demonstrated for a radar waveform application as well as on well-known test problems.

1 Motivation

In recent years, the number of publications on evolutionary multi-objective optimization has been rapidly growing. Most of the studies, however, investigate problems where the number of considered objectives is low, i.e., between two and four, while studies with many objectives are rare, cf. [Coello Coello et al., 2002]. Why do most of the publications address problems with only few objectives? Obviously, problems with many objectives cause additional difficulties; due to the visualization of high-dimensional data, the necessary computational effort, the increasing size of the Pareto-optimal front or the rising number of incomparable solutions in general. That state-of-the-art search algorithms like NSGA-II or SPEA2 do not scale well with an increasing number of objectives was shown several times empirically, e.g., in [Khare et al., 2003], [Purshouse and Fleming, 2003b], and [Wagner et al., 2007].

Contrariwise, a few studies argue the converse, namely that more objectives can improve the performance of evolutionary algorithms. For example, [Jensen, 2004] uses so-called helper-objectives to guide the search of multiobjective evolutionary algorithms, getting stuck in local minima when considering only a few objectives. That the conversion of single-objective problems to multiobjective ones ("multi-objectivization" in [Knowles et al., 2001]) can decrease the running time of evolutionary algorithms was theoretically confirmed for some combinatorial optimization problems, e.g. by [Scharnow et al., 2004], and [Neumann and Wegener, 2006], and also experimentally, e.g., in terms of reducing bloat in Genetic Programming, amongst others in [Bleuler et al., 2001], [Ekárt and Németh, 2001], and [De Jong et al., 2001].

However, the question why many-objective problems are hard to tackle with state-of-the-art EAs remains unsolved. The fact that many of the multiobjective problems with many objectives were solved by aggregating objectives, cf. for example [Coello Coello et al., 2002], leads us to the assertion that reducing the number of objectives can be, in general, a helpful approach when tackling many-objective problems. With the present study, we support this assertion by addressing the field of objective reduction in evolutionary multiobjective optimization with respect to the following open questions:

– What is the effect of joining or omitting objectives?
– What does, in particular, objective conflict mean?
– Can the objective set be reduced? Under which circumstances?
– How can a reduction of the objective set improve evolutionary algorithms?

To this end, in the following sections we will

– give a general definition of objective conflicts, including a measure of conflict,
– show that the presence or absence of pairwise conflicts do not indicate whether objectives can be omitted while the dominance-relation is preserved,
– propose algorithms to compute minimum objective sets, and
– demonstrate how the proposed objective reduction approach can be utilized within applications in terms of
    - learning about a problem by analyzing Pareto front approximations and
    - saving computation time during search.

2 Current State of Research

Why at all are multiple objectives considered simultaneously? An obvious explanation is the inherent multiobjective property of many real-world problems, i.e., the objectives need to be kept separately to gain information about the tradeoff between the objectives while a decision maker wants to learn about the problem and the trade-offs between the objectives in particular. Although in the end a single solution or a small set of solutions has to be chosen by a decision maker, information about the decision maker's preference is insufficient in the beginning to turn the multiobjective problem into a single-objective one in advance. From a practical point of view it is also desirable with most applications to include as many objectives as possible without the need to specify preferences among the different criteria.

When many objectives are considered simultaneously, many obvious problems occur: (i) a human decision maker cannot handle many objectives due to visualization problems and the huge amount of data; (ii) the set of trade-offs becomes larger with more objectives which causes problems both for a (human) decision maker and search algorithms, and (iii) the needed computation effort, to some extent, increases dramatically with the number of objectives, e.g., if hypervolume indicator based algorithms such as SMS-EMOA, proposed by [Emmerich et al., 2005], are used.

In this context, the question arises why at all many objectives are considered and whether a reduction to less objectives is useful both within search and decision making while computation and visualization problems can be avoided.

Since in many fields like statistics, pattern recognition, data mining, and machine learning, dimensionality reduction methods are of broad interest for preprocessing data in real-world problems as pointed out by [Liu and Motoda, 1998], various dimensionality reduction techniques have been proposed and successfully used, e.g., by [Dai et al., 2006] in biology or by [Kaelbling et al., 2003] in text processing. The general idea of dimensionality reduction methods is to reduce large feature spaces to smaller feature spaces, whereas the variables under consideration are called *features*. Two distinct approaches to reduce the dimensionality of the feature space can be distinguished; they are often referred to as *feature extraction* and *feature selection*.

Given a high-dimensional data set with many "features", the task in feature extraction is to find a new feature space, the data can be embedded into and the size of which is as small as possible. In other words, feature extraction tries to extract a set of (arbitrary) features to explain the data. The emerging features are often new and defined as combinations of the original ones. Methods for this task of feature extraction are, e.g., principal component analysis (PCA, see [Jolliffe, 2002]) and independent component analysis, see [Hyvärinen et al., 2001]. In contrast to the feature extraction approach, the task in feature selection is to find the smallest subset of the *given* features, representing the given data best. The task of finding a smallest subset of features is, in general, $\mathcal{NP}$-hard when formalized as an optimization problem as it was shown by [Charikar et al., 2000]. Therefore, an exhaustive search is necessary to solve some instances of feature selection problems optimally. In practice, various methods based on greedy heuristics as well as evolutionary algorithms have been proposed and applied to feature selection problems, cf. for example [Langley, 1994], [Dash and Liu, 1997], and [Vafaie and De Jong, 1993].

Translated to the multiobjective optimization field, one can ask for a set of arbitrary objectives (feature extraction) or for a subset of given objectives (feature selection) which describes the original problem best. Since new objectives—potentially defined as combinations of the given ones—are not easy to handle in the decision making, we focus on finding subsets of the given objectives, (re-)formulating the original problem best.

Various multiobjective optimization problems have been considered in the literature so far. However, most of the problems have only a few objectives, i.e., < 5, cf. [Coello Coello et al., 2002]. Therefore, the development of dimensionality reduction methods for the case of multiobjective optimization had not obtained top priority in research and not many studies deal with objective reduction. In recent years, the interest shifts more and more to problems with many objectives. Unfortunately, it turned out that state-of-the-art evolutionary algorithms, well-suited for bi- or three-objective problems, have difficulties in approximating the Pareto-optimal front of many-objective problems. For example, [Wagner et al., 2007] showed this fact empirically for well-known algorithms like NSGA-II and SPEA2 on a set of test problems. By means of the focus towards many-objective problems in the last years, the question, how an objective reduction can gain a better performance of known algorithms to tackle problems with many objectives, got challenging in the meantime.

When developing an objective reduction method for evolutionary multiobjective optimization, one has to consider the question on which basis the objective set should be reduced. Different notions of objective conflicts yield different objective reduction methods. For example, [Deb and Saxena, 2006] use the principal component analysis method to compute a set of "the most important conflicting objectives" by omitting the redundant ones, whereas an objective is called redundant if its omission will not change the Pareto-optimal front. Since the approach of [Deb and Saxena, 2006] is based on a correlation-based definition of conflicting objectives, the approach cannot guarantee that the Pareto-dominance relation is preserved while omitting objectives. For this case, [Gal and Leberling, 1977] proposed a method to reduce the number of objectives without changing the dominance structure. Nevertheless, the proposed algorithm for computing the smallest set of objectives not changing the dominance structure is only applicable in the case of linear vector maximum problems, where the objective functions are explicitly known as linear combinations of the (real) decision variables, i.e., the Pareto optimal set is determined within the problem formulation. Thus, the approach is inapplicable in a black-box scenario. For the same reason, a generalization of Gal and Leberling's redundancy theory by [Agrell, 1997] is also not applicable in a black-box scenario.

With this paper, we propose an objective reduction method which both is suited for black-box optimization problems and preserves (most of) the Pareto-dominance relation. A generalized definition of conflicting objectives builds the basis of the proposed objective reduction method.

Before we propose the new conflict definition, we will give a brief overview of conflict definitions, known from literature. Some of these definitions state conflict as a property of the entire set ob objectives, some other are restricted to objective pairs. Several publications define conflicts regarding to the set of Pareto optimal solutions, other with respect to the entire search space. We restate the conflict definitions briefly, but cannot go into details[1].

Definition 1 (Conflict by [Deb, 2001]) *A multiobjective optimization problem contains conflicting objectives if and only if there are trade-offs, i. e., there is no single optimal solution.*

Definition 2 (Conflict by [Tan et al., 2005]) *A set $\mathcal{F}$ of objective functions is said to be conflicting if and only if there are incomparable solution pairs*[2].

Definition 3 (Conflict by [Purshouse and Fleming, 2003a]) *Two objectives i and j are conflicting if there exist at least two solutions where the one has a better ith objective value and a worse jth objective value than the other and vice versa.*

The following example shows that a generalization of the conflict definitions of [Deb, 2001], [Purshouse and Fleming, 2003a], and [Tan et al., 2005] to arbitrary objective sets is crucial for a sufficient and necessary criterion to decide whether objectives can be omitted while the dominance relation is preserved.

Example 1. Figure 1 shows the parallel coordinates plot[3] of three solutions $\mathbf{x}_1$ (solid line), $\mathbf{x}_2$ (dotted) and $\mathbf{x}_3$ (dashed) that are pairwise incomparable. Assuming that $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ represent either the entire search space or the Pareto optimal set, the original objective set $\{f_1, f_2, f_3\}$ is conflicting according to [Deb, 2001] as there is no single optimal solution but three Pareto optimal ones. For the same reason of incomparable solution pairs, the objective set is also conflicting according to [Tan et al., 2005]. Every possible objective pair f_i, f_j with $i, j \in \{1, 2, 3\}, i \neq j$ "exhibits evidence of conflict" as defined by [Purshouse and Fleming, 2003a].

Although the three conflict definitions mislead to the assumption that all objectives are necessary, the objective set $\{f_1, f_2, f_3\}$ contains redundant information: the objective f_2 can be omitted, and all solutions remain incomparable to each other with regard to the objective set $\{f_1, f_3\}$, i.e., the dominance relation on the search space stays unaffected.

Note, that not only the consideration of objective pairs is insufficient to consider whether objectives can be omitted as shown in the example, but the consideration of objective sets of any fixed size cannot provide information on redundancy in general.

Since the above mentioned effect of objective subsets inducing the same dominance relation than all objectives is rather rare, it would be a benefit to distinguish between different degrees of conflict. A possible measure of conflict, proposed in [Brockhoff and Zitzler, 2006a], is illustrated in the following example.

---

[1] For a more detailed overview on the definitions, we refer to [Brockhoff and Zitzler, 2006c].

[2] Two solutions are incomparable iff either is better than the other one in some objectives.

[3] cf. [Purshouse and Fleming, 2003a]

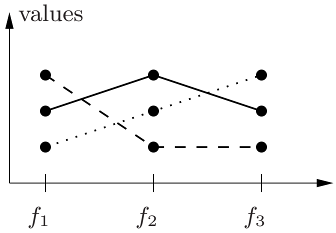

Fig. 1. *Parallel coordinates plot for three solutions and three objectives f_1, f_2, f_3.*

Example 2. Fig. 2 shows the parallel coordinates plot of three solutions $\mathbf{x}_1$ (solid line), $\mathbf{x}_2$ (dashed) and $\mathbf{x}_3$ (dotted) that are pairwise incomparable. At a closer inspection, the objective functions f_1 and f_3 indicate redundancy in the problem formulation, as the corresponding relations are the same. With the above notions, the set $\{f_1, f_2, f_4\}$ is a minimum objective set preserving the dominance structure, i.e., $\mathbf{x}$ weakly dominates $\mathbf{y}$ w.r.t. $\{f_1, f_2, f_4\}$ if and only if $\mathbf{x}$ weakly dominates $\mathbf{y}$ w.r.t. the entire objective set. Because all smaller objective sets yield a changed dominance structure, $\{f_1, f_2, f_4\}$ is minimum.

Considering, for example, the objective subset $\mathcal{F}' := \{f_3, f_4\}$, we observe that, by reducing the set of objectives to $\mathcal{F}'$, the dominances change: on the one hand $\mathbf{x}_1$ weakly dominates $\mathbf{x}_2$ w.r.t. $\mathcal{F}'$; on the other hand $\mathbf{x}_1$ does not weakly dominate $\mathbf{x}_2$ w.r.t. all objectives. In this sense, we make an error: the objective values of $\mathbf{x}_1$ had to be smaller by an additive term of $\delta = 0.5$, such that $\mathbf{x}_1$ would weakly dominate $\mathbf{x}_2$ w.r.t. all objectives. This δ value can be used as a measure to quantify the difference in the dominance structure induced by $\mathcal{F}'$ and the entire objective set. By computing the δ values for all solution pairs $\mathbf{x}, \mathbf{y}$, we can then determine the maximum error. The meaning of the maximum δ value is that whenever we wrongly assume that $\mathbf{x}$ weakly dominates $\mathbf{y}$ w.r.t. $\mathcal{F}'$, we also know that $\mathbf{x}$ is not worse than $\mathbf{y}$ in all objectives by an additive term of δ. For $\mathcal{F}' := \{f_3, f_4\}$, the maximum error is $\delta = 0.5$; for $\mathcal{F}' := \{f_2, f_4\}$, the maximum δ is 4.

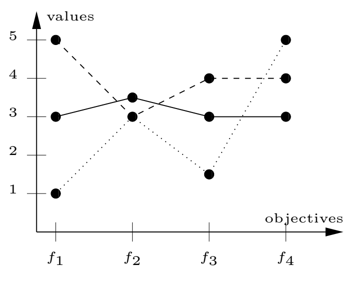

Fig. 2. *Parallel coordinates plot for three solutions and four objectives.*

After these preliminary statements on objective conflicts and objective reduction, we tackle the question whether always all objectives are necessary to induce the dominance relation (with a certain error) by proposing a dimensionality reduction method and applying the approach within evolutionary multiobjective optimization in the remainder of this paper. Before, we briefly restate the general conflict definition of [Brockhoff and Zitzler, 2006a].

3 Objectives, Orders, and a General Notion of Conflict

In the following, we consider—unless otherwise noted—multiobjective minimization problems. The vector function $f : X \to Z \subseteq \mathbb{R}^k$ maps a solution or decision vector $\mathbf{x}$ in the decision space X into an objective vector $f(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})) \in \mathbb{R}^k$ in the objective space $Z = \mathbb{R}^k$.

Furthermore, we assume that the minimal elements of the weak Pareto dominance relation

$$\preceq_{\mathcal{F}} := \{(\mathbf{x}, \mathbf{y}) \,|\, \mathbf{x}, \mathbf{y} \in X \land \forall f \in \mathcal{F} : f(\mathbf{x}) \leq f(\mathbf{y})\}^4$$

are sought, where $\mathcal{F} := \{f_1, \ldots, f_k\}$ is the set of all k objective functions[5]. We say $\mathbf{x}$ *weakly dominates* $\mathbf{y}$ *w.r.t. objective set* $\mathcal{F}$ whenever $\mathbf{x} \preceq_{\mathcal{F}} \mathbf{y}$; if neither $\mathbf{x}$ weakly dominates $\mathbf{y}$ nor $\mathbf{y}$ weakly dominates $\mathbf{x}$, we say $\mathbf{x}$ and $\mathbf{y}$ are *incomparable*. Two solutions which are mutually dominating each other are called *indifferent*.

The minimal elements $\min_{\mathbf{x} \in X}\{\mathbf{x}\}$ of X w.r.t. the weak dominance relation $\preceq_{\mathcal{F}}$ are denoted as Pareto-optimal and constitute the Pareto set, whereas their image $f(\min_{\mathbf{x} \in X}\{\mathbf{x}\})$ in objective space is called Pareto (optimal) front.

The observation used below, that for any objective function set the generalized weak Pareto dominance relation can be derived from the objective-wise less than or equal relation on $\mathbb{R}$, is shown in the following theorem.

Theorem 1. *Let $\mathcal{F} = \{f_1, \ldots, f_k\}$ be a set of k different objective functions. Then it holds:*

$$\preceq_{\mathcal{F}} = \bigcap_{1 \leq i \leq k} \preceq_i$$

Proof. Let $\mathbf{x}, \mathbf{y} \in X$, then $(\mathbf{x}, \mathbf{y}) \in \preceq_{\mathcal{F}} \iff \mathbf{x} \preceq \mathbf{y}$ w.r.t.$\mathcal{F} \iff \forall i \in \{1, \ldots, k\} : f_i(\mathbf{x}) \leq f_i(\mathbf{y}) \iff \forall i \in \{1, \ldots, k\} : \mathbf{x} \preceq \mathbf{y}$ w.r.t.$f_i \iff \forall i \in \{1, \ldots, k\} : (\mathbf{x}, \mathbf{y}) \in \preceq_i \iff (\mathbf{x}, \mathbf{y}) \in \bigcap_{1 \leq i \leq k} \preceq_i$.

Note that the above equivalence also holds for the strict dominance relation $\prec\prec$ and the multiplicative ε-dominance relation $\preceq_{\varepsilon}$, cf. [Zitzler et al., 2003], but does not apply to the regular Pareto dominance relation $\prec$ defined as $\mathbf{x}_1 \prec \mathbf{x}_2 :\iff \mathbf{x}_1 \preceq \mathbf{x}_2 \land \neg(\mathbf{x}_2 \preceq \mathbf{x}_1)$.

