Objective Reduction in Evolutionary Multiobjective Optimization: Theory and Applications

Dimo Brockhoff dimo.brockhoff@tik.ee.ethz.ch
Computer Engineering and Networks Laboratory, ETH Zurich, 8092 Zurich, Switzerland

Eckart Zitzler eckart.zitzler@tik.ee.ethz.ch
Computer Engineering and Networks Laboratory, ETH Zurich, 8092 Zurich, Switzerland

Abstract

Many-objective problems represent a major challenge in the field of evolutionary multiobjective optimization—in terms of search efficiency, computational cost, decision making, visualization, and so on. This leads to various research questions, in particular whether certain objectives can be omitted in order to overcome or at least diminish the difficulties that arise when many, that is, more than three, objective functions are involved. This study addresses this question from different perspectives.

First, we investigate how adding or omitting objectives affects the problem characteristics and propose a general notion of conflict between objective sets as a theoretical foundation for objective reduction. Second, we present both exact and heuristic algorithms to systematically reduce the number of objectives, while preserving as much as possible of the dominance structure of the underlying optimization problem. Third, we demonstrate the usefulness of the proposed objective reduction method in the context of both decision making and search for a radar waveform application as well as for well-known test functions.

Keywords

Multiobjective optimization, many-objective problems, dimensionality reduction, objective conflicts, minimum objective sets.

1 Motivation

In the last decade, there has been growing interest in applying evolutionary algorithms to multiobjective optimization problems, mainly to approximate the set of Pareto-optimal solutions. However, most of the publications in this area deal with problems where only a few, that is, between two and four, objectives are involved, while studies with many objectives are rare (cf. Coello Coello et al., 2002). The reason is that a large number of optimization criteria leads to further difficulties with respect to decision making, visualization, and computation; for instance, it has been shown empirically that state-of-the-art multiobjective evolutionary algorithms (MOEAs) such as NSGA-II and SPEA2 do not scale well with an increasing number of objectives (Khare et al., 2003; Purshouse and Fleming, 2003b; Wagner et al., 2007). Nevertheless, from a practical point of view it is desirable with most applications to include as many objectives as possible without the need to specify preferences among the different criteria. The 2007 conference on evolutionary multi-criterion optimization (Obayashi et al., 2007) revealed that there is a need to handle many-objective scenarios, and the challenge is to develop concepts and methods to tackle the aforementioned difficulties.

An interesting research question that arises in this context is whether all optimization criteria are actually necessary and whether some of the objectives may be omitted without—or with only slightly—changing the problem characteristics. The motivation behind this question lies in the observation that additional objectives cause problems mainly when they are competing with existing ones; a set of nonconflicting criteria can be represented by a single objective. Methods for automated objective reduction can be beneficial for both decision making and search. On the one hand, the decision maker would have to consider fewer objective values per solution, it would be easier to visualize the solutions, and the number of nondominated solutions is likely to decrease as shown by Winkler (1985), resulting in a further reduction of the information that has to be taken into account. On the other hand, search algorithms may work more efficiently and consume less computational resources, if the number of objectives is decreased adaptively. For instance, hypervolume-based MOEAs, for example, IBEA by Zitzler and Künzli (2004) and SMS-EMOA by Emmerich et al. (2005), represent a promising approach to overcome the limitations of density-based MOEAs such as NSGA-II and SPEA2 in many-objective scenarios (Wagner et al., 2007). However, even the best known algorithms for computing the hypervolume have running times exponential in the number of objectives (see While, 2005; While et al., 2005, 2006; Fonseca et al., 2006; Beume and Rudolph, 2006), and therefore objective reduction is of high practical relevance for this type of algorithm. Similar issues emerge with computationally expensive objective functions, for example, when extensive simulations need to be carried out in order to determine the objective function values.

The issue of objective reduction has gained only little attention in the literature so far, and existing methods are either restricted to particular function classes or do not take the underlying dominance structure into account. In this paper and based on work by Brockhoff and Zitzler (2006a), we propose a methodology for objective reduction that allows both to consider black-box optimization criteria and to maintain and control the dominance structure. The key contributions are:

- A theoretical investigation of the effects of adding or omitting objectives, and formal notions of objective conflicts, degree of conflict, redundant objectives, and minimum objective sets;
- A definition of different types of objective reduction problems and the design of corresponding exact and greedy algorithms, including running time analyses and methods for visualizing objective relationships;
- A systematic study of the efficacy of the proposed approach on various benchmark problems and a real-world application;
- A concept for incorporating the objective reduction techniques into an evolutionary algorithm, and an empirical evaluation using a hypervolume-based MOEA.

In the following, we will first review related work before presenting the theoretical foundations (Section 3) and the corresponding algorithms (Section 4). The application of the objective reduction methods in decision making and search is demonstrated in Section 5.

2 Related Work

2.1 Evolutionary Many-Objective Optimization

Up to now, there have only been a few studies that have dealt with applications involving many objectives, ranging from nurse scheduling to aircraft construction (Qiu, 1997; Paechter et al., 1998; Coello Coello and Hernández Aguirre, 2002; Fleming et al., 2005; Hughes, 2007; Sülflow et al., 2007). The corresponding optimization problems have been mainly tackled using aggregation approaches such as weighted sum. In recent studies, density-based MOEAs have also been employed—which were shown to have difficulties when the number of objectives is high (cf. Wagner et al., 2007). Hypervolume-based MOEAs (e.g., Emmerich et al., 2005) can bypass these drawbacks but have the disadvantage of larger running times. However, whether such applications are in general harder to solve than problems with a low number of objectives is an open question. Many researchers argue that more objectives induce additional difficulties for evolutionary algorithms (e.g., Fonseca and Fleming, 1995; Horn, 1997; Deb, 2001; Coello Coello et al., 2002; Coello Coello, 2005); other studies have demonstrated that more objectives can make a problem simpler (Knowles et al., 2001; Jensen, 2004; Scharnow et al., 2004; Neumann and Wegener, 2006). None of the above publications, though, has addressed the issue of objective reduction.

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

2.2 Dimensionality Reduction

Dimensionality reduction is a well-known problem in many areas like statistics and data mining, and various methods to extract and select *features*[1] are known. One can distinguish between two distinct approaches: *feature extraction* and *feature selection*. The task in feature extraction is to determine a (small) set of *arbitrary* features, while the task in feature selection is to find the smallest subset of the *given* features, representing the given data best. Translated to the multiobjective optimization field, one can ask either for a set of arbitrary objectives or for a subset of given objectives that describes the original problem best. The former question has been already addressed in the context of coevolution (de Jong and Bucci, 2006). In this paper, though, we focus on the latter aspect since new objectives—potentially defined as combinations of the given ones—are not easy to handle in the decision making process. As Purshouse and Fleming (2003a) already pointed out, common dimensionality reduction techniques cannot be used directly as an objective reduction technique in evolutionary multiobjective optimization since the Pareto-dominance relation is not taken into account—in other words: it cannot be ensured that the Pareto-dominance relation is maintained while the number of objectives is reduced.

An objective reduction approach that preserves the dominance structure was proposed by Gal and Leberling (1977) for the case that the objective functions are explicitly

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

given as linear combinations of the (real) decision variables, that is, the Pareto optimal set is determined within the problem formulation. Hence, this approach, as well as a generalization by Agrell (1997), is restricted to a narrow class of problems and inapplicable to general black-box scenarios.

A method using principal component analysis to decrease the number of objectives was proposed recently by Deb and Saxena (2006). The method aims at computing a set of "the most important conflicting objectives" by omitting redundant ones, that is, those that are less influential with respect to the principal components. It was incorporated into NSGA-II and used to shrink the objective set iteratively over the course of multiple optimization runs. Furthermore, it was tested on and primarily invented for problems where the Pareto-optimal front has a lower dimension than the problem formulation itself. Since the approach of Deb and Saxena (2006) considers the correlation between objectives as an indicator for the conflict between them, it cannot guarantee that the Pareto-dominance relation, and therefore the Pareto-optimal set, is preserved. In addition, no quantitative measure can be specified by how much the dominance relation changes when objectives are omitted. The same holds for a recently published extension that is based on two nonlinear dimensionality reduction techniques (Saxena and Deb, 2007).

In the following, we propose an approach for objective reduction that is both suited to black-box optimization problems and allows for the maintenance and control of the dominance structure.

3 Theoretical Foundations

What happens if one or several objectives are added to or removed from a problem? Under which circumstances can objectives be omitted? How can one measure the degree of conflict between objectives or objective sets? This section deals with these questions, introduces basic definitions and theorems, and thereby lays the foundation for the objective reduction techniques presented in Section 4.

3.1 Relation Graphs and the Combination of Objectives

Suppose an arbitrary optimization scenario with a decision space X and k objective functions $f_i : X \to \mathbb{R}$ $(1 \leq i \leq k)$, which are without loss of generality to be minimized. Furthermore, assume that the weak Pareto dominance relation is the underlying preference structure according to which the optimization is to be carried out. A solution $\mathbf{x} \in X$ is said to *weakly dominate* another solution $\mathbf{y} \in X$ if and only if $\mathbf{x}$ is not worse than $\mathbf{y}$ in all objectives; here, we consider the notation $\mathbf{x} \preceq_{\mathcal{F}'} \mathbf{y}$ in order to indicate that the weak Pareto dominance relation is used w.r.t. a particular objective set $\mathcal{F}' \subseteq \mathcal{F} := \{f_1, f_2, \ldots, f_k\}$:

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

For better readability, we will sometimes only list the indices of the objective functions instead of the function names themselves, for example, $\preceq_{\{1,2\}}$ instead of $\preceq_{\{f_1,f_2\}}$.[2]

[2]In addition, we will use the following standard terms: (i) $\mathbf{x}$ *dominates* $\mathbf{y}$ if $\mathbf{x} \preceq_{\mathcal{F}'} \mathbf{y}$ and $\mathbf{y} \not\preceq_{\mathcal{F}'} \mathbf{x}$; (ii) $\mathbf{x}$ and $\mathbf{y}$ are *comparable* if either $\mathbf{x} \preceq_{\mathcal{F}'} \mathbf{y}$ or $\mathbf{y} \preceq_{\mathcal{F}'} \mathbf{x}$; (iii) $\mathbf{x}$ and $\mathbf{y}$ are *incomparable* if neither $\mathbf{x} \preceq_{\mathcal{F}'} \mathbf{y}$ nor $\mathbf{y} \preceq_{\mathcal{F}'} \mathbf{x}$; (iv) $\mathbf{x}$ and $\mathbf{y}$ are *indifferent* if both $\mathbf{x} \preceq_{\mathcal{F}'} \mathbf{y}$ and $\mathbf{y} \preceq_{\mathcal{F}'} \mathbf{x}$; (v) the *Pareto(-optimal) set* contains all solutions $\mathbf{x}$ that either weakly dominate or are incomparable to any other solution $\mathbf{y} \in X$, (vi) the *Pareto(-optimal) front* is the image of the Pareto set in the objective space $\mathbb{R}^{|\mathcal{F}'|}$.

