BERGISCHE
UNIVERSITÄT
WUPPERTAL

Adaptive Parametric Scalarizations in Multicriteria Optimization

Dissertation

zur Erlangung des Doktorgrades (Dr. rer. nat.)

Fachbereich C – Mathematik und Naturwissenschaften

Bergische Universität Wuppertal

vorgelegt von

Kerstin Dächert

aus Fürth

Wuppertal, Januar 2014

Die Dissertation kann wie folgt zitiert werden:

urn:nbn:de:hbz:468-20140716-112600-4
[http://nbn-resolving.de/urn/resolver.pl?urn=urn%3Anbn%3Ade%3Ahbz%3A468-20140716-112600-4]

Erstgutachterin : Prof. Dr. Kathrin Klamroth
Zweitgutachterin : Prof. Dr. Margaret M. Wiecek

Tag der mündlichen Prüfung: 25. April 2014

Contents

1 Introduction

1.1 Multiobjective Optimization

Over one hundred years ago, Francis Edgeworth (1845–1926) and Vilfredo Pareto (1848–1923) laid the foundations of what is today called multicriteria decision making. The basic assumption of multicriteria decision making is that whenever a decision has to be taken, not only one but multiple objectives have to be taken into account. Moreover, in general, these objectives are competing, i.e., no solution or decision action exists for which all objectives can be met best simultaneously. An example is given by different products in a market. Since, in general, a cheap product has a rather bad quality while a product of good quality is rather expensive, a compromise between the objectives ‘price’ and ‘quality’ has to be found. We can only exclude products from consideration that are at least as expensive and, simultaneously, of at most the same quality as some other product. This is the basic idea of dominance and nondominance in multiobjective optimization: Of interest are exactly those solutions (products) that cannot be improved with respect to one criterion without being impaired with respect to at least one other criterion. In the literature, these solutions are called ‘nondominated’, ‘efficient’ or, in honor to the fathers of multicriteria decision making, ‘Pareto’ or sometimes also ‘Edgeworth-Pareto optimal’, see Section 2.1 for a precise definition. Due to the conflicting nature of the objectives, there is, in general, not only one but a set of Pareto optimal solutions.

As already suggested by the titles of the early publications *Mathematical psychics* (Edgeworth, 1881) and *Cours d’Economie Politique* (Pareto, 1896), multicriteria decision making is an interdisciplinary field that, from the very beginning up to today, attracts researchers and practitioners from various disciplines as economics, psychology, mathematics and computer as well as engineering science. Thereby, the interests range from the theoretical analysis of multiobjective optimization problems over the practical computation and representation of solutions up to economical utility theory and questions of human behavior in decision making. In brief, this thesis presents new theoretical results for generating Pareto optimal solutions and shows the practical usefulness of the new theory.

1.2 Outline of This Thesis

The content of this thesis is organized in two parts and nine chapters. The first three chapters present the basics. The fourth and fifth chapter, which build Part I, contain new theoretical results. Their practical application is demonstrated in Part II, which consists of chapters six to eight. The last chapter summarizes the results of this thesis. In what follows we describe the content of each chapter in more detail.

Chapter 2 assembles the relevant definitions, notions and concepts from the literature that are needed in the following. First, we provide general definitions from the field of multicriteria optimization. After that we introduce the notions of representations and approximations of the nondominated set and indicate quality criteria from the literature. Then scalarizations as a well known concept to solve multicriteria optimization problems are presented. Finally, the idea of a parametric algorithm that consists in the iterative solution of scalarizations with varying parameters is specified and the notions of a priori and adaptive (a posteriori) parameter schemes are introduced.

Chapter 3 provides a detailed literature review on methods using (adaptive) parametric algorithms. The survey on this topic starts with early publications dating from the sixties of the last century and ends with very recent publications. As several methods are solely applicable to the bicriteria case, we organize the literature review into two sections, one devoted particularly to the bicriteria and the other one to the general multicriteria case.

After the introduction, the preparation of the basics and the presentation of related literature, new theoretical results are presented in **Part I**. In brief, Chapter 4 deals with new adaptive parameter schemes for well-known scalarization methods with augmentation terms, particularly the augmented weighted Tchebycheff method. Chapter 5 is concerned with the general framework of a new parametric algorithm.

In Chapter 4 we derive an adaptive parameter scheme for the augmented weighted Tchebycheff method which is the first classic scalarization for which an augmentation term has been introduced. So far, only the weights have been controlled in an adaptive way but the augmentation parameter has been chosen fixed to a small positive constant. As reported in the literature, on the one hand, numerical issues arise when this constant is too small and, on the other hand, nondominated points are missed when the constant is chosen too large. We construct all parameters of the augmented weighted Tchebycheff method in an adaptive way such that every nondominated point of a discrete multicriteria optimization problem can be generated and, at the same time, the augmentation parameter can be chosen as large as possible up to a feasible upper bound. Besides the classic augmented weighted Tchebycheff method we consider a generalized problem formulation that contains an augmentation parameter for each objective and, thus, provides more flexibility. The generalized formulation is particularly useful for the application to continuous problems, as it allows to incorporate a given trade-off among the objectives. For bicriteria problems it is well known that a prescribed two-sided trade-off can be translated into suitable parameters of a generalized augmented weighted Tchebycheff problem. We improve existing approaches by proposing an adaptive parameter scheme that takes all parameters, i.e., also the weights, into account. Finally, augmented variants of the ε-constraint method from the literature are discussed. We show that the augmentation parameter of an augmented ε-constraint scalarization can be determined in the same way as it is proposed for the augmented weighted Tchebycheff method.

In Chapter 5 we develop the general framework of an adaptive parametric algorithm that is based on a systematic decomposition of the search region, i.e., the region potentially containing further nondominated points. We particularly study the number of subproblems that have to be solved to generate complete representations for discrete problems. In the literature, the best known upper bound on the number of subproblems in the tricriteria case depends quadratically on the number of nondominated points. By indicating a new parametric algorithm in which at most three subproblems are solved per nondominated point, we improve the former quadratic to a linear upper bound. Thereby, the main key is a new decomposition criterion which avoids redundancy. The parametric algorithm can be applied with any scalarization that is suited for non-convex or discrete problems. If the ε-constraint method is used, we can reduce the upper bound further and show that at most two subproblems per nondominated point are sufficient to obtain a complete representation. Finally, we propose an extension of the new algorithm for any number of objectives.

The theoretical results of Part I are validated computationally in **Part II**. Thereby, the results of Part I are combined in the sense that the adaptive parameter scheme from Chapter 4 is employed for each subproblem that is solved in the parametric algorithm derived in Chapter 5.

In Chapter 6 we generate complete representations for discrete problems. In the bicriteria case the performance of different variants of Tchebycheff scalarizations is examined. Besides the validation of the adaptive parameter scheme proposed in Chapter 4 we compare the adaptive parameter scheme to the classic fixed choice of the augmentation parameter which is common in the literature. In particular, we show computationally that already for small instances of knapsack problems nondominated points are missed with the classic approach but not with the adaptive parameter selection. We also study further algorithmic variants with local reference points by which larger values for the augmentation parameter can be obtained. In the tricriteria case we validate the formulas for the parameters of the augmented weighted Tchebycheff method as well as the upper bound on the number of subproblems derived in Chapter 5. In all instances the complete nondominated set is computed reliably with the help of the adaptive parameter scheme. Moreover, the predicted upper bound on the number of subproblems is met exactly in all instances. Besides the validation of our new parametric algorithm, we also compare it with three state of the art methods for the generation of complete representations. Our algorithm clearly outperforms one of the three algorithms and can compete with the other two in the sense that no algorithm outperforms the other with respect to the number of subproblems solved and the required computational time.

In Chapter 7 we apply the new adaptive parametric algorithm to continuous problems, for which incomplete representations of the nondominated set are sought. We use common quality criteria to measure the quality of the representations. In order to refine the representations iteratively, we propose different selection rules based on the volume of the boxes into which the search region is decomposed and the contribution to the dominated hypervolume. Tests with bi- and tricriteria problems from the literature are performed. We compare different variants of Tchebycheff methods employing an adaptive parameter scheme with an a priori ε-constraint method. We observe that with the adaptive methods considerably less infeasible or redundant subproblems are generated than with the a priori method, in general. The adaptive approaches perform particularly well when the nadir point is not known and its estimate is rather bad. Hence, they are particularly useful for problems with more than two criteria.

Chapter 8 treats a real-world problem in which the multicriteria control of sewer networks is considered. Within a preliminary offline analysis we aim at constructing a discrete representation of the nondominated set. Since the single-criterion solver used for the subproblems is interrupted before its termination, it typically does not provide local or global minima but intermediate solutions. These solutions often correspond to dominated or even infeasible points. Therefore, we can only construct a very scarce discrete approximation of the nondominated set. Moreover, due to numerical issues, an a priori parameter scheme yields better results than an adaptive scheme in some test cases. This shows that the performance of the underlying single-criterion solver is crucial for the successful use of adaptive parameter schemes. If the generated points are not nondominated or close to nondominated points, an a priori parameter selection might be preferred.

Chapter 9 contains a summary of the results of this thesis. Ideas for future research are indicated directly at the end of each chapter of Parts I and II.

2 Preliminaries

In this chapter we collect the relevant notions, definitions and concepts that are used in this thesis. They are common knowledge and can be found in textbooks on multicriteria optimization, e.g., in Chankong and Haimes (1983), Steuer (1986), Miettinen (1999), Jahn (2004) or Ehrgott (2005).

2.1 Terminology and Definitions

We consider multiple criteria optimization problems

$$\min_{x \in X} f(x) = (f_1(x), \dots, f_m(x))^\top \tag{2.1}$$

with $m \geq 2$ *objective functions* $f_i : X \to \mathbb{R}$, $i = 1, \dots, m$, and with *feasible set* $X \subseteq \mathbb{R}^n$. We assume that the functions $f_i, i = 1, \dots, m$, are continuous and that X is non-empty and compact. If X is a discrete finite set, we call Problem (2.1) discrete. The image of the feasible set X is denoted by $Z := f(X) \subseteq \mathbb{R}^m$ and is called *set of feasible outcomes*.

To simplify notation, we will often refer to the points in Z without relating them back to their preimages in the feasible set. Consequently, we equivalently formulate Problem (2.1) in the outcome space as

$$\min_{z \in Z} z = (z_1, \dots, z_m)^\top. \tag{2.2}$$

For two vectors $z, \bar{z} \in Z$ we define

$$\begin{aligned}
z < \bar{z} \quad &:\Leftrightarrow \quad z_i < \bar{z}_i \quad \forall\, i = 1, \dots, m, \\
z \leq \bar{z} \quad &:\Leftrightarrow \quad z_i \leq \bar{z}_i \quad \forall\, i = 1, \dots, m \ \text{ and } \ \exists\, j \in \{1, \dots, m\} : z_j < \bar{z}_j, \\
z \leqq \bar{z} \quad &:\Leftrightarrow \quad z_i \leq \bar{z}_i \quad \forall\, i = 1, \dots, m.
\end{aligned} \tag{2.3}$$

The symbols $>$, $\geq$ and $\geqq$ are used accordingly. As there exists no canonical ordering on $\mathbb{R}^m$ for $m \geq 2$, a definition of optimality is required. We use the Pareto concept of optimality: A solution $\bar{x} \in X$ is called *Pareto optimal* or *efficient* if there does not exist a feasible solution $x \in X$ such that $f(x) \leq f(\bar{x})$. The corresponding objective vector $f(\bar{x}) \in \mathbb{R}^m$ is called *nondominated* in this case. If, on the other hand, $f(x) \leq f(\bar{x})$ for some feasible $x \in X$, we say that $f(x)$ *dominates* $f(\bar{x})$, and x *dominates* $\bar{x}$. If strict inequality holds for all m components, i.e., if $f(x) < f(\bar{x})$, then x *strictly dominates* $\bar{x}$. If there exists no feasible solution $x \in X$ that strictly dominates $\bar{x}$, then $\bar{x}$ is called *weakly Pareto optimal* or *weakly efficient.* We denote the set of efficient solutions of (2.1) by X_E and refer to it as the *efficient set*, i.e.,

$$X_E := \{x \in X : \nexists\, \tilde{x} \in X : f(\tilde{x}) \leq f(x)\}. \tag{2.4}$$

The image set of the set of efficient solutions is denoted by

$$Z_N := f(X_E) = \{z \in Z : \nexists\, \tilde{z} \in Z : \tilde{z} \leq z\} \tag{2.5}$$

and is called the *nondominated set* of problem (2.1). In general, one nondominated point $f(\bar{x})$ might have more than one preimage $\bar{x} \in X$. However, throughout this thesis, it is sufficient to know one efficient solution per nondominated point.

A point $\bar{x} \in X$ is called *properly efficient* according to Geoffrion (1968) if it is efficient and if there exists a scalar $M > 0$ such that for each $i = 1, \ldots, m$ and each $x \in X$ satisfying $f_i(x) < f_i(\bar{x})$ there exists an index $j \neq i$ with $f_j(x) > f_j(\bar{x})$ and

$$\frac{f_i(\bar{x}) - f_i(x)}{f_j(x) - f_j(\bar{x})} \leq M. \tag{2.6}$$

An efficient point that is not properly efficient is called *improperly efficient.* Note that if the outcome space Z is discrete and finite, every efficient point is properly efficient.

The notion of *trade-off* is closely related to the definition of proper efficiency. According to Chankong and Haimes (1983), for given $x, \bar{x} \in X$, the ratio of change $T_{ij}(x, \bar{x})$ involving objective functions f_i and f_j, $i, j = 1, \ldots, m$, $i \neq j$, is defined as

$$T_{ij}(x, \bar{x}) := \frac{f_i(x) - f_i(\bar{x})}{f_j(\bar{x}) - f_j(x)} \tag{2.7}$$

for $f_j(x) \neq f_j(\bar{x})$. Note that if $f_i(x) \neq f_i(\bar{x})$, then $T_{ij}(x, \bar{x}) = (T_{ji}(x, \bar{x}))^{-1}$ and $T_{ij}(x, \bar{x}) = T_{ij}(\bar{x}, x)$ hold. In Kaliszewski and Michalowski (1997), for $\bar{z} \in Z$ and a problem in maximization format, the trade-off $T_{ij}^G(\bar{z})$ involving objective functions z_i and z_j, $i, j = 1, \ldots, m$, $i \neq j$, is defined as

$$T_{ij}^G(\bar{z}) := \sup_{z \in Z_j^<(\bar{z})} \frac{z_i - \bar{z}_i}{\bar{z}_j - z_j}, \tag{2.8}$$

where $Z_j^<(\bar{z}) = \{z \in Z : z_j < \bar{z}_j, z_i \geq \bar{z}_i, i = 1, \ldots, m, i \neq j\}$. If $Z_j^<(\bar{z}) = \emptyset$, then $T_{ij}^G(\bar{z}) := \infty$ for all $i = 1, \ldots, m, i \neq j$.

Pareto optimality can also be defined geometrically with the help of *ordering cones*, which are defined as convex cones that characterize a partial ordering in a real linear space. For simplicity, we restrict the description here to the case of pointed ordering cones and use $\mathbb{R}^m$ as a partially ordered linear space. Then, given a nonempty subset $S \subset \mathbb{R}^m$ and a pointed ordering cone $C \subset \mathbb{R}^m$, an element $\bar{y} \in S$ is called *minimal element* (or C-minimal element) of S if

$$(\{\bar{y}\} - C) \cap S = \{\bar{y}\}, \tag{2.9}$$

see, e.g., Jahn (2004). Thereby, $\{\bar{y}\} - C := \{\bar{y} - c : c \in C\}$ denotes the algebraic difference. For $y, \bar{y} \in S$ we say that $\bar{y}$ dominates y if

$$y - \bar{y} \in C \backslash \{0\},$$

where 0 denotes the m-dimensional zero vector. We use the notation

$$y' \leq_C y \;:\Leftrightarrow\; y - y' \in C \tag{2.10}$$

for $y, y' \in \mathbb{R}^m$. If we choose $C := \mathbb{R}^m_+$ with $\mathbb{R}^m_+ := \{y \in \mathbb{R}^m : y_i \geq 0, i = 1, \ldots, m\}$, we obtain the notion of (non)dominance with respect to Pareto optimality. Therefore, the closed positive orthant $\mathbb{R}^m_+$ is sometimes called *Pareto cone*. The more general definition of minimal elements using ordering cones is typically used in, however, not limited to, the field of vector optimization. From this perspective multiobjective optimization can be seen as a special case of vector optimization. For details we refer to the monograph of Jahn (2004).

Nondominated points can be classified further. A nondominated point is called *supported* if it is contained in the set

$$F := \left\{ y \in \operatorname{conv}(Z_N) : (\{y\} - \mathbb{R}^m_+) \cap \operatorname{conv}(Z_N) = \{y\} \right\},$$

see, e.g., Ruzika (2007), where $\operatorname{conv}(S)$ denotes the convex hull of a set S. Otherwise, i.e., if a nondominated point is not contained in F, it is called *unsupported* or *nonsupported*. If the set of feasible outcomes Z of a given problem is $\mathbb{R}^m_+$-convex, then all nondominated points are supported. A set is called $\mathbb{R}^m_+$-convex if $Z + \mathbb{R}^m_+$ is convex (Ehrgott, 2005), where $Z + \mathbb{R}^m_+ := \{z + y : z \in Z, y \in \mathbb{R}^m_+\}$ denotes the algebraic sum of the two sets Z and $\mathbb{R}^m_+$. The breakpoints of the nondominated set of an $\mathbb{R}^m_+$-convex problem are called *extreme* nondominated points. The nondominated set of a discrete or non-convex problem typically contains supported and unsupported points.