[4] We will write $\preceq_i$ if we mean the weak dominance relation w.r.t. $\mathcal{F} = \{f_i\}$; in addition, we define $\preceq_{\emptyset} := X \times X$ for the case that $\mathcal{F}$ is empty.

[5] Other dominance relations such as epsilon dominance, cf. [Zitzler et al., 2003], could be taken as well, and the following discussions apply to any preorder on X that is defined by a corresponding partial order on $\mathbb{R}^k$.

Corollary 1. *The same equivalence as in Theorem 1 holds for the ε-dominance relations*[6] *defined by [Zitzler et al., 2003]:*

$$\preceq_{\mathcal{F}}^{\varepsilon,\circ} = \bigcap_{i \in \mathcal{F}} \preceq_i^{\varepsilon,\circ} \quad \textit{with } \circ \in \{+, \cdot\}$$

To come up with a definition of conflicts between objective sets, we first take a look at the possible changes in the dominance structure, when omitting or adding objectives. We know from Theorem 1 that the weak dominance relation is always an intersection of relations $\preceq_i$. Thus, by considering a larger number of objectives, the number of comparable solution pairs can only decrease; comparable solutions can become incomparable, but also indifferent solutions can become comparable. Contrary, when objectives are omitted, the number of comparable solution pairs can only increase.

Since an omission of objectives will usually result in a changed problem formulation, i.e., the dominance relations $\preceq_{\mathcal{F}' \subset \mathcal{F}}$ and $\preceq_{\mathcal{F}}$ are not exactly the same, the question arises how such a structural change can be quantified. A possible measure for changes in the dominance structure according to Example 2 was recently proposed by [Brockhoff and Zitzler, 2006a]: the definition of δ-conflicting objective sets based on the (additive) ε-dominance relation.

Definition 4 *Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two objective sets. We define*

$$\mathcal{F}_1 \sqsubseteq^{\delta} \mathcal{F}_2 :\Longleftrightarrow \preceq_{\mathcal{F}_1} \subseteq \preceq_{\mathcal{F}_2}^{\delta} .$$

Definition 5 *Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two objective sets. We call $\mathcal{F}_1$* δ-nonconflicting *with $\mathcal{F}_2$ iff $\left(\mathcal{F}_1 \sqsubseteq^{\delta} \mathcal{F}_2\right) \wedge \left(\mathcal{F}_2 \sqsubseteq^{\delta} \mathcal{F}_1\right)$.*

Definition 5 is useful for changing a problem formulation by considering a different objective set. If a multiobjective optimization problem uses the objective set $\mathcal{F}_1$ and one can prove that $\mathcal{F}_1$ is δ-nonconflicting with another objective set $\mathcal{F}_2$, one can easily replace $\mathcal{F}_1$ with $\mathcal{F}_2$ and can be sure that in the new formulation, for any $\mathbf{x}, \mathbf{y} \in X$, $\mathbf{x}$ either weakly dominates $\mathbf{y}$ w.r.t. $\mathcal{F}_2$ or $\mathbf{x}$ ε-dominates $\mathbf{y}$ w.r.t. $\mathcal{F}_2$ if $\mathbf{x}$ weakly dominates $\mathbf{y}$ w.r.t. $\mathcal{F}_1$ and $\varepsilon = \delta$. In the special case of an objective subset $\mathcal{F}' \subseteq \mathcal{F}$, δ-nonconflicting with all objectives $\mathcal{F}$, the definition fits the intuitive measure of error in Example 2. If an objective subset $\mathcal{F}' \subset \mathcal{F}$ is δ-nonconflicting with the set $\mathcal{F}$ of all objectives, $\mathbf{x}$ δ-dominates $\mathbf{y}$, i.e., $\forall i \in \mathcal{F} : f_i(\mathbf{x}) - \delta \leq f_i(\mathbf{y})$, whenever $\mathbf{x}$ weakly dominates $\mathbf{y}$ w.r.t. the reduced objective set $\mathcal{F}'$. We, then, can omit all objectives in $\mathcal{F} \setminus \mathcal{F}'$ without making a larger error than δ in the omitted objectives.

4 Computing Minimum Objective Sets

With the definition of δ-conflict, the question arises whether a smallest set of objectives can be derived which is δ-nonconflicting with the entire objective set.

[6] We distinguish between an additive version of the ε-dominance relation $\preceq_{\mathcal{F}'}^{\varepsilon,+} := \preceq_{\mathcal{F}'}^{\varepsilon} = \{(\mathbf{x}, \mathbf{y}) \,|\, \mathbf{x}, \mathbf{y} \in X \wedge \forall i \in \mathcal{F}' \subseteq \mathcal{F} : f_i(\mathbf{x}) - \varepsilon \leq f_i(\mathbf{y})\}$ and the multiplicative one $\preceq_{\mathcal{F}'}^{\varepsilon,\cdot} := \{(\mathbf{x}, \mathbf{y}) \,|\, \mathbf{x}, \mathbf{y} \in X \wedge \forall i \in F' \subseteq \mathcal{F} : \varepsilon \cdot f_i(\mathbf{x}) \leq f_i(\mathbf{y})\}$

The following section formalizes this topic by defining minimal, minimum, and redundant objective sets (Sec. 4.1) and the optimization problems δ-`MOSS` and `k-EMOSS` (Sec. 4.2). In Sec. 4.3 and Sec. 4.4, we present algorithms to tackle the two objective reduction problems and conclude this section with experiments, comparing the proposed algorithms.

4.1 Minimal, Minimum, and Redundant Objective Sets

Before we define δ-minimal, δ-minimum, and δ-redundant objective sets formally, we want to give an intuitive impression of these three terms. An objective set $\mathcal{F}'$ should be called δ-minimal w.r.t. another set $\mathcal{F}$[7] if

(i) the set $\mathcal{F}'$ is not δ-conflicting with $\mathcal{F}$ but
(ii) for all smaller errors δ', $\mathcal{F}'$ is δ'-conflicting with $\mathcal{F}'$, and
(iii) the set $\mathcal{F}'$ cannot be further reduced without preserving property (i).

This definition of a δ-minimal set is a "local" property, i.e., one can decide only by considering the objective set $\mathcal{F}'$ itself (and the set $\mathcal{F}$ of course) whether $\mathcal{F}'$ is δ-minimal w.r.t. $\mathcal{F}$ or not. In contrast to the notion of δ-minimal objective sets, we want to introduce the notion of a δ-minimum objective set, i.e., a δ-minimal set with the additional property that all other subsets of $\mathcal{F}$ that are also δ-minimal w.r.t. $\mathcal{F}$ contain at least as many objectives as $\mathcal{F}'$. An objective set, δ-minimum w.r.t. a set $\mathcal{F}$, is therefore at all times also δ-minimal w.r.t. $\mathcal{F}$ but not vice versa. Translated into Example 2, the set $\{f_1, f_3\}$ is 2-minimal but not 2-minimum w.r.t. the entire objective set, since the set $\{f_3, f_4\}$ of the same size is 0.5-minimal w.r.t. the entire objective set. For the case of no error, a (0-)minimal objective set is a subset of the original objectives that cannot be further reduced without changing the associated preorder; a minimum objective set, however, is the smallest possible set of original objectives that preserves the original order on the search space. The following definition will formalize the described two forms of minimality.

Definition 6 *Let $\mathcal{F}$ be a set of objectives and $\delta \in \mathbb{R}$. An objective set $\mathcal{F}' \subseteq \mathcal{F}$ is denoted as*

- δ-minimal w.r.t. $\mathcal{F}$ *iff (i) $\mathcal{F}'$ is δ-nonconflicting with $\mathcal{F}$, (ii) $\mathcal{F}'$ is δ'-conflicting with $\mathcal{F}$ for all $\delta' < \delta$, and (iii) there exists no $\mathcal{F}'' \subset \mathcal{F}'$ that is δ-nonconflicting with $\mathcal{F}$;*
- δ-minimum w.r.t. $\mathcal{F}$ *iff (i) $\mathcal{F}'$ is δ-minimal w.r.t. $\mathcal{F}$, and (ii) there exists no $\mathcal{F}'' \subset \mathcal{F}$ with $|\mathcal{F}''| < |\mathcal{F}'|$ that is δ-minimal w.r.t. $\mathcal{F}$.*

Definition 7 *A set $\mathcal{F}$ of objectives is called* δ-redundant *if and only if there exists an $\mathcal{F}' \subset \mathcal{F}$ that is δ-minimal w.r.t. $\mathcal{F}$.*

Before we formalize the problems of finding δ-minimum objective sets, we briefly state the following theorems. They are necessary for the algorithms we propose in the next section. Proofs can be found in the appendix.

[7] Usually, the entire objective set will be used as $\mathcal{F}$.

Theorem 2. *Let $\mathcal{F}' \subseteq \mathcal{F}$. Then $\mathcal{F}'$ is δ-nonconflicting with $\mathcal{F}$ if and only if $\preceq_{\mathcal{F}'} \subseteq \preceq^{\delta}_{\mathcal{F}}$.*

Theorem 3. *Let $\mathcal{F}_1, \mathcal{F}_2$ two objective sets and X a decision space. If*

$$\delta' := \max_{\substack{\mathbf{x},\mathbf{y}\in X \wedge \mathbf{x} \preceq_{\mathcal{F}_1} \mathbf{y} \\ i \in \mathcal{F}_2}} \{f_i(\mathbf{x}) - f_i(\mathbf{y})\} \text{ and } \delta'' := \max_{\substack{\mathbf{x},\mathbf{y}\in X \wedge \mathbf{x} \preceq_{\mathcal{F}_2} \mathbf{y} \\ i \in \mathcal{F}_1}} \{f_i(\mathbf{x}) - f_i(\mathbf{y}),\}$$

then, $\mathcal{F}_1$ is $\overline{\delta}$-nonconflicting with $\mathcal{F}_2$ w.r.t. X for all $\overline{\delta} \geq \max(\delta', \delta'')$ and no $\underline{\delta} < \max\{\delta', \delta''\}$ exists such that $\mathcal{F}_1$ is $\underline{\delta}$-nonconflicting with $\mathcal{F}_2$.

Note, that if $\mathcal{F}_1 \subseteq \mathcal{F}_2$, the theorem can be shortened to $\mathcal{F}_1$ is δ-nonconflicting with $\mathcal{F}_2$ for all $\delta \geq \delta'$ but for no $\delta < \delta'$ if $\delta' := \max_{\mathbf{x},\mathbf{y}\in X \wedge \mathbf{x} \preceq_{\mathcal{F}_1} \mathbf{y}, i \in \mathcal{F}_2}\{f_i(\mathbf{x}) - f_i(\mathbf{y})\}$.

4.2 The MOSS, δ-MOSS, and k-EMOSS Problems

In this section we (re-)formulate three problems, emerging from the above discussion on objective reduction. The problem `MINIMUM OBJECTIVE SUBSET` (`MOSS`) from [Brockhoff and Zitzler, 2006b] asks for a minimum objective set, preserving the dominance structure. The problems δ-`MINIMUM OBJECTIVE SUBSET` and `MINIMUM OBJECTIVE SUBSET OF SIZE k WITH MINIMUM ERROR` already proposed in [Brockhoff and Zitzler, 2006a] correspond to questions arising with the generalization to δ-redundancy.

Based on Sec. 3, the problem `MINIMUM OBJECTIVE SUBSET` (`MOSS`) can be characterized as follows.

Definition 8 *Given a multiobjective optimization problem, the problem `MINIMUM OBJECTIVE SUBSET` (`MOSS`) is defined as follows.*

Instance:	*The set A of solutions, the generalized weak Pareto dominance relation $\preceq_{\mathcal{F}}$, and for all k objective functions $f_i \in \mathcal{F}$ the single relations $\preceq_i$, where $\bigcap_{1\leq i\leq k} \preceq_i = \preceq_{\mathcal{F}}$.*
Task:	*Compute a minimum objective subset $\mathcal{F}' \subseteq \mathcal{F}$ w.r.t. $\mathcal{F}$.*

As the set A of solutions, we can imagine either the entire search space $A = X$ or a Pareto front approximation $A \subseteq X$. The weak dominance relations have to be only defined on the set A of solutions, i.e., $\preceq_i \subseteq A \times A$. Thus, an instance of `MOSS` has a size of $O(k \cdot m^2)$, with $m := |A|$. Because `MOSS` is a kind of set cover problem, it can be proved to be $\mathcal{NP}$-hard, cf. [Brockhoff and Zitzler, 2006c] for details.

When generalizing the `MOSS` problem to deal with dominance structure changes, we can (i) given a δ, ask for a δ-minimum set or (ii) given a certain `k`, ask for an objective set $\mathcal{F}'$ with at most `k` objectives and a minimal δ, such that $\mathcal{F}'$ is δ-nonconflicting with the set of all objectives. Question (i) is denoted as the δ-`MOSS` problem and (ii) as the `k-EMOSS` problem, both already proposed in [Brockhoff and Zitzler, 2006a].

Definition 9 *Given a multiobjective optimization problem, the problem δ-`MINIMUM OBJECTIVE SUBSET` (δ-`MOSS`) is defined as follows.*

Instance: *The objective vectors $f(\mathbf{x}_1), \ldots, f(\mathbf{x}_m) \in \mathbb{R}^k$ of the solutions $\mathbf{x}_1, \ldots, \mathbf{x}_m \in A \subseteq X$ and a $\delta \in \mathbb{R}$.*

Task: *Compute a δ-minimum objective subset $\mathcal{F}' \subseteq \mathcal{F}$ w.r.t. $\mathcal{F}$.*

Definition 10 *Given a multiobjective optimization problem, the problem `MINIMUM OBJECTIVE SUBSET OF SIZE k WITH MINIMUM ERROR` (`k-EMOSS`) is defined as follows.*

Instance: *The objective vectors $f(\mathbf{x}_1), \ldots, f(\mathbf{x}_m) \in \mathbb{R}^k$ of the solutions $\mathbf{x}_1, \ldots, \mathbf{x}_m \in A \subseteq X$ and a $\mathtt{k} \in \mathbb{R}$.*

Task: *Compute an objective subset $\mathcal{F}' \subseteq \mathcal{F}$ which has size $|\mathcal{F}'| \leq \mathtt{k}$ and is δ-nonconflicting with $\mathcal{F}$ with the minimal possible δ.*

The δ-`MOSS` problem contains the `MOSS` problem as the special case with $\delta = 0$, although the two problems differ in its input instances. Nevertheless, the instances can be transformed into each other by a simple algorithm, the running time of which is linear in the input size. A `MOSS` instance, e.g., can be transferred into a δ-`MOSS` instance by using a topological sorting of the orders $\preceq_i$ as objective values $f(\mathbf{x}_1), \ldots, f(\mathbf{x}_m)$. In the following we give attention to the more general problems δ-`MOSS` and `k-EMOSS` and propose an exact algorithm and heuristics for them.

4.3 An Exact Algorithm

Although the problems stated above are $\mathcal{NP}$-hard, we propose with Algorithm 1 an exact algorithm, appropriate to deal with both problem formulations δ-`MOSS`, and `k-EMOSS` respectively. The running time of Algorithm 1 is polynomial in $|A|$ but exponential in the number k of objectives. Nevertheless, the exact algorithm is applicable for instances with only few objectives and a moderate number of solutions as experimental results show (Sec. 4.5).

Instead of simply considering all 2^k possible objective subsets and computing whether they are minimal w.r.t. the set $\mathcal{F}$ of all objectives and the set A of solutions, the basic idea of Algorithm 1 is to consider solution pairs separately. This separate information is then combined to get all minimal objective sets for increasing sets of solution pairs. Algorithm 1 considers all solution pairs $(\mathbf{x}, \mathbf{y})$ successively in arbitrary order and stores in S_M all minimal objective subsets $\mathcal{F}'$ together with the minimal δ' value such that $\mathcal{F}'$ is δ'-nonconflicting with the set $\mathcal{F}$ of all objectives when taking into account only the solution pairs in M, considered so far.