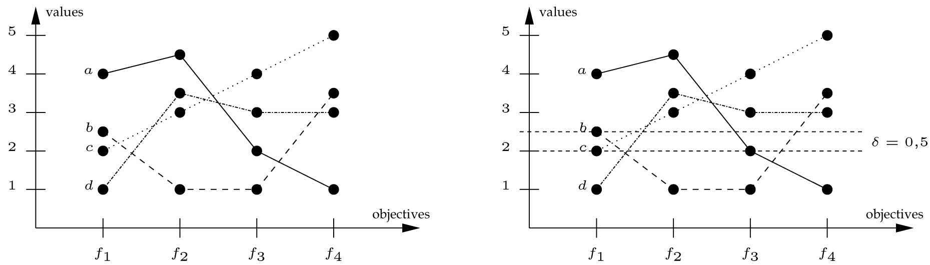

Figure 1: Parallel coordinates plot for the given example with four solutions and four objectives (left); all objectives have to be minimized. On the right, the definition of δ error is illustrated.

Furthermore, we will also use the set notation for relations, that is,

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

where $A \subseteq X$ is a particular set of solutions under consideration; it will be clear from the context which set A is meant.

To illustrate how the weak Pareto dominance relation is modified when objectives are removed from or added to a problem, its representation in terms of a *relation graph* is useful. The relation graph for the weak Pareto dominance relation is given by the tuple $(A, \preceq_{\mathcal{F}'})$ for a solution set A and an objective set $\mathcal{F}'$. It contains for each solution a corresponding node and for each solution pair $\mathbf{x}, \mathbf{y} \in A \subseteq X$ an edge from the node associated with $\mathbf{x}$ to the node associated with $\mathbf{y}$ if and only if $\mathbf{x}$ weakly dominates $\mathbf{y}$ w.r.t. $\mathcal{F}'$.

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

Now, how does adding an objective affect the overall relation graph? Starting with a single-objective problem, the weak Pareto dominance relation $\preceq_{\{f_i\}}$ always forms a total preorder,[4] that is, all solution pairs are comparable (cf. Figure 2(a–d)). With an additional objective, the relation between any two solutions $\mathbf{x}, \mathbf{y} \in X$ can be changed in two ways: (i) $\mathbf{x}$ and $\mathbf{y}$ have been comparable, but not indifferent, and now become incomparable because $\mathbf{x}$ is better regarding the first objective and $\mathbf{y}$ regarding the second (or vice versa), or (ii) $\mathbf{x}$ and $\mathbf{y}$ have been indifferent, but now one solution dominates the other one because it is better regarding the additional objective. The same holds if an

[3]See the paper of Purshouse and Fleming (2003a) for details on parallel coordinates plots.

[4]A relation that is reflexive, transitive and total is called total preorder; if it is also antisymmetric, it is called a partial order. Note, that the weak Pareto-dominance relations for single objectives are usually only total preorders and not partial orders, since solutions with the same objective value can exist, that is, the antisymmetry of $\preceq_{\{f_i\}}$ cannot be guaranteed.

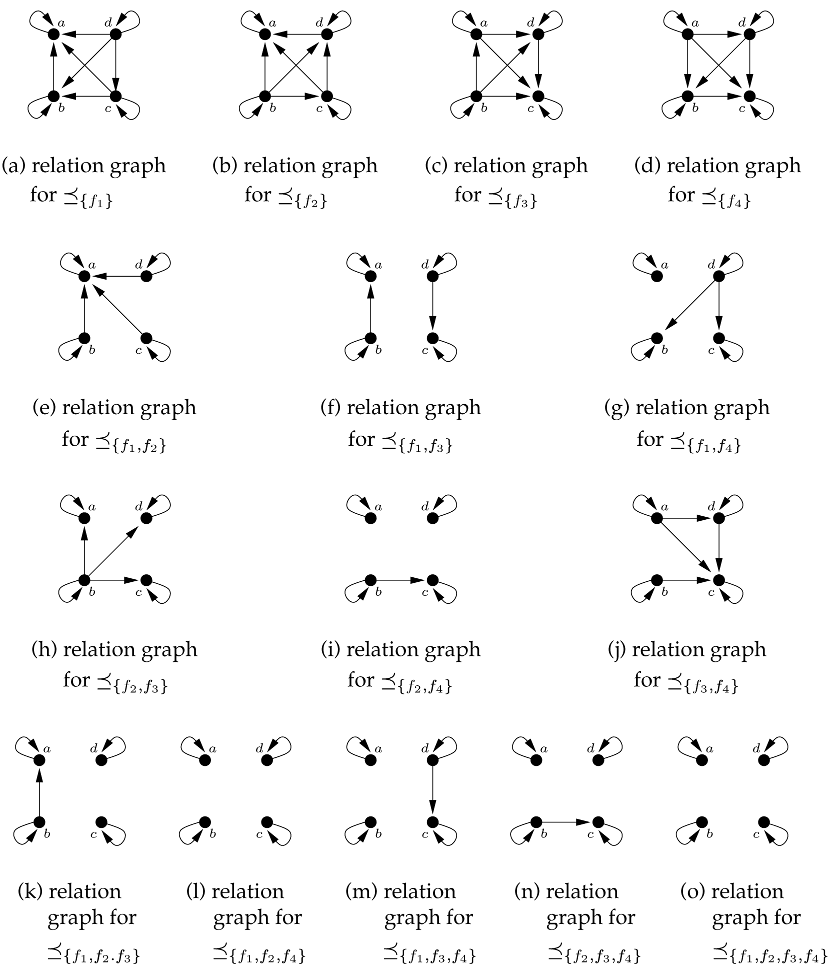

Figure 2: Relation graphs for the solutions depicted in Figure 1.

objective is added to a multiobjective problem. Regarding the relation graph, that means that an additional objective either leaves the edges between two nodes unchanged or removes exactly one edge; overall, adding objectives can only remove edges from the relation graph. Contrariwise, if one or several objectives are omitted, edges are added to the corresponding relation graph: incomparable solutions may become comparable, and a solution dominated by another one may become indifferent to it.

Example 2 Consider the solution pair a, b in Figure 1. When taking only objective f_1 into account as a single-objective minimization problem, solution b is preferred to solution a, that is, b weakly dominates a, see Figure 2(a). If objective f_2 is added, b still weakly dominates a since solution b has smaller objective values than a in both objectives f_1 and f_2 (Figure 2(b) and (e)). In the case of adding objective f_4, the two solutions a and b become incomparable due to the fact that a weakly dominates b w.r.t. f_4. The edge between a and b in the corresponding relation graph disappears, see Figure 2(l).

Since a solution $\mathbf{x}$ weakly dominates another solution $\mathbf{y}$ w.r.t. an objective set if and only if $\mathbf{x}$ weakly dominates $\mathbf{y}$ w.r.t. every single objective, an edge can only be in the relation graph for $\preceq_{\mathcal{F}}$ if for every subset $\mathcal{F}' \subseteq \mathcal{F}$ the corresponding relation graph contains the edge. This can be formalized in the following theorem.

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

PROOF: $\mathbf{x} \preceq_{\mathcal{F}} \mathbf{y} \iff \forall i \in \{1, \ldots, k\} : f_i(\mathbf{x}) \le f_i(\mathbf{y}) \iff \forall i \in \{1, \ldots, k\} : \mathbf{x} \preceq_{\{f_i\}} \mathbf{y} \iff (\mathbf{x}, \mathbf{y}) \in \bigcap_{1 \le i \le k} \preceq_{\{f_i\}}$. □

3.2 Conflicting, Redundant, and Minimum Objective Sets

Based on this result, it is now easy to see under which circumstances objectives can be omitted without changing the problem structure: whenever the underlying relation graph remains the same. We will use the notion of conflicting and nonconflicting objective sets to capture this observation.

DEFINITION 1: *Two objective sets $\mathcal{F}_1, \mathcal{F}_2$ are called* conflicting *if the induced weak Pareto dominance relations differ, that is, $\preceq_{\mathcal{F}_1} \neq \preceq_{\mathcal{F}_2}$ and* nonconflicting *otherwise ($\preceq_{\mathcal{F}_1} = \preceq_{\mathcal{F}_2}$).*

Whenever an objective subset $\mathcal{F}' \subseteq \mathcal{F}$ is nonconflicting with the entire objective set $\mathcal{F}$, an omission of the objectives in $\mathcal{F} \setminus \mathcal{F}'$ will preserve the weak dominance relation, otherwise, the weak dominance relation will change. Other conflict definitions have been proposed in the literature: Deb (2001) and Tan et al. (2005) define conflict only depending on the Pareto-optimal front, while Purshouse and Fleming (2003a) define conflict as a property of objective pairs. As the following example shows, the three mentioned definitions cannot indicate whether objectives can be omitted without affecting the dominance structure.

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

This example indicates that objective conflict appears to be rather a set-based property than a property of objective pairs. Similarly, the question of whether objectives can be omitted while the dominance structure is preserved cannot be decided by considering relations between objective pairs only; the $\mathcal{NP}$-hardness proof of the δ-MOSS problem in Section 4 will support this statement. We will use the term redundancy to state whether objectives in an objective set can be omitted or not (necessary and sufficient criterion).

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

The additional question of which objective set is the smallest one among those nonconflicting with the entire objective set can be denoted as finding a minimum objective set, and will be defined as follows.

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

- minimal w.r.t. *$\mathcal{F}'$ iff $\mathcal{F}''$ is both not redundant and nonconflicting with $\mathcal{F}'$;*
- minimum w.r.t. *$\mathcal{F}'$ iff $\mathcal{F}''$ is the smallest minimal objective set w.r.t. $\mathcal{F}'$.*

A minimal objective set is a subset of the original objectives that cannot be further reduced without changing the associated preorder. A minimum objective set is the smallest possible set of original objectives that preserves the original order on the search space. By definition, every minimum objective set is minimal, but not all minimal sets are at the same time minimum.

Example 4 In the example depicted in Figure 1, the entire objective set is redundant since the objective set $\{f_1, f_2, f_4\}$ induces the same dominance relation as $\{f_1, f_2, f_3, f_4\}$, see Figure 2(l) and (o). The set $\{f_1, f_2, f_4\}$ is at the same time minimal and minimum w.r.t. the entire objective set because no other objective subset with three or fewer objectives induces the same dominance relation as all objectives.

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

3.3 Measuring the Degree of Conflict

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

Example 5 Let us again consider the example in Figure 1 and the corresponding relation graphs from Figure 2. We have seen that the omission of objective f_3 does not change the underlying dominance structure (Figure 2(l)). A further omission of an objective would change the dominance relation by making one (if f_1 is omitted), two (if f_2 is omitted), or even three (if f_4 is omitted) solution pairs comparable. When examining in detail what happens if, for example, f_1 is omitted, we observe that as a result, solution c is weakly dominated by solution b. As b and c are incomparable w.r.t. the entire objective set, we make an error by omitting f_1 and f_3 and wrongly assuming that b weakly dominates c. If the f_1 value of c were larger by an additional term of $\delta = 0.5$, b would weakly dominate c w.r.t. both the set $\{f_2, f_4\}$ and the entire objective set. Thus, we would make no error. The δ value of 0.5 in the example can be used as a measure to quantify the difference in the dominance structure induced by $\{f_2, f_4\}$ and the entire objective set. By computing the δ values for all solution pairs, we can then determine the maximum error. The meaning of the maximum δ value is that whenever we wrongly assume that $\mathbf{x}$ weakly dominates $\mathbf{y}$ w.r.t. an objective subset $\mathcal{F}'$, we also know that $\mathbf{x}$ is not worse than $\mathbf{y}$ in all objectives by an additive term of δ. For $\mathcal{F}' := \{f_2, f_4\}$, the maximum error is $\delta = 0.5$; for $\mathcal{F}' := \{f_1, f_4\}$, the maximum error is $\delta = 2.5$ induced by the solutions b and d and their f_2 values.[5]

This idea of determining the maximum error can be generalized to a continuous definition of objective conflict, namely *δ-conflicting objective sets*, which is based on the weak (additive) ε-dominance relation $\preceq^{\varepsilon}_{\mathcal{F}'}$ defined as[6]

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

where $A \subseteq X$ and $\mathcal{F}' \subseteq \mathcal{F}$ (cf. Zitzler et al., 2003). Instead of using the number of edges in which the corresponding relation graphs differ as a *degree* of conflict, we take a look at the objective values and how much they have to be adjusted by an additive term δ such that the corresponding dominance relations are identical.