An order of special interest is the lexicographic order. For two vectors $z, \bar{z} \in Z$ we define "$\leq_{\text{lex}}$" as

$$z \leq_{\text{lex}} \bar{z} \;:\Leftrightarrow\; z = \bar{z} \;\text{ or }\; z_i < \bar{z}_i \;\text{ for } i = \min\{j : z_j \neq \bar{z}_j, j = 1, \ldots, m\}.$$

Let $\pi = (\pi_1, \dots, \pi_m)$ be any permutation of $(1, \dots, m)$. Then $\bar{x} \in X$ is said to be *lexicographically optimal* with respect to π if there exists no $x \in X, x \neq \bar{x}$, such that $f_\pi(x) \leq_{\text{lex}} f_\pi(\bar{x})$, where $f_\pi(x) = (f_{\pi(1)}(x), \dots, f_{\pi(m)}(x))^\top$, see, e.g., Gorski (2010). It is immediately clear that every lexicographically optimal solution is also efficient.

Under the given assumption that Z is compact, lower and upper bounds on Z_N can be computed. A sharp lower bound on the nondominated set is given by the *ideal point* which we denote by z^I. The i-th component of the ideal point is defined as the minimum of the i-th objective, i.e.,

$$z_i^I := \min\{z_i \,:\, z \in Z\} \quad \forall\, i = 1, \dots, m. \tag{2.11}$$

In general, it holds that $z^I \notin Z$. If Z is replaced by a subset $S \subset Z$ in (2.11), then we call the resulting point a *local ideal point* with respect to S. A point z^U that strictly dominates z^I is called a *utopia point*. A sharp upper bound on the nondominated set is given by the *nadir point* z^N with components

$$z_i^N := \max\{z_i \,:\, z \in Z_N\} \quad \forall\, i = 1, \dots, m. \tag{2.12}$$

Note that in case of the nadir point the maximum over the nondominated set is built which is a complicated task in general. However, for bicriteria problems the nadir point can be easily determined in the following way. The two lexicographically optimal points are computed with respect to $\pi = (1, 2)$ and $\pi = (2, 1)$ by solving first $z_i^I := \min\{z_i \,:\, z \in Z\}$ for each $i = 1, 2$ and then, again for each $i = 1, 2$, $z_j^* := \min\{z_j \,:\, z_i \leq z_i^I, z \in Z\}$ with $j \in \{1, 2\} \backslash \{i\}$. The nadir point then equals z^*. For more than two objectives the nadir point can no longer be computed with the help of all lexicographically optimal points as the following example of Szczepanski and Wierzbicki (2003) shows. The considered tricriteria linear optimization problem, transformed into minimization format, reads

$$\begin{array}{rrcl}
\min\limits_{x \in \mathbb{R}^3} & -\begin{pmatrix} 100 - 7x_1 - 20x_2 - 9x_3 \\ 4x_1 + 5x_2 + 3x_3 \\ x_3 \end{pmatrix} & & \\
\text{s.t.} & \frac{3}{2}x_1 + x_2 + \frac{8}{5}x_3 & \leq & 9, \\
& x_1 + 2x_2 + x_3 & \leq & 10, \\
& x_i & \geq & 0, \quad i = 1, 2, 3.
\end{array} \tag{2.13}$$

The nondominated set of Problem (2.13) consists of two faces, which are defined by the extreme points $z^1 = -(100, 0, 0)^\top$, $z^2 = -(58, 24, 0)^\top$, $z^3 = -(49\frac{3}{8}, 16\frac{7}{8}, 5\frac{5}{8})^\top$, $z^4 = -(12, 31, 0)^\top$ and $z^5 = -(3\frac{7}{11}, 26\frac{9}{11}, 3\frac{7}{11})^\top$, see Figure 2.1. The points defining $z_i^I, i = 1, 2, 3$, are z^1, z^4 and z^3, respectively. The local nadir point with respect to

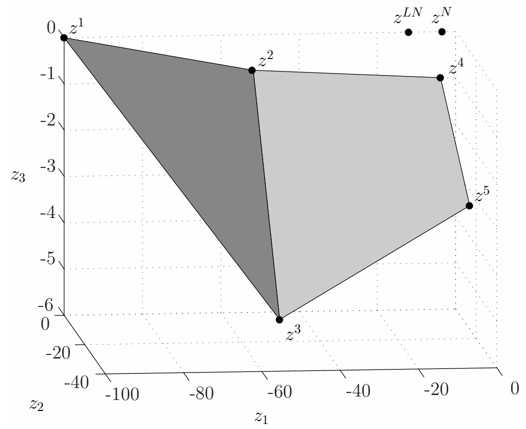


Figure 2.1: Example of Szczepanski and Wierzbicki (2003): The local nadir point z^{LN}, computed from the three lexicographically optimal points z^1, z^3 and z^4, underestimates the true nadir point z^N, defined by z^1, z^3 and z^5.

these three points is $-(12, 0, 0)^\top$. However, the true nadir point is defined by z^1, z^3 and z^5 and equals $z^N = -(3\frac{7}{11}, 0, 0)^\top$. This simple example shows that for $m \geq 3$ the local nadir point defined by the lexicographically optimal points is only a lower bound on the nadir point, in general.

An estimate on the nadir point that may under- or overestimate the true nadir point is computed with the help of the well-known *payoff-table*, see, e.g., Isermann and Steuer (1987). Each objective is individually minimized and the resulting outcome is inserted as corresponding column of an $(m \times m)$-matrix. While the diagonal of the payoff-table contains the components of the ideal point, an estimate on the nadir point is obtained by computing the maximum entry of each row and combining these maxima to a corresponding vector. Note that the resulting vector might overestimate the true nadir point if some of the individual minima correspond to weakly nondominated points.

A guaranteed upper bound on Z_N is given by

$$z_i^M := \max\{z_i \,:\, z \in Z\} + \delta \quad \forall\, i = 1, \ldots, m \tag{2.14}$$

with $\delta > 0$. We will use this vector of individual maxima whenever the nadir point is not available.

2.2 Representation and Approximation of the Nondominated Set

We call a finite set of nondominated points *(discrete) representation*, *representative subset* or *representative system* in the following, see, e.g., Armann (1989), Sayın (2000), Ruzika (2007) or the survey of Faulkenberg and Wiecek (2010).

Definition 2.1 (Representation)**.** *A finite set $\mathcal{R} \subseteq Z_N$ is called a (discrete) representation of Z_N. If $\mathcal{R} = Z_N$, we call $\mathcal{R}$ a complete representation, otherwise an incomplete representation.*

The notion of an *approximation* of the nondominated set is more general: An approximation might contain points that are not nondominated. We borrow the definition of Hansen and Jaszkiewicz (1998) but modify it slightly to our purpose as we do not assume $\mathcal{A}$ to be a finite subset of $f(X)$ and use the Pareto cone $C = \mathbb{R}^m_+$.

Definition 2.2 (Approximation)**.** *A set $\mathcal{A} \subseteq \mathbb{R}^m$ is called an approximation of the set Z_N if for all points $z^1, z^2 \in \mathcal{A}$, $z^1 \neq z^2$ it holds that*

$$z^1 \nleq z^2 \quad \text{and} \quad z^2 \nleq z^1,$$

i.e., if no point in $\mathcal{A}$ is dominated by any other point in $\mathcal{A}$.

Note that, according to Definitions 2.1 and 2.2, a representation is also an approximation. Nevertheless, we will use the term approximation in the following only when the set of computed points contains at least one point that is not part of Z_N or if we cannot guarantee that all points are nondominated. In the literature, there is no unifying notion and often the term approximation is used for both, i.e., a set of approximating or representing points. An approximation might be constructed from a discrete representation whose points are connected, e.g., by piecewise linear functions. However, an approximation might also consist of a set of discrete points which are not nondominated themselves but close to nondominated points. We refer to the survey paper of Ruzika and Wiecek (2005) for more details.

From a practical perspective incomplete representations and approximations are important for several reasons. If the nondominated set is not finite and no method is known to find the nondominated set by solving a finite number of auxiliary problems (as is, e.g., possible for linear bicriteria problems), then we are naturally limited to generate only an incomplete representation or an approximation. But even if the nondominated set is finite, it may still grow exponentially with the problem size and, hence, be intractable (Ehrgott and Gandibleux, 2000). This means that, on the one hand, the computational effort of generating a complete representation is too high, while, on the other hand, no sense is seen in presenting an exponentially large set of points to a decision maker.

From a theoretical perspective it is typically of interest whether a method is able to find a complete representation for finite problems, even if the latter is not generated in practice. Thereby, the generation of a complete representation can also be seen as a sort of validation of a solution method.

Quality Criteria

In Definitions 2.1 and 2.2 no quality criteria are given. However, in order to have meaningful substitutes of the nondominated set, indicators measuring the quality of a representation or approximation have been introduced. In Sayın (2000) the three criteria *coverage, uniformity and cardinality* are proposed as quality measures for representations. The coverage error of a representation $\mathcal{R}$ is defined by

$$d_C(\mathcal{R}, Z_N) := \sup_{z \in Z_N} \min_{y \in \mathcal{R}} d(z, y), \tag{2.15}$$

with d being a norm. Coverage is a measure for the worst represented nondominated point. Note that (2.15) corresponds to the Hausdorff metric for the two sets Z_N and $\mathcal{R}$

$$d_H(\mathcal{R}, Z_N) := \max \left\{ \sup_{y \in \mathcal{R}} \inf_{z \in Z_N} d(z, y), \sup_{z \in Z_N} \inf_{y \in \mathcal{R}} d(z, y) \right\} = \sup_{z \in Z_N} \inf_{y \in \mathcal{R}} d(z, y), \tag{2.16}$$

as $\mathcal{R} \subseteq Z_N$. The uniformity level is defined by

$$d_U(\mathcal{R}) := \min_{z, y \in \mathcal{R}, z \neq y} d(z, y) \tag{2.17}$$

and measures the distance between the closest representing points. Lastly, cardinality is given by the number of representing points, hence, it equals $|\mathcal{R}|$. In order to obtain a 'good' representation, the coverage error is minimized, the uniformity level is maximized and the cardinality is minimized. As pointed out, e.g., by Eichfelder (2006) and Ruzika (2007), these three criteria are competing, i.e., the generation of a representation that meets all criteria is again a multiobjective problem. In particular, minimizing the coverage error on the one hand and maximizing uniformity and minimizing cardinality on the other hand are contrary, in general. A good compromise would be a representation that contains sufficiently many but not too many evenly distributed points which cover all parts of the nondominated set 'sufficiently well'.

In general, it is difficult to compute the coverage error as the nondominated set is typically not available (Armann, 1989). Therefore, Eichfelder (2006) proposes to replace (2.15) by

$$\max_{j \in \{1,\dots,R\}} \max_{z \in \mathcal{N}(z^j)} d(z^j, z), \tag{2.18}$$

where $\mathcal{R} = \{z^1, \dots, z^R\}$ is the representation and $\mathcal{N}(z^j), j = 1, \dots, R$, denotes the set of representatives that are neighbors of $z^j \in \mathcal{R}$. Thereby, it is assumed that the representatives cover the entire nondominated set sufficiently well. However, a difficulty consists in the determination of neighbored representatives. While this task is easy in the bicriteria case, it is not evident for three or more objectives. Another problem that is also mentioned in Eichfelder (2006), is that for non-convex problems the nondominated set is not connected, in general. This must be respected in the determination of neighboring points, too.

A further quality measure, introduced by Zitzler and Thiele (1998), is the so-called *hypervolume indicator*. It was originally intended for (and is mainly used in) the context of evolutionary algorithms. The (dominated) hypervolume denotes the m-dimensional volume that is dominated by a given set of nondominated points. Therefore, typically a reference point $r \in \mathbb{R}^m$ is chosen. Then, the dominated hypervolume with respect to a finite set of nondominated points $\mathcal{R}$ is given by the set

$$\bigcup_{z \in \mathcal{R}} \{y \in \mathbb{R}^m : z_i \leq y_i \leq r_i \; \forall \, i = 1, \dots, m\}. \tag{2.19}$$

If available, the nadir point can be chosen as reference point. For further quality criteria for representations we refer to Faulkenberg and Wiecek (2010).

If a continuous (e.g., a piecewise linear) approximation is sought, the criteria uniformity and cardinality are not meaningful, but coverage is still a useful criterion. In the literature, the notions of an *inner* and *outer approximation* are used, where — roughly spoken — an inner approximation contains feasible points, while an outer approximation contains mostly infeasible points. If both approximations are constructed in parallel, the notion *sandwich approximation* is common as the nondominated set is bounded from below and above. In this case the Hausdorff distance between these two sets serves as quality criterion. In this thesis we only focus on representations. Therefore, we omit a detailed discussion of quality criteria for approximations and refer to Ruzika and Wiecek (2005) for further information.

2.3 Scalarization Methods

A common technique to solve problems of the form (2.1) is to replace the original multiple objective problem by one or a series of parametric single objective problems with a scalar-valued objective function, a so-called *scalarization* (see, e.g., Ehrgott, 2005; Miettinen, 1999). Then well-known (single criterion) methods can be applied to solve the problem. A variety of different scalarization methods exists.

However, scalarizations are not the only concept available to solve multiobjective optimization problems. A different, very common methodology is *evolutionary multiobjective optimization* (EMO), see, e.g., Deb (2001) for an overview. Thereby, not only one but a set of initial solutions is created. A so-called fitness, that is related to the values of the objective functions evaluated for the respective solution, is assigned to each solution. By specific mechanisms which mimic a genetic process, the fitness of the maintained solutions is improved so that the population tends to efficient solutions. Evolutionary methods can be easily applied to multicriteria optimization problems, as the method deals with a set of points at every step of the algorithm. Therefore, typically a good diversity among the solutions can be achieved. However, as the solution process is stochastic, there is no guarantee to obtain efficient solutions. Nevertheless, EMO is very popular and widely used, especially in engineering applications.

A second non-scalarizing methodology consists in the generalization of the theory of single objective optimization to the multiobjective case. This results in multicriteria variants of the steepest descent or Newton method. We refer to Fliege and Svaiter (2000) and Fliege et al. (2009) for details.

In this thesis we focus on scalarization methods for solving multicriteria optimization problems. In the following we present a selection of well-known scalarization approaches from the literature and state their theoretical properties. Of particular importance is the question whether the outcomes generated by a specific method always correspond to nondominated points of (2.1) and whether all nondominated points of (2.1) can be generated by appropriately varying the involved parameters of the respective scalarization. Besides, also the structure of the scalarization plays a role, as it is typically correlated to the effort needed to solve the problem computationally.

The Weighted Sum Method

Probably the most widely used scalarization is the *weighted sum method* where a convex combination of the objective functions is built. In Gass and Saaty (1955) the weighted sum is introduced as 'the parametric function' for linear programming problems with two objectives. In general, the weighted sum problem is formulated as

$$\min_{x \in X} \sum_{i=1}^{m} \lambda_i f_i(x), \tag{2.20}$$

where $\lambda \in \mathbb{R}^m_+$, i.e., $\lambda_i \geq 0$ for all $i = 1, \ldots, m$, and $\sum_{i=1}^m \lambda_i = 1$. The parameters λ_i, $i = 1, \ldots, m$, are usually called weights since they indicate which relative importance the objective f_i, to which λ_i is associated, has with respect to the other objectives. One important advantage of the weighted sum approach is that the constraints of the problem do not change with respect to the underlying multicriteria problem, i.e., that the feasible set of (2.20) is the same as of (2.1). This implies that the scalarized problem does not become more difficult to solve than its multicriteria counterpart. This property is particularly important when a combinatorial problem is considered whose constraint set has a specific structure.

It is well-known (Geoffrion, 1968) that for $\lambda \in \mathbb{R}^m_>$, i.e., $\lambda_i > 0$ for all $i = 1, \ldots, m$, every solution of (2.20) is properly efficient. If $\lambda_i = 0$ for at least one $i = 1, \ldots, m$, then a solution of (2.20) is weakly efficient, and efficient if the solution is unique. Conversely, every supported efficient solution can be obtained as a solution of (2.20) with suitable weights. This implies that for convex problems every properly efficient solution can be obtained for some $\lambda \in \mathbb{R}^m_>$ and every weakly efficient solution for some $\lambda \in \mathbb{R}^m_+$. However, no unsupported efficient solution can be generated by a weighted sum regardless of the choice of the weights, which represents the main drawback of this method. Nevertheless, because of its simple construction and structure-preserving nature, it is frequently used in practice. Besides, since all supported nondominated points can be computed by a weighted sum, the method is often used for that purpose, see, e.g., the two phase method of Ulungu and Teghem (1995).

The ε-Constraint Method

The *ε-constraint method* was first proposed in Haimes et al. (1971) and is discussed in more detail in Chankong and Haimes (1983). In this method one of the objectives f_i with $i \in \{1, \ldots, m\}$ is selected and minimized whereas bounds are imposed on all other objectives, which yields

$$\begin{aligned} \min \quad & f_i(x) \\ \text{s.t.} \quad & f_k(x) \leq \varepsilon_k \quad \forall\, k = 1, \ldots, m,\, k \neq i, \\ & x \in X, \end{aligned} \tag{2.21}$$

where $\varepsilon \in \mathbb{R}^m$. Note that component i of vector ε is not used in (2.21). It is well-known that every feasible solution of (2.21) is weakly efficient. If the solution is unique, then it is efficient.

On the other hand, for every efficient solution $\bar{x} \in X_E$ there exists a vector $\varepsilon \in \mathbb{R}^m$ such that $\bar{x}$ solves (2.21) for any $i = 1, \ldots, m$. More precisely, every efficient solution $\bar{x} \in X_E$ is an optimal solution of (2.21) for any $i = 1, \ldots, m$ and $\varepsilon = f(\bar{x})$.