The algorithm uses a subfunction $\delta_{\min}(\mathcal{F}_1, \mathcal{F}_2)$, that computes the minimal δ error for two solutions $\mathbf{x}, \mathbf{y} \in A$, such that $\mathcal{F}_1$ is δ-nonconflicting with $\mathcal{F}_2$ w.r.t. $\mathbf{x}, \mathbf{y}$ according to Theorem 3. Furthermore, Algorithm 1 computes the union $\sqcup$ of two sets of objective subsets with simultaneous deletion of not δ'-minimal pairs $(\mathcal{F}', \delta')$:

Algorithm 1 An exact algorithm for the problems δ-MOSS and k-EMOSS

1: Init:
2: $\quad M := \emptyset, \quad S_M := \emptyset$
3: **for all** pairs $\mathbf{x}, \mathbf{y} \in A$, $\mathbf{x} \neq \mathbf{y}$ of solutions **do**
4: $\quad S_{\{(\mathbf{x},\mathbf{y})\}} := \emptyset$
5: $\quad$ **for all** objective pairs $i, j \in \mathcal{F}$, not necessary $i \neq j$ **do**
6: $\quad\quad$ compute $\delta_{ij} := \delta_{\min}(\{i\} \cup \{j\}, \mathcal{F})$ w.r.t. $\mathbf{x}, \mathbf{y}$
7: $\quad\quad S_{\{(\mathbf{x},\mathbf{y})\}} := S_{\{(\mathbf{x},\mathbf{y})\}} \sqcup (\{i\} \cup \{j\}, \delta_{ij})$
8: $\quad$ **end for**
9: $\quad S_{M \cup \{(\mathbf{x},\mathbf{y})\}} := S_M \sqcup S_{\{(\mathbf{x},\mathbf{y})\}}$
10: $\quad M := M \cup \{(\mathbf{x}, \mathbf{y})\}$
11: **end for**
12: Output for δ-MOSS: $\quad (s_{\min}, \delta_{\min})$ in S_M with minimal size $|s_{\min}|$ and $\delta_{min} \leq \delta$
13: Output for k-EMOSS: $\quad (s, \delta)$ in S_M with size $|s| \leq$ k and minimal δ

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

The correctness proof of Algorithm 1—as well as the proof of its running time of $O(m^2 \cdot k \cdot 2^k)$—can be found in Appendix B. Note, that the exact algorithm can be easily parallelized, as the computation of the sets $S_{\{(\mathbf{x},\mathbf{y})\}}$ are independent for different pairs $(\mathbf{x}, \mathbf{y})$. It can also be accelerated if line 9 of Algorithm 1 is tailored to either the δ-MOSS or the k-EMOSS problem by including a pair $(\mathcal{F}', \delta')$ into $S_{M \cup \{(\mathbf{x},\mathbf{y})\}}$ only if $\delta' \leq \delta$, and $|\mathcal{F}'| \leq$ k respectively. Input instances for which Algorithm 1 provably needs exponential time are known; we refer to [Brockhoff and Zitzler, 2006c] for an example.

4.4 Heuristics

The two heuristic algorithms, we propose in this section, are better suited for large instances of the δ-MOSS problem, and k-EMOSS respectively, than the above exact algorithm. They are much faster but therefore do not guarantee to find a minimum objective set. Furthermore, not even a minimal set can be assured. Nevertheless, the computed objective sets can keep up with the sets computed with the exact algorithm (Sec. 4.5).

A Greedy Algorithm for δ-MOSS Before we propose an approximation algorithm for the δ-MOSS problem, we introduce a generalization of the weak ε-dominance $\preceq^{\varepsilon}_{\mathcal{F}}$, used in the algorithm. We also present observations on the new dominance relation which are necessary for the correctness proof of the algorithm.

Algorithm 2 A Greedy Algorithm for δ-`MOSS`.

Init:
  compute the relations $\preceq_i$ for all $1 \leq i \leq k$ and $\preceq_{\mathcal{F}}$
  $\mathcal{F}' := \emptyset$
  $R := X \times X \setminus \preceq_{\mathcal{F}}$
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

Definition 11 *Let $\delta_1, \ldots, \delta_k \in \mathbb{R}$ and $\mathcal{F}_1, \ldots, \mathcal{F}_k$ objective subsets. We define the $(\delta_1, \ldots, \delta_k)$-dominance relation on X for all $\mathbf{x}, \mathbf{y} \in X$ as*

$$\mathbf{x} \preceq^{\delta_1, \ldots, \delta_k}_{\mathcal{F}_1, \ldots, \mathcal{F}_k} \mathbf{y} :\Longleftrightarrow \forall 1 \leq i \leq k : \forall j \in \mathcal{F}_i : f_j(\mathbf{x}) - \delta_i \leq f_j(\mathbf{y}).$$

Observation 1 *Let $\delta_1, \ldots, \delta_k, \delta'_1, \ldots, \delta'_k \in \mathbb{R}$ with $\forall 1 \leq i \leq k : \delta_i \leq \delta'_i$, and $\mathcal{F}_1, \ldots, \mathcal{F}_k, \mathcal{F}'_1, \ldots, \mathcal{F}_k$ objective sets with $\forall 1 \leq i \leq k : \mathcal{F}'_i \subseteq \mathcal{F}_i$. Then both $\preceq^{\delta_1, \ldots, \delta_k}_{\mathcal{F}_1, \ldots, \mathcal{F}_k} \subseteq \preceq^{\delta'_1, \ldots, \delta'_k}_{\mathcal{F}_1, \ldots, \mathcal{F}_k}$ and $\preceq^{\delta_1, \ldots, \delta_k}_{\mathcal{F}_1, \ldots, \mathcal{F}_k} \subseteq \preceq^{\delta_1, \ldots, \delta_k}_{\mathcal{F}'_1, \ldots, \mathcal{F}'_k}$ holds.*

Observation 2 *Furthermore, $\preceq^{\delta_1, \ldots, \delta_k}_{\mathcal{F}_1, \ldots, \mathcal{F}_k} = \bigcap_{1 \leq i \leq k} \preceq^{\delta_i}_{\mathcal{F}_i}$ and $\preceq^{\delta, \ldots, \delta}_{\mathcal{F}_1, \ldots, \mathcal{F}_k} = \preceq^{\delta}_{\bigcup_i \mathcal{F}_i}$.*

Observation 3 *Let $\delta \in \mathbb{R}$ and $f_i \in \mathcal{F}$ for all $1 \leq i \leq k$. Then*

$$\bigcap_{i \in \mathcal{F}} \preceq^{\delta}_i = \preceq^{\delta}_{\mathcal{F}} .$$

Algorithm 2, as an approximation algorithm for δ-`MOSS`, computes an objective subset $\mathcal{F}'$, δ-nonconflicting with the set $\mathcal{F}$ of all objectives in a greedy way. Starting with an empty set $\mathcal{F}'$, Algorithm 2 chooses in each step the objective f_i which yields the smallest set $\preceq_{\mathcal{F}'} \cap \preceq_i$ without considering the relationships in $\preceq^{0,\delta}_{\mathcal{F}' \cup \{i\}, \mathcal{F}}$ until $\mathcal{F}'$ is δ-nonconflicting with $\mathcal{F}$. For the correctness proof of Algorithm 2 and the proof of its running time of $O(\min\{k^3 \cdot m^2, k^2 \cdot m^4\})$ we once again refer to Appendix B. Note, that Algorithm 2 not necessarily yields a δ-minimal or even δ-minimum objective set w.r.t. $\mathcal{F}$.

A Greedy Algorithm for `k-EMOSS` Algorithm 3 is an approximation algorithm for the `k-EMOSS` problem. It supplies always an objective subset of size `k` which is δ-nonconflicting with the entire objective set but does not guarantee to find the set with minimal δ. The greedy algorithm needs time $O(m^2 \cdot k^3)$ since at most $\mathtt{k} \leq k$ 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. Note, that Algorithm 3 can be accelerated in a concrete implementation as the while loop can be aborted if either $|\mathcal{F}'| = \mathtt{k}$ or $\delta_{min}(\mathcal{F}', \mathcal{F}) = 0$.

Algorithm 3 A greedy algorithm for `k-EMOSS`

1: Init:
2: $\mathcal{F}' := \emptyset$
3: **while** $|\mathcal{F}'| < \mathtt{k}$ **do**
4: $\mathcal{F}' := \mathcal{F}' \cup \underset{i \in \mathcal{F} \setminus \mathcal{F}'}{\operatorname{argmin}} \{\delta_{\min}(F' \cup \{i\}, \mathcal{F}) \text{ w.r.t. } X\}$
5: **end while**

4.5 Comparison of Exact and Greedy Methods

The experiments, presented in this section, serve two goals: (i) to investigate the size of a δ-minimum objective subset as well as the error of computed objective sets for `k-EMOSS` depending on the size of the search space and the number of original objective functions, and (ii) to compare the exact and the approximative algorithms with respect to the size of the generated objective subsets and the corresponding running times. Both issues have been considered on various test problems.

`MOSS`, $\boldsymbol{\delta}$-`MOSS`, and `k-EMOSS` Problems To show the potentials of our objective reduction method, we test the proposed approach of objective reduction in a simple scenario. We generate the objective values for a set of solutions at random where the objective values are chosen uniformly distributed in $[0, 1] \subset \mathbb{R}$. This corresponds to randomly chosen solutions of a problem with objectives, the induced linear (pre)orders of which are chosen uniform randomly from the set of all linear preorders. For different solution set sizes and increasing number of objectives, we use the exact Algorithm 1 to compute a minimum objective set. The sizes of the minimum objective sets, averaged over 100 independent random samples, are shown in Fig. 3.

For all tested solution set sizes, the resulting sizes of the minimum objective subsets behave similar: 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. Nevertheless, we expect that the size of a minimum set decreases down to 2 if the number of objectives is increased to infinity, independently of the number of solutions. The reason is the increasing probability of an objective pair, the intersection of which yields the given preorder on the solutions, if more and more objectives are taken into account.

In addition to the objective omission on the random problem where we tried to preserve the underlying preorder, we also look at the generalized objective reduction methods, based on δ-`MOSS` and `k-EMOSS` for six different test problems. We use the indicator based evolutionary algorithm IBEA, proposed in [Zitzler and Künzli, 2004], to generate a Pareto front approximation for each of the six test problems first. The population size μ is varying with the number k of objectives, i.e., $\mu = 100$ for $k = 5$, $\mu = 200$ for $k = 15$, and $\mu = 300$ for

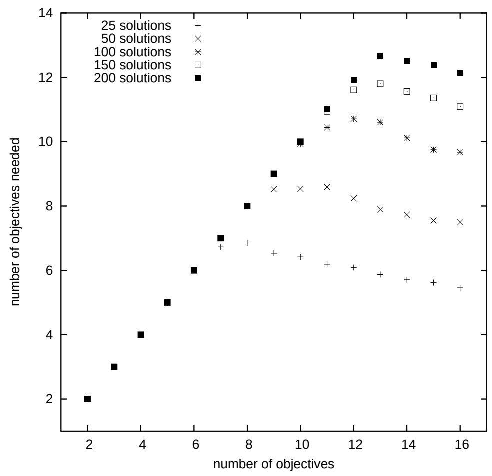

Fig. 3. Size of the computed minimum sets for different number k of randomly chosen objectives and the number $|A|$ of solutions.

$k = 25$. The other parameters are chosen according to the standard settings of the PISA package presented in [Bleuler et al., 2003]. The generated Pareto front approximations are, then, used as inputs for the 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 for the three DTLZ problems DTLZ2, DTLZ5, and DTLZ7, introduced in [Deb et al., 2005], as well as for three 0-1-knapsack instances with 100, 250 and 500 items, denoted as KP100, KP250, and KP500, cf. [Laumanns et al., 2004a].

With $\delta = 0$, the six test problems show similar results than 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. 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.

	δ-MOSS				k-EMOSS		
	0%	10%	20%	40%	30%	60%	90%
knapsack: 100 items, 5 objectives, 100 solutions	5	5	5	5	0.926	0.516	0.486
knapsack, 100 items, 15 objectives, 200 solutions	11	10	10	9	0.818	0.348	0.000
knapsack, 100 items, 25 objectives, 300 solutions	13	13	13	11	0.597	0.000	0.000
knapsack: 250 items, 5 objectives, 100 solutions	5	5	5	4	0.859	0.697	0.280
knapsack, 250 items, 15 objectives, 200 solutions	11	11	10	9	0.762	0.342	0.000
knapsack, 250 items, 25 objectives, 300 solutions	12	12	12	11	0.575	0.000	0.000
knapsack: 500 items, 5 objectives, 100 solutions	5	5	5	4	0.748	0.504	0.237
knapsack, 500 items, 15 objectives, 200 solutions	15	15	14	10	0.643	0.435	0.278
knapsack, 500 items, 25 objectives, 300 solutions	25	23	17	13	0.472	0.320	0.138
DTLZ2: 5 objectives, 100 solutions	5	5	5	5	0.991	0.970	0.920
DTLZ2: 15 objectives, 200 solutions	13	13	13	13	0.942	0.891	0.000
DTLZ2: 25 objectives, 300 solutions	18	18	18	18	0.832	0.782	0.000
DTLZ5: 5 objectives, 100 solutions	5	5	5	5	0.952	0.906	0.896
DTLZ5: 15 objectives, 200 solutions	11	11	11	11	0.860	0.803	0.000
DTLZ5: 25 objectives, 300 solutions	13	13	13	13	0.820	0.000	0.000
DTLZ7: 5 objectives, 100 solutions	5	5	1	1	0.135	0.134	0.132
DTLZ7: 15 objectives, 200 solutions	10	1	1	1	0.078	0.070	0.000
DTLZ7: 25 objectives, 300 solutions	11	1	1	1	0.050	0.000	0.000

Table 1. Sizes (for δ-MOSS) and relative errors (for k-EMOSS) of objective subsets for different problems, computed with the greedy algorithms. 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 noted relatively to the problem's number of objectives.

Comparison of the Algorithms To compare the exact Algorithm 1 with the greedy Algorithm 2 on δ-MOSS, we both use the random objective problem, described above, and the 0-1-knapsack problem the results of which are shown in Fig. 4, and Fig. 5 respectively.

For both problems, the comparison shows the same two aspects. Firstly, 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. Secondly, 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. 5.

5 Applications

In this last section, we provide examples where the above algorithms and the definition of conflict can be useful. In the case of offline analysis where a search method offers a set of non-dominated solutions, the proposed approach can not

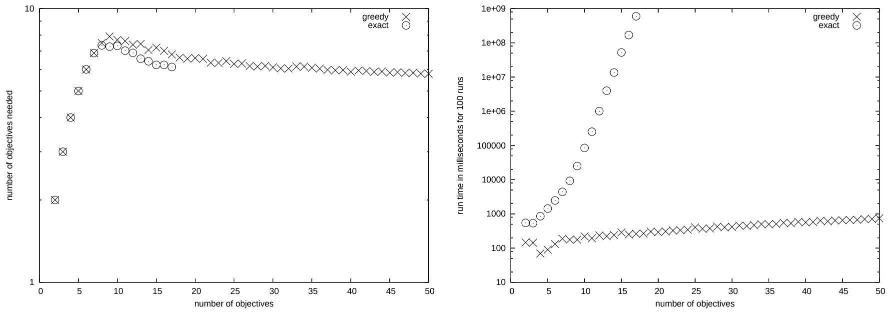

Fig. 4. Comparison between the exact and the greedy algorithm for the `MOSS` problem on sets with 32 solutions and random objective values. The left plot shows the size of the computed objective sets averaged over 100 runs for different number of objectives. In the right plot, the average running times of both algorithms are shown for 100 runs on each number of objectives.

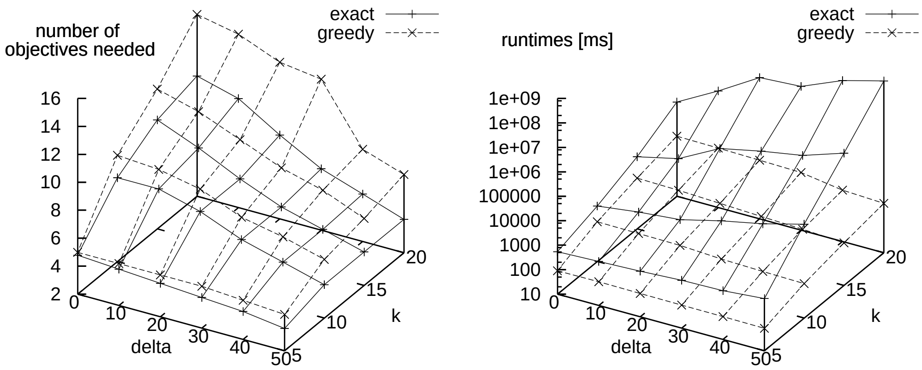

Fig. 5. Comparison of the exact and the greedy algorithm for δ-`MOSS` on the 0-1-knapsack problem.

only indicate which objectives are redundant but can also provide insights in the problem itself to make the decision making process easier. Section 5.1 will show these benefits exemplary for a radar waveform problem with nine objectives, recently proposed by [Hughes, 2007].

The general question whether objective reduction is useful during the search is subject of Sec. 5.2 where we show experimentally that additional objectives can both improve and worsen the running time behavior of a search method. How the integration of an online objective reduction can drastically improve the running time of a simple hypervolume-based search algorithm is, in addition, shown in Sec. 5.2.

5.1 Offline Objective Reduction

How the proposed approach of objective reduction can be used offline, i.e., to effectively assist in the decision making after the search, is presented in the following for a radar waveform optimization problem, recently proposed by [Hughes, 2007]. The real and unmodified engineering problem, described in [Hughes, 2007], is to design a waveform for a pulsed Doppler radar, i.e., to decide on how to choose a set of waveforms 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. For example, objectives 1 & 3, 2 & 4, 5 & 7, and 6 & 8 “tend to have a degree of correlation” because they are metrics associated with the performance in range and velocity respectively.

We received a set of more than $22,000$ non-dominated solutions for the radar waveform problem from the author. With this data, we try to confirm his statements about the relationship between the objectives quantitatively and get a deeper insight into the problem itself.

To apply the greedy algorithms for δ-`MOSS` and `k-EMOSS`, a reduction of the $> 22,000$ solutions to less solutions is necessary and performed by computing the ε-nondominated solutions out of the normalized original ones. The error ε is chosen as 0.062 yielding 107 solutions in the reduced set. The computed ε-Pareto front approximation with 107 solutions is used as input for the greedy algorithms, the results of which are presented in the following. Note, that the used error of 0.062 and the resulting solution set size of 107 is more or less arbitrary. Smaller errors, i.e., larger solution sets with up to 500 solutions yield similar results.