DEFINITION 4: *Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be two objective sets. We call $\mathcal{F}_1$* δ-nonconflicting *with $\mathcal{F}_2$ if and only if both ($\preceq_{\mathcal{F}_1} \subseteq \preceq^{\delta}_{\mathcal{F}_2}$) and ($\preceq_{\mathcal{F}_2} \subseteq \preceq^{\delta}_{\mathcal{F}_1}$) holds; otherwise $\mathcal{F}_1$ and $\mathcal{F}_2$ are denoted as* δ-conflicting.

Definition 4 is useful for changing a problem formulation by considering a different objective set. When replacing an objective set $\mathcal{F}_1$ by another objective set $\mathcal{F}_2$ which is δ-nonconflicting with $\mathcal{F}_1$, then after the replacement one can be sure that for any $\mathbf{x}, \mathbf{y} \in X$, $\mathbf{x}$ either weakly dominates $\mathbf{y}$ w.r.t. both objective sets and we make no error, or $\mathbf{x}$ dominates $\mathbf{y}$ w.r.t. $\mathcal{F}_2$ and $\mathbf{x}$ weakly δ-dominates $\mathbf{y}$ w.r.t. $\mathcal{F}_1$. In other words, we make an error by considering $\mathcal{F}_2$ instead of $\mathcal{F}_1$ only if we wrongly assume that $\mathbf{x}$ weakly dominates $\mathbf{y}$, although $\mathbf{x}$ does not weakly dominate $\mathbf{y}$ w.r.t. $\mathcal{F}_1$. In this case, the error is bounded by δ: $\mathbf{y}$ is not better than $\mathbf{x}$ in any objective in $\mathcal{F}_1$ by an additive term of δ. As a consequence, we know that for any Pareto-optimal solution w.r.t. $\preceq_{\mathcal{F}_1}$ there exists a Pareto-optimal solution w.r.t. $\preceq_{\mathcal{F}_2}$ that weakly δ-dominates the former w.r.t. $\mathcal{F}_1$ (and vice versa). When replacing an objective set by a δ-nonconflicting subset of this objective set, one can guarantee that the resulting Pareto-optimal set is not worse than the original Pareto-optimal set by an an additive term of δ in any omitted objective.

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

DEFINITION 5: *A set $\mathcal{F}' \subseteq \mathcal{F}$ of objectives is called* δ-redundant *if and only if there exists an objective subset $\mathcal{F}'' \subset \mathcal{F}'$ that is δ-nonconflicting with $\mathcal{F}'$.*

[5]Note that we always assume that all objective values have the same scale and reference point such that the small errors δ are comparable among the objectives.

In addition, we assume that an error made close to the Pareto-optimal front is of the same importance as the same error made far away from the Pareto-optimal front. Situations where a decision maker prefers extremal solutions with maximal objective function values are not considered here. The same holds for objective functions for which the possible objective function values are not equally distributed: the case that, for example, solutions close to extremal values are more unlikely than ones with midrange values, is not considered in this study. Nevertheless, an incorporation of the decision maker's preference and nonlinear objective functions would be extremely useful but remains future work.

[6]Note that also the multiplicative ε-dominance relation can be used; all the following results apply to this relation as well.

DEFINITION 6: *Let* $\delta \geq 0$. *An objective set* $\mathcal{F}'' \subseteq \mathcal{F}'$ *is denoted as*

- δ-minimal w.r.t. $\mathcal{F}'$ *iff* $\mathcal{F}''$ *is both not δ-redundant and δ-nonconflicting with* $\mathcal{F}'$*;*
- δ-minimum w.r.t. $\mathcal{F}'$ *iff* $\mathcal{F}''$ *is the smallest δ-minimal objective set w.r.t.* $\mathcal{F}'$*.*

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

Example 6 Regarding Figure 1, the set $\{f_1, f_3, f_4\}$ is 0.5-minimal but not 0.5-minimum w.r.t. the entire objective set, since the smaller set $\{f_2, f_4\}$ is 0.5-minimal as well. Because no objective set with one objective only induces an error smaller than or equal to 0.5, the set $\{f_2, f_4\}$ is also 0.5-minimum w.r.t. the entire objective set.

4 Computing Minimum Objective Sets

Regarding objective reduction together with the measure of conflict, as defined in the previous section, there are two perspectives: on the one hand, given an error δ, one may ask for a δ-minimum objective set; on the other hand, one can ask for a δ-minimal objective subset of predefined size k with the smallest possible δ-error. These problems can be formalized as follows.

DEFINITION 7: *Given a* $\delta \in \mathbb{R}$ *and a set* $A \subseteq X$ *of* m *solutions, together with the objective values* $f_i(\mathbf{x}) \in \mathbb{R}$ *where* $1 \leq i \leq k$ *and* $\mathbf{x} \in A$*, the problem* δ-MINIMUM OBJECTIVE SUBSET, δ-MOSS *for short, is to compute an objective subset* $\mathcal{F}' \subseteq \mathcal{F}$ *which is δ-minimum w.r.t.* $\mathcal{F}$*.*

DEFINITION 8: *Given a* $\mathtt{k} \in \mathbb{N}$ *and a set* $A \subseteq X$ *of* m *solutions, together with the objective values* $f_i(\mathbf{x}) \in \mathbb{R}$ *where* $1 \leq i \leq k$ *and* $\mathbf{x} \in A$*, the problem* MINIMUM OBJECTIVE SUBSET OF SIZE k WITH MINIMUM ERROR*, or* k-EMOSS *for short, is to compute an objective subset* $\mathcal{F}' \subseteq \mathcal{F}$ *which has size* $|\mathcal{F}'| \leq \mathtt{k}$ *and is δ-nonconflicting with* $\mathcal{F}$ *with the minimal possible δ.*

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

THEOREM 2: *Both the* δ-MOSS *problem and the* k-EMOSS *problem are* $\mathcal{NP}$*-hard.*

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

4.1 An Exact Algorithm

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

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

The algorithm uses a subfunction $\delta_{\min}(\mathcal{F}_1, \mathcal{F}_2)$, that computes for two solutions $\mathbf{x}, \mathbf{y} \in A$ and two objective sets $\mathcal{F}_1, \mathcal{F}_2$ the minimal δ error such that $\mathcal{F}_1$ and $\mathcal{F}_2$ are δ-nonconflicting w.r.t. the solution set $\{\mathbf{x}, \mathbf{y}\}$. To guarantee that the set S_M contains only pairs $(\mathcal{F}', \delta')$ such that $\mathcal{F}'$ is always δ'-minimal w.r.t. $\mathcal{F}$ with the smallest δ' possible, the union $\sqcup$ of two sets of objective subsets is done with simultaneous deletion of not δ'-minimal pairs $(\mathcal{F}', \delta')$ as follows:

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

The full procedure is detailed in Algorithm 1. Note that the running time of Algorithm 1 is polynomial in the number $m := |A|$ of solutions but exponential in the number k of objectives. Nevertheless, the exact algorithm is applicable for instances with only a few objectives and a moderate number of solutions as the experimental results show (Section 4.3).

THEOREM 3: *Algorithm 1 solves both the* δ-MOSS *and the* k-EMOSS *problem exactly in time* $O(m^2 \cdot k \cdot 2^k)$.

For details and the very technical correctness proof, we refer to Appendix C. The upper bound for the running time of the exact algorithm can be derived by computing the maximum size of the set S_M. As S_M contains at most $O(2^k)$ objective subsets of size $O(k)$, the computation of $S_M \sqcup S_{\{(\mathbf{x},\mathbf{y})\}}$ in line 9 is possible in time $O(k \cdot 2^k)$. The outer loop will be finished after at most $O(m^2)$ iterations. Thus, the entire algorithm runs in time $O(m^2 \cdot k \cdot 2^k)$. Note that the exact algorithm can be easily parallelized, as the computation of the sets $S_{\{(\mathbf{x},\mathbf{y})\}}$ 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 $|\mathcal{F}'| \leq \mathtt{k}$, respectively.

4.2 Heuristics

The three heuristic algorithms we propose in this section are better suited for large instances of the δ-MOSS problem and k-EMOSS, respectively, than the proposed exact

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

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

algorithm. They are much faster but therefore do not guarantee to find a δ-minimum objective set. Nevertheless, the sizes of the objective sets and the δ errors are close to the sizes and errors of the δ-minimal sets found by the exact algorithm, respectively, see Section 4.3. In addition, for the case of 0-MOSS, the greedy algorithm proposed below has the best approximation ratio possible (see Brockhoff and Zitzler, 2006b, for details).

4.2.1 A Greedy Algorithm for δ-MOSS

The general idea of the proposed approximation algorithm for δ-MOSS is to adapt the well-known greedy algorithm for the set cover problem[7] to compute an objective subset $\mathcal{F}'$, δ-nonconflicting with the set $\mathcal{F}$ of all objectives in a greedy way. Starting with an empty set $\mathcal{F}'$ of objectives, the algorithm chooses in each step the objective f_i the addition of which removes most of the edges in the relation graph of $\preceq_{\mathcal{F}'}$ that are not contained in the relation graph for all objectives, that is, $\mathcal{F}$. Since we are interested in approximating the δ-MOSS problem, that is, finding a δ-nonconflicting objective set, we do not care about the remaining edges in $\preceq_{\mathcal{F}'}$ which imply an error of at most δ. This idea is formalized with the following generalization of the weak ε-dominance, the (δ_1, δ_2)-dominance relation.

DEFINITION 9: *Let $\delta_1, \delta_2 \in \mathbb{R}$ and $\mathcal{F}_1, \mathcal{F}_2$ be two objective subsets. The (δ_1, δ_2)-dominance relation $\preceq^{\delta_1,\delta_2}_{\mathcal{F}_1,\mathcal{F}_2}$ on X is defined as $\mathbf{x} \preceq^{\delta_1,\delta_2}_{\mathcal{F}_1,\mathcal{F}_2} \mathbf{y} :\Longleftrightarrow f_1(\mathbf{x}) - \delta_1 \leq f_1(\mathbf{y}) \wedge f_2(\mathbf{x}) - \delta_2 \leq f_2(\mathbf{y})$ for all $\mathbf{x}, \mathbf{y} \in X$.*

The (δ_1, δ_2)-dominance relation states that w.r.t. objective set $\mathcal{F}_1$ a solution δ_1-dominates another and w.r.t. objective set $\mathcal{F}_2$ the same solution δ_2-dominates the second. Within the greedy algorithm for δ-MOSS, the details of which are depicted as Algorithm 2, all edges in $\preceq^{0,\delta}_{\mathcal{F}' \cup \{i\}, \mathcal{F} \setminus (\mathcal{F}' \cup \{i\})}$ are not considered; that means we do not care about solutions inducing an error of at most δ in the objectives in $\mathcal{F} \setminus (\mathcal{F}' \cup \{i\})$, those which are not taken. For the proof of the polynomial running time and the correctness stated in Theorem 4, we refer to Appendix B.