The Hybrid Method

Wendell and Lee (1977) and Corley (1980) propose to combine the weighted sum and the ε-constraint problem. The resulting problem, known as *hybrid approach*, see, e.g., Chankong and Haimes (1983), reads

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{m} \lambda_i f_i(x) \\
\text{s.t.} \quad & f_i(x) \leq \varepsilon_i \quad \forall\, i = 1, \ldots, m, \\
& x \in X
\end{aligned}
\tag{2.22}
$$

with $\lambda_i > 0$ for all $i = 1, \ldots, m$. Due to the weighted sum objective and the positive weights, every feasible solution of (2.22) is properly efficient. On the other hand, every properly efficient solution can be obtained by a suitable choice of $\varepsilon \in \mathbb{R}^m$ and arbitrarily chosen $\lambda \in \mathbb{R}^m_{>}$. Note that, in contrast to the weighted sum, also all non-supported properly efficient solutions can be computed thanks to the ε-constraints.

Compromise Programming

The ideal point is the most desired point of a multicriteria problem as it meets all objectives best. However, in the presence of conflicting objectives, the ideal point is not feasible, in general. A point as close as possible to the ideal point is seen as a good compromise. This is the basic idea of *Compromise programming* according to Zeleny (1973), also called *method of the global criterion*, see, e.g., Miettinen (1999).

Thereby, the distance is measured with the help of a norm, typically an l_p-norm. Of special interest are the cases $p = 1, 2, \infty$. In general, the formulation of compromise programming reads

$$
\min_{x \in X} \left(\sum_{i=1}^{m} |\, f_i(x) - z_i^I \,|^p \right)^{\frac{1}{p}} \tag{2.23}
$$

for $1 \leq p < \infty$ and

$$
\min_{x \in X} \max_{i=1,\ldots,m} \left\{ \left| \, f_i(x) - z_i^I \right| \right\} \tag{2.24}
$$

for $p = \infty$. Problem (2.24) is also called Tchebycheff problem and will be treated separately below. Note that $f_i(x) - z_i^I \geq 0$ holds for all $i = 1, \dots, m$ according to the definition of the ideal point, i.e., the absolute values in the objective function can be dropped. From a more general perspective the ideal point can be seen as a *reference point* and the objective function with absolute values dropped as an *achievement scalarizing function* as introduced by Wierzbicki (1980).

It is well-known that every solution of (2.23) is efficient, see, e.g., Miettinen (1999). Note that the use of the ideal point as reference point in (2.23) is important. If, for example, the ideal point was replaced by any feasible point $f(\bar{x})$, $\bar{x} \in X$, then $\bar{x}$ would be optimal for (2.23) regardless whether $f(\bar{x})$ is (weakly) nondominated or not. If we were allowed to vary the reference point, then it would be immediately clear that every efficient solution can be obtained as a solution of (2.23). However, if the reference point remains fixed to the ideal point, we might only obtain different outcomes by varying the norm, or, more precisely, the value of p. Thereby, typically, only a small subset of the nondominated set can be obtained. Thus, a more flexible problem formulation is desirable, which is, for example, given by a weighted variant of (2.23) that reads

$$\min_{x \in X} \left(\sum_{i=1}^{m} w_i |\, f_i(x) - z_i^I \,|^p \right)^{\frac{1}{p}} \tag{2.25}$$

with $w \in \mathbb{R}^m_+$. However, note that the objective function of (2.25) does not necessarily constitute a norm if some components of w are zero. Therefore, in some formulations, the reference point z^I is replaced by a utopian point z^U, as then also those points that equal z^I in at least one component can be generated with positive weights, i.e., $w \in \mathbb{R}^m_>$ can be set. Now, similar to the unweighted case, it holds that every solution of (2.25) is efficient if either $w \in \mathbb{R}^m_>$ or the solution of (2.25) is unique.

Note that for $p = 1$, the weighted sum method is obtained, as the objective function then equals (2.20) up to the constant $-\sum_{i=1}^{m} w_i z_i^I$, that can be omitted for optimization. Therefore, converse results for $p = 1$ are the same as for the weighted sum method.

The Weighted Tchebycheff Method and Variants

The *weighted Tchebycheff problem* was introduced in Bowman (1976) and studied in detail in Steuer and Choo (1983). It is defined as

$$\min_{x \in X} \max_{i=1,\dots,m} \left\{ w_i \left| \, f_i(x) - z_i^U \right| \right\} \tag{2.26}$$

with $w \in \mathbb{R}^m_>$. The weights are typically normalized, i.e., $\sum_{i=1}^{m} w_i = 1$. Note that, as already discussed for the weighted variant of compromise programming, a utopian point z^U instead of the ideal point is chosen as reference point, as otherwise the lexicographically optimal solutions could not necessarily be obtained by (2.26) due to the positive weights.

It is well-known (Bowman, 1976) that every solution of (2.26) is weakly efficient, and efficient if the solution is unique. Conversely, for every efficient solution $\bar{x} \in X_E$ there is some $w \in \mathbb{R}^m_>$ such that $\bar{x}$ solves (2.26). The corresponding weights are explicitly stated in Steuer and Choo (1983).

As, by definition, $z_i^U < f_i(x)$ for all $i = 1, \dots, m$ holds, the absolute values in the objective function of (2.26) can be dropped. Moreover, the max-function in the objective can be replaced by inequalities. An alternative formulation of (2.26) is, thus, given by

$$\begin{aligned} \min \quad & t \\ \text{s.t.} \quad & t \geq w_i \left(f_i(x) - z_i^U \right), \quad i = 1, \dots, m, \\ & t \in \mathbb{R}, x \in X. \end{aligned} \tag{2.27}$$

Formulation (2.27), presented in Steuer and Choo (1983), which employs an additional variable t and introduces m new constraints, is particularly used when all underlying functions are differentiable. While the objective function of problem (2.26) with the absolute values dropped is not differentiable due to the l_∞-norm, the objective and all constraints of problem (2.27) are, see, e.g., Miettinen (1999). Since $w \in \mathbb{R}^m_>$, another equivalent reformulation of (2.26) is given by

$$\begin{aligned} \min \quad & t \\ \text{s.t.} \quad & z_i^U + \tfrac{1}{w_i} t \geq f_i(x), \quad i = 1, \dots, m, \\ & t \in \mathbb{R}, x \in X, \end{aligned} \tag{2.28}$$

where $d := (\frac{1}{w_1}, \dots, \frac{1}{w_m})^\top$ can be interpreted as search direction. Formulations (2.27) and (2.28) have the same theoretical properties as (2.26), i.e., the solutions are weakly efficient, but not necessarily efficient. Therefore, Steuer and Choo (1983) propose the following two modifications of the weighted Tchebycheff problem so that a solution is guaranteed to be efficient.

The Lexicographic or Two-Stage Weighted Tchebycheff Method

The first approach is called *lexicographic weighted Tchebycheff method* and consists of two stages. In the first stage a weighted Tchebycheff problem is solved. In the second stage the (weakly efficient) solution x^* of the first stage is used to solve the problem

$$\begin{aligned} \min \quad & \sum_{i=1}^{m} f_i(x) \\ \text{s.t.} \quad & f_i(x) \leq f_i(x^*), \quad i = 1, \dots, m, \\ & x \in X \end{aligned} \tag{2.29}$$

to optimality. It is shown in Steuer and Choo (1983) that every optimal solution of (2.29) is efficient. Conversely, every nondominated point can be found by selecting appropriate parameters $w \in \mathbb{R}^m_>$ and solving (2.26) and (2.29). Note that (2.29) can be seen as a hybrid approach with $\varepsilon = f(x^*)$ and $w = e = (1, \dots, 1)^\top$.

According to the terminology used in Sayın and Kouvelis (2005), we also refer to this method as the *two-stage weighted Tchebycheff method* in the following, which reflects the fact that, independently of the number of objectives, two optimization problems are solved to obtain one nondominated point. Note that, in general, a second stage problem of the form (2.29) can be combined with any scalarization that yields a weakly nondominated point in the first stage, for example also with the classic ε-constraint method.

The Augmented Weighted Tchebycheff Method

A second variant of the weighted Tchebycheff method that avoids weakly nondominated points consists in modifying the objective function of the scalarization method slightly by adding a so-called augmentation term such that all objective functions become involved. The *augmented weighted Tchebycheff method* is given by

$$\min_{x \in X} \max_{i=1,\dots,m} \left\{ w_i \left(f_i(x) - z_i^U \right) \right\} + \rho \sum_{j=1}^{m} \left(f_j(x) - z_j^U \right) \tag{2.30}$$

with $w \in \mathbb{R}^m_>$, $\sum_{i=1}^m w_i = 1$ and $\rho > 0$, where ρ is a sufficiently small scalar. Likewise we might set $\rho \geq 0$ such that (2.26) becomes a special case of (2.30). Sometimes z^I is used as a reference point instead of z^U in (2.30), and $w \in \mathbb{R}^m_+$ is assumed. Then we require that either $w \in \mathbb{R}^m_>$ or $\rho > 0$.

It is shown in Steuer and Choo (1983) that for a sufficiently small choice of $\rho > 0$ every optimal solution of (2.30) is properly efficient. Conversely, every properly nondominated point can be obtained by solving Problem (2.30) with an appropriate choice of the involved parameters. An improperly nondominated point cannot be generated with a positive value of ρ. In this case, the two-stage approach must be used instead of the augmented approach. The advantage of the augmented variant in comparison to the two-stage variant is that only one stage, i.e., one scalarized problem, must be solved. On the contrary, the determination of an appropriate value of ρ is required.

The Modified Weighted Tchebycheff Method

Kaliszewski (1987) proposes the *modified weighted Tchebycheff problem.* It is given by

$$\min_{x\in X}\max_{i=1,\ldots,m}\left\{w_i\left(\left(f_i(x)-z_i^U\right)+\rho\sum_{j=1}^{m}\left(f_j(x)-z_j^U\right)\right)\right\} \tag{2.31}$$

with $w \in \mathbb{R}^m_>$, $\sum_{i=1}^m w_i = 1$ and $\rho > 0$ sufficiently small. This scalarization has the same theoretical properties as the augmented weighted Tchebycheff method.

The Method of Pascoletti and Serafini

In Pascoletti and Serafini (1984) the scalarization

$$\begin{aligned} \max_{(\xi,x,\lambda)}\quad & \xi \\ \text{s.t.}\quad & f(x) = p + \xi q + \lambda,\ \lambda \in \Lambda, \\ & (\xi,x,\lambda) \in \mathbb{R}\times X\times Y \end{aligned} \tag{2.32}$$

is considered for maximization problems, where X is a set, Y a finite-dimensional real linear space, $f : X \to Y$ a map and $\Lambda \subset Y$ a closed convex cone. Moreover, $(p,q) \in Y \times L(\Lambda)$ denotes the given parameter set with $L(\Lambda)$ being the smallest subspace of Y containing Λ. It is shown that for any Λ-minimal x^0 there exists some pair (p,q) such that (2.32) has a solution (ξ,x,λ) with $x = x^0$. Conversely, for any solution (ξ,x,λ) of (2.32), x is Λ'-optimal, where Λ' denotes the relative interior of Λ. If Λ equals the Pareto cone, the result implies that x is weakly efficient. However, note that (2.32) does not necessarily have a solution. In Eichfelder (2006), see also Eichfelder (2009a) and Eichfelder (2009b), (2.32) is restricted to the case $Y = \mathbb{R}^m$, $q \in \mathrm{int(C)} \neq \emptyset$, where $C \subset \mathbb{R}^m$ is a closed pointed convex cone and int(C) denotes the interior of C. Moreover, (2.32) is reformulated for minimization problems as

$$\begin{aligned} \min_{(\xi,x)}\quad & \xi \\ \text{s.t.}\quad & p + \xi q - f(x) \in C, \\ & \xi \in \mathbb{R}, x \in X. \end{aligned} \tag{2.33}$$

According to (2.10), $p+\xi q-f(x) \in C$ is equivalent to $f(x) \leq_C p+\xi q$. For $C := \mathbb{R}^m_+$, we equivalently obtain $p + \xi q \geqq f(x)$.

In Eichfelder (2009b) it is shown that for a specific choice of p, q and C, most of the classic scalarizations can be obtained. For example, the ε-constraint method (2.21) results from (2.33) by setting $C = \mathbb{R}^m_+$, $p_k = \varepsilon_k$ for all $k \neq i$, $p_i = 0$ and $q = e_i$, where e_i denotes the i-th unit vector. The weighted Tchebycheff method is derived for $C = \mathbb{R}^m_+$, $p = z^U$ and $q_i = 1/w_i$ for all $i = 1, \dots, m$. Other scalarizations that are proven to be special cases of (2.33) are, among others, the weighted sum method, the hybrid approach and the modified weighted Tchebycheff problem.

2.4 Parametric Algorithms

The solution of one scalarization yields one (weakly) efficient solution and, thus, one (weakly) nondominated point in the objective space. In order to compute a set of nondominated points with the help of a scalarization method, the parameters of the chosen scalarization have to be varied. Therefore, the scalarization is typically embedded in an algorithm that repeatedly solves the same scalarized problem for different choices of its parameters. We refer to such an algorithm as a *parametric algorithm* in the following, independent of the specific scalarization chosen. We also use the term *parametric scalarization* in order to emphasize that the parameters of the chosen scalarization are varied. A scalarization with a certain parameter choice is called a *subproblem*. Hence, a parametric algorithm consists in the successive solution of subproblems.

Thereby, the question arises how the parameters should be varied. Following the terminology of Hamacher et al. (2007), we distinguish a priori and a posteriori parameter schemes. For brevity, we will often simply speak of a priori and a posteriori methods. Note that, throughout this thesis, these notions always refer to the parameter selection and not, as in other classifications, to the articulation of preferences (Miettinen, 1999) or the incorporation of error measures (Faulkenberg and Wiecek, 2012).

A Priori Parameter Selection

We call a parameter choice *a priori* if the parameters of all subproblems are fixed before the iterative solution procedure starts. Bounds on the nondominated set might already be available and used. Examples are a random parameter selection, see, e.g. Steuer and Choo (1983), or a uniform parameter choice. For example, in case of the weighted sum method for bicriteria problems with normalized parameters $\lambda_1 = \lambda$, $\lambda \in [0, 1]$, $\lambda_2 = 1 - \lambda$, the interval $[0, 1]$ is divided into N subintervals $[\lambda^i, \lambda^{i+1}]$, $i = 0, 1, \dots, N-1$, where $\lambda^i = i/N$ for all $i = 0, 1, \dots, N$. However, this simple technique has the drawback that even for $\mathbb{R}^m_+$-convex nondominated sets, an evenly distributed set of weights does typically not produce an even distribution of nondominated points from all parts of the nondominated set, as discussed in Das and Dennis (1997).

An a priori parameter selection for the ε-constraint method can be found, e.g., in Hamacher et al. (2007) or Eichfelder (2009a). As the ε-constraints are related to the values of the respective objectives, bounds on the latter should be included into the computation of a priori parameters. Otherwise, a high number of infeasible subproblems might be obtained. Therefore, parameters of an ε-constraint method are typically computed based on available bounds. The subdivision is then achieved with a specified fixed stepsize for each objective, see, e.g., Chankong and Haimes (1983).

A further drawback of a priori approaches is that different choices of parameters might yield the same nondominated point, which causes an unnecessary computational effort.

A Posteriori or Adaptive Parameter Selection

We call a parameter selection scheme *a posteriori* or *adaptive* if the computation of the parameters is based on already known (nondominated) points. This implies that the parameters are updated *during* the run of the algorithm.

If the parameters are chosen appropriately, then both disadvantages of an a priori method, the unnecessary investigation of infeasible subproblems and the repeated computation of the same nondominated points, can be avoided. However, the successful application of adaptive methods depends on the quality of the computed points. If the generated points are not nondominated, but, e.g., dominated or have infeasible preimages, then parameters based on these 'wrong' points might be misleading in the sense that regions containing further nondominated points are excluded from further investigation. However, when nondominated points can be generated, adaptive parameter schemes are very appealing as they automatically adapt to the shape of the nondominated set.

Moreover, if a complete representation of a discrete problem is sought, the number of subproblems can be bounded by the number of nondominated points when an adaptive parameter scheme is used. This implies that it is possible to indicate the number of subproblems solved in the worst case dependent on the cardinality of the nondominated set.

In this thesis we propose a new adaptive parametric algorithm for generating a representation of the nondominated set. Before presenting our main results in Part I, we provide a detailed survey on related literature in the next chapter.

3 Literature Review on Parametric Algorithms

3.1 Introduction

In this chapter we give a detailed survey on (adaptive) parametric algorithms for bicriteria and multicriteria optimization problems. While the methods in the multicriteria case are also applicable in the bicriteria case, we review methods which are explicitly designed for bicriteria problems separately.

All reviewed methods generate a representation or a piecewise linear approximation that is constructed from a discrete representation. Most of the presented approaches either generate complete or incomplete representations or consider either discrete or continuous problems. In the bicriteria case we do not classify the methods further. With regard to the topic of this thesis we focus on methods that generate a complete representation in the multicriteria case. While, in general, methods designed for generating complete representations can be stopped prematurely resulting in an incomplete representation and, conversely, methods that yield an incomplete representation can also be used to generate a complete representation, if the specified maximal error is selected small enough, the course of the corresponding algorithms is typically different. If an incomplete representation is sought, quality measures as discussed in Section 2.2 guide the search for new nondominated points. If a complete representation is aimed at, quality criteria as discussed in Section 2.2 are not meaningful. Recall that coverage, uniformity, cardinality and the dominated hypervolume are possible measures for a representation. These measures are useless in the situation where a complete representation is sought, as they are intrinsically given by the structure of the nondominated set. In contrast, the number of iterations is of particular interest as it measures the number of subproblems solved and, hence, influences the computational time of an underlying algorithm.