Table 2 shows the δ-minimal sets, computed with the greedy Algorithm 2, together with the corresponding δ-errors. It 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 objectives, that means that we make only an error of at least 6.2% when omitting the right two objectives. Nevertheless, this is not really useful at all. 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.

Algorithm 4 A second greedy algorithm for `k-EMOSS`, based on omitting objectives.

Init:
  $\mathcal{F}' := \mathcal{F}$
while $|\mathcal{F}'| > \mathtt{k}$ **do**
  $(f_r, f_s) := \underset{f_i, f_j \in \mathcal{F}'}{\operatorname{argmin}} \{\delta_{\min}(i, j) \text{ w.r.t. } X\}$
  if $\max_{\mathbf{x},\mathbf{y} \in X \wedge \mathbf{x} \preceq_{f_r} \mathbf{y}} \{f_s(\mathbf{x}) - f_s(\mathbf{y})\} < \max_{\mathbf{x},\mathbf{y} \in X \wedge \mathbf{x} \preceq_{f_s} \mathbf{y}} \{f_r(\mathbf{x}) - f_r(\mathbf{y})\}$ **then**
    $\mathcal{F}' := \mathcal{F}' \setminus \{f_s\}$
  else
    $\mathcal{F}' := \mathcal{F}' \setminus \{f_r\}$
  end if
end while

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 as stated in [Hughes, 2007] qualitatively. The approach of δ-conflict can provide such quantitative statements on objective pairs. For example, we can compute the maximum δ-error between all possible objective pairs and illustrate them as in Fig. 6. A low δ-error between an objective pair predicts that the consideration of only one of both objectives does not change the dominance relation with an error of more than δ. When illustrating the minimal δ-error between objective pairs, as in Fig. 7, additional information on the objectives can be presented; with the additional arrows, we can indicate which objective is to choose (and which to omit) to yield the minimal δ-error. 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]; Fig. 7 shows the computed δ-errors graphically.

A New Greedy Algorithm for `k-EMOSS` As the last method in support of decision making, 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. 8. Instead of greedily constructing a not δ-conflicting objective set of size `k` by adding objectives in Algorithm 3, the new Algorithm 4 removes objectives greedily until the resulting subset has `k` objectives. At each step, the objective pair f_i, f_j with the smallest δ-error is selected and the objective that minimizes the error between f_i and f_j is omitted. Each of these steps can then be visualized as an inner node in a tree, like in Fig. 8. Starting with the set of all objectives at the leafs, on the way to the root more and more objectives are omitted until a single objective—for the 107 solutions of the radar problem it is f_6—is computed.

Table 2. All δ-minimal objective sets with $\delta < 0.66$ for the 107 solutions of the radar waveform problem. For the overall error, 0.062 has to be added to the denoted values. See the text for details.

# objectives	objective set	δ-error
7	{ 2, 3, 4, 5, 6, 7, 8 }	0.0
7	{ 1, 2, 3, 4, 5, 8, 9 }	0.0
7	{ 1, 2, 3, 5, 6, 8, 9 }	0.0
7	{ 1, 2, 3, 5, 7, 8, 9 }	0.0
7	{ 2, 3, 4, 5, 7, 8, 9 }	0.0
6	{ 1, 2, 3, 5, 8, 9 }	0.137734775
6	{ 2, 3, 5, 7, 8, 9 }	0.159793814
7	{ 1, 2, 3, 5, 6, 7, 9 }	0.203106603
7	{ 1, 2, 3, 4, 5, 6, 8 }	0.208309618
7	{ 1, 3, 4, 5, 7, 8, 9 }	0.215672228
6	{ 1, 3, 5, 6, 7, 9 }	0.215672228
6	{ 3, 5, 6, 7, 8, 9 }	0.221649484
7	{ 1, 2, 3, 4, 5, 6, 9 }	0.361418472
6	{ 2, 3, 4, 5, 6, 8 }	0.408081958
6	{ 1, 2, 3, 4, 5, 6 }	0.418133084
6	{ 1, 2, 3, 5, 7, 9 }	0.43225554
5	{ 2, 3, 4, 5, 6 }	0.437106497
5	{ 3, 5, 6, 7, 9 }	0.437106497
5	{ 2, 3, 5, 7, 9 }	0.437106497
6	{ 1, 2, 3, 4, 5, 9 }	0.462720546
5	{ 1, 2, 3, 5, 9 }	0.503130336
6	{ 1, 3, 4, 5, 7, 9 }	0.506718475
6	{ 1, 2, 3, 4, 6, 7 }	0.510638306
6	{ 1, 2, 3, 4, 6, 8 }	0.510638306
6	{ 1, 2, 3, 6, 7, 9 }	0.510638306
6	{ 1, 2, 3, 6, 8, 9 }	0.510638306
6	{ 3, 4, 5, 6, 7, 8 }	0.51392599
5	{ 1, 2, 3, 4, 6 }	0.529302392
5	{ 1, 2, 3, 6, 9 }	0.529302392
6	{ 1, 3, 4, 6, 7, 9 }	0.531914881
6	{ 3, 4, 6, 7, 8, 9 }	0.531914882
6	{ 2, 3, 6, 7, 8, 9 }	0.531914882
5	{ 1, 3, 6, 7, 9 }	0.553191477
5	{ 3, 6, 7, 8, 9 }	0.553191477
5	{ 3, 4, 5, 7, 9 }	0.580335506
5	{ 3, 4, 5, 6, 7 }	0.602954888
4	{ 3, 6, 7, 9 }	0.617021266
6	{ 2, 3, 4, 7, 8, 9 }	0.639294927

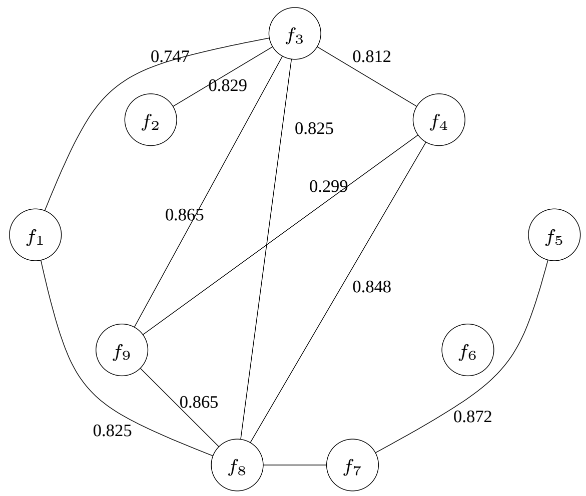

Fig. 6. Radar waveform problem: visualization of maximum delta error between objective pairs. Errors larger than 0.9 are omitted for clarity.

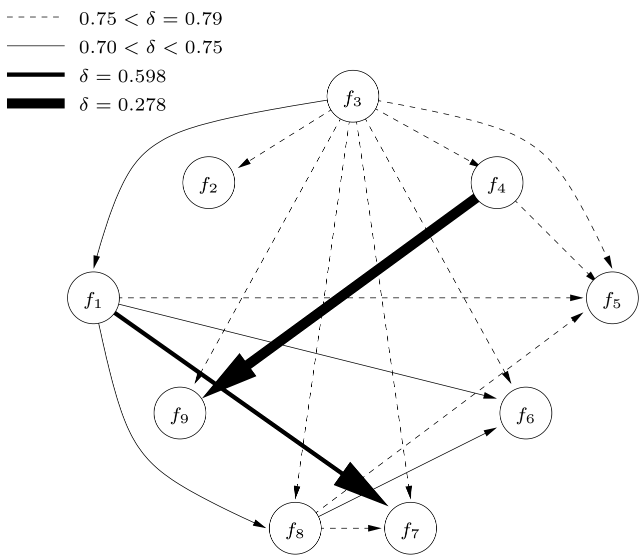

Fig. 7. Radar waveform problem: visualization of minimum delta error between objective pairs. Errors larger than 0.8 are omitted for clarity and the line width indicates the delta-error (the thicker, the smaller the error). The arrows point at the objectives which should be used.

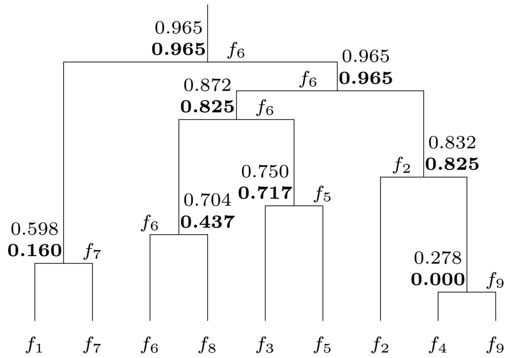

Fig. 8. Radar waveform problem: tree visualization of the greedy Algorithm 4 considering the δ-error for paired objectives. The δ-errors are written at the tree's inner nodes, in bold font the exact values.

5.2 Online Objective Reduction

Besides the advantages of objective reduction in the decision making, one may ask whether objective reduction can also improve the search itself. To answer this question, we provide—in addition to a brief review of related work—a simple test problem, based on `LOTZ`, showing that both an improvement and a worsening of the running time behavior of a simple indicator-based evolutionary algorithm can happen when different objectives are added to the test problem. Nevertheless, the main pursuit of reducing the algorithm complexity by online objective reduction can be achieved for evolutionary algorithms the running times of which highly depend on the number of objectives, as it is the case for some hypervolume-based algorithms. Section 5.2 gives an example how the running time of a hypervolume-based evolutionary algorithm can be reduced drastically by objective reduction.

General Considerations To answer the question whether the number of objectives and/or the number of incomparable solutions make a multiobjective problem difficult, we give a brief overview of recent work regarding the complexity of problems depending on the number of objectives.

The more objectives the harder the problem? Many publications, amongst others [Fonseca and Fleming, 1995], [Horn, 1997], [Deb, 2001], [Coello Coello et al., 2002], [Coello Coello, 2005], conclude that a problem becomes harder to solve if objectives are added to the problem formulation. It is often mentioned that the increasing size of the Pareto front makes the problem harder when more objectives are taken into account. This assumption is valid if the entire Pareto front is sought. In practice, this aim is often not feasible. We are, in most scenarios, interested in a good approximation of the Pareto front with a solution set of predefined size; the population size. Assuming, that we are interested in a good approximation of the Pareto front, it is not clear, in general, that additional objectives make the problem harder to approximate. A theoretical result on random linear orders can emphasize this.

[Winkler, 1985] analyzed some basic properties of (finite) random partial orders $P_k(n)$, like width and height, whereas $P_k(n)$ is the intersection of k randomly chosen linear orders on $\{1, \dots, n\}$. We, here, briefly present the most interesting results of [Winkler, 1985], translated to the field of multiobjective optimization.

The first result points out that additional arbitrary objectives will probably increase the number of Pareto optimal solutions:

Theorem 4 ([Winkler, 1985]). *The expected number $M_k(n)$ of Pareto optimal elements for a multiobjective optimization problem with search space S, $|S| = n$, and k random objectives is asymptotic to $(\ln(n))^{k-1}/(k-1)!$ for large n, i.e., $\lim_{n\to\infty} M_k(n) \cdot (k-1)!/(\ln(n))^{k-1} = 1$.*

If we are interested in finding the entire Pareto-optimal set, Theorem 4 indicates that problems become harder with more objectives, assuming that additional objectives behave like random objectives. We remark that it is not clear whether objectives in practical problems behave like random objectives. Otherwise, if we are only interested in finding a good approximation of the Pareto-optimal front, additional objectives can help, as the following theorem shows.

Theorem 5 ([Winkler, 1985]). *For a multiobjective optimization problem with search space S, $|S| = n$, and k random objectives, there is a constant c (depending only on k) with $0 < c < e$, such that for almost every random objectives a solution needs at most between $cn^{1/k}$ and $en^{1/k}$ improvements to become Pareto-optimal.*

The two presented Theorems of [Winkler, 1985] are connected to each other and can be applied to problems with non-random objectives: With constant search space, it is clear that a larger Pareto front cause a smaller average distance between a solution and the front, i.e., less improvement steps are necessary to reach the Pareto front starting from a randomly chosen solution.

That the computation of the entire front becomes harder with more objectives was shown in [Laumanns et al., 2004b] for the artificial `COUNTING ONES COUNTING ZEROS` (`COCZ`) problem. But for the `COCZ` problem, the search space size increases with larger number of objectives. In addition, the `COCZ` problem with many objectives do not emerge from the problem with less objectives by adding additional objectives, i.e., the first objective of the three-objective `COCZ` problem is not the same than the first objective of the bi-objective `COCZ` problem and so forth.

To sum up, there are no hints saying that adding objectives to a problem makes it, in general, harder to solve. In the following section we give a brief review of hints indicating that additional objectives can simplify problems.

The more objectives the easier the problem? Some theoretical results on the runtime of simple evolutionary algorithms are known which show that additional objectives can make the problem easier, i.e., it is easier to solve for a simple

EA if more objectives are taken into account. One of these problems is the MINIMUM SPANNING TREE problem. [Neumann and Wegener, 2006] showed that the global SEMO algorithm, as the multiobjective version of the (1+1) EA, solves the bi-objective formulation of the problem asymptotically in less time than the (1+1) EA needs for the single-objective formulation. The bi-objective problem formulation is derived from the single-objective one by splitting the two terms of the single-objective formulation into different objectives. It is, therefore, not a correct hint that the addition of objectives can simplify a problem, but it shows that more objectives are not always a problem.

Another problem, where the change from a single-objective problem to a multiobjective formulation can improve the runtime of a simple EA drastically is the problem SINGLE SOURCE SHORTEST PATH (SSSP) as it was proved in [Scharnow et al., 2004]. But also here, the multiobjective problem is derived from the original single-objective one by decomposition of the single objective, like in [Neumann and Wegener, 2006], instead of adding additional objectives.

The above mentioned approach of using multiobjective formulations instead of optimizing single-objective problems is known, in general, as *multi-objectivization*, introduced by [Knowles et al., 2001]. Knowles et al. distinguish between decomposition of the single objective and the addition of objectives but focus only on decomposition examples. [Jensen, 2004] proposed a generalization of the idea of additional objectives, called *helper-objectives*. Jensen's helper-objectives are constructed during the search to fasten the computation of good solutions by escaping local optima. In the end of his paper, Jensen mentions that his experiments "indicate that when many helper-objectives are used simultaneously, the disadvantage of the bad moves outweighs the advantage of escaping local optima". Nevertheless, both [Knowles et al., 2001] and [Jensen, 2004] showed with experimental studies that (some) problems can be solved faster or with a better quality of the Pareto front approximation if adequate additional objectives are considered.

Generalization is not possible The various examples, mentioned above, do not show a general behavior if additional objectives are used in multiobjective optimization. Whether a problem becomes harder or easier to solve with more objectives highly depends on the problem and the form of the additional objectives. The addition of only a few objectives can, if the objectives are designed well, make a problem easier. But wrongly chosen objectives can also increase the difficulty of a problem. That the form of the additional objective really has an influence on the running time, was recently shown by [Brockhoff et al., 2007] by proving rigorous running time bounds for the SEMO algorithm on a simple PLATEAU problem. Starting from a single-objective problem, the PLATEAU problem gets either harder or easier, depending which objective function is added as the second objective.

To get a better insight how an additional objective can influence a problems complexity, we exemplify the possible changes when adding a new objective to a simple bi-objective test problem in the next section.

Additional Objectives: Experimental Study As we have seen, there is no general answer to the question whether a problem becomes harder with more objectives. With the experimental study in this section we, furthermore, reveal that the number of incomparable solutions can also not indicate whether a problem becomes harder or easier with additional objectives.

In general, the addition of objective functions can only change the dominance structure in two ways as we learned in Sec. 3. Either indifferent solutions become comparable or comparable solutions become incomparable. Here, we propose four problem formulations where the problem becomes harder (P_1,P_3) or easier (P_2,P_4) by adding an additional third objective function and therefore making indifferent solutions comparable (P_1,P_2) or comparable solutions incomparable (P_3,P_4). The problems P_1 and P_2 are based on the two-objective `LOTZ` problem, defined in [Laumanns et al., 2004b], whereas the `LOTZ` problem is modified for the problems P_3 and P_4. In the following definitions, let $|\mathbf{x}|$ the length of bitstring $\mathbf{x}$.

Definition 12 *Let $X = \{0,1\}^n$. We define the following objective functions* `LEADING ONES (LO)` *and* `TRAILING ZEROS (TZ)` *on a bitstring $\mathbf{x} := (x_1, \ldots, x_n)$ as defined in [Laumanns et al., 2004a].*

$$\mathtt{LO}(\mathbf{x}) := \sum_{i=1}^{n} \prod_{j=1}^{i} x_j \qquad \mathtt{TZ}(\mathbf{x}) := \sum_{i=1}^{n} \prod_{j=i}^{n} (1 - x_j)$$

In addition, we define the middle block $\mathbf{x}_M$ of a bitstring $\mathbf{x} := (x_1, \ldots, x_n)$ as the substring $x_{\mathtt{LO}(\mathbf{x})+1}, \ldots, x_{n-\mathtt{TZ}(\mathbf{x})}$ of size $n - \mathtt{LO}(\mathbf{x}) - \mathtt{TZ}(\mathbf{x})$ which is simply $\mathbf{x}$ excluding its leading ones and the trailing zeros.