[7]See for example Garey and Johnson (1990) for details.

Algorithm 2 A greedy algorithm for δ-`MOSS`.

Init:
  compute the relations $\preceq_i$ for all $1 \le i \le k$ and $\preceq_{\mathcal{F}}$
  $\mathcal{F}' := \emptyset$
  $R := A \times A \setminus \preceq_{\mathcal{F}}$
while $R \ne \emptyset$ **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

Algorithm 3 A greedy algorithm for `k-EMOSS`

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

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 $\mathcal{F} := \{f_1, \ldots, f_k\}$ in time $O(\min\{k^3 \cdot m^2, k^2 \cdot m^4\})$.*

Note that Algorithm 2 does not necessarily yield a δ-minimal objective set. However, by simply checking whether an additional omission of single objectives in the computed set $\mathcal{F}'$ leaves the dominance relation unchanged, a δ-minimal set can be guaranteed. The asymptotic running time will stay the same, since the additional check of δ-minimality can be done in time $O(k^2 \cdot m^2)$. Furthermore, for a slightly modified version of Algorithm 2, known results for the set cover problem (Slavík, 1996; Feige, 1998) can be used to prove that the algorithm's $\Theta(\log |A|)$ approximation ratio is optimal for the case of $\delta = 0$ (see Brockhoff and Zitzler, 2006b, for details).

4.2.2 A Greedy Algorithm for `k-EMOSS`

A simple greedy heuristic to approximate the `k-EMOSS` problem is to choose the `k` objectives iteratively. Starting with an empty set $\mathcal{F}'$ of objectives in each of the `k` steps, the algorithm chooses the next objective f_i to be included into $\mathcal{F}'$ as the objective yielding the smallest δ such that $\mathcal{F}' \cup \{f_i\}$ is δ-nonconflicting with the entire objective set, see Algorithm 3. Algorithm 3 obviously always computes 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.

THEOREM 5: *Algorithm 3 needs time $O(m^2 \cdot k^3)$ to compute an objective subset of size* `k`.

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

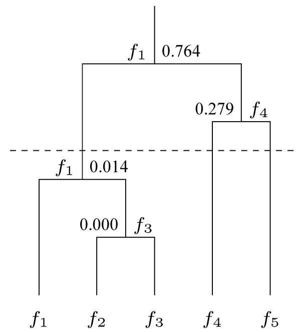

Figure 3: Example of a tree computed by Algorithm 4. More and more objectives are omitted from bottom to top. The dashed line corresponds to the situation after the second objective is omitted, that is, the objective set is reduced to $\{f_1, f_4, f_5\}$. The numbers on the inner nodes denote the δ errors.

4.2.3 A Second Greedy Algorithm for **k-EMOSS** based on Omission of Objectives

To support the 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 Figure 3. Instead of constructing a δ-nonconflicting objective set by adding objectives as in Algorithm 3, the second algorithm 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 between f_i and f_j is selected and the objective that maximizes the error between f_i and f_j is omitted. Algorithm 4 provides the details. If the algorithm is run with $\mathtt{k} = 1$, each of its steps can be visualized as an inner node in a tree (cf. Figure 3) which can support the decision maker with useful information on the measure of conflict between objective pairs. Starting with the set of all objectives at the leaves, each iteration of the algorithm corresponds to an inner node where one objective is omitted; the later an objective is omitted, the closer the corresponding node is to the root.

THEOREM 6: *Algorithm 4 needs time $O((k - \mathtt{k}) \cdot k^2 \cdot m^2) = O(k^3 \cdot m^2)$ to compute an objective subset of size* k*.*

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

4.3 Validation of the Algorithms

Regarding the proposed objective reduction algorithms, two main questions remain. First, what is the usefulness of these algorithms regarding concrete problems, in particular how much can the objective set be reduced? Second, how good are the

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

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

objective sets computed by the greedy methods in comparison with the exact algorithm? This section provides first experimental results for both questions, whereas Section 5 shows how the algorithms can be employed both in decision making and during search.

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

4.3.1 Investigating δ-Minimum Objective Sets

To show the potentials of our objective reduction approach, we use the exact Algorithm 1 to compute 0-minimum objective sets in the random scenario and δ-minimum sets for the entire search space of the 0-1-knapsack problem with seven items.[8] The sizes of the 0-minimum objective sets in the random scenario, averaged over 100 independent random samples, are shown in Figure 4; the sizes of the δ-minimum sets, averaged over five knapsack instances, can be found in the left part of Figure 5.

Regarding the random scenario, the resulting sizes of the minimum objective subsets behave similarly for all tested solution set sizes $|A|$: with increasing numbers of objectives, the size of the computed minimum set increases up to a specific point,

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

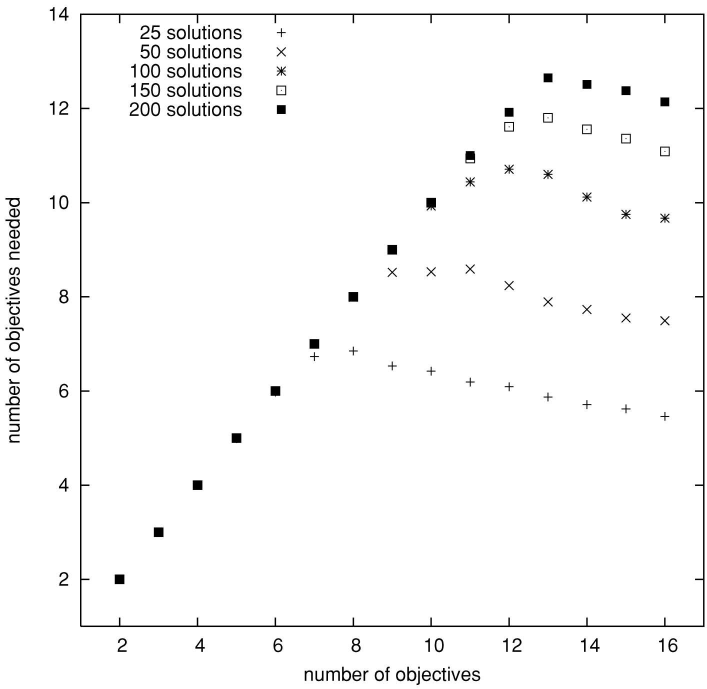

Figure 4: Size of the computed minimum sets for different number k of randomly chosen objectives and the number $|A|$ of solutions.

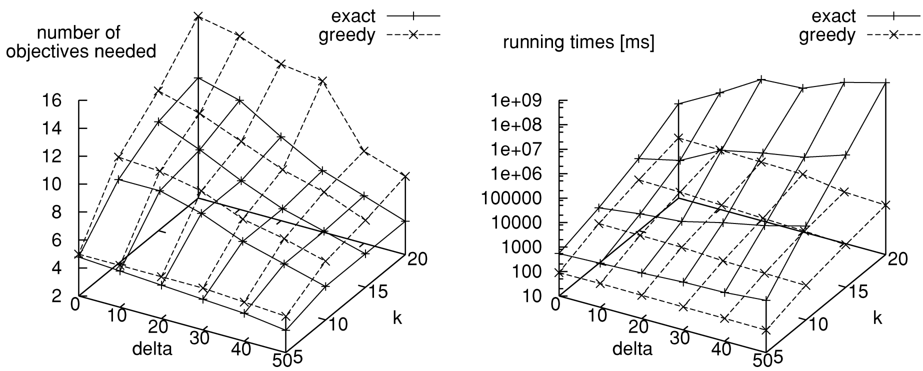

Figure 5: Comparison of the exact and the greedy algorithm for δ-`MOSS` on the 0-1-knapsack problem.

depending on the number of solutions, and further decreases with more objectives. The larger the search space, that is, the more solutions we generate, the fewer objectives can be omitted. With 200 solutions and 16 objectives, however, about 25% of the objectives can be omitted without changing the underlying dominance structure. The investigation of the 0-1-knapsack problem indicates a similar behavior of the δ-minimum objective sets: the more objectives are used in the problem formulation, the more objectives can be

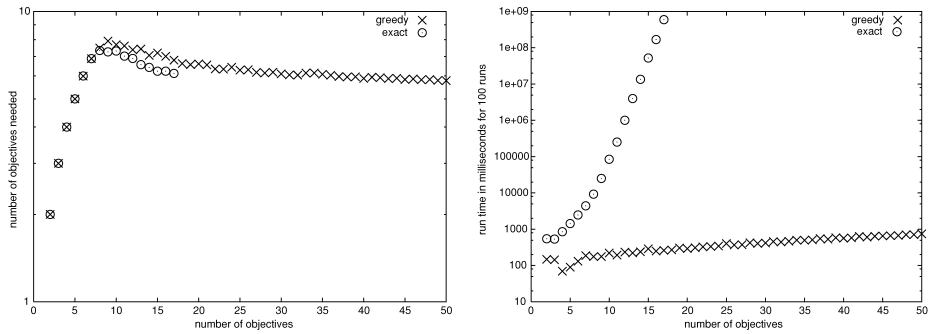

Figure 6: Comparison between the exact and the greedy algorithm for the 0-`MOSS` problem on sets with 32 solutions and random objective values. The left plot shows the size of the computed objective sets averaged over 100 runs for different numbers of objectives. In the right plot, the average running times of both algorithms are shown for 100 runs on each number of objectives.

omitted. Similarly, by increasing the allowed δ error, more objectives can be omitted. For example, the δ-minimum sets contain only 4.4 objectives on average for the 20-objective knapsack problem if we allow an error of $\delta = 50$; instead, 10.6 objectives are needed to preserve the dominance relation with no error.

As the running times, depicted in the right-hand plot of Figure 5 for the knapsack instances and in Figure 6 for the random scenario, indicate, the exact algorithm is not applicable for larger instances of practical size. Therefore, the greedy algorithms with their smaller running times have been developed to cope with problem instances with hundreds of solutions and a few tens of objectives in reasonable time. Because we are—regarding the quality of the algorithms and the error we make—more interested in applying the δ-`MOSS` algorithms within decision making rather than using the `k-EMOSS` algorithms during search, we will compare only the algorithms for the δ-`MOSS` problem in the following.[9]

4.3.2 Investigating Approximate Objective Reduction

Before we investigate approximate objective reduction by applying the greedy algorithms, we briefly compare the exact Algorithm 1 to the greedy Algorithm 2 on δ-`MOSS`. To this end we use both the random objective problem and the 0-1-knapsack problem with seven items as described above and the results of which are shown in Figure 6 and Figure 5, respectively.

For both problems, the comparison shows the same two aspects. First, the objective sets computed with the greedy algorithm are not too large in comparison to the minimum sets computed with the exact algorithm. Nevertheless, the difference between the sizes of the objective sets computed by the two algorithms increase with more objectives. Secondly, the greedy algorithm is—as expected—much faster than the exact algorithm. The running time is a large advantage of the greedy algorithm, especially for larger

[9] In the end it is the error we make in the decision making that matters whereas the quality of the algorithms is not that important if they are used during search.