3.2 Bicriteria Approaches

In **Geoffrion (1967)** a problem

$$\max\{h(f_1(x), f_2(x)), x \in X\} \tag{3.1}$$

is numerically solved with the help of a weighted sum problem. Thereby, h is an increasing utility function, preferably quasiconcave, its arguments f_1 and f_2 are concave (objective) functions and X is a convex set. Instead of the original problem (3.1), $\max\{wf_1(x) + (1-w)f_2(x), x \in X\}$ with $w \in [0,1]$ is solved by known parametric programming algorithms, e.g., parametric linear or quadratic programming. As a utility function is given explicitly, the method of Geoffrion (1967) computes only one representing or approximating point.

Pasternak and Passy (1973) build on the approach of Geoffrion (1967). They also assume that a utility function is given. In a first stage, they solve weighted sum problems where the weight is initially set to one half. A particular bisection approach is applied for varying the weight subsequently. In a second stage, a parametric hybrid scalarization of the form

$$\begin{aligned} \max \quad & wf_1(x) + (1-w)f_2(x) \\ \text{s.t.} \quad & f_2(x) \geq \theta, \\ & x \in X \end{aligned} \tag{3.2}$$

is used, where the values of $\theta \in \mathbb{R}$ are selected dependent on the values of the second objective of the nondominated points obtained in the first stage. Implicit enumeration is used to solve the subproblems.

Cohon et al. (1979), see also **Cohon (1978)**, design an algorithm called NISE (NonInferior Set Estimation) that generates a piecewise linear inner and outer approximation of the nondominated set of bicriteria linear programs. Therefore, a sequence of parametric weighted sum problems is solved. The algorithm starts by computing the lexicographic maxima. While the line connecting these two points serves as the initial inner approximation of the nondominated set, the two lines connecting the two points with the ideal point, respectively, define an initial outer approximation. The error is measured by the Hausdorff metric (2.16) between the inner and outer approximation using the l_2-norm. Initially, the error equals the length of the perpendicular from the ideal point on the inner approximation. The normal of the inner approximation is used to define the weights of the next subproblem. In each iteration, the inner and outer approximation are updated based on the solution of the subproblem. The next subproblem is defined by the segment of the inner approximation with the largest error. The algorithm terminates when a specified error is reached.

Aneja and Nair (1979) also use the weighted sum method to solve bicriteria linear programs. They focus on the generation of the extreme nondominated points which are sufficient to describe the complete nondominated set. First, the two lexicographic minima are determined. The points are saved in increasing order with respect to the first objective. Furthermore, the pair of indices of the two points is saved. In all subsequent iterations, a pair of indices (r, s), that corresponds to a pair of (temporary) adjacent nondominated points z^r, z^s, is selected arbitrarily. The weights are set to $w_1 := |z_2^s - z_2^r|$ and $w_2 := |z_1^s - z_1^r|$, i.e., the objective value of the corresponding weighted sum problem is the same for both points z^r and z^s. If an extreme nondominated point exists between z^r and z^s, it must be generated by the weighted sum problem, as a smaller objective value of the weighted sum is obtained. Note that it is assumed that alternative optima of the weighted sum problem can be determined, and that, in case of the existence of alternative optima, the nondominated point with minimal value in the first objective is computed. If the outcome of the subproblem equals z^r, the pair of indices (r, s) is removed from the list since no further extreme nondominated point exists between z^r and z^s. Otherwise, i.e., if a new nondominated point z^* is detected, this point is saved and two new pairs of indices corresponding to z^r and z^* and z^* and z^s are saved for an investigation in a later iteration. The authors show that if the bicriteria problem has N extreme nondominated points ($N > 2$), the algorithm performs exactly $2N - 3$ iterations after having determined the lexicographic minima. Note that Aneja and Nair (1979) motivate their approach for a bicriteria transportation problem. However, as pointed out by Ulungu and Teghem (1995), only supported nondominated points of the latter can be found by this approach.

Chalmet et al. (1986) propose an algorithm for solving bicriteria integer problems in maximization format. They focus on a complete representation of the finite nondominated set. While the proposed algorithm is very similar to the one of Aneja and Nair (1979), an important difference is that constraints are imposed that eliminate all known nondominated points as well as the regions dominated by them. In each iteration, a subproblem

$$\begin{aligned} \max \quad & w_1 f_1(x) + w_2 f_2(x) \\ \text{s.t.} \quad & f_i(x) \geq z_i^{LN} + 1, \quad i = 1, 2, \\ & x \in X \end{aligned} \tag{3.3}$$

is solved, where $w_1, w_2 > 0$ and z^{LN} denotes the local nadir point with respect to the pair of known nondominated points (z^r, z^s) that has been selected for investigation. Assuming without loss of generality that $z_1^r < z_1^s$, it holds that $z^{LN} = (z_1^r, z_2^s)^\top$. With the help of the hybrid formulation it is possible to generate all supported and unsupported nondominated points. It is shown that $2|Z_N| + 1$ integer programs are solved in total. Note that this bound is compatible with the one of Aneja and Nair (1979) since four integer problems need to be solved to determine the lexicographic maxima. Chalmet et al. (1986) also extend their approach to problems with more than two criteria. We will discuss this extension in Section 3.3 below.

Eswaran et al. (1989) propose an algorithm for nonlinear integer bicriteria problems in maximization format. As scalarization a weighted Tchebycheff method with normalized weights is used, i.e., the weight parameter is chosen from the initial parametric space $W := [0, 1]$. Since the efficient set is assumed to be uniformly dominant, which excludes the existence of weakly efficient points, every solution of the weighted Tchebycheff method is efficient and its image is nondominated. In the algorithm the initial parametric space is subsequently decomposed into subintervals. Thereby, the limits of each subinterval correspond to instances of weighted Tchebycheff problems that have already been solved. For example, $w = 0$ and $w = 1$ correspond to the two lexicographic maxima. In every iteration a subinterval is chosen for further refinement and a new weight is determined by simple bisection. If the solution of the corresponding subproblem is a new nondominated point, then the current subinterval is divided into two new subintervals. Thereby, with every subinterval $[w^r, w^s] \subset W$, always the two nondominated points that were generated for w^r and w^s, respectively, are saved. If one of the two nondominated points that are associated with a subinterval is recomputed, the respective part of the subinterval is discarded from further consideration. The algorithm terminates when $w^s - w^r \leq \xi$ holds for all (non discarded) subintervals of W, where ξ is a prescribed small positive number. If ξ is chosen sufficiently small, all nondominated points can be generated.

Solanki (1991) generate incomplete representations of mixed integer bicriteria linear programs in maximization format. The presented approach is similar to the algorithm of Cohon (1978), but instead of a weighted sum scalarization an augmented weighted Tchebycheff method is used. After having computed the lexicographic maxima and, hence, z^I and z^N, an initial rectangle containing further nondominated points is computed. This rectangle is refined subsequently in the following. In general, each rectangle is defined by two adjacent nondominated points. In each iteration, the next rectangle is always selected as the one with the largest width or height scaled by the ranges of the objective, i.e., scaled by $|z_i^N - z_i^I|$ for each

$i = 1, 2$. The weights are then chosen such that the inflection point of the weighted Tchebycheff contour lies on the diagonal of the considered rectangle. The algorithm terminates when a prescribed maximal width or height is reached in all rectangles. The augmentation parameter ρ is chosen fixed from the interval $[10^{-3}, 10^{-2}]$. The author states that, on the one hand, numerical difficulties may arise if ρ is chosen too small and, on the other hand, some nondominated points might be unreachable if the value of ρ is too large.

Ulungu and Teghem (1995) address bicriteria combinatorial optimization problems. Combinatorial problems are discrete problems with a particular structure for which efficient single-criterion methods exist, as, e.g., the Hungarian method for the assignment problem (Kuhn, 1955). While the weighted sum method is well suited for combinatorial problems in general since the structure of the underlying multicriteria problem is not destroyed by additional constraints, only supported nondominated points can be computed. In contrast, when scalarizations are applied that generate supported as well as non-supported points, at least one additional constraint is introduced and, consequently, efficient combinatorial methods can not be used directly, in general. Ulungu and Teghem (1995) propose a two phase procedure. In the first phase, all supported nondominated points are determined with a weighted sum scalarization, where the weights are varied as in Aneja and Nair (1979). Note that, as in Aneja and Nair (1979), it is assumed that all alternative optima of the weighted sum problem are determined, respectively. When the first phase stops, a set of triangles between adjacent supported points remains in which further unsupported nondominated points might lie. In the second phase, all triangles are investigated with a problem-specific combinatorial procedure.

In **Schandl et al. (2001)** an algorithm that generates a piecewise linear approximation for continuous and discrete bicriteria problems in minimization format is proposed. A scalarization of the general form

$$\max \left\{ \gamma(z) : z \in Z \cap \left(\{r\} - \mathbb{R}^2_+ \right) \right\}$$

is used, where $\gamma : \mathbb{R}^2 \to \mathbb{R}$ is an oblique norm (i.e., a norm with a polyhedral unit ball where no facet is parallel to any coordinate axis), and $r \in Z + \mathbb{R}^2_+$ is a given reference point. W.l.o.g., the origin is taken as reference point. The principal idea is to use a part of the unit ball of the oblique norm as (convex) approximation of the nondominated set. The unit ball, which is assumed to be polyhedral, is decomposed into a set of cones, each of which is defined by a pair of two adjacent nondominated points. For each cone, a candidate is computed by solving

$$\begin{aligned} \max \quad & \gamma(z) = \lambda_i + \lambda_j \\ \text{s.t.} \quad & z = \lambda_i z^i + \lambda_j z^j, \\ & \lambda_i, \lambda_j \geq 0, \\ & z \in Z. \end{aligned} \tag{3.4}$$

Note that the 'weights' λ_i and λ_j that are associated with the known nondominated points z^i and z^j are variables themselves. In (3.4) a nondominated point z^* is determined that lies in the cone defined by z^i and z^j and, thus, can be expressed as a linear combination of z^i and z^j with non-negative weights. The deviation of the candidate from the current approximation is given by $|\gamma(z^*) - 1|$, i.e., it is computed with the help of the optimal objective value of (3.4). In each iteration, the candidate with the largest deviation is inserted, the approximation is updated by connecting the new point with the two points defining the corresponding cone, respectively, and the cone is subdivided into two new cones. Then, a new candidate is computed in each of the two new cones and a new iteration starts. The procedure stops when either a prescribed number of cones or a desired maximal deviation is obtained. However, only supported nondominated points can be computed by (3.4). Therefore, if a general, possibly non-convex or discrete problem is considered, a second stage is performed when the desired accuracy has been obtained. Now, a lexicographic weighted Tchebycheff method with a local ideal point as reference point and the weights defined with respect to the local nadir and the local ideal point is employed in order to generate additional unsupported points. The main difference of the approach of Schandl et al. (2001) from all previously presented methods is the property that the approximation error is provided by the scalarization itself.

Sayın and Kouvelis (2005) use a two-stage weighted Tchebycheff scalarization to solve bicriteria discrete optimization problems in minimization format. Thereby, two variants are employed, where the first uses the ideal point and the second the origin as fixed reference point. Note that, w.l.o.g., it is assumed that $Z \subseteq \mathbb{R}^2_>$. The algorithm is similar to the algorithm of Eswaran et al. (1989). It improves the latter, as, instead of simple bisection, the weights of the current subproblem are determined based on the fixed reference point and the local nadir point with respect to the selected pair of adjacent points. However, the authors do not discard the complete rectangle between two adjacent points when one of its defining points is computed in the current subproblem. Instead, the algorithm keeps on subdividing the parametric space, even if certain subproblems can not contain further nondominated points. Similar to Eswaran et al. (1989), the algorithm terminates when a prescribed distance between every pair of subsequent weights is reached. A numerical study reveals that the variant which uses the origin as reference point requires considerably less computational time than the original Tchebycheff method. While no explanation for this observation is given, the difference is probably caused by the fact that the weights depend on the chosen reference point while the termination criterion does not.

In **Eichfelder (2006)**, see also **Eichfelder (2009a)**, an equidistant representation of the nondominated set of continuous multicriteria optimization problems is generated in which the representing points have a predefined l_2-distance $\alpha > 0$. As scalarization the method of Pascoletti and Serafini (2.32) is used which, among others, also comprises the weighted Tchebycheff method as a special case. We describe the general method with the help of this scalarization in the bicriteria case in the following. Note that while the method can be used for an arbitrary number of objectives, a special algorithm is derived in the bicriteria case. Therefore, we discuss the bicriteria case separately and will state the algorithm for the general case in Section 3.3 below. In the bicriteria case, the considered Tchebycheff subproblems are of the form

$$\begin{array}{ll} \min & t \\ \text{s.t.} & r_i + t\, d_i \geq f_i(x), \quad i = 1, 2, \\ & t \in \mathbb{R}, x \in X. \end{array} \tag{3.5}$$

Thereby, the direction $d \in \mathbb{R}^2_+, d_1 > 0$, is given as input and kept constant throughout the algorithm. The reference points $r \in \mathbb{R}^2$ are chosen adaptively from a hyperplane $H := \{y \in \mathbb{R}^2 : b^\top y = \beta\}$ whose parameters $b \in \mathbb{R}^2$ and $\beta \in \mathbb{R}$ are also given as input. First, the lexicographic minima are determined. The corresponding reference points $r^1 \in H$ and $r^E \in H$ are computed as well as the Lagrangian multiplier μ^1, corresponding to the constraints $r^1 + t^1\, d \geqq f(x^1)$ in (3.5). In each iteration, the next reference point on the line connecting r^1 and r^E is determined adaptively based on the previously computed point. Thereby, a first or second order approximation of the nondominated set is implicitly constructed and, with the help of the Lagrangian multipliers, a reference point $r \in H$ is determined such that the solution of the next subproblem (3.5) yields a point that has approximately the desired distance α from the previously computed point. Based on this approximated reference point, (3.5) is solved in the following. If, for the resulting point $f(x^*)$, all constraints in (3.5) are satisfied with equality, the algorithm proceeds by computing the next estimated reference point. Otherwise, a correction of the next reference point is applied which avoids that the algorithm gets stuck in non-convex parts of the nondominated set. The algorithm proceeds until the reference point that corresponds to x^E is reached.

Laumanns et al. (2006), see also **Laumanns et al. (2005)**, propose an approach to generate a complete representation for problems in maximization format with any number of criteria. However, as their method yields the best known upper bound on the number of subproblems in the bicriteria case, we review their approach for the bi- and the general multicriteria case separately. Laumanns et al. (2006) employ a lexicographic ε-constraint method

$$\begin{aligned} \text{lex max} \quad & (f_1(x), f_2(x))^\top \\ \text{s.t.} \quad & l_2 < f_2(x), \\ & x \in X \end{aligned} \tag{3.6}$$

as scalarization. A first (unconstrained) scalarization with $l_2 :=$ '∞' is solved that yields the nondominated point z^* with minimal second component. In all iterations, $l_2 := z_2^*$ is set, where z^* denotes the nondominated point obtained in the previous iteration. Thereby, all nondominated points are generated sorted in increasing order with respect to the second component. Every subproblem yields a new nondominated point besides the last subproblem which is infeasible. Hence, a complete representation is obtained within the solution of $|Z_N| + 1$ subproblems.

Ralphs et al. (2006) propose an algorithm for integer bicriteria optimization problems in maximization format. The (augmented) weighted Tchebycheff method is used as scalarization. The authors improve the approach of Eswaran et al. (1989) by computing the weights of the subproblems as presented in Solanki (1991), i.e., based on the local ideal and local nadir point with respect to the two nondominated points between which a new nondominated points is sought. Thereby, in each iteration, either a new nondominated point is generated or the pair of nondominated points that define the weights of the current subproblem can be discarded from further consideration. The algorithm is shown to solve $2|Z_N| - 1$ subproblems , where the generation of the lexicographic maxima is included and the computation of each lexicographic maximum is counted as one subproblem. Hence, the stated number of subproblems equals the one presented in Chalmet et al. (1986). If the efficient set is uniformly dominant (see Eswaran et al. (1989)), a weighted Tchebycheff method is used, otherwise an augmented weighted Tchebycheff method with $\rho \in \{10^{-4}, 10^{-3}, 10^{-2}\}$ is employed. As an alternative, a weighted Tchebycheff method that enumerates all optimal outcomes is proposed. If more than one nondominated point is found, weakly nondominated points are eliminated by imposing a cut during the branch and bound procedure. As already stated in Solanki (1991), the choice of the augmentation parameter in the augmented weighted Tchebycheff method is crucial. Even for $\rho = 10^{-4}$, the authors experience numerical problems in the sense that in some instances not only some nondominated points are missed but also weakly nondominated points are computed. They conclude that it is not possible to choose a proper fixed augmentation parameter for these instances.