We also define the two modified `LO` *and* `TZ` *functions* `MoLO` *and* `MoTZ`*. They emerge from the* `LOTZ` *problem by grouping 2* `LOTZ` *levels together and reflecting every other 2-party layer at the origin. Figure 9 illustrates these objective functions.*

$$\mathtt{MoLO}(\mathbf{x}) := \begin{cases} \mathtt{LO}(\mathbf{x}) & \text{iff } \; 0 \equiv (n - |\mathbf{x}_M|) \bmod 4 \\ & \text{or } \; 1 \equiv (n - |\mathbf{x}_M|) \bmod 4 \\ (-n) \cdot \left\lfloor \frac{\left\lfloor \frac{n-|\mathbf{x}_M|}{2} \right\rfloor}{2} \right\rfloor - \mathtt{LO}(\mathbf{x}) & \text{else} \end{cases}$$

$$\mathtt{MoTZ}(\mathbf{x}) := \begin{cases} \mathtt{TZ}(\mathbf{x}) & \text{iff } \; 0 \equiv (n - |\mathbf{x}_M|) \bmod 4 \\ & \text{or } \; 1 \equiv (n - |\mathbf{x}_M|) \bmod 4 \\ (-n) \cdot \left\lfloor \frac{\left\lfloor \frac{n-|\mathbf{x}_M|}{2} \right\rfloor}{2} \right\rfloor - \mathtt{TZ}(\mathbf{x}) & \text{else} \end{cases}$$

Definition 13 *According to [Laumanns et al., 2004b], the pseudo-Boolean function* `LOTZ`*: $\{0,1\}^n \to \mathbb{N}^2$ is defined as*

$$\mathtt{LOTZ}(x_1, \ldots, x_n) := (\mathtt{LO}, \mathtt{TZ}).$$

The modified `LOTZ` *function* `moLOTZ`*: $\{0,1\}^n \to \mathbb{N}^2$ is defined as*

$$\mathtt{moLOTZ}(x_1, \ldots, x_n) := (\mathtt{MoLO}, \mathtt{MoTZ}).$$

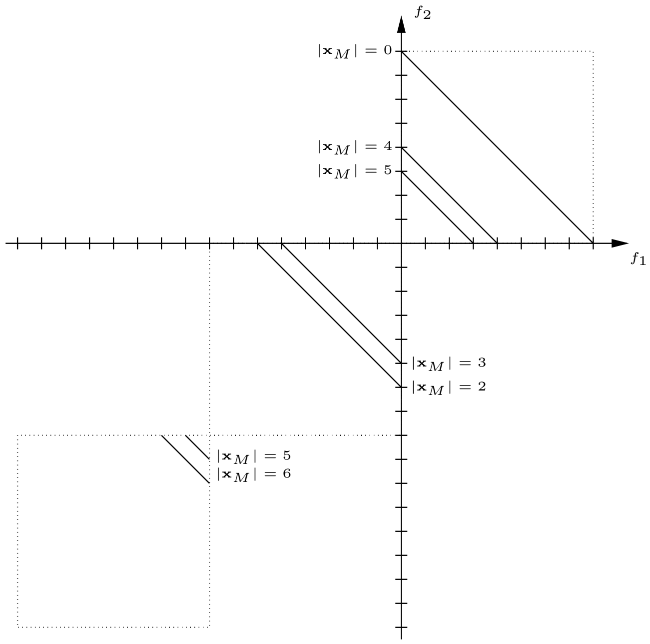

Fig. 9. Illustration of the objective space for the modified LOTZ.

Definition 14 *We define the following four three-objective problems, based on the LOTZ and the modified LOTZ problem respectively, where the first two objective functions have to be maximized and the third one has to be minimized:*

(i) The problem P_1 is to maximize LOTZ and in addition, to minimize the additional third objective

$$f_{(i)}(\mathbf{x}) = |\mathbf{x}_M| - \mathit{LZ}(\mathbf{x}_M) - \mathit{TO}(\mathbf{x}_M).$$

where the functions LZ (LEADING ZEROS) and TO (TRAILING ONES) are defined as

$$\mathit{LZ}(\mathbf{x}) := \sum_{i=1}^{n} \prod_{j=1}^{i} (1 - x_j) \qquad \mathit{TO}(\mathbf{x}) := \sum_{i=1}^{n} \prod_{j=i}^{n} x_j$$

(ii) The problem P_2 is defined as the LOTZ problem with additional third objective

$$f_{(ii)}(\mathbf{x}) = \mathit{ONES}(\mathbf{x}_M),$$

where $\mathit{ONES}((x_1, \ldots, x_n)) := \sum_{i=0}^{n} x_i$.

(iii) We define the problem P_3 as the moLOTZ problem with additional objective

$$f_{(iii)}(\mathbf{x}) = \frac{n}{2} - |\frac{n}{2} - |\mathbf{x}_M||.$$

(iv) Problem P_4 is defined as the `moLOTZ` *problem with additional objective*

$$f_{(iv)}(\mathbf{x}) := |\mathbf{x}_M|.$$

Note, that all four problems have the same Pareto-optimal front $\{\mathbf{x} = 1^i 0^{n-i} \mid 0 \le i \le n\}$ as `LOTZ` *and* `moLOTZ`*, since the Pareto-optimal points of* `LOTZ` *and* `moLOTZ` *minimize each of the additional objective functions.*

In the `LOTZ` problem, all bitstrings $\mathbf{x} = 1^i 0 \dots 1 0^j$ for fixed i, j are mapped to the same objective vector; they are indifferent. Since $f_{(i)}$ and $f_{(ii)}$ are only dependent on the middle block, the additional objective function can only make indifferent solutions comparable. When using $f_{(i)}$ as the third objective, the additional comparability between search points let an evolutionary algorithm, like SEMO (defined by [Laumanns et al., 2004b]), prefer solutions the middle block of which starts with many zeros and ends with many ones. This makes the way to the Pareto-optimal front harder, because almost every bit in the middle block has to flip successively. In the other case, when using $f_{(ii)}$ as third objective, search points with less ones in the middle block are preferred which cause a lower running time of evolutionary algorithms like SEMO compared to the two-objective problem because variation steps improving the objectives will benefit from larger blocks of zeros, see Fig. 12 for an example. Therefore, the `LOTZ` problem will become harder, when optimizing with the additional objective $f_{(i)}$, and easier if the objective $f_{(ii)}$ is added. Note, that the statements on the difficulty of the problems only refer to the time until a Pareto-optimal search point is found and, in principle, only for mutation based algorithms like SEMO. Regarding the complexity of finding the entire Pareto-optimal front, the problems' complexity for P_1 and P_2 are asymptotically equal for simple algorithms like SEMO[8].

Similar thoughts hold for the problems P_3 and P_4, whereas solutions, which are comparable with `moLOTZ`, become incomparable in P_3 and P_4. Due to the fact, that an additional objective $f_{(iii)}$ makes solutions with $|\mathbf{x}_M| = 4i$ and $|\mathbf{x}_M| = 4i+1$, $0 \le i \le \lfloor \frac{n}{4} \rfloor - 1$ incomparable, the emerging large plateaus of incomparable solutions makes the problem P_3 harder to solve for evolutionary algorithms than the bi-objective `moLOTZ`. On the other hand, the `moLOTZ` problem becomes easier if the objective $f_{(iv)}$ is added because the additional objective cause solutions within the levels with $|\mathbf{x}_M| = 4i+1$ and $|\mathbf{x}_M| = 4i+2$, $0 \le i \le \lfloor \frac{n}{4} \rfloor - 1$ to become incomparable. As the comparability between a solution in level $|\mathbf{x}_M| = 4i + 1$ and a solution in level $|\mathbf{x}_M| = 4i+2$ within the `moLOTZ` problem is misleading for an evolutionary algorithm, the algorithm, if running on P_4, has the opportunity to search on a plateau of incomparable solutions instead of getting stuck in a local minimum in the search space.

[8] We omit the proofs for the expected running time of local SEMO, since we are not interested in asymptotic running times for a theoretical and specific algorithm like local SEMO, but in experiments showing that adding an objective can make a problem either harder or easier for a simple EA like IBEA. The proofs can be done based on techniques of [Droste et al., 2002] and [Laumanns et al., 2004b]. Note, that the running time analysis for global SEMO on the mentioned problems is not trivial and remains future work.

Instead of proving asymptotical runtime bounds for the impractical[9] SEMO algorithm, we use simulation runs of the IBEA algorithm to show that additional objectives can either increase or decrease the time until a Pareto-optimal search point is found. To this end, 10 IBEA runs with a population size of 200 are performed on each of the six problems `LOTZ`, `moLOTZ`, P_1, P_2, P_3, and P_4 for various bitstring lengths. We use the standard settings from the PISA package of [Bleuler et al., 2003] and compute the number of generations needed to reach the Pareto-optimal front, i.e., the time until a first Pareto-optimal search point is found. The figures Fig. 10 and Fig. 11 show the results for various bit string lengths.

As expected, for both `LOTZ` and `moLOTZ`, an additional objective can improve or worsen the performance of IBEA. IBEA needs for problem P_1 more time than for the bi-objective `LOTZ` problem because the third objective $f_{(i)}$ leads the evolutionary algorithm away from the Pareto-optimal front, whereas with the addition of $f_{(ii)}$, IBEA finds a Pareto-optimal point faster than for `LOTZ`, because the additional objective give hints in direction to the Pareto-optimal front. In comparison to the `moLOTZ` problem, the addition of $f_{(iii)}$ cause a greater running time for IBEA, because solution pairs become incomparable the corresponding dominance relation of which gives the correct direction to the Pareto front in `moLOTZ`. Therefore, large plateaus have to be exploited if $f_{(iii)}$ is added. With the addition of objective function $f_{(iv)}$, the `moLOTZ` problem becomes easier for IBEA because the misleading dominance relation between solution pairs with $|\mathbf{x}_M| = 4i+1$ and $|\mathbf{x}_M| = 4i+2$, $0 \leq i \leq \lfloor \frac{n}{4} \rfloor - 1$, is changed to incomparability.

Objective Reduction During Hypervolume-Based Search The main focus of the preliminary study, we present in this section, is to show that an objective reduction can be useful if the running time of an evolutionary algorithm highly depends on the number of considered objectives. Since the best known algorithm for computing the hypervolume indicator exactly, proposed by [Beume and Rudolph, 2006], is exponential in the number of objectives, evolutionary algorithms which are based on this indicator, such as SMS-EMOA of [Emmerich et al., 2005] or SIBEA by [Zitzler et al., 2007], can benefit from an objective reduction during search.

The following preliminary study shows as a proof-of-principle that a simple indicator based evolutionary algorithm together with an online objective reduction according to the greedy algorithm from Sec. 4.4 outperforms the algorithm without any objective reduction.

The study is based on a simple indicator based evolutionary algorithm, namely SIBEA from [Zitzler et al., 2007], enhanced with two different objective reduction strategies and a slightly modified version of the DTLZ2 test function, proposed by [Deb et al., 2005]. The following paragraphs explain the main issues of both the used algorithms and the used test function.

[9] In general, the population size of the SEMO algorithm is not bounded by a constant, i.e., the population can become arbitrary large.

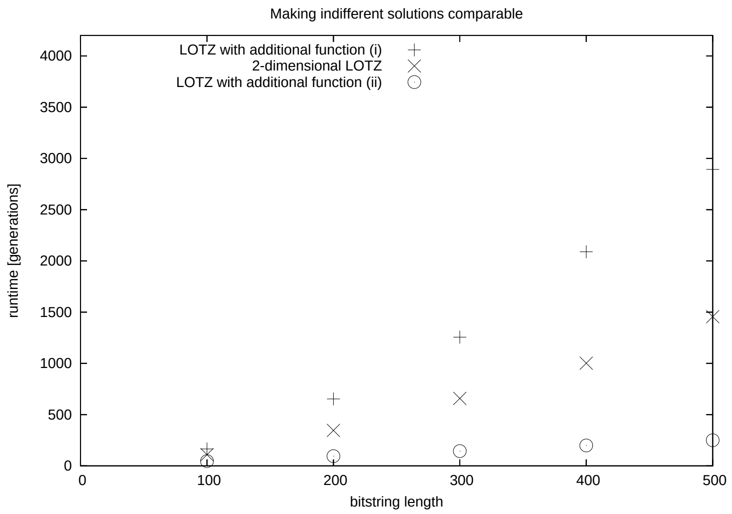

Fig. 10. If an additional objective makes indifferent solutions comparable the resulting problem can be either harder or easier than the original one. The plot shows the number of generations the algorithm IBEA needs to reach a Pareto-optimal point on the `LOTZ` problem itself ($\times$), on the `LOTZ` problem with additional objective $f_{(i)}$ (+), and on the `LOTZ` problem with additional objective $f_{(ii)}$ ($\circ$) averaged over 10 runs for each of the five bitstring lengths.

The Algorithms $SIBEA_{no}$, $SIBEA_{random}$, and $SIBEA_{online}$ Algorithm 5 shows the organization of the original SIBEA, extended with a general objective reduction functionality. SIBEA starts with randomly choosing a set of μ solutions, the population P. Until a certain time limit T is reached, the μ solutions of the current population P are randomly selected for recombination and mutation, the variated solutions are inserted into the population, and the population of the next generation is determined by successively removing the individuals with the worst hypervolume losses. The hypervolume loss $d(\mathbf{x})$ of a solution $\mathbf{x}$ is defined as the difference between the hypervolume of the population P and the hypervolume of P without $\mathbf{x}$:

$$d(\mathbf{x}) := I_H(P) - I_H(P \setminus \{\mathbf{x}\}).$$

In addition, SIBEA applies two different objective reduction strategies to improve the running time of the hypervolume computation. To this end, every G generations it is decided which objectives are chosen for optimization and which ones are neglected during the next G generations. In the following, we distinguish between three versions of SIBEA: SIBEA_{no}, $\text{SIBEA}_{\text{random}}$ and $\text{SIBEA}_{\text{online}}$. The SIBEA algorithm from [Zitzler et al., 2007] with “no” objective reduction is referred to as SIBEA_{no}. $\text{SIBEA}_{\text{random}}$ chooses the `r` considered objectives every G generations randomly, where the number `r` of considered objectives is given in advance. The algorithm $\text{SIBEA}_{\text{online}}$ performs an objective reduction by apply-

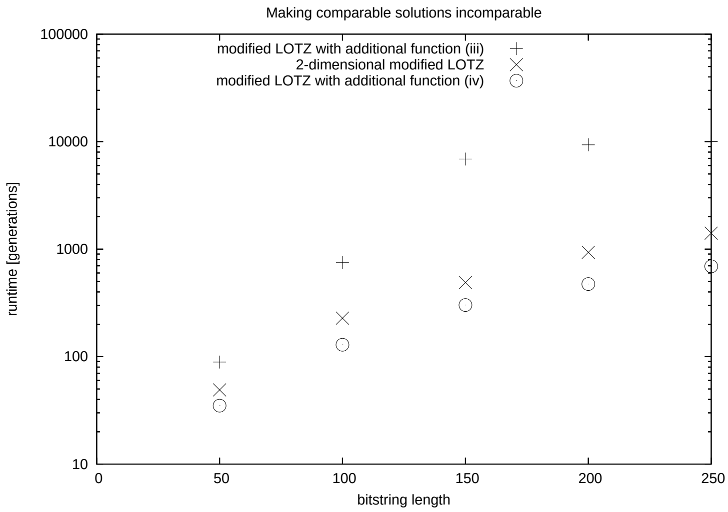

Fig. 11. If an additional objective makes comparable solutions incomparable the resulting problem can be either harder or easier than the original one. The plot shows the number of generations the algorithm IBEA needs to reach a Pareto-optimal point on the modified `LOTZ` problem ($\times$), the modified `LOTZ` problem with additional objective $f_{(iii)}$ ($+$), and the modified `LOTZ` problem with additional objective $f_{(iv)}$ ($\circ$) averaged over 10 runs for each of the five bitstring lengths. Note, that an IBEA run is aborted when no Pareto-optimal solution is found within 10000 generations.

$$\mathbf{x}_{P_1} = 11\underline{0001111}0 \quad \Rightarrow \quad \mathbf{x}'_{P_1} = 11\underline{000111}00$$
$$\mathbf{x}_{P_2} = 11\underline{0100001}0 \quad \Rightarrow \quad \mathbf{x}'_{P_2} = 11\underline{01}000000$$

Fig. 12. An example of the influence of $f_{(i)}$ and $f_{(ii)}$ on the objective function improvement for a certain mutation. The middle block is always underlined. Because an evolutionary algorithm, optimizing P_1, prefers solutions with starting ones and leading zeros in the middle block as $\mathbf{x}_{P_1}$, a mutation of the second last bit to $\mathbf{x}'_{P_1}$ will cause an improvement of one in the second and third objective function. Since an evolutionary algorithm running on P_2 prefers solutions with less ones in the middle block like in $\mathbf{x}_{P_2}$, a bit flip of the second last bit to $\mathbf{x}'_{P_2}$ will cause an improvement of 5 in the second and third objective.

ing the greedy algorithm for `k-EMOSS` on the current population to compute the objectives which are considered in the next G generations.