Table 1: Sizes (for δ-MOSS) and relative errors (for k-EMOSS) of objective subsets for different problems, computed with the Algorithm 2 and 3 respectively. For δ-MOSS, the δ value is chosen relatively to the maximum spread of the IBEA population after 100 generations; in the case of k-EMOSS the specified size k of the output subset is denoted relatively to the problem's number of objectives.

	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

values of δ because the heuristic's running time decreases with larger δ (cf. the right-hand plot in Figure 5).

With the scenario of Pareto-front approximations for the DTLZ and knapsack instances with various numbers of objectives, we investigate the ability of the greedy objective reduction methods to approximate the generalized δ-MOSS and k-EMOSS problems. To be able to compare the results for the different test problems and the varying number of objectives, we choose the δ and k values on a relative basis. On the one hand, the error δ is chosen relative to the spread of the IBEA population after 100 generations, that is, 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 relative to the number $k \in \{5, 15, 25\}$ of objectives in the problem formulation. We choose four different δ values for the δ-MOSS problem (0%, 10%, 20%, 40%) and three different values for k (30%, 60%, 90%). Table 1 shows the results.

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

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 set of nondominated solutions is given, the proposed approach cannot 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 of whether objective reduction is useful during the search is the subject of Section 5.2 where we show experimentally that the integration of an online objective reduction can drastically improve the running time of a hypervolume-based search algorithm.

5.1 Offline Objective Reduction

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

With the set of more than 22,000 nondominated solutions, collected from multiple MOEA runs,[10] we investigate the usefulness of the objective reduction approach proposed above in a decision maker scenario, where a set of nondominated solutions is used to learn about the problem and get a deeper insight into the problem itself. To apply both the exact and the greedy algorithms for δ-MOSS and k-EMOSS, a reduction of the large set of solutions to a smaller set is necessary and performed by computing the ε-nondominated solutions out of the normalized original ones. The error ε is chosen as 6.2% yielding 107 solutions in the reduced set.[11] The computation of the smaller set of ε-nondominated solutions out of the entire set of >22,000 solutions means that whenever we make a statement of δ error w.r.t. the set of 107 solutions, we can ensure that the error w.r.t. the set of all known solutions is at most $\varepsilon + \delta$.

Computing all δ-minimal sets with the exact algorithm shows that for the reduced set of 107 solutions two objectives can be omitted without changing the dominance structure. With respect to the entire set of >22,000 known solutions, that means that we make only an error of at most 6.2% when omitting the correct two objectives. Nevertheless, the use of such a reduction is limited. Reducing the set of objectives from

[10] As provided by Evan Hughes.

[11] Note that the used error of 6.2% and the resulting solution set size of 107 is more or less arbitrary. Smaller errors, that is, larger solution sets with up to 5,000 solutions, yield similar results.

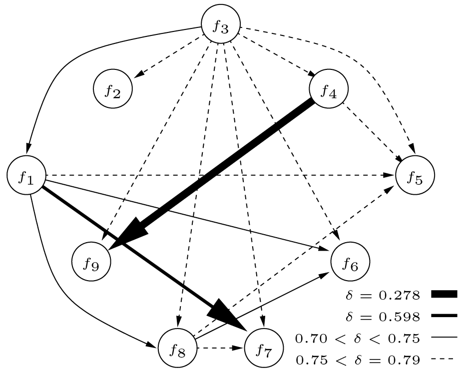

Figure 7: Radar waveform problem: visualization of minimum δ-error between objective pairs. Errors larger than 0.8 are omitted for clarity and the line width indicates the δ-error (the thicker, the smaller the error). The arrows point at the objectives that should be used.

nine to seven still yields a huge amount of information that a decision maker has to survey, especially if more than 22,000 solutions are to be compared. More useful for a decision maker would be to learn about the problem, that is, 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 minimum δ error between all possible objective pairs and illustrate them as in Figure 7. A low δ error between an objective pair predicts that the consideration of only the one objective the arrow points at in Figure 7 while the other objective is omitted does not change the dominance relation with an error of more than δ. Surprisingly, the smallest error occurs between objectives 4 and 9, the second smallest between objective pair 1& 7, in contrast to the prediction of Hughes (2007). These pairwise δ errors can, in addition, be used within the greedy Algorithm 4 to obtain a tree visualization of the objective conflicts. Figure 8 depicts the resulting tree for the 107 solutions in the radar waveform example together with the δ value on the inner nodes (bold) such that the corresponding objective set is δ-nonconflicting with the entire objective set. The tree visualization identifies f_4 and f_1 as the most unimportant and f_6 as the most important objective(s) regarding the dominance structure between the 107 objective vectors used.

This illustrates how the proposed objective reduction approach can be used to analyze objective relations and to assist in decision making in various application domains.

5.2 Online Objective Reduction

Besides the advantages of objective reduction in decision making, one may ask whether objective reduction can also improve the search itself. The following study shows as a proof-of-principle that the objective reduction methods proposed above can improve a simple hypervolume-based MOEA by reducing the number of objectives during search. To this end, we slightly modify the simple indicator-based evolutionary algorithm SIBEA, proposed by Zitzler et al. (2007). SIBEA starts with randomly choosing the population P of size μ. Until a certain time limit T is reached, the μ solutions of

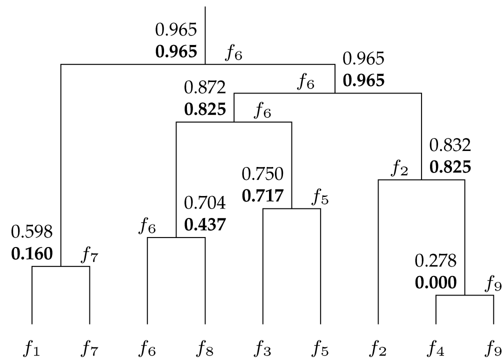

Figure 8: Radar waveform problem: tree visualization of the greedy Algorithm 4 considering the δ-error for paired objectives. The δ-errors are written at the tree's inner nodes (the exact values are given in bold face).

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 the following procedure: after a nondominated sorting of the population, the nondominated fronts are, starting with the best front, completely inserted into the new population until the size of the new population is at least μ. For the first front F, the inclusion of which yields a population size of more than μ, the solutions $\mathbf{x}$ in this front with the smallest hypervolume loss $d(\mathbf{x}) := I_H(F) - I_H(F \setminus \{\mathbf{x}\})$ are successively removed from the new population where the hypervolume loss is recalculated after each removal. This original SIBEA algorithm from Zitzler et al. (2007) without any objective reduction is used as a reference and is denoted by $\text{SIBEA}_{\text{ref}}$. For the hypervolume computation, the algorithm from the performance assessment package of Knowles et al. (2006) is used. By deciding every G generations which objectives are chosen for optimization and which ones are neglected during the next G generations, we introduce two modified versions of $\text{SIBEA}_{\text{ref}}$: On the one hand, $\text{SIBEA}_{\text{random}}$ chooses the k considered objectives every G generations randomly, where the number k of considered objectives is given in advance. On the other hand, $\text{SIBEA}_{\text{online}}$ performs an objective reduction by applying the greedy Algorithm 3 for k-EMOSS on the current population to compute the objectives which are considered in the next G generations. For a detailed description of SIBEA, we refer to Zitzler et al. (2007).

To show that $\text{SIBEA}_{\text{ref}}$ can be improved by reducing the number of considered objectives, we use a slightly modified version of the DTLZ2 problem, known from Deb et al. (2005) as a test function.[12] Figure 9 shows the formal definition of the used function

[12] Due to two main properties of the original DTLZ2 function, the original DTLZ2 function is slightly modified toward the used DTLZ2_{BZ} function: On the one hand, the original DTLZ2 function has the property that the projection of the Pareto front to fewer than k objectives collapses to one optimal point, that is, when omitting arbitrary objectives, the search always converges to one solution. Every multiobjective function has this property if all objectives except one are omitted. For the DTLZ function suite, however, this property even holds for every subset of objectives. To eliminate this property, we limit the range of the variables. On the other hand, when optimizing only a subset of fewer than k objectives, the neglected objectives are also optimized at the same time. The reason is the scaling of all objectives by a function $g(\mathbf{x}_M)$, 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 $g(\mathbf{x}_M)$.

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

Figure 9: Definition of the modified DTLZ2$_{BZ}$ problem.

DTLZ2$_{BZ}$. In addition, we differentiate in the experiments between two versions of the DTLZ2$_{BZ}$ function, one unscaled and one scaled version.[13] The idea behind the scaled version is that not all objectives are, in general, equally scaled in practical problems. A different scaling of the objectives leads to some objectives that have a higher impact on the hypervolume than other objectives. We expect that especially in this case of not equally scaled objectives, the objective reduction approach proposed above can improve hypervolume-based MOEAs considerably compared to a randomly performed objective reduction.

5.2.1 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 $\mathtt{k}$. As algorithms we use the three SIBEA versions described above. As problems, we use the above defined DTLZ2$_{BZ}$ test problem with three, five, and nine objectives, where the chosen objective set sizes $\mathtt{k}$ 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 $k/\mathtt{k}$: 3/2, 5/2, 5/3, 9/2, 9/3, 9/4. We fix the computation time to $T = 300$ s 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 Wilcoxon rank sum test[14] is used to support the hypothesis that one random variable, for example, the hypervolume indicator of algorithm $\mathcal{A}$, "systematically" produces larger values than another one,

[13]The unscaled version is described in Figure 9 whereas in the scaled version, the objective value $f_i(\mathbf{x})$ $(1 \le i \le k)$ is scaled to $f_i^*(\mathbf{x}) = \text{maxValue} \cdot (f_i(\mathbf{x})/\text{maxValue})^i$ if i is even, and to $f_i^*(\mathbf{x}) = \text{maxValue} \cdot (f_i(\mathbf{x})/\text{maxValue})^{1/i}$ if i is odd, where $\text{maxValue} = 1 + (n - M + 1)/4$ is an upper bound for the original $f_i(\mathbf{x})$ values.

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

Table 2: Ranking between the hypervolume indicator values of the three algorithms $\text{SIBEA}_{\text{ref}}$ ($I_{\text{H, no}}$), $\text{SIBEA}_{\text{random}}$ ($I_{\text{H, random}}$), and $\text{SIBEA}_{\text{online}}$ ($I_{\text{H, online}}$) based on the results of the Wilcoxon rank sum tests on a significance level of $p = 0.05$. If the test between two indicator samples is significant, the sample with the larger mean is assigned a smaller rank; otherwise the two samples get the same rank. A lower rank is always better. (For details see Brockhoff and Zitzler, 2007b.)

			unscaled DTLZ2_{BZ}			scaled DTLZ2_{BZ}		
k	/	$\mathtt{k}$	$I_{\text{H, online}}$	$I_{\text{H, no}}$	$I_{\text{H, random}}$	$I_{\text{H, online}}$	$I_{\text{H, no}}$	$I_{\text{H, random}}$
3	/	2	2	1	3	1	2	3
5	/	2	2	1	3	1	2	3
5	/	3	2	1	3	1	2.5	2.5
9	/	2	2	1	3	1	2	3
9	/	3	1.5	1.5	3	1	2	3
9	/	4	1	2	3	1	2	3

for example, the hypervolume indicator of algorithm $\mathcal{B}$ by ranking all values and comparing the rank sums for both samples.