Hamacher et al. (2007), see also **Ruzika (2007)**, present an a priori and an a posteriori algorithm for generating an incomplete representation for discrete bicriteria optimization problems in minimization format. In both algorithms, a lexicographic ε-constraint method

$$\begin{aligned} \operatorname{lex\,min} \quad & (f_2(x), f_1(x)) \\ \text{s.t.} \quad & f_1(x) \geq \varepsilon, \\ & x \in X \end{aligned} \tag{3.7}$$

is used to generate nondominated points. First, the lexicographic minima are determined which define the initial rectangle in which possibly further nondominated points lie. In the a priori variant, equidistant values for ε are determined based on z_1^I and z_1^N, and a corresponding subproblem is solved. In the a posteriori variant, the area of the initial rectangle is computed which serves as quality measure in the following. In each iteration, the rectangle with the largest area is selected for further refinement. Let z^i and z^{i+1} be two nondominated points defining the current rectangle, where, w.l.o.g., $z_1^i < z_1^{i+1}$ holds. Then, (3.7) is solved with $\varepsilon := \lfloor z_1^i + (z_1^{i+1} - z_1^i)/2 \rfloor$. Let z^* be the resulting nondominated point. The authors show that it is sufficient to consider the two rectangles defined by the pairs of points $(z^i, z^*)^\top$ and $((\varepsilon, z_2^*)^\top, z^{i+1})^\top$, respectively, using not only the nondominance of z^* but also the properties of the ε-constraint method (cf. Laumanns et al. (2006)). If an integer-valued problem is assumed, the rectangles can even be reduced further. In any case the area of the considered rectangle can be reduced at least by a factor of two. As termination criterion either a desired accuracy, i.e., a bound on the area of all rectangles, or a desired maximal cardinality of the representation can be set. If the accuracy is set to one, all nondominated points are computed by both variants.

Faulkenberg and Wiecek (2012) address the problem of generating equidistant representations for continuous bicriteria optimization problems. Two alternative methods to the approach of Eichfelder (2009a) are proposed. In the first method, called Constraint Controlled-Spacing Formulation, the desired distances among the generated points are obtained by imposing appropriate constraints. However, problems arise for non-convex problems for which the nondominated set is not connected. Additional scalarizations are needed to check whether a point is dominated. The second approach is denoted as Bilevel Controlled-Spacing method and addresses convex problems. A bilevel problem is formulated in which the nondominated point is generated in the lower level and the spacing is controlled in the upper level. The lower level is then reformulated with the help of Lagrangian multipliers.

3.3 Multicriteria Approaches

In the following, we review literature that deals with (adaptive) parametric algorithms for optimization problems with an arbitrary number of objectives. We concentrate on approaches that generate a complete representation of the nondominated set for discrete multicriteria optimization problems. These methods are presented first, while approaches for generating incomplete representations or piecewise linear approximations are discussed at the end of this section.

Generation of complete representations

Klein and Hannan (1982) generalize the bicriteria approach of Pasternak and Passy (1973) to a procedure for multicriteria linear integer problems in minimization format. In every iteration $s \geq 1$, they solve a problem of the form

$$\begin{aligned} \min \quad & f_k(x) \\ \text{s.t.} \quad & \bigwedge_{i=1}^{s-1} \left(\bigvee_{\substack{j=1,\ldots,m \\ j \neq k}} f_j(x) \leq f_j(x^i) - \delta_j \right), \\ & x \in X, \end{aligned} \tag{3.8}$$

where $k \in \{1, \ldots, m\}$ is an arbitrarily chosen index, $x^1, \ldots, x^{s-1}$ denote the efficient solutions found in the previous iterations, and $\delta_j, j = 1, \ldots, m$, is a positive integer. Since the objective function comprises only one single objective, outcomes of (3.8) might also be weakly nondominated. With the help of the disjunctive constraints in (3.8) it is possible to consider all remaining parts of the search region simultaneously. However, thereby, in each iteration, a set of $m - 1$ constraints is added to the problem. Numerical results for problems with two to five objectives, the largest one having on average less than 45 efficient solutions, are presented. Besides CPU time, the emphasis lies on the number of nodes visited in the corresponding enumeration tree.

The method of **Chalmet et al. (1986)**, discussed in the previous section, mainly addresses bicriteria problems. Therefore, their idea how to extend their algorithm to the general multicriteria case probably remained unnoticed by most authors. A recursive procedure is proposed which, on the basic level, uses the approach for bicriteria problems. We describe the recursive procedure exemplary for the tricriteria case. First, all points are determined that are nondominated with respect to the first two objectives and that are lexicographically maximal with respect to the third objective. Among these nondominated points, the point with the minimal value in the third objective is identified. Let $\bar{z}$ be this point. The constraint $f_3(x) \geq \bar{z}_3 + 1$ is added to the formulation of the subproblem and again, all nondominated points with respect to f_1 and f_2 can be determined for the modified subproblem. In order to avoid that the same nondominated points are recomputed, additional constraints based on known nondominated points that are adjacent in the f_1-f_2-projection are imposed. Therefore, it can be guaranteed that no nondominated point is computed more than once. For more than three criteria, further levels of recursion are required. The authors show that at least $m|Z_N| + 1$ integer programs need to be solved. This lower bound occurs when all nondominated points are found on the lowest level of recursion and the remaining iterations are required to verify that no further nondominated points exist.

In the technical report of **Tenfelde-Podehl (2003)** a recursive approach for combinatorial multicriteria optimization problems is proposed. In a first stage, for a given problem with m criteria, all m corresponding $(m-1)$-criteria problems are solved which might include further recursions until bicriteria problems are obtained. These are solved by known methods. According to Ehrgott and Tenfelde-Podehl (2003) the points obtained by solving all $(m-1)$-criteria problems represent a subset of the nondominated set of the original problem and, in particular, all points defining the ideal and nadir point belong to this subset. Based on these results, the regions in which all remaining nondominated points might lie can be described after having solved all $(m-1)$-criteria problems. In the second stage, for each nondominated point $\bar{z}$ computed in the first stage, m hyperplanes $h^i(\bar{z}) = \{z \in \mathbb{R}^m : z_i = \bar{z}_i\}$, $i = 1, \ldots, m$, are introduced which decompose the set of feasible outcomes that is bounded by the ideal and the nadir point into boxes. A part of these boxes can be directly excluded due to the points obtained in the first stage being nondominated. All boxes that remain after this reduction are regrouped according to several criteria. As observed in the computational tests of Przybylski et al. (2009), the limitation of this method is the memory requirement needed for saving the huge number of boxes.

Sylva and Crema (2004) revisit the idea of Klein and Hannan (1982) for solving multiple objective integer linear programs in maximization format. They change the objective to a weighted sum in order to avoid weakly nondominated points. Moreover, they use the common reformulation (see, e.g. Nemhauser and Wolsey (1999)) of the disjunctive constraints in (3.8) with the help of binary variables. Hence, the subproblem to be solved in iteration $s \geq 1$ reads

$$\begin{aligned}
\max \quad & \sum_{j=1}^{m} w_j f_j(x) \\
\text{s.t.} \quad & f_j(x) \geq \Big(f_j(x^i) + 1\Big)\, y_j^i - M_j(1 - y_j^i), \quad && j = 1, \ldots, m,\ i = 1, \ldots, s-1, \\
& \sum_{j=1}^{m} y_j^i \geq 1, && i = 1, \ldots, s-1, \\
& y_j^i \in \{0, 1\}, && j = 1, \ldots, m,\ i = 1, \ldots, s-1, \\
& x \in X,
\end{aligned} \tag{3.9}$$

where $w \in \mathbb{R}^m_>$ are given weights, $-M_j$ is a lower bound on f_j for every $j = 1, \ldots, m$, e.g., $M := -z^I$, and $x^1, \ldots, x^{s-1}$ denote the efficient solutions found in the previous iterations. In every iteration, m binary variables as well as $m + 1$ constraints are added, which makes this approach computationally demanding. Indeed, Sylva and Crema (2004) only generate complete representations for the bicriteria case. For three objectives they restrict the numerical study to the generation of incomplete representations.

In **Laumanns et al. (2006)** multicriteria optimization problems in maximization format are addressed. The proposed method relies on the decomposition of the objective space into disjoint cells based on already computed nondominated points. Since a lexicographic ε-constraint method is used as scalarization, the search region can be projected to an $(m-1)$-dimensional subspace, see Section 5.4 for further details. The subproblems to be solved are of the form

$$\begin{aligned}
\operatorname{lex\,max} \quad & f(x) = (f_1(x), \ldots, f_m(x))^\top \\
\text{s.t.} \quad & l_i < f_i(x) \leq u_i \quad \forall\, i = 2, \ldots, m, \\
& x \in X,
\end{aligned} \tag{3.10}$$

where l and u denote the vertices of the current $(m-1)$-dimensional cell. The latter is initialized by $l := -(\infty, \ldots, \infty)^\top$ and $u := (\infty, \ldots, \infty)^\top$. In every iteration, the algorithm investigates all current cells in a specified order until a new point is obtained as solution of (3.10). Note that the point corresponding to the solution of (3.10) is not necessarily nondominated due to the two-sided constraints. However, it is excluded by the order in which the cells are investigated that a dominated point is generated before a point dominating it is known. Every new nondominated point is inserted into the $(m-1)$-dimensional search region which, thereby, is decomposed into $(n+1)^{m-1}$ cells in total, where n denotes the number of currently known nondominated points. After the update of the cells the algorithm restarts with the first cell according to the specified order. Whenever the subproblem in a cell is infeasible or a dominated point is obtained, the cell is marked as empty, i.e., the bounds of the current cell are saved such that this cell or a cell that is completely contained in it is not investigated in later iterations. Moreover, when a new nondominated point z^* is detected in a certain cell, the subcell that is dominated by the projection of z^* is discarded as well. Before solving (3.10) the list of discarded cells is scanned in order to prevent the unnecessary solution of subproblems. In the worst case, a subproblem is solved in every cell. Hence, at most $(|Z_N| + 1)^{m-1}$ subproblems are solved in the course of the algorithm. Therefore, a complexity of $\mathcal{O}(|Z_N|^{m-1} \cdot T)$ is derived where T denotes the running time of the single-objective optimizer. To the best of our knowledge, Laumanns et al. (2006) state the first upper bound on the number of subproblems to be solved for any $m \geq 2$. The two-sided bounds in (3.10) which might cause the generation of dominated points are removed in Laumanns et al. (2005). Moreover, the lexicographic objective is replaced by a two-stage formulation, and a kind of relaxation test is performed based on saved subproblems. However, the (worst-case) complexity is not improved and the main drawback, the huge amount of cells to be investigated, remains.

Özlen and Azizoğlu (2009) propose a recursive algorithm for generating the entire nondominated set of integer multicriteria optimization problems in minimization format. Their approach is similar to the approach of Chalmet et al. (1986) for more than two criteria. On the lowest level of recursion, constraint problems of the form

$$\begin{aligned} \min \quad & f_1(x) + w_2 f_2(x) + \cdots + w_m f_m(x) \\ \text{s.t.} \quad & f_j(x) \leq u_j, \quad j = 2, \ldots, m, \\ & x \in X \end{aligned} \tag{3.11}$$

are solved. Thereby, the coefficients w_i, $i = 2, \ldots, m$, are set to

$$w_i := \left(\prod_{j=2}^{i} \left(f_j^{GUB} - f_j^{GLB} + 1 \right) \right)^{-1},$$

where f_j^{GLB} and f_j^{GUB}, $j = 2, \ldots, m$, denote global lower and upper bounds on objective f_j, respectively, which are computed before the recursion starts. The parameters u_j, $j = 2, \ldots, m$, are set to the respective upper bounds f_j^{GUB} in the beginning. By systematically varying u_2 and keeping u_j, $j = 3, \ldots, m$, fixed, all points that are nondominated with respect to the first two objectives and which satisfy the imposed bounds on the remaining objectives are determined. Thereby, in each iteration $s \geq 2$, the value of u_2 is subsequently updated by setting $u_2 := f_2(x^{s-1}) - 1$, where x^{s-1} denotes the efficient solution of the previous iteration. Note that due to the augmented objective of (3.11) the outcomes of (3.11) are nondominated with respect to all criteria. When (3.11) becomes infeasible, the maximal value of the third component of the obtained points is determined. Let x^* be the corresponding solution. Then u_3 is updated by setting $u_3 := f_3(x^*) - 1$, u_2 is reset to f_2^{GUB} and a new sequence of problems (3.11) is solved. The same principle applies to all recursion levels, i.e., as soon as the problem of the current level becomes infeasible, the value of the bound u_j corresponding to the next higher level is updated based on the nondominated points found in the current level. It is shown in Özlen and Azizoğlu (2009) that the entire nondominated set can be computed by this recursive procedure. However, the algorithm might generate the same nondominated point several times during the algorithm. In the worst case, only one nondominated point is excluded when the bound u_j, $j = 3, \dots, m$, is updated, i.e., for a tricriteria problem up to $1 + 2 + \cdots + |Z_N| = \frac{1}{2}|Z_N|(|Z_N| + 1) = \mathcal{O}(|Z_N|^2)$ subproblems need to be solved. For general $m \geq 2$, a complexity of $\mathcal{O}(|Z_N|^{m-1})$ with respect to the number of subproblems is derived.

Dhaenens et al. (2010) extend the approach of Tenfelde-Podehl (2003) by proposing a three-stage procedure. The first stage is identical to the first stage in Tenfelde-Podehl (2003), the second stage consists in selecting a well-dispersed subset from the points computed in the first stage. In the third stage, all remaining points are computed. Thereby, an improved decomposition of the search region is proposed. However, no comparative computational studies are provided.

Przybylski et al. (2010a) propose an extension of the two phase method of Ulungu and Teghem (1995) for multicriteria combinatorial optimization problems and apply it to tricriteria assignment problems. In the first phase, a weighted sum method is used to determine all supported nondominated points (Przybylski et al., 2010b) as well as the hyperplanes of the facets of the convex hull of these points. Analogously to the bicriteria case the search area can also be bounded from below with the help of the hyperplanes determined in the first phase. However, different from the bicriteria case, the nadir point is not necessarily available after the computation of all supported points. Therefore, the authors continue by solving all m problems with $m - 1$ objectives. In doing so, the nadir point is determined, see also Ehrgott and Tenfelde-Podehl (2003) and Tenfelde-Podehl (2003). Based on all nondominated points found so far and the hyperplanes determined in the first stage, a so-called search area is defined that is described by an upper bound set. Thereby, only upper bound vectors u are maintained that do not equal the ideal point in any component and for which no vector $u' \neq u$ in the set of upper bounds exists with $u \leqq u'$. Then, the original second phase starts by computing all remaining nondominated points with the help of a problem specific method. In a computational study the authors compare their multicriteria version of the two phase method with the approaches of Sylva and Crema (2004), Laumanns et al. (2006) and Tenfelde-Podehl (2003). They report that their two phase method outperforms all other methods considerably.

Lokman and Köksalan (2013) present two algorithms for integer multicriteria optimization problems in maximization format. The first directly builds upon the method of Sylva and Crema (2004). Lokman and Köksalan (2013) state that when using a particular choice of the weight in the objective function of the approach of Sylva and Crema (2004), one constraint and one binary variable can be saved per iteration. In other words the authors use the original formulation of Klein and Hannan (1982), but augment the objective by all other objectives scaled by a small constant. Thereby, the weakly dominated outcomes of (3.8) can be avoided. While better computational times compared to the algorithm of Sylva and Crema (2004) are reported and also tricriteria problems are solved, the algorithm still suffers from the quickly increasing number of constraints and binary variables. The second algorithm proposed in Lokman and Köksalan (2013) relies on the observation that for each feasible point, at most one constraint from (3.9) is sufficient for each of the $m-1$ criteria. Note that this fact has already been used in Laumanns et al. (2005), as each cell is described by $m-1$ constraints. In each iteration, $n+1$ subproblems of the form

$$\begin{aligned} \max \quad & f_m(x) + \varepsilon \sum_{j=1}^{m-1} f_j(x) \\ \text{s.t.} \quad & f_j(x) \geq b_j^k, \quad j = 1, 2, \dots, m-1, \\ & x \in X \end{aligned} \tag{3.12}$$

with $k = 0, \dots, n$ are considered. The parameter ε denotes a small positive constant. The bounds b^k, $k = 0, \dots, n$, are computed based on the set $S = (z^1, \dots, z^n)$ which contains the n nondominated points found so far in the order in which they have been generated. In the tricriteria case the bounds $b^k = (b_1^k, b_2^k)^\top$, $k = 0, \dots, n$, are defined by

$$b_1^k := \begin{cases} -M, & \text{if } k = 0, \\ z_1^k + 1, & \text{otherwise,} \end{cases}$$

where the constant $-M$ denotes a global lower bound on every component of every feasible point, and

$$b_2^k := \begin{cases} -M, & \text{if } S^k = \emptyset, \\ \max\limits_{z \in S^k} \{z_2\} + 1, & \text{otherwise,} \end{cases}$$

where $S^k := \{z \in S : z_1^i \geq b_1^k\}$ for every $k = 0, \ldots, n$. Note that the method is applicable to problems with an arbitrary number of criteria, but that we restrict the description here to $m = 3$ for simplicity. Before the current subproblem is solved, a suitable relaxation is searched among the list of already investigated subproblems. Numerical results of Lokman and Köksalan (2013) show that their second approach outperforms the algorithm of Sylva and Crema (2004), their own improvement of the latter as well as the algorithm of Özlen and Azizoğlu (2009). Moreover, for $m = 3$, on average 2.13 subproblems are solved per nondominated point on randomly generated instances of multiobjective knapsack, minimum spanning tree as well as shortest path problems. However, the theoretical upper bound on the number of subproblems is again $\mathcal{O}(|Z_N|^{m-1})$. For example, in the tricriteria case $1 + 2 + \cdots + (|Z_N| + 1) = \frac{1}{2} \cdot (|Z_N| + 1) \cdot (|Z_N| + 2)$ subproblems are solved in the worst case.