Algorithm 5 Simple Indicator Based Evolutionary Algorithm (**SIBEA**)

Input: population size μ; running time T in seconds; indicator function I; reduction frequency G in generations; size $\mathtt{r}$ of reduced objective set;
Output: approximation of Pareto-optimal set A;

Step 1 (Initialization):
Generate an initial set of decision vectors P of size μ; set the current time t_0; set generation counter $m := 0$.
Step 2 (Dimensionality reduction):
If $m \equiv 0 \bmod G$: Either do nothing (SIBEA$_{\text{no}}$), choose $\mathtt{r}$ objectives randomly (SIBEA$_{\text{random}}$), or use the greedy `k-EMOSS` algorithm to compute an objective subset of size $\mathtt{r}$ according to all solutions in P (SIBEA$_{\text{online}}$). In the following G generations, use only the chosen objectives for hypervolume computation.
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 [Deb et al., 2000] is used.
2. For each solution $\boldsymbol{x} \in P'$ determine the loss $d(\boldsymbol{x})$ w.r.t. the hypervolume indicator I_H if it is removed from P', i.e., $d(\boldsymbol{x}) := I_H(P') - I_H(P' \setminus \{x\})$.
3. Remove the solution with the smallest loss $d(\boldsymbol{x})$ 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 variation operators such as recombination and mutation to the mating pool Q which yields Q'. Set $P := P + Q'$ (multi-set union) and continue with Step 2.

DTLZ2 and DTLZ2$_{BZ}$ Test Functions As test function, we use a slightly modified version of the DTLZ2 function, known from [Deb et al., 2005]. Due to two main properties of the original DTLZ2 function, the original DTLZ2 function is slightly modified towards the used DTLZ2$_{BZ}$ function:

– On the one hand, the original DTLZ2 function proposed in [Deb et al., 2005] has the property that the projection of the Pareto front to $k < M$ objectives collapses to one optimal point, i.e., when omitting arbitrary objectives, the search always converge to one solution. Every multiobjective function has this property if all objectives except one are omitted, i.e., optimizing a single-objective problem. For the DTLZ function suite, however, this property even holds for every subset of objectives. To eliminate this property, we limit the range of the variables, i.e., we cut corners of the Pareto-optimal front, cf. Fig. 13.

– On the other hand, when optimizing only a subset of $k < M$ objectives, the neglected objectives are also optimized at the same time. The reason is the scaling of all objectives by a function $g(\mathbf{x}_M)$, indicating the distance to the real Pareto front. To come up with a problem where all single objectives have to be optimized simultaneously to reach the Pareto front, we introduce different scaling functions $g_i(\mathbf{x})$, instead of one single scaling function $g(\mathbf{x}_M)$. Figure 14 shows the formal definition of the original DTLZ2 in comparison to the new function DTLZ2$_{BZ}$.

In addition, we differentiate in the experiments between two versions of the DTLZ2$_{BZ}$ function, one unscaled as described in Fig. 14 and one scaled version DTLZ2$^*_{BZ}$ where the objective value $f_i(\mathbf{x})$ $(1 \leq i \leq k)$ is scaled to

$$f_i^*(\mathbf{x}) = \text{maxValue} \cdot \left(\frac{f_i(\mathbf{x})}{\text{maxValue}}\right)^i$$

if i is even, and to

$$f_i^*(\mathbf{x}) = \text{maxValue} \cdot \left(\frac{f_i(\mathbf{x})}{\text{maxValue}}\right)^{1/i}$$

if i is odd, where maxValue $= 1 + (n - M + 1)/4$ is an upper bound for the original $f_i(\mathbf{x})$ values.

Experimental Settings To compare the three versions of SIBEA experimentally, we perform 11 runs for each combination of algorithm, problem, and chosen objective set size `r`. As algorithms we use the three proposed SIBEA versions. As problems, we use the above defined DTLZ2$_{BZ}$ test problem with 3, 5, and 9 objectives, where the chosen objective set sizes `r` depend on the number k of all objectives. The 11 runs are performed for both the unscaled and the scaled version of DTLZ2$_{BZ}$ and the following combinations of k/`r`: 3/2, 5/2, 5/3, 9/2, 9/3, 9/4. We fix the computation time to $T = 300$ seconds and use a population size of $\mu = 50$, just as an objective reduction frequency of $G = 50$.

The populations of the three algorithms after the given time T are compared by computing their hypervolume indicator. The nonparametric Mann-Whitney U test[10] is used to confirm the hypothesis that one random variable, e.g., the hypervolume indicator of algorithm $\mathcal{A}$, "systematically" produces larger values than another one, e.g., the hypervolume indicator of algorithm $\mathcal{B}$ by ranking all values and comparing the rank sums for both samples.

Results The statistical tests, the results of which are shown in Table 3, confirm in most cases the hypothesis derived from preliminary experiments. For all considered problem sizes, the random version SIBEA$_{\text{random}}$ yields the worst results

[10] As implemented in the statistical package SPSS, version 15.0, cf. `www.spss.com`.

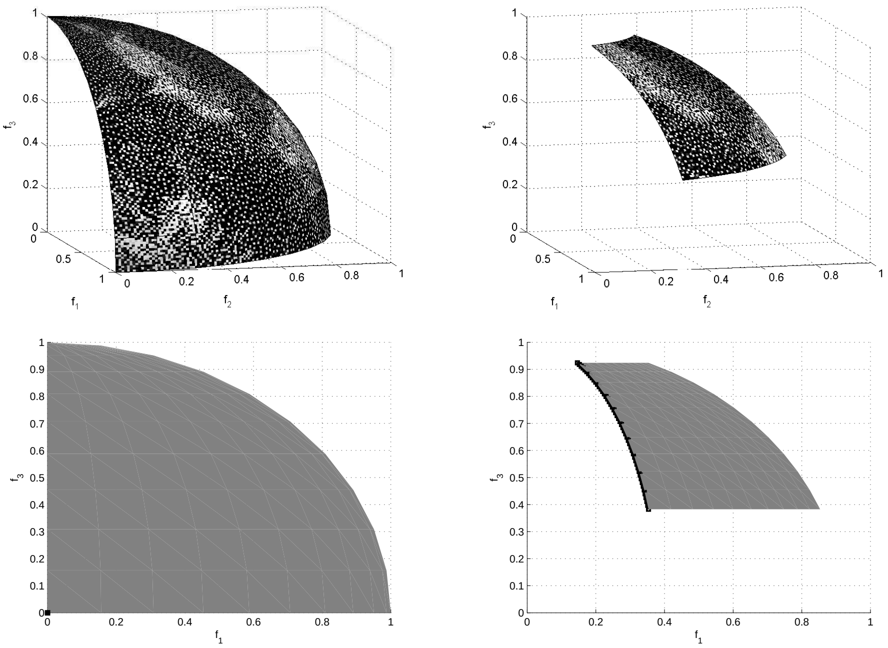

Fig. 13. Visual comparison between the Pareto-optimal fronts of original DTLZ2 (left) and modified DTLZ2$_{BZ}$ (right). The first row shows the Pareto-optimal fronts for the three-dimensional problems, whereas the second row shows the same fronts projected to the f_1/f_3 plane: if objective f_2 is omitted 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$_{BZ}$ (right). Note that the surfaces of the Pareto fronts are textured for illustration purpose.

$$
\begin{array}{ll}
\text{Min} & f_1(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos(\theta_1) \cos(\theta_2) \cdots \cos(\theta_{M-2}) \cos(\theta_{M-1}), \\
\text{Min} & f_2(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos(\theta_1) \cos(\theta_2) \cdots \cos(\theta_{M-2}) \sin(\theta_{M-1}), \\
\text{Min} & f_3(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos(\theta_1) \cos(\theta_2) \cdots \sin(\theta_{M-2}), \\
\vdots & \vdots \\
\text{Min} & f_{M-1}(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos(\theta_1) \sin(\theta_2), \\
\text{Min} & f_M(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \sin(\theta_1), \\
\text{where} & g(\mathbf{x}_M) = \sum_{x_i \in \mathbf{x}_M} (x_i - 0.5)^2, \\
& \theta_i = \frac{\pi}{2} \cdot x_i \text{for } i = 1, \ldots, M-1 \\
& 0 \leq x_i \leq 1, \quad \text{for } i = 1, 2, \ldots, n.
\end{array}
$$

(a) original DTLZ2 function of [Deb et al., 2005]

$$
\begin{array}{ll}
\text{Min} & f_1(\mathbf{x}) = (1 + g_1(\mathbf{x}_M)) \cos(\theta_1) \cos(\theta_2) \cdots \cos(\theta_{M-2}) \cos(\theta_{M-1}), \\
\text{Min} & f_2(\mathbf{x}) = (1 + g_2(\mathbf{x}_M)) \cos(\theta_1) \cos(\theta_2) \cdots \cos(\theta_{M-2}) \sin(\theta_{M-1}), \\
\text{Min} & f_3(\mathbf{x}) = (1 + g_3(\mathbf{x}_M)) \cos(\theta_1) \cos(\theta_2) \cdots \sin(\theta_{M-2}), \\
\vdots & \vdots \\
\text{Min} & f_{M-1}(\mathbf{x}) = (1 + g_{M-1}(\mathbf{x}_M)) \cos(\theta_1) \sin(\theta_2), \\
\text{Min} & f_M(\mathbf{x}) = (1 + g_M(\mathbf{x}_M)) \sin(\theta_1), \\
\text{where} & g_i(\mathbf{x}_M) = \sum_{j=M+(i-1)\cdot\lfloor\frac{n-M+1}{M}\rfloor}^{M+i\cdot\lfloor\frac{n-M+1}{M}\rfloor-1} \left(\left(\frac{x_i}{2} + \frac{1}{4}\right) - 0.5\right)^2 \text{for } i = 1, \ldots, M-1, \\
& g_M(\mathbf{x}_M) = \sum_{j=M+(M-1)\cdot\lfloor\frac{n-M+1}{M}\rfloor}^{n} \left(\left(\frac{x_i}{2} + \frac{1}{4}\right) - 0.5\right)^2, \\
& \theta_i = \frac{\pi}{2} \cdot \left(\frac{x_i}{2} + \frac{1}{4}\right) \text{for } i = 1, \ldots, M-1 \\
& 0 \leq x_i \leq 1, \quad \text{for } i = 1, 2, \ldots, n.
\end{array}
$$

(b) new DTLZ2$_{BZ}$ function

Fig. 14. The original DTLZ2 function (a) in comparison to the modified DTLZ2$_{BZ}$ (b).

on the $DTLZ2_{BZ}$ problem (all tests are significant or even highly significant, except for the scaled $DTLZ2^*_{BZ}$ problem with $k = 5$ objectives and $\mathtt{r} = 3$). On the scaled $DTLZ2^*_{BZ}$ problems, $SIBEA_{online}$ performs best (all tests significant or highly significant), whereas on the unscaled problems, $SIBEA_{no}$ beats $SIBEA_{online}$ except for the 9-objective problem and $\mathtt{r} = 3$ and $\mathtt{r} = 4$.

Two main statements can be derived from the presented comparison. Firstly, the integration of online reduction methods into a hypervolume-based evolutionary algorithm makes sense and can improve the quality of the computed Pareto front approximations. Secondly, the used objective reduction strategy highly influences the outcome of the evolutionary algorithm. The usage of an advanced objective reduction technique as the δ-conflict based one, presented in this paper, is preferred to a random choice of the objectives to optimize.

6 Conclusion

With many objectives in a multiobjective optimization problem, the question arises whether the omission of objectives can help both in the decision making (in terms of visualizing high-dimensional data and the unmanageable, large amount of data) and the search with evolutionary algorithms (in terms of computation time spent for the hypervolume indicator or to search in large plateaus of incomparable solutions). In this work, we investigated the effect of adding or omitting objectives on the Pareto dominance relation and proposed an objective reduction technique based on objective conflicts which finds objective sets of minimum size ensuring that the Pareto dominance relation is preserved or only slightly changed with a certain, predefined error. To demonstrate how the proposed objective reduction method can be utilized in an offline scenario, where the omission of objectives assist in the decision making process, we analyzed the application of designing a radar waveform in terms of conflicting objectives and minimum objective sets. That the proposed objective reduction technique is also useful within an online scenario, i.e., within an evolutionary algorithm, is shown empirically for a Pareto-dominance based algorithm and in the special case of a simple hypervolume based algorithm.

References

[Agrell, 1997] Agrell, P. J. (1997). On redundancy in multi criteria decision making. *European Journal of Operational Research*, 98(3):571–586.

[Beume and Rudolph, 2006] 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. shorter version published at IASTED International Conference on Computational Intelligence (CI 2006).

[Bleuler et al., 2001] 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)*, pages 536–543, Piscataway, NJ. IEEE.

Table 3. Results of Mann-Whitney U tests on the hypervolume indicators for different $k/\mathtt{k}$ combinations. The hypervolume indicator values are denoted by $I_{\text{H, no}}$ for SIBEA_{no}, $I_{\text{H, random}}$ for $\text{SIBEA}_{\text{random}}$, and $I_{\text{H, online}}$ for $\text{SIBEA}_{\text{online}}$. The tests were performed with the SPSS 15.0 software.

*Note, that usually 11 runs are performed and therefore the considered samples are of size 11. Due to the long running time for 9 objectives, this sample size differs. For SIBEA_{no}, all generations within the first hour of each run are considered. For $\text{SIBEA}_{\text{random}}$, $k = 9$, $\mathtt{k} = 4$, and unscaled DTLZ2_{BZ}, the populations after 300 seconds of only 6 runs are considered.

k / $\mathtt{k}$	DTLZ2 version	ranking	outcome of Mann-Whitney U test
3 / 2	unscaled	$I_{\text{H, no}} > I_{\text{H, online}} > I_{\text{H, random}}$	all tests highly significant ($p < 0.001$)
3 / 2	scaled	$I_{\text{H, online}} > I_{\text{H, no}} > I_{\text{H, random}}$	$I_{\text{H, online}} > I_{\text{H, no}}$: $p = 0.008$ $I_{\text{H, online}} > I_{\text{H, random}}$: $p < 0.001$ $I_{\text{H, no}} > I_{\text{H, random}}$: $p < 0.001$
5 / 2	unscaled	$I_{\text{H, no}} > I_{\text{H, online}} > I_{\text{H, random}}$	all tests highly significant ($p < 0.001$)
5 / 2	scaled	$I_{\text{H, online}} > I_{\text{H, no}} > I_{\text{H, random}}$	all tests highly significant ($p < 0.001$)
5 / 3	unscaled	$I_{\text{H, no}} > I_{\text{H, online}} > I_{\text{H, random}}$	$I_{\text{H, no}} > I_{\text{H, online}}$: $p = 0.033$ $I_{\text{H, no}} > I_{\text{H, random}}$: $p < 0.001$ $I_{\text{H, online}} > I_{\text{H, random}}$: $p = 0.012$
5 / 3	scaled	$I_{\text{H, online}} > I_{\text{H, no}} > I_{\text{H, random}}$	$I_{\text{H, online}} > I_{\text{H, no}}$: $p < 0.001$ $I_{\text{H, online}} > I_{\text{H, random}}$: $p < 0.001$ $I_{\text{H, no}} > I_{\text{H, random}}$: $p < 0.108$
9 / 2	unscaled	$I_{\text{H, no}} > I_{\text{H, online}} > I_{\text{H, random}}$	all tests highly significant ($p < 0.001$) Note, that #samples for $I_{\text{H, no}}$ is 14*
9 / 2	scaled	$I_{\text{H, online}} > I_{\text{H, no}} > I_{\text{H, random}}$	$I_{\text{H, online}} > I_{\text{H, no}}$: $p = 0.003$ $I_{\text{H, online}} > I_{\text{H, random}}$: $p < 0.001$ $I_{\text{H, no}} > I_{\text{H, random}}$: $p < 0.001$ Note, that #samples for $I_{\text{H, no}}$ is 12*
9 / 3	unscaled	$I_{\text{H, online}} > I_{\text{H, no}} > I_{\text{H, random}}$	$I_{\text{H, online}} > I_{\text{H, no}}$: $p = 0.101$ $I_{\text{H, online}} > I_{\text{H, random}}$: $p < 0.001$ $I_{\text{H, no}} > I_{\text{H, random}}$: $p < 0.001$ Note, that #samples for $I_{\text{H, no}}$ is 14*
9 / 3	scaled	$I_{\text{H, online}} > I_{\text{H, no}} > I_{\text{H, random}}$	$I_{\text{H, online}} > I_{\text{H, no}}$: $p < 0.001$ $I_{\text{H, online}} > I_{\text{H, random}}$: $p < 0.001$ $I_{\text{H, no}} > I_{\text{H, random}}$: $p = 0.006$ Note, that #samples for $I_{\text{H, no}}$ is 12*
9 / 4	unscaled	$I_{\text{H, online}} > I_{\text{H, no}} > I_{\text{H, random}}$	$I_{\text{H, online}} > I_{\text{H, no}}$: $p < 0.001$ $I_{\text{H, online}} > I_{\text{H, random}}$: $p = 0.001$ $I_{\text{H, no}} > I_{\text{H, random}}$: $p < 0.001$ Note, that #samples for $I_{\text{H, no}}$ is 14 and #samples for $I_{\text{H, random}}$ is 6*
9 / 4	scaled	$I_{\text{H, online}} > I_{\text{H, no}} > I_{\text{H, random}}$	all tests highly significant ($p < 0.001$) Note, that #samples for $I_{\text{H, no}}$ is 12*

[Bleuler et al., 2003] 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., Fleming, P. J., Zitzler, E., Deb, K., and Thiele, L., editors, *Evolutionary Multi-Criterion Optimization (EMO 2003)*, volume 2632 of *Lecture Notes in Computer Science*, pages 494–508, Berlin. Springer.