5.2.2 Results

The statistical tests, the results of which are shown in Table 2 as a ranking between the three algorithms, support in most cases the hypothesis derived from preliminary experiments. For all considered problem sizes, the random version $\text{SIBEA}_{\text{random}}$ yields the worst results 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 $\mathtt{k} = 3$). On the scaled DTLZ2_{BZ} problems, $\text{SIBEA}_{\text{online}}$ performs best (all tests significant or highly significant), whereas on the unscaled problems, $\text{SIBEA}_{\text{ref}}$ beats $\text{SIBEA}_{\text{online}}$ except for the nine-objective problem and $\mathtt{k} = 3$ and $\mathtt{k} = 4$. The more objectives are considered, the more $\text{SIBEA}_{\text{online}}$ can gain from the performed objective reduction, for example, $\text{SIBEA}_{\text{ref}}$ can only run for 32 generations in 300 s on scaled DTLZ2_{BZ} with $k = 5$ objectives whereas $\text{SIBEA}_{\text{online}}$ performs more than 600 generations in the same time if the objective set is reduced to $\mathtt{k} = 2$ objectives (and about 300 generations if $\mathtt{k} = 3$). These differences become even larger for higher dimensions (see Brockhoff and Zitzler, 2007a, for details).

Two main statements can be derived from the comparison presented. First, the integration of online reduction methods into a hypervolume-based evolutionary algorithm appears to be a promising approach that can improve the quality of the computed Pareto front approximations with fixed computational resources especially if a large number of objectives is considered. Second, the used objective reduction strategy highly influences the outcome of the evolutionary algorithm: using an advanced objective reduction technique as the δ-conflict based one, presented in this paper, is clearly preferable over a random choice of the objectives to optimize.

6 Conclusion

This paper addresses the issue of objective reduction in many-objective optimization. We have investigated the effect of adding or omitting objectives on the Pareto dominance relation and proposed an objective reduction approach that is based on a general notion of objective conflict. The approach allows us to identify objective sets of minimum size, while ensuring that the Pareto dominance relation is preserved or only slightly changed according to a certain, predefined error. To this end, an exact algorithm as well as several heuristics have been proposed, and corresponding implementations are freely available for download at `http://www.tik.ee.ethz.ch/sop/download/supplementary/objectiveReduction/`.

The experimental results have demonstrated that the proposed methodology can be useful in various domains. On the one hand, the algorithms can be employed to support the decision making process: they not only may substantially reduce the number of objectives, but they also reveal relationships between objectives and objective sets, which may provide valuable information about the underlying problem. On the other hand, integrating the objective reduction techniques into an evolutionary algorithm can significantly improve the search efficiency and thereby the quality of the outcome with fixed computational resources—when the runtime of the selection procedure is strongly affected by the number of objectives as with hypervolume-based multiobjective optimizers. There may be many more application areas, for example, in many-objective scenarios where the computation times needed for the distinct objective functions vary highly. These topics are the subject of future research.

Acknowledgments

The authors would like to thank the anonymous reviewers for their help in improving the readability and understandability of the paper and Evan J. Hughes for providing data for and assistance with the radar waveform application. We also thank Lothar Thiele for valuable discussions during this project and Nikolaus Hansen for his advice concerning the statistical tests. Dimo Brockhoff has been supported by the Swiss National Science Foundation (SNF) under grant 112079.

References

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

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.

Bleuler, S., Laumanns, M., Thiele, L., and Zitzler, E. (2003). PISA—A platform and programming language independent interface for search algorithms. In C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele (Eds.), *Conference on Evolutionary Multi-Criterion Optimization (EMO 2003)*, volume 2632 of *LNCS*, pages 494–508. Berlin: Springer.

Brockhoff, D., and Zitzler, E. (2006a). Are all objectives necessary? On dimensionality reduction in evolutionary multiobjective optimization. In T. Runarsson, H.-G. Beyer, E. Burke, J.-J. Merelo-Guervós, L. D. Whitley, and X. Yao (Eds.), *Conference on Parallel Problem Solving from Nature (PPSN IX)*, volume 4193 of *LNCS*, pages 533–542. Berlin: Springer.

Brockhoff, D., and Zitzler, E. (2006b). On objective conflicts and objective reduction in multiple criteria optimization. TIK Report 243, Computer Engineering and Networks Laboratory (TIK), ETH Zurich.

Brockhoff, D., and Zitzler, E. (2007a). Improving hypervolume-based multiobjective evolutionary algorithms by using objective reduction methods. In *Congress on Evolutionary Computation (CEC 2007)*, pp. 2086–2093. Piscataway, NJ: IEEE Press.

Brockhoff, D., and Zitzler, E. (2007b). Offline and online objective reduction in evolutionary multiobjective optimization based on objective conflicts. TIK Report 269, Computer Engineering and Networks Laboratory (TIK), ETH Zurich.

Coello Coello, C. A. (2005). Recent trends in evolutionary multiobjective optimization. In *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, pp. 7–32. Berlin: Springer.

Coello Coello, C., and Hernández Aguirre, A. (2002). Design of combinational logic circuits through an evolutionary multiobjective optimization approach. *Artificial Intelligence for Engineering, Design, Analysis and Manufacture, 16*(1):39–53.

Coello Coello, C. A., Van Veldhuizen, D. A., and Lamont, G. B. (2002). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. New York: Wiley.

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)*, pp. 3352–3360. Piscataway, NJ: IEEE Press.

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2005). Scalable test problems for evolutionary multi-objective optimization. In A. Abraham, R. Jain, and R. Goldberg (Eds.), *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, chapter 6, pp. 105–145. Berlin: Springer.

de Jong, E. D., and Bucci, A. (2006). DECA: Dimension extracting coevolutionary algorithm. In *Genetic and Evolutionary Computation Conference (GECCO 2006)*, pp. 313–320. New York: ACM.

Emmerich, M., Beume, N., and Naujoks, B. (2005). An EMO algorithm using the hypervolume measure as selection criterion. In *Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, volume 3410 of *LNCS*, pp. 62–76. Berlin: Springer.

Feige, U. (1998). A threshold of ln n for approximating set cover. *J. ACM, 45*(4):634–652.

Fleming, P. J., Purshouse, R. C., and Lygoe, R. J. (2005). Many-objective optimization: An engineering design perspective. In C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler (Eds.), *Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, volume 3410 of *LNCS*, pp. 14–32. Berlin: Springer.

Fonseca, C. M., and Fleming, P. J. (1995). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation, 3*(1):1–16.

Fonseca, C. M., Paquete, L., and López-Ibáñez, M. (2006). An improved dimension-sweep algorithm for the hypervolume indicator. In *Congress on Evolutionary Computation (CEC 2006)*, pp. 1157–1163. Piscataway, NJ: IEEE Press.

Gal, T., and Leberling, H. (1977). Redundant objective functions in linear vector maximum problems and their determination. *European Journal of Operational Research, 1*(3):176–184.

Garey, M. R., and Johnson, D. S. (1990). *Computers and intractability; A guide to the theory of NP-completeness*. New York: W. H. Freeman.

Horn, J. (1997). Multicriterion decision making. In T. Bäck, D. B. Fogel, and Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation*. Bristol, UK: IOP Publishing.

Hughes, E. J. (2007). Radar waveform optimization as a many-objective application benchmark. In *Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, volume 4403 of *LNCS*, pp. 700–714. Berlin: Springer.

Jensen, M. T. (2004). Helper-objectives: Using multi-objective evolutionary algorithms for single-objective optimisation. *Journal of Mathematical Modelling and Algorithms, 3*(4):323–347.

Khare, V. R., Yao, X., and Deb, K. (2003). Performance scaling of multi-objective evolutionary algorithms. In *Conference on Evolutionary Multi-Criterion Optimization (EMO 2003)*, volume 2632 of *LNCS*, pp. 376–390. Berlin: Springer.

Knowles, J., Thiele, L., and Zitzler, E. (2006). A tutorial on the performance assessment of stochastic multiobjective optimizers. TIK Report 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich.

Knowles, J. D., Watson, R. A., and Corne, D. W. (2001). Reducing local optima in single-objective problems by multi-objectivization. In E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello, and D. Corne (Eds.), *Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, volume 1993 of *LNCS*, pp. 269–283. Berlin: Springer.

Laumanns, M., Thiele, L., and Zitzler, E. (2004). Running time analysis of evolutionary algorithms on a simplified multiobjective knapsack problem. *Natural Computing, 3*(1):37–51.

Neumann, F., and Wegener, I. (2006). Minimum spanning trees made easier via multi-objective optimization. *Natural Computing, 5*(3):305–319.

Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., and Murata, T. (Eds.). (2007). *Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, volume 4403 of *LNCS*. Berlin: Springer.

Paechter, B., Rankin, R. C., Cumming, A., and Fogarty, T. C. (1998). Timetabling the classes of an entire university with an evolutionary algorithm. In *Conference on Parallel Problem Solving From Nature (PPSN V)*, pp. 865–874. Berlin: Springer.

Purshouse, R. C., and Fleming, P. J. (2003a). Conflict, harmony, and independence: Relationships in evolutionary multi-criterion optimisation. In *Conference on Evolutionary Multi-Criterion Optimization (EMO 2003)*, volume 2632 of *LNCS*, pp. 16–30. Berlin: Springer.

Purshouse, R. C., and Fleming, P. J. (2003b). Evolutionary many-objective optimisation: An exploratory analysis. In *Congress on Evolutionary Computation (CEC 2003)*, pp. 2066–2073. Piscataway, NJ: IEEE Press.

Qiu, M. (1997). Prioritizing and scheduling road projects by genetic algorithm. *Mathematics and Computers in Simulation, 43*:569–574.

Saxena, D. K., and Deb, K. (2007). Non-linear dimensionality reduction procedures for certain large-dimensional multi-objective optimization problems: Employing correntropy and a novel maximum variance unfolding. In *Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, volume 4403 of *LNCS*, pp. 772–787. Berlin: Springer.

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.

Slavík, P. (1996). A tight analysis of the greedy algorithm for set cover. In *Symposium on Theory of Computing (STOC 1996)*, pp. 435–441. New York: ACM.

Sülflow, A., Drechsler, N., and Drechsler, R. (2007). Robust multi-objective optimization in high dimensional spaces. In *Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, volume 4403 of *LNCS*, pp. 715–726. Berlin: Springer.

Tan, K. C., Khor, E. F., and Lee, T. H. (2005). *Multiobjective evolutionary algorithms and applications*. Berlin: Springer.

Wagner, T., Beume, N., and Naujoks, B. (2007). Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata

(Eds.), *Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, volume 4403 of *LNCS*, pp. 742–756. Berlin: Springer.

While, L. (2005). A new analysis of the LebMeasure algorithm for calculating hypervolume. In *Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, volume 3410 of *LNCS*, pp. 326–340. Berlin: Springer.

While, L., Bradstreet, L., Barone, L., and Hingston, P. (2005). Heuristics for optimising the calculation of hypervolume for multi-objective optimisation problems. In *Congress on Evolutionary Computation (CEC 2005)*, pp. 2225–2232. Piscataway, NJ: IEEE Press.