Ozlen et al. (2014) improve the method of Özlen and Azizoğlu (2009). The main drawback of the latter is the solution of subproblems which have already been solved before. In Ozlen et al. (2014) the right-hand side vectors in (3.11) are saved together with either the corresponding nondominated point or the information that the subproblem is infeasible. Before a new subproblem with bound u is solved, it is checked whether a *relaxation* exists, i.e., a saved subproblem with bound u' where $u' \leqq u$. Note that the authors require '$\leq$', which is naturally satisfied since at least one component must have changed. Two cases may occur: If the relaxation is infeasible, the current problem must be infeasible as well. If the relaxation is feasible and all outcomes of this relaxation are feasible for the current problem, the set of outcomes of the relaxation equals the set of outcomes of the current problem. Note that as soon as only one outcome of the potential relaxation is not feasible, the current subproblem must be resolved. In the computational tests in Ozlen et al. (2014) significant savings of up to 95% with respect to the algorithm of Özlen and Azizoğlu (2009) are reported. For a knapsack problem with three objectives used as a test problem in Laumanns et al. (2005), the improved approach is additionally compared to the algorithm of Laumanns et al. (2005) and a significant saving in computational time is demonstrated.

The approach of **Kirlik and Sayın (2014)** is very similar to the algorithm of Laumanns et al. (2005), but removes the fixed order in which the cells are investigated. Instead, as next cell always the one with the largest $(m-1)$-dimensional volume that is defined by

$$\prod_{j=1}^{m-1} \left(u_j - z_j^I\right)$$

is selected, where u denotes the upper bound of the corresponding cell. By this small change the number of subproblems solved is reduced drastically. However, the theoretical worst case bound is not improved in comparison to the approach of Laumanns et al. (2005) and Laumanns et al. (2006). Numerical tests are carried out for knapsack and assignment problems with three and four objectives. For tricriteria problems the authors compare their approach with the ones of Sylva and Crema (2004), Laumanns et al. (2005) and Özlen and Azizoğlu (2009). The four-objective problem can only be solved by the new approach of the authors and the one of Özlen and Azizoğlu (2009). The results show that the proposed method clearly outperforms all other ones. Moreover, for $m = 3$, at most 1.99 subproblems are solved per nondominated point.

Generation of incomplete representations and approximations

In the remainder of this chapter we discuss methods generating incomplete representations or approximations of the nondominated set. We limit the discussion to parametric algorithms that are applicable to general multicriteria optimization problems, i.e., in particular, to discrete or non-convex problems. We do not review special approaches for linear or convex problems, but refer to Solanki et al. (1993), Rennen et al. (2009), Özpeynirci and Köksalan (2010b) and Przybylski et al. (2010b) and references therein.

In **Das and Dennis (1998)**, the *normal boundary intersection method*, known under its acronym NBI, is proposed. For different values of the normalized parameter vector $\beta \in R^m_+$, a scalarization of the form

$$\begin{aligned} \max \quad & t \\ \text{s.t.} \quad & \phi\beta + t\,\hat{n} = f(x), \\ & t \in \mathbb{R}, x \in X \end{aligned} \tag{3.13}$$

is solved. Thereby, ϕ is an $(m \times m)$-matrix whose i-th column consists of a point $f(x)$ with $f_i(x) = z^I_i$. Hence, ϕ can be chosen as the payoff-table given in Section 2.1. With the help of the parameter $\beta \in R^m_+$, $\sum_{i=1}^m \beta_i = 1$, theoretically all convex combinations

$$\left\{ \phi\beta : \beta \in R^m_+, \sum_{i=1}^m \beta_i = 1 \right\}$$

of the column vectors of ϕ that represent the convex hull of the individual minima, called *CHIM*, can be constructed. Finally, $\hat{n} \in \mathbb{R}^m$ denotes the unit normal of the CHIM pointing towards the ideal point. The basic idea of the method is to choose a set of (reference) points $\phi\beta$ for specific choices of β on the CHIM and to project them in the direction given by the normal to the boundary of the outcome space. Therefore, the NBI method can be classified as an a priori method.

While the principal idea of the NBI method is very appealing, the method has some shortcomings. Unfortunately, as already stated by the authors themselves, not every solution of (3.13) is weakly efficient. The reason is the equality in the constraints which causes that dominated points might be optimal for (3.13) in non-convex regions of the outcome space. However, as the authors particularly deal with nonlinear multicriteria problems, they argue that a solution of (3.13) might always be only a local optimum of the single-objective problem and therefore computational problems typically occur anyway. Also for the converse result there are some limitations. While in the bicriteria case every nondominated point can be obtained as a solution of (3.13) for an appropriate parameter choice, this is not true for $m \geq 3$, at least when the reference points are restricted to the CHIM. Also this fact has already been noticed by the authors who use the sphere of the unit ball in $\mathbb{R}^3$ as an example of the set Z. None of the points lying on the boundary of Z_N except the three individual minima can be computed by (3.13) with a non-negative choice of β, however, all these points are nondominated.

Schandl et al. (2002) generalize the approach of Schandl et al. (2001) to approximate the nondominated set of problems with an arbitrary number of objectives. The two main differences with respect to the bicriteria case are that, even in the $\mathbb{R}^m_+$-convex case, an outcome of the *gauge method*

$$\begin{aligned} \max \quad & \gamma(z) = \sum_{i=1}^{m} \lambda_i \\ \text{s.t.} \quad & z = \sum_{i=1}^{m} \lambda_i z^i, \\ & \lambda_i \geq 0, \quad i = 1, \dots, m, \\ & z \in Z \end{aligned} \tag{3.14}$$

is not necessarily nondominated. Moreover, the update of the piecewise linear approximation becomes more involved.

Klamroth et al. (2002) extend Schandl et al. (2002) and provide several further ideas. For the $\mathbb{R}^m_+$-convex and $\mathbb{R}^m_+$-non-convex case, the construction of an inner and outer piecewise linear approximation is proposed, respectively. Note that only an inner approximation is provided in Schandl et al. (2001) and Schandl et al. (2002). Moreover, a convergence rate is given for the bicriteria $\mathbb{R}^m_+$-convex case. In the $\mathbb{R}^m_+$-non-convex case, weighted Tchebycheff problems as in Schandl et al. (2002) are used. However, the reference points and the coefficients of the search directions are chosen such that the objective function value of the scalarization captures the distance information. Hence, the unsupported points are measured in the same way as the supported ones and it is possible to insert the point with the largest deviation from the current approximation regardless whether the point is supported or not.

The approach of **Eichfelder (2006)**, see also **Eichfelder (2009a)**, is intended to generate an equidistant representation of the nondominated set of continuous multi-criteria optimization problems. If more than two criteria are considered, the method is subdivided into two stages. In the first stage, a coarse initial representation is computed with the help of the ε-constraint method with an a priori parameter selection. In the second stage, a refinement with equidistant points is constructed around points of the initial representation. Note that no first stage is needed for bicriteria problems, but that the method constructs points with almost equal spacing based on the two lexicographic minima as described in Section 3.2.

Sylva and Crema (2007) present a variant of the method of Sylva and Crema (2004) and apply it to generate incomplete representations. In each iteration, the point which maximizes the infinity-norm distance from the dominated set, i.e., the set dominated by all previously generated points, is computed.

Part I

Theoretical Findings for Adaptive Parametric Algorithms

4 Adaptive Parameters for Scalarizations with Augmentation

4.1 Introduction

In this chapter we consider the computation of adaptive parameters for scalarizations with a so-called augmentation term. In particular, we study the weighted Tchebycheff method for which the concept of augmentation was introduced originally.

The weighted Tchebycheff method and its variations are among the most common scalarization methods in multiple criteria optimization, see, e.g., Steuer and Choo (1983), Eswaran et al. (1989), Solanki (1991), Alves and Climaco (2000), Schandl et al. (2001), Schandl et al. (2002), Klamroth et al. (2002), Sayın and Kouvelis (2005), Ralphs et al. (2006), Bozkurt et al. (2010) and Luque et al. (2010). As already stated in Section 2.3, every nondominated point of a general multiple criteria optimization problem can be generated with the help of the weighted Tchebycheff method. This explicitly includes non-convex and discrete problems which may have a large percentage of unsupported nondominated points, which are not computable by a weighted sum method. Since the decision maker can easily interpret the reference point and the distance information, the method is frequently used within interactive approaches, see, e.g., Miettinen et al. (2006) or the survey of Alves and Climaco (2007). In the context of multiple criteria combinatorial optimization problems, Tchebycheff scalarizations often lead to NP-complete problems, see, e.g., Murthy and Her (1992). Nevertheless, Tchebycheff scalarizations are also used in this context due to their general applicability.

As the outcomes of the weighted Tchebycheff method may be weakly nondominated, two variants are proposed in Steuer and Choo (1983), the lexicographic and the augmented weighted Tchebycheff method, see problems (2.29) and (2.30). While the main advantage of the lexicographic method is that no additional parameter has to be chosen, its drawback is that two single optimization problems have to be solved to determine one nondominated point, which means a higher computational effort. In the augmented approach only one scalarization is solved per nondominated point, but the additional augmentation parameter has to be chosen appropriately.

In the literature it is typically assumed that the augmentation parameter can be chosen 'sufficiently small'. In Steuer (1986) it is recommended to choose ρ between 0.0001 and 0.01. However, when the augmentation parameter is not related to the given data, either nondominated points might be missed (when ρ is not small enough), or numerical problems may occur (when ρ is too small), see, e.g., Steuer (1986), Solanki (1991) or Ralphs et al. (2006). Based on the complete set of nondominated points, Steuer and Choo (1983) calculate an upper bound on ρ such that for all smaller values of ρ all nondominated points can be found. However, this theoretical upper bound has no practical application if the nondominated set is not known in advance.

Given a discrete problem with a finite nondominated set we show how all parameters of the augmented weighted Tchebycheff method (2.30), including the augmentation parameter ρ, can be computed in an adaptive way such that the entire nondominated set can be computed within a parametric algorithm. In particular, for each subproblem solved within a parametric algorithm we derive a (local) upper bound on the value of ρ such that for all values of ρ smaller than this upper bound either a new nondominated point is generated or a certain part of the outcome space can be discarded from further consideration. Moreover, among all feasible choices of ρ, we study how ρ can be chosen largest in order to avoid numerical difficulties. While an appropriate parameter scheme has already been studied for the bicriteria case in Dächert et al. (2012), the analysis derived in the following is carried out for an arbitrary number of criteria. Besides the augmented weighted Tchebycheff method we also study a generalization of it as well as an augmented ε-constraint method.

The remainder of this chapter is organized as follows. In Section 4.1.1 we discuss the geometry of the contour of an augmented weighted Tchebycheff norm and motivate our construction in the discrete case. In Section 4.2, the main section of this chapter, we propose a new adaptive parameter scheme for the augmented weighted Tchebycheff method. This includes the computation of feasible upper bounds on ρ which, on the one hand, are used to show maximality of ρ and, on the other hand, serve to determine appropriate values for the practical application. In Section 4.3 we study a generalization of the augmented weighted Tchebycheff method which permits a more intuitive understanding of the parameters. Furthermore, we discuss the well-known relation between augmentation parameters and prescribed trade-offs on pairs of objectives. The use of trade-offs permits an adaptive parameter computation for continuous problems. In Section 4.4 we consider an augmented ε-constraint method. We review existing formulations and propose an adaptive parameter scheme analogous to the construction in Section 4.2. In Section 4.5 we summarize the content of

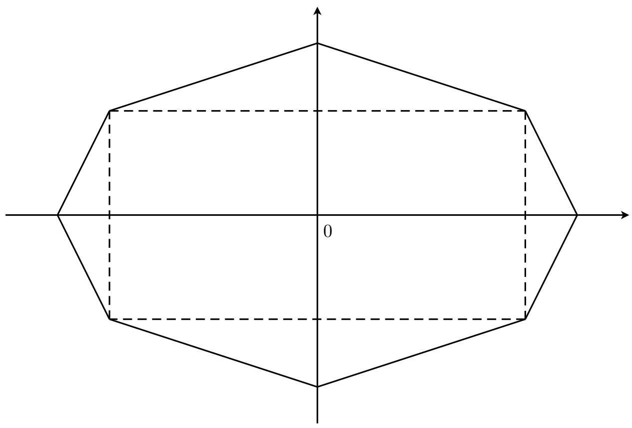


Figure 4.1: Contour of an augmented (solid) and a weighted Tchebycheff norm (dashed) for $m = 2$

this chapter and suggest further directions of research.

4.1.1 Geometry of the Augmented Weighted Tchebycheff Norm

We first state some well-known geometrical properties of the augmented weighted Tchebycheff norm which are the basis of our construction. We use the following notation:

Definition 4.1 (Weighted Tchebycheff Norm). *The weighted Tchebycheff norm $\|\cdot\|_{w,\infty} : \mathbb{R}^m \to \mathbb{R}_+$ is defined by*

$$\|z\|_{w,\infty} := \max_{i=1,\dots,m} \{w_i |z_i|\}, \tag{4.1}$$

where $w \in \mathbb{R}^m_>$ and $\sum_{i=1}^m w_i = 1$.

Definition 4.2 (Augmented Weighted Tchebycheff Norm). *The augmented weighted Tchebycheff norm $\|\cdot\|_{w,\rho} : \mathbb{R}^m \to \mathbb{R}_+$ is defined by*

$$\|z\|_{w,\rho} := \max_{i=1,\dots,m} \{w_i |z_i|\} + \rho \, \|z\|_1, \tag{4.2}$$

where $w \in \mathbb{R}^m_+$, $\sum_{i=1}^m w_i = 1$, $\rho \geq 0$ and $w_i + \rho > 0$, $i = 1, \dots, m$.

It is well-known (Steuer, 1986) that the contour $\mathcal{L}_\alpha := \{z \in \mathbb{R}^m : \|z\|_{w,\rho} = \alpha\}$ of the augmented weighted Tchebycheff norm with respect to a certain level $\alpha \in \mathbb{R}_+$ is piecewise linear for all $\alpha > 0$ and symmetric with respect to the origin. For an illustration in the bicriteria case, see Figure 4.1. In the classic (augmented) weighted Tchebycheff method, the ideal point or an utopian point is taken as the point from which the distance to the feasible set is minimized. Consequently, as already discussed in Section 2.3 and stated, e.g., in Steuer (1986), the absolute values in (4.2) can be dropped. Geometrically, this implies that only the portion of the contour in the positive orthant is relevant.

In order to understand how the weights w and the augmentation parameter ρ influence the shape of the contour, consider $m = 2$ and let $z \in \mathbb{R}^2_+$, $w \in \mathbb{R}^2_>$ and $\rho > 0$. Then

$$\|z\|_{w,\rho} = \max\{w_1 z_1, w_2 z_2\} + \rho(z_1 + z_2) = \alpha$$

is equivalent to

$$\begin{cases} z_2 = [\alpha - (w_1 + \rho) z_1] \cdot \rho^{-1}, & \text{if } w_1 z_1 \geq w_2 z_2, \\ z_2 = (\alpha - \rho z_1)\,(w_2 + \rho)^{-1}, & \text{otherwise.} \end{cases}$$

Thus, the contour of an augmented weighted Tchebycheff norm in the positive orthant is represented by segments of the two lines

$$h_1(t) = \frac{\alpha}{w_2 + \rho} - \frac{\rho}{w_2 + \rho} \cdot t \quad \text{and} \quad h_2(t) = \frac{\alpha}{\rho} - \frac{w_1 + \rho}{\rho} \cdot t,$$

which are depicted in Figure 4.2 for $\alpha = 1$. For arbitrary $m \geq 2$ and $w \in \mathbb{R}^2_>$, the contour is composed of m hyperplanes of dimension $m - 1$ in each orthant. The intersection of these m hyperplanes, which we call inflection point, and which we denote by z^q in the following, is characterized as the point for which $w_1 z_1 = w_2 z_2 = \cdots = w_m z_m$ holds. For a small value of ρ the hyperplanes are nearly parallel to the coordinate axes, and for $\rho = 0$ the contour of the weighted Tchebycheff norm is obtained. Note that the slope of the hyperplanes depends not only on ρ but also on the weights w, see the definition of $h_1(t)$ and $h_2(t)$ above. This dependence is removed in the modified weighted Tchebycheff norm, cf. Problem (2.31). For $m = 2$, the two lines of which the contour is constituted in the positive orthant have a slope of $(1 + \rho)/\rho$ and $\rho/(1 + \rho)$, respectively, i.e., the slope only depends on ρ.

The augmented weighted Tchebycheff method consists in minimizing the distance between the reference point $r := z^U$ and the set of feasible outcomes. Analogous to (2.28), for $w \in \mathbb{R}^m_>$ the augmented weighted Tchebycheff method (2.30) can be equivalently written as

$$\begin{aligned} \min \quad & \lambda + \rho \sum_{j=1}^{m} \left(f_j(x) - z_j^U \right) \\ \text{s.t.} \quad & z_i^U + \tfrac{1}{w_i} \lambda \geq f_i(x), \quad i = 1, \ldots, m, \\ & \lambda \in \mathbb{R}, x \in X. \end{aligned} \tag{4.3}$$

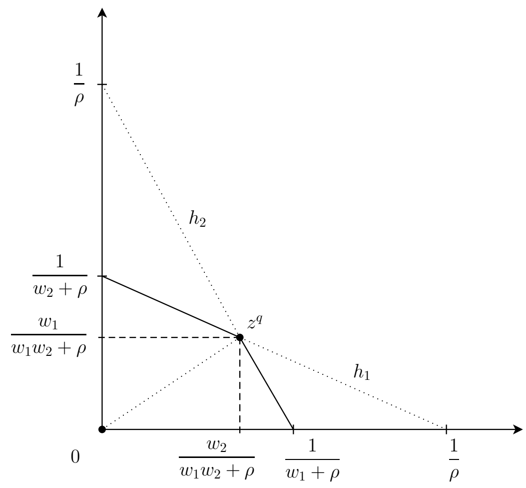


Figure 4.2: Geometry of the unit ball ($\alpha = 1$) of the augmented weighted Tchebycheff norm restricted to the positive orthant for $m = 2$

Geometrically, starting from the given reference point z^U, the contour $\mathcal{L}_\alpha$ (with fixed parameters w and ρ) is shifted along the direction $d := (w_1^{-1}, \ldots, w_m^{-1})^\top$ towards $f(X)$ until the intersection of $f(X)$ and $\mathcal{L}_\alpha$ becomes non-empty for some $\alpha > 0$. This is illustrated in Figure 4.3. Note that the point where $f(X)$ and $\mathcal{L}_\alpha$ intersect with minimal α does not necessarily coincide with the inflection point z^q of the contour, which is also illustrated in Figure 4.3.