[Brockhoff et al., 2007] Brockhoff, D., Friedrich, T., Hebbinghaus, N., Klein, C., Neumann, F., and Zitzler, E. (2007). Do Additional Objectives Make a Problem Harder? In *Genetic and Evolutionary Computation Conference (GECCO 2007)*. accepted.

[Brockhoff and Zitzler, 2006a] Brockhoff, D. and Zitzler, E. (2006a). Are All Objectives Necessary? On Dimensionality Reduction in Evolutionary Multiobjective Optimization. In Runarsson, T. et al., editors, *Parallel Problem Solving from Nature (PPSN IX)*, volume 4193 of *LNCS*, pages 533–542, Berlin, Germany. Springer.

[Brockhoff and Zitzler, 2006b] Brockhoff, D. and Zitzler, E. (2006b). Dimensionality Reduction in Multiobjective Optimization: The Minimum Objective Subset Problem. In *Proceedings of the Operations Research 2006 conference*. Springer. to appear.

[Brockhoff and Zitzler, 2006c] Brockhoff, D. and Zitzler, E. (2006c). On Objective Conflicts and Objective Reduction in Multiple Criteria Optimization. TIK Report 243, Institut für Technische Informatik und Kommunikationsnetze, ETH Zürich.

[Charikar et al., 2000] Charikar, M., Guruswami, V., Kumar, R., Rajagopalan, S., and Sahai, A. (2000). Combinatorial feature selection problems. In *IEEE Symposium on Foundations of Computer Science*, pages 631–640.

[Coello Coello, 2005] Coello Coello, C. A. (2005). Recent Trends in Evolutionary Multiobjective Optimization. In *Evolutionary Multiobjective Optimization: Theoretical Advances And Applications*, pages 7–32. Springer-Verlag, London.

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

[Dai et al., 2006] Dai, J. J., Lieu, L., and Rocke, D. (2006). Dimension reduction for classification with gene expression microarray data. *Statistical Applications in Genetics and Molecular Biology*, 5(1). article 6.

[Dash and Liu, 1997] Dash, M. and Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis*, 1(3):131–156.

[De Jong et al., 2001] 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)*, pages 11–18. Morgan Kaufmann Publishers.

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

[Deb et al., 2000] Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In Schoenauer, M. et al., editors, *Parallel Problem Solving from Nature (PPSN VI)*, Lecture Notes in Computer Science Vol. 1917, pages 849–858. Springer.

[Deb and Saxena, 2006] 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)*, pages 3352–3360. IEEE Press.

[Deb et al., 2005] 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.

[Droste et al., 2002] Droste, S., Jansen, T., and Wegener, I. (2002). On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276:51–81.

[Ekárt and Németh, 2001] 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 et al., 2005] Emmerich, M., Beume, N., and Naujoks, B. (2005). An EMO Algorithm Using the Hypervolume Measure as Selection Criterion. In *Evolutionary Multi-Criterion Optimization (EMO 2005)*, volume 3410 of *Lecture Notes in Computer Science*, pages 62–76. Springer-Verlag Berlin.

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

[Gal and Leberling, 1977] 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.

[Horn, 1997] Horn, J. (1997). Multicriterion decision making. In Bäck, T., Fogel, D. B., and Michalewicz, Z., editors, *Handbook of Evolutionary Computation*. CRC Press.

[Hughes, 2007] Hughes, E. J. (2007). Radar Waveform Optimization as a Many-Objective Application Benchmark. In *Evolutionary Multi-Criterion Optimization (EMO 2007)*, volume 4403 of *Lecture Notes in Computer Science*.

[Hyvärinen et al., 2001] Hyvärinen, A., Karhunen, J., and Oja, E. (2001). *Independent Component Analysis*. John Wiley & Sons.

[Jensen, 2004] 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. Online Date Wednesday, February 23, 2005.

[Jolliffe, 2002] Jolliffe, I. T. (2002). *Principal component analysis*. Springer.

[Kaelbling et al., 2003] Kaelbling, L. P. et al., editors (2003). *The Journal of Machine Learning Research: Special Issue on Variable and Feature Selection*. MIT Press.

[Khare et al., 2003] Khare, V. R., Yao, X., and Deb, K. (2003). Performance Scaling of Multi-objective Evolutionary Algorithms. In *Evolutionary Multi-Criterion Optimization (EMO 2003)*, volume 2632 of *Lecture Notes in Computer Science*, pages 376–390. Springer.

[Knowles et al., 2001] 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, *Evolutionary Multi-Criterion Optimization (EMO 2001)*, volume 1993 of *Lecture Notes in Computer Science*, pages 269–283, Berlin. Springer-Verlag.

[Langley, 1994] Langley, P. (1994). Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall Symposium on Relevance*, pages 140–144.

[Laumanns et al., 2004a] 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 et al., 2004b] 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.

[Liu and Motoda, 1998] Liu, H. and Motoda, H., editors (1998). *Feature Extraction, Construction and Selection: A Data Mining Perspective*. Kluwer Academic Publishers, Norwell, MA, USA.

[Neumann and Wegener, 2006] Neumann, F. and Wegener, I. (2006). Minimum Spanning Trees Made Easier Via Multi-Objective Optimization. *Natural Computing*, 5(3):305–319. Conference version in Beyer et al. (Eds.): Genetic and Evolutionary

Computation Conference - GECCO 2005, Volume 1, ACM Press, New York, USA, pages 763–770.

[Purshouse and Fleming, 2003a] Purshouse, R. C. and Fleming, P. J. (2003a). Conflict, Harmony, and Independence: Relationships in Evolutionary Multi-criterion Optimisation. In *Evolutionary Multi-Criterion Optimization (EMO 2003)*, volume 2632 of *Lecture Notes in Computer Science*, pages 16–30. Springer, Berlin.

[Purshouse and Fleming, 2003b] Purshouse, R. C. and Fleming, P. J. (2003b). Evolutionary many-objective optimisation: an exploratory analysis. In *Congress on Evolutionary Computation (CEC-03)*, pages 2066–2073.

[Scharnow et al., 2004] 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. Online Date Tuesday, December 28, 2004.

[Tan et al., 2005] Tan, K. C., Khor, E. F., and Lee, T. H. (2005). *Multiobjective Evolutionary Algorithms and Applications*. Springer, London, UK.

[Vafaie and De Jong, 1993] Vafaie, H. and De Jong, K. (1993). Robust feature selection algorithms. In *Tools with Artificial Intelligence (TAI '93)*, pages 356–363.

[Wagner et al., 2007] Wagner, T., Beume, N., and Naujoks, B. (2007). Pareto-, Aggregation-, and Indicator-based Methods in Many-objective Optimization. In Obayashi, S. et al., editors, *Evolutionary Multi-Criterion Optimization (EMO 2007)*, volume 4403 of *Lecture Notes in Computer Science*, pages 742–756, Berlin Heidelberg, Germany. Springer. extended version published as internal report of Sonderforschungsbereich 531 Computational Intelligence CI-217/06, Universität Dortmund, September 2006.

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

[Zitzler et al., 2007] 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, *Evolutionary Multi-Criterion Optimization (EMO 2007)*, volume 4403 of *LNCS*, pages 862–876, Berlin. Springer.

[Zitzler and Künzli, 2004] Zitzler, E. and Künzli, S. (2004). Indicator-based selection in multiobjective search. In *Proc. 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, Birmingham, UK.

[Zitzler et al., 2003] Zitzler, E., Thiele, L., Laumanns, M., Foneseca, 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 Proofs omitted in Section 3

Theorem 2. *Let $\mathcal{F}' \subseteq \mathcal{F}$. Then $\mathcal{F}'$ is δ-nonconflicting 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 $\delta \geq 0$ $\preceq_{\mathcal{F}} \subseteq \preceq_{\mathcal{F}'} \subseteq \preceq^{\delta}_{\mathcal{F}'}$, because $\forall i \in \mathcal{F} : \mathbf{x} \preceq_i \mathbf{y} \Rightarrow \forall i \in \mathcal{F}' \subseteq \mathcal{F} : \mathbf{x} \preceq_i \mathbf{y} \Rightarrow \forall i \in \mathcal{F}' : f_i(\mathbf{x}) \leq f_i(\mathbf{y}) \Rightarrow \forall i \in \mathcal{F}' : f_i(\mathbf{x}) - \delta \leq f_i(\mathbf{y}) \Rightarrow \forall i \in \mathcal{F}' : \mathbf{x} \preceq^{\delta}_i \mathbf{y}$ for all $x, y \in X$ and $\delta > 0$. But then $\mathcal{F}'$ δ-nonconflicting with $\mathcal{F} \iff \mathcal{F}' \sqsubseteq^{\delta} \mathcal{F} \wedge \mathcal{F} \sqsubseteq^{\delta} \mathcal{F}' \iff \preceq_{\mathcal{F}'} \subseteq \preceq^{\delta}_{\mathcal{F}} \wedge \preceq_{\mathcal{F}} \subseteq \preceq^{\delta}_{\mathcal{F}'} \iff \preceq_{\mathcal{F}'} \subseteq \preceq^{\delta}_{\mathcal{F}}$.

Theorem 3. *Let $\mathcal{F}_1, \mathcal{F}_2$ two objective sets and X a decision space. If*

$$\delta' := \max_{\substack{\mathbf{x},\mathbf{y} \in X \wedge \mathbf{x} \preceq_{\mathcal{F}_1} \mathbf{y} \\ i \in \mathcal{F}_2}} \{f_i(\mathbf{x}) - f_i(\mathbf{y})\} \quad \textit{and} \quad \delta'' := \max_{\substack{\mathbf{x},\mathbf{y} \in X \wedge \mathbf{x} \preceq_{\mathcal{F}_2} \mathbf{y} \\ i \in \mathcal{F}_1}} \{f_i(\mathbf{x}) - f_i(\mathbf{y})\},$$

then, $\mathcal{F}_1$ is $\overline{\delta}$-nonconflicting with $\mathcal{F}_2$ w.r.t. X for all $\overline{\delta} \geq \max(\delta', \delta'')$ and no $\underline{\delta} < \max\{\delta', \delta''\}$ exists such that $\mathcal{F}_1$ is $\underline{\delta}$-nonconflicting with $\mathcal{F}_2$.

Proof. Let $\delta', \delta'' \in \mathbb{R}$ as defined above. Then

$$\begin{aligned}
&\forall \mathbf{x}, \mathbf{y} \in X : [\mathbf{x} \preceq_{\mathcal{F}_1} \mathbf{y} \Rightarrow \forall i \in \mathcal{F}_2 : f_i(\mathbf{x}) - f_i(\mathbf{y}) \leq \delta'] \\
&\qquad\qquad \wedge [\mathbf{x} \preceq_{\mathcal{F}_2} \mathbf{y} \Rightarrow \forall i \in \mathcal{F}_1 : f_i(\mathbf{x}) - f_i(\mathbf{y}) \leq \delta''] \\
&\iff \forall \mathbf{x}, \mathbf{y} \in X : [\mathbf{x} \preceq_{\mathcal{F}_1} \mathbf{y} \Rightarrow \forall i \in \mathcal{F}_2 : f_i(\mathbf{x}) - \delta' \leq f_i(\mathbf{y})] \\
&\qquad\qquad \wedge [\mathbf{x} \preceq_{\mathcal{F}_2} \mathbf{y} \Rightarrow \forall i \in \mathcal{F}_1 : f_i(\mathbf{x}) - \delta'' \leq f_i(\mathbf{y})] \\
&\overset{(*)}{\Longrightarrow} \forall \overline{\delta} \geq \max\{\delta', \delta''\} : \forall \mathbf{x}, \mathbf{y} \in X : \left[\mathbf{x} \preceq_{\mathcal{F}_1} \mathbf{y} \Rightarrow \forall i \in \mathcal{F}_2 : f_i(\mathbf{x}) - \overline{\delta} \leq f_i(\mathbf{y})\right] \\
&\qquad\qquad \wedge \left[\mathbf{x} \preceq_{\mathcal{F}_2} \mathbf{y} \Rightarrow \forall i \in \mathcal{F}_1 : f_i(\mathbf{x}) - \overline{\delta} \leq f_i(\mathbf{y})\right] \\
&\iff \forall \overline{\delta} \geq \max\{\delta', \delta''\} : \forall \mathbf{x}, \mathbf{y} \in X : [\mathbf{x} \preceq_{\mathcal{F}_1} \mathbf{y} \Rightarrow \mathbf{x} \preceq^{\overline{\delta}}_{\mathcal{F}_2} \mathbf{y}] \wedge [\mathbf{x} \preceq_{\mathcal{F}_2} \mathbf{y} \Rightarrow \mathbf{x} \preceq^{\overline{\delta}}_{\mathcal{F}_1} \mathbf{y}] \\
&\iff \forall \overline{\delta} \geq \max\{\delta', \delta''\} : \preceq_{\mathcal{F}_1} \subseteq \preceq^{\overline{\delta}}_{\mathcal{F}_2} \wedge \preceq_{\mathcal{F}_2} \subseteq \preceq^{\overline{\delta}}_{\mathcal{F}_1} \\
&\iff \forall \overline{\delta} \geq \max\{\delta', \delta''\} : \mathcal{F}_1 \sqsubseteq^{\overline{\delta}} \mathcal{F}_2 \wedge \mathcal{F}_2 \sqsubseteq^{\overline{\delta}} \mathcal{F}_1 \\
&\iff \mathcal{F}_1 \ \overline{\delta}\text{-nonconflicting with } \mathcal{F}_2 \text{ for all } \overline{\delta} \geq \max\{\delta', \delta''\}
\end{aligned}$$

As a result of implication $(*)$, it is clear that $\mathcal{F}_1$ is $\underline{\delta}$-conflicting with $\mathcal{F}_2$ for any $\underline{\delta} < \max\{\delta', \delta''\}$ if δ' and δ'' are defined as above.

B Correctness proofs

In this section we provide the correctness proofs for the algorithms proposed in Sec. 4.1.

B.1 Greedy Algorithm for δ-MOSS

Before proving the correctness of Algorithm 2, we prove Lemma 1 first.

Lemma 1. *Let $\mathcal{F}' \subseteq \mathcal{F}$ and $\delta > 0$. Then*

$$\Big(\forall \mathbf{x}, \mathbf{y} \in X : \mathbf{x} \preceq_{\mathcal{F}'} \mathbf{y} \Longleftrightarrow \mathbf{x} \preceq^{0,\delta}_{\mathcal{F}',\mathcal{F}\setminus\mathcal{F}'} \mathbf{y}\Big) \Longrightarrow \mathcal{F}' \textit{ is } \delta\textit{-nonconflicting with } \mathcal{F}.$$

Proof. Let $\mathcal{F}' \subseteq \mathcal{F}$ and $A := \Big(\forall \mathbf{x}, \mathbf{y} \in X : \mathbf{x} \preceq_{\mathcal{F}'} \mathbf{y} \Longleftrightarrow \mathbf{x} \preceq^{0,\delta}_{\mathcal{F}',\mathcal{F}\setminus\mathcal{F}'} y\Big)$. Then $\preceq_{\mathcal{F}'} = \preceq^0_{\mathcal{F}'} \overset{A}{=} \preceq^{0,\delta}_{\mathcal{F}',\mathcal{F}\setminus\mathcal{F}'} = \ (\preceq^0_{\mathcal{F}'} \cap \preceq^\delta_{\mathcal{F}\setminus\mathcal{F}'}) \subseteq \preceq^\delta_{\mathcal{F}'} \cap \preceq^\delta_{\mathcal{F}\setminus\mathcal{F}'} = \preceq^\delta_{\mathcal{F}}$, i.e., $\mathcal{F}'$ is δ-nonconflicting with $\mathcal{F}$ according to Theorem 2.

Theorem 4. *Given the objective vectors $f(\mathbf{x}_1), \ldots, f(\mathbf{x}_m) \in \mathbb{R}^k$ and a $\delta \in \mathbb{R}$, Algorithm 2 always provides an objective subset $\mathcal{F}' \subseteq \mathcal{F}$, δ-nonconflicting with $\mathcal{F} := \{1, \ldots, k\}$ in time $O(\min\{k^3 \cdot m^2, k^2 \cdot m^4\})$.*

Proof. If we show that the invariant

$$\forall (\mathbf{x}, \mathbf{y}) \in \overline{R} := (X \times X) \setminus R : \quad \mathbf{x} \preceq_{\mathcal{F}'} \mathbf{y} \Longleftrightarrow \mathbf{x} \preceq^{0,\delta}_{\mathcal{F}',\mathcal{F}} \mathbf{y} \tag{I}$$

holds during each step of Algorithm 2, the theorem is proved, due to Lemma 1 and the fact that $\mathbf{x} \preceq_{\mathcal{F}'} \mathbf{y} \Longleftrightarrow \mathbf{x} \preceq^{0,\delta}_{\mathcal{F}',\mathcal{F}\setminus\mathcal{F}'} \mathbf{y}$ holds for all $(\mathbf{x}, \mathbf{y}) \in X \times X$ if Algorithm 2 terminates, i.e., if $R = \emptyset$. We proof the invariant with induction over $|\overline{R}|$.