While, L., Hingston, P., Barone, L., and Huband, S. (2006). A faster algorithm for calculating hypervolume. *IEEE Transactions on Evolutionary Computation*, *10*(1):29–38.

Winkler, P. (1985). Random orders. *Order*, *1*:317–331.

Zitzler, E., Brockhoff, D., and Thiele, L. (2007). The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. In S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata (Eds.), *Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, volume 4403 of *LNCS*, pages 862–876. Berlin: Springer.

Zitzler, E., and Künzli, S. (2004). Indicator-based selection in multiobjective search. In *Conference on Parallel Problem Solving from Nature (PPSN VIII)*, volume 3242 of *LNCS*, pp. 832–842. Berlin: Springer.

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.

Appendix A $\mathcal{NP}$-Hardness Proofs

THEOREM 2: *Both the* δ-`MOSS` *problem and the* `k-EMOSS` *problem are* $\mathcal{NP}$*-hard.*

PROOF: First, we prove the $\mathcal{NP}$-hardness of δ-`MOSS` by a Turing reduction from the $\mathcal{NP}$-hard `MOSS` problem, see Definition 10 and Theorem 7 below. Secondly, we prove the $\mathcal{NP}$-hardness of `k-EMOSS` via a Turing reduction from δ-`MOSS`.

`MOSS` $\leq_T$ δ-`MOSS`

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

δ-`MOSS` $\leq_T$ `k-EMOSS`

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

DEFINITION 10: *Given a multiobjective optimization problem with the objective set $\mathcal{F} = \{f_1, \ldots, f_k\}$, the problem* MINIMUM OBJECTIVE SUBSET (MOSS) *is defined as follows:*

Instance: *The set $A \subseteq X$ of solutions, the generalized weak Pareto dominance relation $\preceq_{\mathcal{F}}$ and for all objective functions $f_i \in \mathcal{F}$ the single relations $\preceq_i$ where $\bigcap_{1 \le i \le k} \preceq_i = \preceq_{\mathcal{F}}$.*

Task: *Compute an index $I \subseteq \{1, \ldots, k\}$ of minimum size with $\bigcap_{i \in I} \preceq_i = \preceq_{\mathcal{F}}$.*

THEOREM 7: *The problem* MOSS *is $\mathcal{NP}$-hard.*

PROOF: First, we define the $\mathcal{NP}$-hard problem SET COVER, or SCP for short, as follows (see Garey and Johnson, 1990).

Given a collection $C = \{C_1, \ldots, C_k\}$ of subsets of a finite set $S = \{1, \ldots, m\}$, compute an index $I \subseteq \{1, \ldots, k\}$ of minimum size with $\bigcup_{i \in I} C_i = S$.

A Turing transformation SCP $\le_T$ MOSS proves the $\mathcal{NP}$-hardness of MOSS. Starting from the SCP instance consisting of the set $S = \{s_1, \ldots, s_m\}$ and the subsets C_i with $1 \le i \le k$, all relations $\preceq_i$ as well as $\preceq_{\mathcal{F}}$ in the MOSS instance are defined on the basic set $A := \{\mathbf{x}_1, \ldots, \mathbf{x}_m, \mathbf{x}'_1, \ldots, \mathbf{x}'_m\}$. The relation $\preceq_{\mathcal{F}}$ will be the reflexive closure of the antichain on A, that is, $\preceq_{\mathcal{F}}$ only contains the elements $(\mathbf{x}_j, \mathbf{x}_j)$ and $(\mathbf{x}'_j, \mathbf{x}'_j)$ for $1 \le j \le m$. The relations $\preceq_i$ with $1 \le i \le k$ are all constructed in the same way. They include the linear order $[\mathbf{x}_1, \mathbf{x}'_1, \mathbf{x}_2, \mathbf{x}'_2, \ldots, \mathbf{x}_m, \mathbf{x}'_m]$ as well as the reflexive relations. Additionally, relation $\preceq_i$ contains the element $(\mathbf{x}'_j, \mathbf{x}_j)$ iff $s_j \notin C_i$. In addition, we have to compute another relation $\preceq_{k+1}$ which is the reverse linear order $[\mathbf{x}'_m, \mathbf{x}_m, \mathbf{x}'_{m-1}, \mathbf{x}_{m-1}, \ldots, \mathbf{x}'_1, \mathbf{x}_1]$. After this transformation, we question our MOSS oracle once. The resulting index $I_{\mathtt{SCP}}$ for the SCP problem will be then $I_{\mathtt{SCP}} := I_{\text{oracle}} \setminus \{k+1\}$ if the oracle produces I_{oracle} as its output.

It remains to be shown that the transformation yields an exact algorithm for SCP with polynomial running time, under the assumption that there is an exact polynomial time algorithm $\mathcal{A}$ for MOSS. Let us assume that $(S = \{s_1, \ldots, s_m\}, C_1, \ldots, C_l)$ is the SCP instance with $C_i = \{c_1, \ldots, c_{|C_i|}\} \subseteq S$. Via the described transformation and the hypothetical algorithm $\mathcal{A}$, we can compute the index $I_{\mathtt{SCP}} := I_{\mathcal{A}} \setminus \{k+1\}$ as the output corresponding to the SCP instance S. Obviously, the computation of $I_{\mathtt{SCP}}$ is possible in polynomial time using a polynomial algorithm for MOSS. To complete the proof, we still have to show (i) why $k+1 \in I_{\mathcal{A}}$ is always true, (ii) why $I_{\mathcal{A}} \setminus \{k+1\}$ is a correct output for our SCP instance, and (iii) why the computed index $I_{\mathcal{A}} \setminus \{k+1\}$ is minimum.

First, we will take a look at question (i), that is, why $\preceq_{k+1}$ is always needed to yield $\preceq_{\mathcal{F}}$ as the intersection of some $\preceq_i$. Because in $\preceq_{\mathcal{F}}$ no pair $\mathbf{x}, \mathbf{y} \in A$ with $\mathbf{x} \neq \mathbf{y}$ is comparable, for each pair $\mathbf{x}, \mathbf{y} \in A$, $\mathbf{x} \neq \mathbf{y}$, there has to be at least one $i \in I_{\mathcal{A}}$ where $\mathbf{x} \not\preceq_i \mathbf{y}$ and at least one $j \in I_{\mathcal{A}}$ with $\mathbf{y} \not\preceq_j \mathbf{x}$. Considering a pair $\mathbf{x}, \mathbf{y}$, for all $\preceq_i$ with $i \in \{1, \ldots, k\}$, $\mathbf{x} \preceq_i \mathbf{y}$ holds. By construction, only $\mathbf{x} \not\preceq_{k+1} \mathbf{y}$. Consequently, $\preceq_{k+1}$ is always needed, to construct $\preceq_{\mathcal{F}}$ as the intersection of single $\preceq_i$'s. Now we show (ii) why $I := I_{\mathcal{A}} \setminus \{k+1\}$ is always a correct output for the given SCP instance. As we have seen before, $k+1 \in I_{\mathcal{A}}$ and therefore, the intersection of the $\preceq_i$'s contains no pairs $(\mathbf{x}_\nu, \mathbf{x}_\mu)$ and $(\mathbf{x}'_\nu, \mathbf{x}'_\mu)$ with $1 \le \nu < \mu \le m$ and no pairs $(\mathbf{x}_\nu, \mathbf{x}'_\nu)$ with $1 \le \nu \le m$. The construction of the relations $\preceq_i$ with $i \in \{1, \ldots, k\}$ results in the absence of pairs $(\mathbf{x}_\nu, \mathbf{x}_\mu)$ and $(\mathbf{x}'_\nu, \mathbf{x}'_\mu)$ with $1 \leq \mu < \nu \leq m$ in the intersection if there will be at least one $i \in I_\mathcal{A}$ with $1 \leq i \leq k$. There only remains the possibility of pairs $(\mathbf{x}'_\nu, \mathbf{x}_\nu)$ with $1 \leq \nu \leq m$ in the intersection. To avoid this, for each $\nu \in \{1, \ldots, m\}$ there must be at least one $i \in \{1, \ldots, k\}$ in $I_\mathcal{A}$ with $\mathbf{x}'_\nu \not\preceq_i \mathbf{x}_\nu$. By construction of the Turing transformation, this can only occur if $c_\nu \in C_i$. Thus, $\bigcup_{i \in I_\mathcal{A} \setminus \{k+1\}} C_i = \{1, \ldots, m\} = S$. Last, we have to show (iii) why the computed index $I_\mathcal{A} \setminus \{k+1\}$ is a minimum index for SCP. Assume that $I_\mathcal{A} \setminus \{k+1\}$ is not a minimum index for SCP, that is, there is a smaller index J with $|J| < |I|$ and $\bigcup_{j \in J} C_j = S$. As one can easily see from the above transformation, $J \cup \{k+1\}$ would be a smaller index for MOSS than $I_\mathcal{A}$. □

Appendix B Correctness and Running Time Proof of the Greedy Algorithm on δ-MOSS

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: We prove the theorem with the help of the Lemmata 1 and 2. If we show that the invariant

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

holds during each step of Algorithm 2, the theorem is proved, due to Lemma 2 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 $(\mathbf{x}, \mathbf{y}) \in A \times A$ if Algorithm 2 terminates, that is, if $R = \emptyset$. We prove the invariant with induction over $|\overline{R}|$.

Induction basis: When the algorithm starts, $R = (A \times A) \setminus \preceq_\mathcal{F}$, that is, $\overline{R} = \preceq_\mathcal{F}$. For each $(\mathbf{x}, \mathbf{y}) \in \overline{R} = \preceq_\mathcal{F}$ with $\mathbf{x} \preceq_{\mathcal{F}'} \mathbf{y}$, that is, $\mathbf{x} \preceq_\emptyset \mathbf{y}$ with $\preceq_\emptyset := A \times A$, $\mathbf{x} \preceq_\mathcal{F} \mathbf{y}$ 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, that is, $\overline{R}$ contains more and more pairs $(\mathbf{x}, \mathbf{y}) \in A \times A$. 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 a function f_i is inserted into $\mathcal{F}'$ 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 $\mathcal{F}'' \supseteq \mathcal{F}'$ 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 $(\mathbf{x}, \mathbf{y}) \in A \times A$ 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, $(\mathbf{x}, \mathbf{y}) \notin \cap_{i \in (\mathcal{F}'\cup\{i^*\})} \preceq_i = \preceq_{\mathcal{F}'}$ 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 $(\mathbf{x}, \mathbf{y}) \in \preceq_{\mathcal{F}'\cup\{i^*\}}$, that is, 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 $R = \emptyset$, that is, after at most $|X \times X| = O(m^2)$ cycles of the while loop, if in each cycle $|R|$ decreases by at least one—which is true due to Theorem 1. The total running time of Algorithm 2 is, therefore, $O(\min\{k, m^2\} \cdot k^2 \cdot m^2) = O(\min\{k^3 \cdot m^2, k^2 \cdot m^4\})$. □