4.1.2 Motivation in the Discrete Bicriteria Case

Let a discrete multicriteria optimization problem be given. Without loss of generality, we restrict the general discrete case to the integer-valued case in the following. We motivate our approach with the help of a bicriteria problem. Let $z^1, z^2 \in Z, Z \subseteq \mathbb{Z}^2$ denote two nondominated points of (2.1) where, w.l.o.g., $z_1^1 < z_1^2$. Furthermore, let $r \in Z^2$ denote the local ideal point with respect to $\{z^1, z^2\}$. Without loss of generality, we assume $r = 0$ in the following, as for arbitrary $m \geq 2$ we can apply the linear mapping $\psi : \mathbb{Z}^m \to \mathbb{Z}^m$, $z_i \mapsto z_i - r_i$, for $i = 1, \ldots, m$ to the given problem such that the reference point coincides with the origin. Thereby, $z^1 = (0, z_2^1)^\top$ and $z^2 = (z_1^2, 0)^\top$ hold. Let

$$B(u) := \{z \in \mathbb{R}^m_+ : z < u\}$$

with $u := (z_1^2, z_2^1)^\top$ be the region 'between' z^1 and z^2 in which further nondominated points are searched for. We assume that $u_1, u_2 \geq 2$ holds, as, otherwise, $B(u)$ can not contain further nondominated points due to the integrality assumption on the

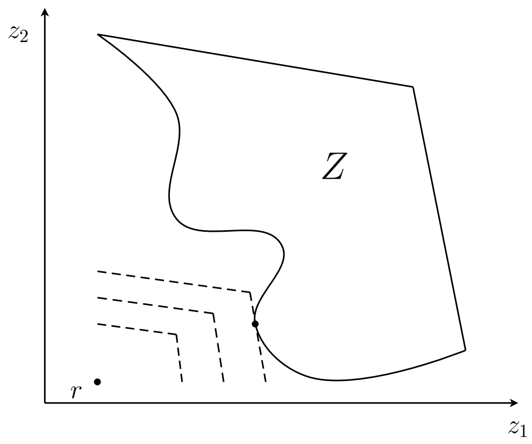


Figure 4.3: Working principle of the augmented weighted Tchebycheff method for $m = 2$

nondominated points. For the same reason the set

$$\tilde{B}(u) := \{z \in B(u) : \tilde{u}_1 < z_1 < u_1\} \cup \{z \in B(u) : \tilde{u}_2 < z_2 < u_2\},$$

with $\tilde{u} := u - e = (z_1^2 - 1, z_2^1 - 1)^\top$ does not contain further nondominated points. The set is depicted as the shaded region in Figure 4.4. Hence, if $B(u)$ contains a nondominated point, then this point must be contained in

$$B(u) \backslash \tilde{B}(u) = \{z \in B(u) : z \leqq \tilde{u}\}.$$

By computing the parameters w and ρ of an augmented weighted Tchebycheff norm such that every possibly existing nondominated point $z \in B(u)$ has a strictly smaller

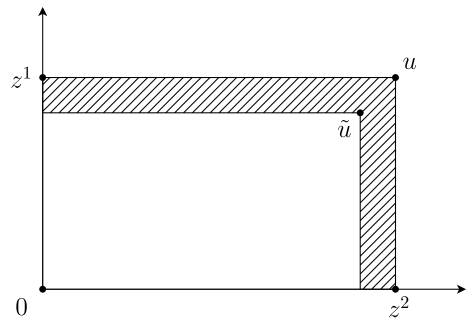


Figure 4.4: Empty region (shaded) between two nondominated points z^1 and z^2 of an integer-valued, bicriteria problem; due to the integrality assumption the sets defined by u and $\tilde{u} = u - e$ cannot contain further nondominated points

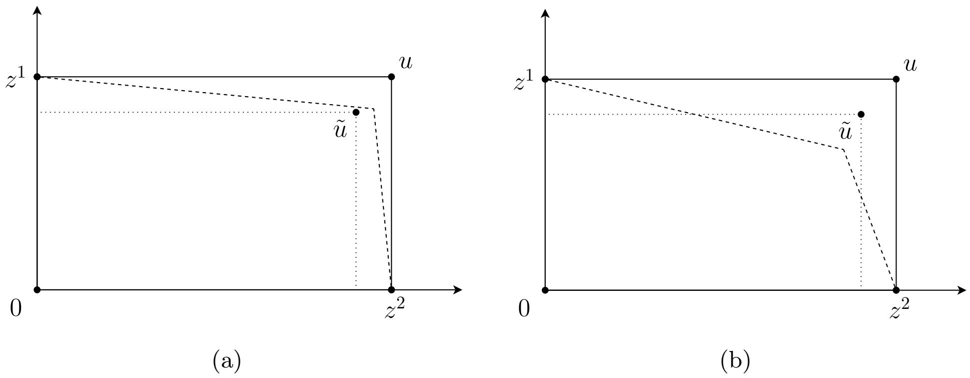


Figure 4.5: Two contours of an augmented weighted Tchebycheff norm (dashed curve); in (a) the point $\tilde{u}$ has a smaller level than z^1 and z^2, in (b) not, hence, (4.6) is not satisfied

level than z^1 and z^2, i.e.,

$$\|z\|_{w,\rho} < \min\{\|z^1\|_{w,\rho}, \|z^2\|_{w,\rho}\} \tag{4.4}$$

holds for all $z \in Z_N \cap B(u)$, and by solving an augmented weighted Tchebycheff problem with these parameters, we can guarantee that a nondominated point in $B(u)$ is detected, whenever there exists one. As we do not know possible nondominated points in $B(u)$ before solving the corresponding subproblem(s), we replace (4.4) by a 'worst-case' scenario which consists in the situation that $\tilde{u} = u - e$ is a nondominated point. Hence, we compute parameters of an augmented weighted Tchebycheff norm such that

$$\|\tilde{u}\|_{w,\rho} < \min\{\|z^1\|_{w,\rho}, \|z^2\|_{w,\rho}\} \tag{4.5}$$

holds. Obviously, every nondominated point in $B(u)\setminus\{\tilde{u}\}$ has a strictly smaller level than $\tilde{u}$ with respect to the constructed augmented weighted Tchebycheff norm. This is the basic idea of our construction. Since the strict inequality in (4.5) is not practical, in particular not with regard to the determination of a maximal value of ρ, we first construct the parameters such that

$$\|\tilde{u}\|_{w,\rho} \leq \min\{\|z^1\|_{w,\rho}, \|z^2\|_{w,\rho}\} \tag{4.6}$$

holds. By a slight perturbation of the values of z^1 and z^2 we can later achieve that (4.5) holds. Figure 4.5 shows an example of two level curves of an augmented weighted Tchebycheff norm, one satisfying the condition in (4.6), the other not.

Among the feasible choices of the parameters satisfying (4.6) we further aim at maximizing the value of ρ in order to avoid numerical difficulties that may arise when too small values for ρ are used. In the second part of our analysis, we will discuss the question on how large a value of ρ can be selected at maximum such that finding all nondominated points of the given problem can still be guaranteed.

4.2 Parameters of the Augmented Weighted Tchebycheff Norm

For a general number of objectives $m \geq 2$, let a vector $u \in \mathbb{N}^m$ be given, where, without loss of generality, $u_1 \geq u_2 \geq \cdots \geq u_m \geq 2$. Let $u^1, \ldots, u^m \in \mathbb{N}_0^m$ be m vectors with $u^i := (0, \ldots, 0, u_i, 0, \ldots, 0)^\top$, i.e., with the only non-zero entry $u_i \geq 2$ at the i-th position, $i = 1, \ldots, m$. These m points denote extreme points of the box $B(u)$ to be considered. Furthermore, let $U := \sum_{j=1}^m \frac{1}{u_j}$.

As motivated in Section 4.1.2 for the bicriteria case, we construct the contour of an augmented weighted Tchebycheff norm such that

$$\|\tilde{u}\|_{w,\rho} \leq \min_{i=1,\ldots,m} \{\|u^i\|_{w,\rho}\} \tag{4.7}$$

with $\tilde{u} := u - e$, $e = (1, \ldots, 1)^\top$. In order to simplify the analysis, we first consider the special case

$$\|\tilde{u}\|_{w,\rho} = \|u^1\|_{w,\rho} = \|u^2\|_{w,\rho} = \cdots = \|u^m\|_{w,\rho} \tag{4.8}$$

in the next subsection.

4.2.1 Feasible Parameter Choice

In contrast to the augmented weighted Tchebycheff norm, the construction of a contour of a weighted Tchebycheff norm that comprises all m points $u^i, i = 1, \ldots, m$, or, equivalently, whose vertex is u, is well-known.

Lemma 4.3 (Parameters of the Weighted Tchebycheff Norm (Steuer and Choo, 1983))**.** *Let $\alpha \in \mathbb{R}_+$. It holds that the points $u^i, i = 1, \ldots, m$, lie on a common level curve of a weighted Tchebycheff norm with level α, i.e., $\|u^i\|_{w,\infty} = \alpha$ for all $i = 1, \ldots, m$, if and only if*

$$\alpha = U^{-1} \quad \wedge \quad w_i = \frac{1}{u_i} \cdot U^{-1} \quad \forall\, i = 1, \ldots, m.$$

Proof. By definition it holds for all $i = 1, \ldots, m$ that $\|u^i\|_{w,\infty} = w_i u_i$, thus, $\|u^i\|_{w,\infty} = \alpha$ if and only if $w_i = \alpha \cdot (u_i)^{-1}$. Using $\sum_{j=1}^m w_j = 1$ yields $1 = \alpha \cdot \sum_{j=1}^m (u_j)^{-1}$, so $\alpha = U^{-1}$ holds. Then $w_i = (u_i)^{-1} \cdot U^{-1}$. The converse statement is obvious. □

Remark 4.4. *For $m = 2$ we obtain*

$$w_1 = \frac{u_2}{u_1 + u_2} \quad \textit{and} \quad w_2 = \frac{u_1}{u_1 + u_2}, \tag{4.9}$$

see also Solanki (1991), Sayın and Kouvelis (2005) or Ralphs et al. (2006).

Note that the level α and the parameters w_i, $i = 1, \ldots, m$, are uniquely defined by the coordinates of the given points u^i, $i = 1, \ldots, m$, in case of a weighted Tchebycheff norm. This property no longer holds true when the augmented weighted Tchebycheff norm is considered. In more detail, for fixed u^i, $i = 1, \ldots, m$, the weights w_i, $i = 1, \ldots, m$, and the parameter ρ can be chosen dependent on an appropriately chosen value of α.

Theorem 4.5 (Parameters of the Augmented Weighted Tchebycheff Norm)**.** *Let $\alpha \in \mathbb{R}_+$. It holds that the points $u^i, i = 1, \ldots, m$, lie on a common level curve of an augmented weighted Tchebycheff norm with level α, i.e., $\|u^i\|_{w,\rho} = \alpha$ for all $i = 1, \ldots, m$, if and only if*

$$\rho(\alpha) = \frac{\alpha U - 1}{m} \quad \wedge \quad w_i(\alpha) = \frac{\alpha}{u_i} - \rho(\alpha) \quad \forall\, i = 1, \ldots, m, \tag{4.10}$$

where $\alpha \in I$ with

(i) $I := \left[U^{-1}, \infty\right)$, if $u_1 = u_m$, and

(ii) $I := \left[U^{-1}, \frac{u_1}{u_1 U - m}\right]$ otherwise, i.e., if $u_1 > u_m$.

Proof. By definition, it holds that $\|u^i\|_{w,\rho} = \alpha$ if and only if $(w_i + \rho)u_i = \alpha$ for all $i = 1, \ldots, m$, which is equivalent to $w_i = \frac{\alpha}{u_i} - \rho$, as $u_i > 0$. As $\sum_{i=1}^m w_i = 1$, it follows that $1 = \alpha U - \rho m$, or, equivalently,

$$\rho(\alpha) = \frac{\alpha U - 1}{m}, \tag{4.11}$$

and, thus,

$$w_i(\alpha) = \frac{\alpha}{u_i} - \frac{\alpha U - 1}{m},\ i = 1, \ldots, m. \tag{4.12}$$

However, the parameters $w_i(\alpha)$, $i = 1, \ldots, m$, and $\rho(\alpha)$ only define a valid augmented weighted Tchebycheff norm, if additionally the conditions $\rho(\alpha) \geq 0$, $w_i(\alpha) \geq 0$ and $w_i(\alpha) + \rho(\alpha) > 0$, $i = 1, \ldots, m$, are satisfied. Hence, we have to restrict the value of α such that these conditions are valid. It holds that

$$\rho(\alpha) \geq 0 \iff \alpha \geq U^{-1}, \tag{4.13}$$

which implies the lower bound on the level α. For all $i = 1, \dots, m$ it holds that

$$w_i(\alpha) \geq 0 \iff \alpha\,(m - u_i \cdot U) \geq -u_i. \tag{4.14}$$

For all $i = 1, \dots, m$ for which $m - u_i U \geq 0$ holds, this inequality does not impose a restriction since $u_i > 0$ for all $i = 1, \dots, m$. For all $i = 1, \dots, m$ for which $m - u_i U < 0$ holds, we obtain the upper bound

$$\alpha \leq \frac{u_i}{u_i U - m}.$$

If $u_1 = u_m$, then $m - u_i U = m - u_1 \cdot m \cdot \frac{1}{u_1} = 0$. Therefore, no upper bound on α is imposed in this case. If $u_1 > u_m$, it holds that $m - u_1 U = m - \sum_{i=1}^{m} \frac{u_1}{u_i} < 0$ due to the general assumption $u_1 \geq u_j$ for all $j = 2, \dots, m$. Furthermore,

$$\min_{i=1,\dots,m} \left\{ \frac{u_i}{u_i U - m} \;:\; u_i U - m > 0 \right\} = \min_{i=1,\dots,m} \left\{ \frac{1}{U - \frac{m}{u_i}} \;:\; u_i U - m > 0 \right\} = \frac{1}{U - \frac{m}{u_1}},$$

which is, thus, the smallest upper bound on α if $u_1 > u_m$. Finally, $w_i(\alpha) + \rho(\alpha) = \frac{\alpha}{u_i} > 0$ holds for all $i = 1, \dots, m$ since $\alpha \geq U^{-1} > 0$. The converse results stated in the theorem are obvious. $\square$

Remark 4.6.

1. *For fixed α it follows from $u_1 \geq u_2 \geq \dots \geq u_m$ that $w_1 \leq w_2 \leq \dots \leq w_m$. If $u_1 = u_m$, then $w_1 = \dots = w_m = \frac{1}{m}$.*
2. *The lower bound for α in Theorem 4.5 and the uniquely determined value of α in Lemma 4.3 coincide, thus, $\alpha = U^{-1}$ corresponds to the case $\rho = 0$. If $u_1 > u_m$ and $\alpha = u_1/(u_1 U - m)$, then $w_1 = 0$. If $u_1 = u_m$, then the parameter ρ tends to infinity when $\alpha \to \infty$ is considered. In this case, the influence of the l_∞-norm vanishes and in the limit, the distance is measured by a pure l_1-norm.*

For fixed and finite $\alpha \in I$, the contour

$$\mathcal{L}_\alpha^+ := \{z \in \mathbb{R}_+^m : \|z\|_{w,\rho} = \alpha\}$$

can be represented as the union of the m sets

$$\begin{aligned} \mathcal{L}_{\alpha,k}^+ \;&:=\; \{z \in \mathbb{R}_+^m : \|z\|_{w,\rho} = \alpha, w_k z_k \geq w_i z_i \;\forall\, i = 1, \dots, m\} \\ &=\; \{z \in \mathbb{R}_+^m : (w_k + \rho) z_k + \rho \sum_{\substack{j=1,\dots,m,\\ j \neq k}} z_j = \alpha, w_k z_k \geq w_i z_i \;\forall\, i = 1, \dots, m\}, \end{aligned}$$

$k = 1, \dots, m$, which are subsets of hyperplanes of dimension $m - 1$ for $w \in \mathbb{R}_>^m$. The point lying at the intersection of these m hyperplanes, the inflection point $z^q(\alpha)$, can be characterized as follows.

Lemma 4.7 (Inflection point of the contour)**.** *Let $w \in \mathbb{R}^m_>$, and let $z^q(\alpha) \in \mathbb{R}^m$ denote the intersection point of the m sets $\mathcal{L}^+_{\alpha,k}$, $k = 1, \dots, m$. Then the coordinates of $z^q(\alpha)$ are given by*

$$z^q_i(\alpha) = \frac{\alpha}{1+\rho W} \cdot \frac{1}{w_i} \quad \forall\, i = 1, \dots, m \tag{4.15}$$

with $W := \sum_{j=1}^m \frac{1}{w_j}$.