Induction basis: When the algorithm starts, $R = X \times X \setminus \preceq_{\mathcal{F}}$, i.e., $\overline{R} = \preceq_{\mathcal{F}}$. For each $(\mathbf{x}, \mathbf{y}) \in \overline{R} = \preceq_{\mathcal{F}}$ with $\mathbf{x} \preceq_{\mathcal{F}'} \mathbf{y}$, i.e., $\mathbf{x} \preceq_{\emptyset} \mathbf{y}$ with $\preceq_{\emptyset} = X \times X$, $\mathbf{x} \preceq_{\mathcal{F}} \mathbf{y}$ holds and therefore $\mathbf{x} \preceq^{0,\delta}_{\mathcal{F}',\mathcal{F}\setminus\mathcal{F}'} \mathbf{y}$. The other direction $\mathbf{x} \preceq^{0,\delta}_{\mathcal{F}',\mathcal{F}\setminus\mathcal{F}'} \mathbf{y} \Rightarrow \mathbf{x} \preceq_{\mathcal{F}'} \mathbf{y}$ 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 2. Note, first, that R becomes only smaller by-and-by, i.e., $\overline{R}$ contains more and more pairs $(\mathbf{x}, \mathbf{y}) \in X \times X$. Such a pair $(\mathbf{x}, \mathbf{y})$, 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 an $\{i\}$ is inserted in $\mathcal{F}'$ to gain $\mathcal{F}'' \supseteq \mathcal{F}'$, two possibilities for a pair $(\mathbf{x}, \mathbf{y}) \in \overline{R}$ exist. First, if $\mathbf{x} \not\preceq_{\mathcal{F}'} \mathbf{y}$, then $\mathbf{x} \not\preceq_{\mathcal{F}''} \mathbf{y}$ for any $\mathcal{F}'' \supseteq \mathcal{F}'$ and also $\mathbf{x} \not\preceq^{0,\delta}_{\mathcal{F}'',\mathcal{F}\setminus\mathcal{F}''} \mathbf{y}$. Second, if $\mathbf{x} \preceq_{\mathcal{F}'} \mathbf{y}$, then $\mathbf{x} \preceq^{0,\delta}_{\mathcal{F}',\mathcal{F}\setminus\mathcal{F}'} \mathbf{y}$ by induction hypothesis. Thus, $\mathbf{x} \preceq^{\delta}_{\mathcal{F}\setminus\mathcal{F}'} \mathbf{y}$ and $\mathbf{x} \preceq^{\delta}_{\mathcal{F}\setminus\mathcal{F}''} \mathbf{y}$ for any $\mathcal{F}'' \supseteq \mathcal{F}'$. If $\mathbf{x} \preceq_{\mathcal{F}''} \mathbf{y}$ for any $\mathcal{F}'' \supseteq \mathcal{F}'$, then $\mathbf{x} \preceq^{0,\delta}_{\mathcal{F}'',\mathcal{F}\setminus\mathcal{F}''} \mathbf{y}$ and if $\mathbf{x} \not\preceq_{\mathcal{F}'} \mathbf{y}$ for any $\mathcal{F}'' \subseteq \mathcal{F}'$ then $\mathbf{x} \not\preceq^{0,\delta}_{\mathcal{F}'',\mathcal{F}\setminus\mathcal{F}''} \mathbf{y}$. Thus, a pair $(\mathbf{x}, \mathbf{y}) \in \overline{R}$ will always fulfill the implication in (I) for any $\mathcal{F}'' \supseteq \mathcal{F}'$ if it fulfills it for $\mathcal{F}'$. Beyond, a pair $(\mathbf{x}, \mathbf{y}) \in X \times X$ will only be included in $\overline{R}$ during the update of R in line 7 if

(i) $(\mathbf{x},\mathbf{y}) \notin (R \cap \preceq_{i^*})$ or if
(ii) $(\mathbf{x},\mathbf{y}) \in \preceq^{0,\delta}_{\mathcal{F}' \cup \{i^*\}, \mathcal{F} \setminus (\mathcal{F}' \cup \{i^*\})}$

In case (i), the invariant stays true because for all new pairs $(\mathbf{x},\mathbf{y})$ in $\overline{R}$, $(\mathbf{x},\mathbf{y}) \in R \wedge (\mathbf{x},\mathbf{y}) \notin \preceq_{i^*}$ holds. Thus, $(\mathbf{x},\mathbf{y}) \notin \cap_{i \in (\mathcal{F}' \cup \{^*\})} \preceq_i = \preceq_{\mathcal{F}'}$ and, therefore, $(\mathbf{x},\mathbf{y}) \notin \preceq^{0,\delta}_{\mathcal{F}' \cup \{i^*\}, \mathcal{F} \setminus (\mathcal{F}' \cup \{i^*\})}$ as well. In the case (ii), $(\mathbf{x},\mathbf{y}) \in \preceq^{0,\delta}_{\mathcal{F}' \cup \{i^*\}, \mathcal{F} \setminus (\mathcal{F}' \cup \{i^*\})}$ and trivially $(\mathbf{x},\mathbf{y}) \in \preceq_{\mathcal{F}' \cup \{i^*\}}$, i.e., the invariant remains true, too.

The running time of Algorithm 2 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 2 terminates. On the other hand, Algorithm 2 terminates if $R = \emptyset$, i.e., after at most $|X \times X| = O(m^2)$ cycles of the while loop, if in each cycle $|R|$ decreases by at least one—what is true due to Theorem 1. The total running time of Algorithm 2 is, therefore, $O(\max\{k, m^2\} \cdot k^2 \cdot m^2) = O(\max\{k^3 \cdot m^2, k^2 \cdot m^4\})$.

B.2 Exact Algorithm

Theorem 5. *Algorithm 1 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 1, we use Lemma 2. It states that Algorithm 1 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 1 follows directly from the lines 12 and 13.

The upper bound on the running time of Algorithm 1 results from the size of the set S_M. For all of the $O(m^2)$ solution pairs, the set $S_{\{(\mathbf{x},\mathbf{y})\}}$ can be computed in time $O(k^3) = o(k \cdot 2^k)$, but the computation time for $S_M \sqcup S_{\{(\mathbf{x},\mathbf{y})\}}$ can be exponential in k. As S_M contains at most $O(2^k)$ objective subsets of size $O(k)$, the computation of $S_M \sqcup S_{\{(\mathbf{x},\mathbf{y})\}}$ in line 9 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 15 *Let $\mathcal{F}' \subseteq \mathcal{F}$ and $M \subseteq X \times X$. Then $\delta(\mathcal{F}', M) := \delta_{\min}(\mathcal{F}', \mathcal{F})$ w.r.t. all solution pairs $(\mathbf{x},\mathbf{y}) \in M$.*

Lemma 2. *Given an instance of the δ-`MOSS` or the `k-EMOSS` problem. Let $\mathcal{F}' \subseteq \mathcal{F}$, $\mathcal{F}' \neq \emptyset$, an arbitrary objective set and*

$$M := \{(\mathbf{x}, \mathbf{y}) \in X \times X \,|\, (\mathbf{x}, \mathbf{y}) \text{ considered in Algorithm 1 so far}\}.$$

Then there exists always a $(\mathcal{F}'' \subseteq \mathcal{F}', \delta'') \in S_M$, such that the following six statements hold.

(i) $\delta(\mathcal{F}'', M) = \delta''$
(ii) $\delta(\mathcal{F}', M) = \delta''$
(iii) $\nexists(\mathcal{F}''', \delta''') \in S_M : \mathcal{F}''' \subset \mathcal{F}' \wedge \delta''' \leq \delta''$
(iv) $\nexists(\mathcal{F}''', \delta''') \in S_M : \mathcal{F}''' \subseteq \mathcal{F}' \wedge \delta''' < \delta''$
(v) $\nexists(\mathcal{F}''', \delta''') \in S_M : \mathcal{F}''' \supset \mathcal{F}' \wedge \delta''' \geq \delta''$
(vi) $\nexists(\mathcal{F}''', \delta''') \in S_M : \mathcal{F}''' \supseteq \mathcal{F}' \wedge \delta''' > \delta''$

Proof. The statements (iii)-(vi) hold for any M due to the definition of the $\sqcup$-union in line 9. We, therefore, prove only (i) and (ii) by mathematical induction on $|M|$.
Induction basis: Let $|M| = 1$, i.e., $M := \{(\mathbf{x}, \mathbf{y})\}$.

(a) $\mathbf{x} \sim_{\mathcal{F}} \mathbf{y}$: Thus, $\forall i \in \mathcal{F} : f_i(\mathbf{x}) = f_i(\mathbf{y})$ and $\forall \mathcal{F}' \subseteq \mathcal{F}, \mathcal{F}' \neq \emptyset : \delta(\mathcal{F}', \{(\mathbf{x}, \mathbf{y})\}) = 0$. By definition of $\sqcup$, Algorithm 1 computes $S_{\{(\mathbf{x},\mathbf{y})\}} = \{(\{i\}, 0) \,|\, 1 \leq i \leq k\}$ correctly according to (i) and (ii).

(b) Without loss of generality $\mathbf{x} \preceq_{\mathcal{F}} \mathbf{y} \wedge \neg(\mathbf{y} \preceq_{\mathcal{F}} \mathbf{x})$: We can divide $\mathcal{F}$ into two disjoint sets $\mathcal{F}_=, \mathcal{F}_<$ with $\mathcal{F}_= \cup \mathcal{F}_< = \mathcal{F}$, $\mathcal{F}_< \neq \emptyset$, $\forall i \in \mathcal{F}_= : \mathbf{x} \preceq_i \mathbf{y} \wedge \mathbf{y} \preceq_i \mathbf{x}$, and $\forall i \in \mathcal{F}_< : \mathbf{x} \preceq_i \mathbf{y} \wedge \neg(\mathbf{y} \preceq_i \mathbf{x})$, i.e., $\forall i \in \mathcal{F}_= : f_i(\mathbf{x}) = f_i(\mathbf{y})$ and $\forall i \in \mathcal{F}_< : f_i(\mathbf{x}) < f_i(\mathbf{y})$. Furthermore, $\forall i \in \mathcal{F}_< : \delta(\{i\}, \{(\mathbf{x}, \mathbf{y})\}) = 0$ and $\forall i \in \mathcal{F}_= : \delta(\{i\}, \{(\mathbf{x}, \mathbf{y})\}) = \delta > 0$ with $\delta := \max_{j \in \mathcal{F}_<}\{f_j(\mathbf{y}) - f_j(\mathbf{x})\}$ independent of the choice of i. Therefore, $S_{\{(\mathbf{x},\mathbf{y})\}}$ 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 $\mathcal{F}' \subseteq \mathcal{F}$, $\mathcal{F}' \neq \emptyset$, $\delta' := \delta(\mathcal{F}', \{(\mathbf{x}, \mathbf{y})\})$ 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) $\mathbf{x} \parallel_{\mathcal{F}} \mathbf{y}$: We can divide $\mathcal{F}$ into three well-defined disjoint sets $\mathcal{F}_<$, $\mathcal{F}_>$, and $\mathcal{F}_=$ with $\mathcal{F}_< \cup \mathcal{F}_> \cup \mathcal{F}_= = \mathcal{F}$, $\mathcal{F}_< \neq \emptyset$, $\mathcal{F}_> \neq \emptyset$, $\forall i \in \mathcal{F}_< : f_i(\mathbf{x}) < f_i(\mathbf{y})$, $\forall i \in \mathcal{F}_> : f_i(\mathbf{x}) > f_i(\mathbf{y})$, and $\forall i \in \mathcal{F}_= : f_i(\mathbf{x}) = f_i(\mathbf{y})$. For all singletons $\{i\}$ with $1 \leq i \leq k$, $\delta_i := \delta(\{i\}, \{(\mathbf{x}, \mathbf{y})\}) > 0$ holds, i.e., $(\{i\}, \delta_i) \in S_{\{(\mathbf{x},\mathbf{y})\}}$ for all $i \in \mathcal{F}$ and

$$\delta_i := \begin{cases} \delta_< := \max_{j \in \mathcal{F}_>}\{f_j(\mathbf{x}) - f_j(\mathbf{y})\} & \text{if } i \in \mathcal{F}_< \\ \delta_> := \max_{j \in \mathcal{F}_<}\{f_j(\mathbf{y}) - f_j(\mathbf{x})\} & \text{if } i \in \mathcal{F}_> \\ \delta_= := \max_{j \in \mathcal{F} \setminus \{i\}}\{|f_j(\mathbf{x}) - f_j(\mathbf{y})|\} & \text{if } i \in \mathcal{F}_= \end{cases}.$$

In addition, $S_{\{(\mathbf{x},\mathbf{y})\}}$ 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 $S_{\{(\mathbf{x},\mathbf{y})\}}$ 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 $\delta(\mathcal{F}', \{(\mathbf{x}, \mathbf{y})\})$ can only take a value $\delta' \in \{0, \delta_<, \delta_>, \delta_=\}$ and a pair $(\mathcal{F}'' \subseteq, \delta')$ exists in $S_{\{(\mathbf{x},\mathbf{y})\}}$:

1. $\delta(\mathcal{F}', \{(\mathbf{x}, \mathbf{y})\}) = 0$ if $\mathcal{F}'_> \neq \emptyset \wedge \mathcal{F}'_< \neq \emptyset$. But then, $i \in \mathcal{F}'_>$ and $j \in \mathcal{F}'_<$ exist and $(\{i, j\}, 0) \in S_{\{(\mathbf{x},\mathbf{y})\}}$.
2. Without loss of generality $\delta(\mathcal{F}', \{(\mathbf{x}, \mathbf{y})\}) = \delta_<$ if $\mathcal{F}'_> = \emptyset \wedge \mathcal{F}'_< \neq \emptyset$. Then there exists an $i \in \mathcal{F}'_<$ and $(\{i\}, \delta_<) \in S_{\{(\mathbf{x},\mathbf{y})\}}$
3. $\delta(\mathcal{F}', \{(\mathbf{x}, \mathbf{y})\}) = \delta_=$ 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 $(\{i\}, \delta_=) \in S_{\{(\mathbf{x},\mathbf{y})\}}$.

Induction step: Let $\mathcal{F}' \subseteq \mathcal{F}$ an arbitrary objective set with $\delta(\mathcal{F}', M \cup \{(\mathbf{x}, \mathbf{y})\})$. Assume that (i)-(vi) holds for M and $\{\mathbf{x}, \mathbf{y}\}$. Thus, $\exists(\mathcal{F}''', \delta''') \in S_M$ with $\mathcal{F}''' \subseteq \mathcal{F}'$ and (i)-(vi) and $\exists(\mathcal{F}'''', \delta'''') \in S_{\{(\mathbf{x},\mathbf{y})\}}$ with $\mathcal{F}'''' \subseteq \mathcal{F}'$ and (i)-(vi).

To show that an $(\mathcal{F}'' \subseteq \mathcal{F}', \delta'')$ exists in $S_{M \cup \{(\mathbf{x},\mathbf{y})\}} := S_M \sqcup S_{\{(\mathbf{x},\mathbf{y})\}}$ that fulfills (i) and (ii), we define $\mathcal{F}'' := \mathcal{F}''' \cup \mathcal{F}'''' \subseteq \mathcal{F}'$ and $\delta'' := \max\{\delta''', \delta''''\}$. Because $\delta(\mathcal{F}''', M) = \delta(\mathcal{F}', M)$, $\delta(\mathcal{F}''', M) = \delta(\hat{\mathcal{F}}, M)$ holds for any $\mathcal{F}''' \subseteq \hat{\mathcal{F}} \subseteq \mathcal{F}'$ and because of $\delta(\mathcal{F}'''', \{(\mathbf{x}, \mathbf{y})\}) = \delta(\mathcal{F}', \{(\mathbf{x}, \mathbf{y})\})$, $\delta(\mathcal{F}'''', \{(\mathbf{x}, \mathbf{y})\}) = \delta(\hat{\mathcal{F}}, \{(\mathbf{x}, \mathbf{y})\})$ holds for any $\mathcal{F}'''' \subseteq \hat{\mathcal{F}} \subseteq \mathcal{F}'$. Together with $\mathcal{F}''' \cup \mathcal{F}'''' \subseteq \mathcal{F}'$, this yields $\delta(\mathcal{F}''' \cup \mathcal{F}'''', M) = \delta(\mathcal{F}', M) = \delta'''$ as well as $\delta(\mathcal{F}''' \cup \mathcal{F}'''', \{(\mathbf{x}, \mathbf{y})\}) = \delta(\mathcal{F}', M) = \delta''''$. This follows (i) and (ii):

$$\begin{aligned}
\delta'' &= \max\{\delta(\mathcal{F}''' \cup \mathcal{F}'''', M), \delta(\mathcal{F}''' \cup \mathcal{F}'''', \{(\mathbf{x}, \mathbf{y})\})\} \\
&= \delta(\mathcal{F}''' \cup \mathcal{F}'''', M \cup \{(\mathbf{x}, \mathbf{y})\}) && \text{(i)} \\
&= \max\{\delta(\mathcal{F}', M), \delta(\mathcal{F}', \{(\mathbf{x}, \mathbf{y})\})\} = \delta(\mathcal{F}', M \cup \{(\mathbf{x}, \mathbf{y})\}) && \text{(ii)}
\end{aligned}$$