LEMMA 1: *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 $\delta \geq 0$ the relation $\preceq_{\mathcal{F}}$ is always a subset of or equal to $\preceq^{\delta}_{\mathcal{F}'}$, because $\mathbf{x} \preceq_{\mathcal{F}} \mathbf{y}$ implies that $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$ for all $f_i \in \mathcal{F}'$ and also $f_i(\mathbf{x}) - \delta \leq f_i(\mathbf{y})$ holds for all $f_i \in \mathcal{F}'$, that is, $\mathbf{x} \preceq^{\delta}_{\mathcal{F}'} \mathbf{y}$ for all $\mathbf{x}, \mathbf{y} \in X$. Thus, $\mathcal{F}'$ is δ-nonconflicting with $\mathcal{F}$ iff $\preceq_{\mathcal{F}'} \subseteq \preceq^{\delta}_{\mathcal{F}} \wedge \preceq_{\mathcal{F}} \subseteq \preceq^{\delta}_{\mathcal{F}'}$, that is, iff $\preceq_{\mathcal{F}'} \subseteq \preceq^{\delta}_{\mathcal{F}}$. □

LEMMA 2: *Let $\mathcal{F}' \subseteq \mathcal{F}$ and $\delta > 0$. Then*

$$\left(\forall \mathbf{x}, \mathbf{y} \in A : \mathbf{x} \preceq_{\mathcal{F}'} \mathbf{y} \Longleftrightarrow \mathbf{x} \preceq^{0,\delta}_{\mathcal{F}', \mathcal{F} \setminus \mathcal{F}'} \mathbf{y}\right) \Longrightarrow \mathcal{F}' \textit{ is } \delta\textit{-nonconflicting with } \mathcal{F} \textit{ w.r.t. } A.$$

PROOF: Let $\mathcal{F}' \subseteq \mathcal{F}$, $\delta > 0$, and $(\forall \mathbf{x}, \mathbf{y} \in A : \mathbf{x} \preceq_{\mathcal{F}'} \mathbf{y} \Longleftrightarrow \mathbf{x} \preceq^{0,\delta}_{\mathcal{F}', \mathcal{F} \setminus \mathcal{F}'} \mathbf{y})$, denoted by $(*)$. We observe the following two statements:

- Let $\delta_1, \delta_2, \delta_1', \delta_2' \in \mathbb{R}$ with $\delta_1 \leq \delta_1'$ and $\delta_2 \leq \delta_2'$, and $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_1', \mathcal{F}_2'$ be objective sets with $\mathcal{F}_1' \subseteq \mathcal{F}_1$ and $\mathcal{F}_2' \subseteq \mathcal{F}_2$. Then both $\preceq^{\delta_1,\delta_2}_{\mathcal{F}_1,\mathcal{F}_2} \subseteq \preceq^{\delta_1',\delta_2'}_{\mathcal{F}_1,\mathcal{F}_2}$ and $\preceq^{\delta_1,\delta_2}_{\mathcal{F}_1,\mathcal{F}_2} \subseteq \preceq^{\delta_1,\delta_2}_{\mathcal{F}_1',\mathcal{F}_2'}$ holds.
- Furthermore, $\preceq^{\delta_1,\delta_2}_{\mathcal{F}_1,\mathcal{F}_2} = \preceq^{\delta_1}_{\mathcal{F}_1} \cap \preceq^{\delta_2}_{\mathcal{F}_2}$ and $\preceq^{\delta,\delta}_{\mathcal{F}_1,\mathcal{F}_2} = \preceq^{\delta}_{\mathcal{F}_1 \cup \mathcal{F}_2}$.

With these observations, $\preceq_{\mathcal{F}'} \stackrel{(*)}{=} \preceq^{0,\delta}_{\mathcal{F}', \mathcal{F} \setminus \mathcal{F}'} = (\preceq^{0}_{\mathcal{F}'} \cap \preceq^{\delta}_{\mathcal{F} \setminus \mathcal{F}'}) \subseteq \preceq^{\delta}_{\mathcal{F}'} \cap \preceq^{\delta}_{\mathcal{F} \setminus \mathcal{F}'} = \preceq^{\delta}_{\mathcal{F}}$, that is, $\mathcal{F}'$ is δ-nonconflicting with $\mathcal{F}$ according to Lemma 1. □

Appendix C Correctness and Running Time Proof for the Exact Algorithm

THEOREM 3: *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 3. 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 3, the correctness of Algorithm 1 follows directly from 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 11: *Let $\mathcal{F}' \subseteq \mathcal{F}$ and $M \subseteq A \times A$. Then $\delta(\mathcal{F}', M) := \delta_{\min}(\mathcal{F}', \mathcal{F})$ w.r.t. all solution pairs $(\mathbf{x}, \mathbf{y}) \in M$.*

LEMMA 3: *Given an instance of the* δ-MOSS *or the* k-EMOSS *problem, let $\mathcal{F}_1 \subseteq \mathcal{F}$, $\mathcal{F}_1 \neq \emptyset$, an arbitrary objective set and*

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

Then there always exists a $(\mathcal{F}_2 \subseteq \mathcal{F}_1, \delta_2) \in S_M$, such that the following six statements hold.

(i) $\delta(\mathcal{F}_2, M) = \delta_2$

(ii) $\delta(\mathcal{F}_1, M) = \delta_2$

(iii) $\nexists(\mathcal{F}_3, \delta_3) \in S_M : \mathcal{F}_3 \subset \mathcal{F}_1 \wedge \delta_3 \leq \delta_2$

(iv) $\nexists(\mathcal{F}_3, \delta_3) \in S_M : \mathcal{F}_3 \subseteq \mathcal{F}_1 \wedge \delta_3 < \delta_2$

(v) $\nexists(\mathcal{F}_3, \delta_3) \in S_M : \mathcal{F}_3 \supset \mathcal{F}_1 \wedge \delta_3 \geq \delta_2$

(vi) $\nexists(\mathcal{F}_3, \delta_3) \in S_M : \mathcal{F}_3 \supseteq \mathcal{F}_1 \wedge \delta_3 > \delta_2$

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

(a) $\mathbf{x}$ and $\mathbf{y}$ are indifferent: 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 ⊔, Algorithm 1 computes $S_{\{(\mathbf{x},\mathbf{y})\}} = \{(\{i\}, 0) \mid 1 \leq i \leq k\}$ 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 $\forall i \in \mathcal{F}_< : \mathbf{x} \preceq_i \mathbf{y} \wedge \neg(\mathbf{y} \preceq_i \mathbf{x})$, that is, $\forall i \in \mathcal{F}_= : f_i(\mathbf{x}) = f_i(\mathbf{y})$ 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 $1 \leq i \leq k$

$$\text{and } \delta_i := \begin{cases} 0 & \text{if} \quad i \in \mathcal{F}_< \\ \delta & \text{if} \quad i \in \mathcal{F}_= \end{cases}.$$

Thus (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 $\mathcal{F}' \nsubseteq \mathcal{F}_= (\Rightarrow \delta' = 0)$.

(c) $\mathbf{x}$ and $\mathbf{y}$ are incomparable: We can divide $\mathcal{F}$ into three well-defined disjoint sets $\mathcal{F}_<$, $\mathcal{F}_>$, and $\mathcal{F}_=$ with $\mathcal{F}_< \cup \mathcal{F}_> \cup \mathcal{F}_= = \mathcal{F}$, $\mathcal{F}_< \neq \emptyset$, $\mathcal{F}_> \neq \emptyset$, $\forall i \in \mathcal{F}_< : f_i(\mathbf{x}) <$

$f_i(\mathbf{y}), \forall i \in \mathcal{F}_> : f_i(\mathbf{x}) > f_i(\mathbf{y})$, and $\forall i \in \mathcal{F}_= : f_i(\mathbf{x}) = f_i(\mathbf{y})$. For all singletons $\{i\}$ with $1 \le i \le k$, $\delta_i := \delta(\{i\}, \{(\mathbf{x}, \mathbf{y})\}) > 0$ holds, that is, $(\{i\}, \delta_i) \in S_{\{(\mathbf{x},\mathbf{y})\}}$ 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 $(\mathcal{F}_2 \subseteq \mathcal{F}', \delta')$ 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 $\mathcal{F}_1 \subseteq \mathcal{F}$ be an arbitrary objective set with error $\delta(\mathcal{F}_1, M \cup \{(\mathbf{x}, \mathbf{y})\})$. Assume that (i)–(vi) holds for M and $\{(\mathbf{x}, \mathbf{y})\}$. Thus, $\exists(\mathcal{F}_M, \delta_M) \in S_M$ with $\mathcal{F}_M \subseteq \mathcal{F}_1$ and (i)–(vi) and $\exists(\mathcal{F}_{\mathbf{xy}}, \delta_{\mathbf{xy}}) \in S_{\{(\mathbf{x},\mathbf{y})\}}$ with $\mathcal{F}_{\mathbf{xy}} \subseteq \mathcal{F}_1$ and (i)–(vi). To show that an $(\mathcal{F}_2 \subseteq \mathcal{F}_1, \delta_2)$ exists in $S_{M\cup\{(\mathbf{x},\mathbf{y})\}} := S_M \sqcup S_{\{(\mathbf{x},\mathbf{y})\}}$ that fulfills (i) and (ii), we define $\mathcal{F}_2 := \mathcal{F}_M \cup \mathcal{F}_{\mathbf{xy}} \subseteq \mathcal{F}_1$ and $\delta_2 := \max\{\delta_M, \delta_{\mathbf{xy}}\}$. Because $\delta(\mathcal{F}_M, M) = \delta(\mathcal{F}_1, M)$, $\delta(\mathcal{F}_M, M) = \delta(\mathcal{G}, M)$ holds for any $\mathcal{F}_M \subseteq \mathcal{G} \subseteq \mathcal{F}_1$ and because of $\delta(\mathcal{F}_{\mathbf{xy}}, \{(\mathbf{x}, \mathbf{y})\}) = \delta(\mathcal{F}_1, \{(\mathbf{x}, \mathbf{y})\})$, $\delta(\mathcal{F}_{\mathbf{xy}}, \{(\mathbf{x}, \mathbf{y})\}) = \delta(\mathcal{H}, \{(\mathbf{x}, \mathbf{y})\})$ holds for any $\mathcal{F}_{\mathbf{xy}} \subseteq \mathcal{H} \subseteq \mathcal{F}_1$. Together with $\mathcal{F}_M \cup \mathcal{F}_{\mathbf{xy}} \subseteq \mathcal{F}_1$, this yields $\delta(\mathcal{F}_M \cup \mathcal{F}_{\mathbf{xy}}, M) = \delta(\mathcal{F}_1, M)$ as well as $\delta(\mathcal{F}_M \cup \mathcal{F}_{\mathbf{xy}}, \{(\mathbf{x}, \mathbf{y})\}) = \delta(\mathcal{F}_1, M)$. This follows (i) and (ii):

$$\begin{aligned} \delta_2 &= \max\{\delta(\mathcal{F}_M \cup \mathcal{F}_{\mathbf{xy}}, M), \delta(\mathcal{F}_M \cup \mathcal{F}_{\mathbf{xy}}, \{(\mathbf{x}, \mathbf{y})\})\} \\ &= \delta(\mathcal{F}_M \cup \mathcal{F}_{\mathbf{xy}}, M \cup \{(\mathbf{x}, \mathbf{y})\}) && \text{(i)} \\ &= \max\{\delta(\mathcal{F}_1, M), \delta(\mathcal{F}_1, \{(\mathbf{x}, \mathbf{y})\})\} = \delta(\mathcal{F}_1, M \cup \{(\mathbf{x}, \mathbf{y})\}) && \text{(ii)} \end{aligned}$$

□