Proof. As $z^q(\alpha) \in \bigcap_{k=1,\dots,m} \mathcal{L}^+_{\alpha,k}$, it holds that $(w_k+\rho) z^q_k(\alpha) + \rho \sum_{i=1,\, i\neq k}^m z^q_i(\alpha) = \alpha$ for all $k = 1, \dots, m$. We claim that the solution of the linear system

$$A_m z^q(\alpha) = \alpha \cdot e \tag{4.16}$$

with

$$A_m := \begin{pmatrix} w_1+\rho & \rho & \cdots & \cdots & \rho \\ \rho & w_2+\rho & \rho & \cdots & \rho \\ \rho & \rho & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \rho \\ \rho & \cdots & \rho & \rho & w_m+\rho \end{pmatrix}$$

and $e = (1, \dots, 1)^\top$ is given by (4.15). First, we show that (4.16) admits a unique solution since A_m has full rank. Therefore, we prove the more general result that $\det(A_m) > 0$ holds if $w \in \mathbb{R}^m_+$, $w_i = 0$ for at most one index $i \in \{1, \dots, m\}, \rho \geq 0$ and $w_i + \rho > 0$ for all $i = 1, \dots, m$. First, consider $\rho = 0$. Then $w_i > 0$ for all $i = 1, \dots, m$ must hold and, thus, $\det(A_m) = \Pi_{i=1}^m w_i > 0$. The case $\rho > 0$ is shown by induction on m :

$\underline{m=1}$: $\det(A_1) = w_1 + \rho > 0$

$\underline{m \to m+1}$: We assume that $\det(A_m) > 0$ and show that $\det(A_{m+1}) > 0$ holds. Let $w_1, \dots, w_{m+1} \geq 0$ with at most one $w_j = 0$. Furthermore, define $w_0 := 0$. Then,

$$\det(A_{m+1}) = \det \begin{pmatrix} w_1 & -w_2 & 0 & \cdots & 0 \\ \rho & w_2+\rho & \rho & \cdots & \rho \\ \vdots & \ddots & \ddots & & \rho \\ \rho & \cdots & \cdots & \rho & w_{m+1}+\rho \end{pmatrix} = w_1 \det(B_m) + w_2 \det(C_m)$$

with

$$B_m := \begin{pmatrix} w_2+\rho & \rho & \cdots & \rho \\ \rho & w_3+\rho & \cdots & \rho \\ \vdots & \ddots & \ddots & \rho \\ \rho & \cdots & \rho & w_{m+1}+\rho \end{pmatrix}$$

and

$$C_m := \begin{pmatrix} \rho & \rho & \cdots & \rho \\ \rho & w_3+\rho & \cdots & \rho \\ \vdots & \ddots & \ddots & \rho \\ \rho & \cdots & \rho & w_{m+1}+\rho \end{pmatrix}.$$

Note that $\det(B_m) > 0$ holds by the induction hypothesis.

i) If $w_1 = 0$, then $w_2, \ldots, w_{m+1} > 0$ follows. With $w_0 = 0$ we have

$$C_m = \begin{pmatrix} w_0+\rho & \rho & \cdots & \rho \\ \rho & w_3+\rho & \cdots & \rho \\ \vdots & \ddots & \ddots & \rho \\ \rho & \cdots & \rho & w_{m+1}+\rho \end{pmatrix}.$$

Using the induction hypothesis we see that $\det(C_m) > 0$. Hence, it holds that $\det(A_{m+1}) = w_2 \det(C_m) > 0$.

ii) If $w_1 > 0$, then $w_1 \det(B_m) > 0$. Consequently, $\det(A_{m+1}) = w_1 \det(B_m) > 0$ if, additionally, $w_2 = 0$ holds. The same holds if $w_2 > 0$ but $w_j = 0$ for some $j = 3, \ldots, m+1$, as then C_m contains two rows with all entries being ρ, thus, $\det(C_m) = 0$. If $w_i > 0$ for all $i = 2, \ldots, m$, then $\det(B_m) > 0$ and $\det(C_m) > 0$ by induction, thus, $\det(A_{m+1}) > 0$.

Therefore, $\det(A_m) > 0$ holds for arbitrary m, which implies that the solution of (4.16) is unique. Hence, it suffices to verify (4.15). For every $k = 1, \ldots, m$ it holds that

$$\begin{aligned} w_k z_k^q(\alpha) + \rho \sum_{i=1}^m z_i^q(\alpha) &= w_k \cdot \frac{\alpha}{1+\rho W} \cdot \frac{1}{w_k} + \rho \sum_{i=1}^m \frac{\alpha}{1+\rho W} \cdot \frac{1}{w_i} \\ &= \frac{\alpha}{1+\rho W}\left[1 + \rho \sum_{i=1}^m \frac{1}{w_i}\right] = \alpha. \end{aligned}$$

□

We have shown that for a given level α the inflection point $z^q(\alpha)$ is uniquely determined. For the case $m = 2$, a visualization of the curve γ which describes the location of the inflection points $z^q(\alpha)$ for all feasible values $\alpha \in I$ is given in Figure 4.6. For an explicit representation of γ for $m = 2$ we refer to Dächert et al. (2012).

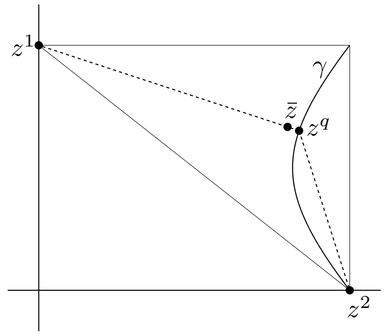


Figure 4.6: Example for the curve γ for $u_1 = z_1^2 = 5$ and $u_2 = z_2^1 = 3$. γ describes the location of the inflection point $z^q(\alpha)$ for $\alpha \in I = [1.875; 7.5]$.

Summarizing the discussion above, the formulas for w and ρ stated in Theorem 4.5 imply that we can compute parameters of a valid augmented weighted Tchebycheff norm such that all $u^i, i = 1, \dots, m$, lie on the same level curve. However, our derivations show that under this assumption the inflection point of the contour cannot be chosen arbitrarily, but lies on a curve dependent on u and α. In particular, this implies that the parameters cannot be set such that the point $\tilde{u} = (u_1 - 1, u_2 - 1, \dots, u_m - 1)^\top$ is the inflection point of this level curve, in general. This can be seen in Figure 4.6 where the point $\tilde{u} = \bar{z} = (4, 2)^\top$ is not an element of $z^q(\alpha), \alpha \in I$.

This particular point $\tilde{u}$ is now included into the construction, i.e., in the following we compute parameters w and ρ such that

$$\|\tilde{u}\|_{w,\rho} = \|u^1\|_{w,\rho} = \|u^2\|_{w,\rho} = \cdots = \|u^m\|_{w,\rho}$$

holds. As mentioned above, typically $\tilde{u} \neq z^q(\alpha)$, $\alpha \in I$, but $\tilde{u}$ lies on only one of the hyperplanes which constitute the contour. By definition, $\tilde{u}$ lies on the hyperplane $\mathcal{L}^+_{\alpha,l}$ with index $l \in \{1, \dots, m\}$ for which $w_l(\alpha)\tilde{u}_l \geq w_i(\alpha)\tilde{u}_i$ for all $i = 1, \dots, m$ holds. This particular hyperplane, i.e., the index $l \in \{1, \dots, m\}$, will be specified later. First, we treat l as a parameter.

Lemma 4.8. *Let $m \geq 3$ or $u_1 > u_m$ hold. Furthermore, let $\alpha \in \mathbb{R}_+$, and let $\tilde{\sigma} := \sum_{i=1}^m \tilde{u}_i$. If $\|\tilde{u}\|_{w,\rho} = \alpha$ and $\|u^i\|_{w,\rho} = \alpha$ for all $i = 1, \dots, m$ hold, then*

$$\alpha = \frac{u_l(\tilde{\sigma} - \tilde{u}_l)}{u_l U(\tilde{\sigma} - \tilde{u}_l) - m}, \tag{4.17}$$

where $l \in \{1, \dots, m\}$ such that $w_l(\alpha)\tilde{u}_l \geq w_i(\alpha)\tilde{u}_i$ for all $i = 1, \dots, m$.

Proof. We recall from Theorem 4.5 that $\|u^i\|_{w,\rho} = \alpha$ for all $i = 1, \dots, m$ if and only if

$$w_i(\alpha) = \frac{\alpha(m - u_i U) + u_i}{m u_i} \quad \text{and} \quad \rho(\alpha) = \frac{\alpha U - 1}{m}$$

hold for $\alpha \in I$ as specified in Theorem 4.5. Assume that $l \in \{1, \dots, m\}$ is chosen such that $w_l(\alpha)\tilde{u}_l = \max_{i=1,\dots,m}\{w_i(\alpha)\tilde{u}_i\}$ is satisfied. Then

$$\|\tilde{u}\|_{w,\rho} = w_l(\alpha)\tilde{u}_l + \rho(\alpha)\sum_{i=1}^{m}\tilde{u}_i = \frac{\alpha(m - u_l U) + u_l}{m u_l} \cdot \tilde{u}_l + \frac{\alpha U - 1}{m} \cdot \tilde{\sigma}$$

and the following equivalences are valid:

$$\begin{aligned}
\|\tilde{u}\|_{w,\rho} = \alpha \iff & \alpha\left[\frac{(m - u_l U)\tilde{u}_l}{m u_l} + \frac{U}{m}\tilde{\sigma} - 1\right] = \frac{1}{m}\tilde{\sigma} - \frac{1}{m}\tilde{u}_l \\
\iff & \alpha\left(m\tilde{u}_l - u_l U\tilde{u}_l + U u_l\tilde{\sigma} - m u_l\right) = u_l(\tilde{\sigma} - \tilde{u}_l) \\
\iff & \alpha\underbrace{\left[u_l U(\tilde{\sigma} - \tilde{u}_l) - m\right]}_{=:A} = u_l(\tilde{\sigma} - \tilde{u}_l).
\end{aligned}$$

If $m = 2$ and $u_1 = u_2 = 2$, then $A = 0$. Since $u_l(\tilde{\sigma} - \tilde{u}_l) = 2 \neq 0$ there does not exist a level α such that $\|\tilde{u}\|_{w,\rho} = \alpha$ in this case. In all other cases, $A > 0$ holds: If there exists some $j \neq l$ such that $u_j \geq 3$, then

$$\tilde{\sigma} - \tilde{u}_l = \sum_{\substack{i=1,\dots,m,\\ i\neq l}} \tilde{u}_i \geq m,$$

and with

$$u_l U = 1 + \sum_{\substack{i=1,\dots,m\\ i\neq l}} \underbrace{\frac{u_l}{u_i}}_{>0} > 1$$

it follows that $A > 0$. Otherwise, i.e., if $u_i = 2 \;\forall\, i = 1, \dots, m, i \neq l$, then $u_l U \geq 2$ and $\tilde{\sigma} - \tilde{u}_l = m - 1$, thus, $A \geq m - 2$. We see that $A > 0$ if $m \geq 3$ or if $m = 2$ and $u_l > 2$, as then $u_l U > 2$. So it holds that

$$\alpha = \frac{u_l(\tilde{\sigma} - \tilde{u}_l)}{u_l U(\tilde{\sigma} - \tilde{u}_l) - m}$$

in all cases except $m = 2$ and $u_1 = u_2 = 2$. □

The value of α derived in the previous lemma is only formal, i.e., it is not clear whether $\alpha \in I$. However, the lower bound can be easily verified: As $u_l(\tilde{\sigma} - \tilde{u}_l) > 0$,

$$\alpha = \left(U - \frac{m}{u_l(\tilde{\sigma} - \tilde{u}_l)}\right)^{-1} > U^{-1}. \tag{4.18}$$

It remains to show that the upper bound is valid in the case $u_1 > u_m$. Moreover, the index l has to be specified, i.e., the hyperplane on which $\tilde{u}$ lies. Inserting (4.17) into the formulas for $w_i(\alpha)$ and $\rho(\alpha)$ derived in Theorem 4.5 directly yields

Corollary 4.9. *Let the assumption of Lemma 4.8 hold and let α be defined according to* (4.17)*. Then*

$$w_i = \frac{u_l(\tilde{\sigma} - \tilde{u}_l) - u_i}{u_i(u_l U(\tilde{\sigma} - \tilde{u}_l) - m)}, \; i = 1, \ldots, m, \quad \text{and} \quad \rho = \frac{1}{u_l U(\tilde{\sigma} - \tilde{u}_l) - m}$$

hold, where $l \in \{1, \ldots, m\}$ such that $w_l(\alpha)\tilde{u}_l \geq w_i(\alpha)\tilde{u}_i$ for all $i = 1, \ldots, m$.

Proof. A short calculation yields

$$\begin{aligned}
\rho = \rho(\alpha) = \alpha \cdot \frac{U}{m} - \frac{1}{m} &= \frac{u_l(\tilde{\sigma} - \tilde{u}_l)}{u_l U(\tilde{\sigma} - \tilde{u}_l) - m} \cdot \frac{U}{m} - \frac{1}{m} \\
&= \frac{u_l U(\tilde{\sigma} - \tilde{u}_l) - (u_l U(\tilde{\sigma} - \tilde{u}_l) - m)}{m(u_l U(\tilde{\sigma} - \tilde{u}_l) - m)} = \frac{1}{u_l U(\tilde{\sigma} - \tilde{u}_l) - m}
\end{aligned}$$

and, for all $i = 1, \ldots, m$,

$$w_i = w_i(\alpha) = \frac{\alpha}{u_i} - \rho(\alpha) = \frac{\alpha}{u_i} - \frac{1}{u_l U(\tilde{\sigma} - \tilde{u}_l) - m} = \frac{u_l(\tilde{\sigma} - \tilde{u}_l) - u_i}{u_i(u_l U(\tilde{\sigma} - \tilde{u}_l) - m)}.$$

□

We finally specify the index l for which $w_l(\alpha)\tilde{u}_l \geq w_i(\alpha)\tilde{u}_i$ for all $i = 1, \ldots, m$ is satisfied.

Lemma 4.10. *Let $m \geq 3$ or $u_1 > u_2$ be satisfied. Furthermore, let α be defined according to* (4.17)*. Then it holds*

1. *for $m = 2$ that $w_2\tilde{u}_2 \geq w_1\tilde{u}_1$, i.e., $l = 2$, and*
2. *for $m \geq 3$ that $w_1\tilde{u}_1 \geq w_i\tilde{u}_i$ for all $i = 1, \ldots, m$, i.e., $l = 1$,*

where $l \in \{1, \ldots, m\}$ such that $w_l\tilde{u}_l \geq w_i\tilde{u}_i$ for all $i = 1, \ldots, m$.

Proof. Let $\tilde{\sigma} := \sum_{i=1}^{m} \tilde{u}_i$ and $A := u_l U(\tilde{\sigma} - \tilde{u}_l) - m$ as in the proof of Lemma 4.8. From Corollary 4.9 we see that for all $i = 1, \ldots, m$

$$w_i\tilde{u}_i = \frac{\tilde{u}_i}{u_i} \cdot \frac{u_l(\tilde{\sigma} - \tilde{u}_l) - u_i}{A}.$$

Thus, using $u_i > 0$ for all $i = 1, \ldots, m$ and $A > 0$ (c.f. the proof of Lemma 4.8), the following equivalences hold for all $i = 1, \ldots, m, i \neq l$:

$$\begin{aligned}
w_l\tilde{u}_l \geq w_i\tilde{u}_i &\iff \tilde{u}_l \cdot u_i \cdot [u_l(\tilde{\sigma} - \tilde{u}_l) - u_l] \geq \tilde{u}_i \cdot u_l \cdot [u_l(\tilde{\sigma} - \tilde{u}_l) - u_i] \\
&\iff (\tilde{\sigma} - \tilde{u}_l)\, [\tilde{u}_l u_i - \tilde{u}_i u_l] \geq u_i(\tilde{u}_l - \tilde{u}_i) \\
&\iff (\tilde{\sigma} - \tilde{u}_l)\, [-u_i + u_l)] \geq u_i(u_l - u_i)
\end{aligned}$$

$$\iff (\tilde{\sigma} - \tilde{u}_l - u_i)(u_l - u_i) \geq 0. \tag{4.19}$$

We therefore investigate for which choices of l (4.19) is valid.

For $m = 2$, it holds that $\tilde{\sigma} - \tilde{u}_l - u_i = -1 < 0$ for any choice of l since $i \neq l$ is assumed. If $u_1 > u_2$, we see that (4.19) holds if and only if $i = 1$ and $l = 2$, thus, by definition of l we have $w_2\tilde{u}_2 \geq w_1\tilde{u}_1$. If $u_1 = u_2$, then $w_1\tilde{u}_1 = w_2\tilde{u}_2$.

Now, consider $m \geq 3$. In this case, as $i \neq l$,

$$\tilde{\sigma} - \tilde{u}_l - u_i = \sum_{\substack{j=1,\dots,m,\\ j\neq l,i}} (u_j - 1) - 1 \geq m - 2 - 1 \geq 0.$$

Thus, (4.19) holds if $u_l \geq u_i$ for all $i \neq l$ which is satisfied for $l = 1$. □

Note that for $m \geq 3$, $l = 1$ is not necessarily the only valid choice but that in some cases (4.19) might also be satisfied for some $l \neq 1$.

Lemma 4.11. *Let $m \geq 3$ or $u_1 > u_2$ be satisfied. Let $\alpha \in \mathbb{R}_+$ and $\tilde{\sigma} := \sum_{i=1}^m \tilde{u}_i$. If $\|\tilde{u}\|_{w,\rho} = \alpha$ and $\|u^i\|_{w,\rho} = \alpha$ for all $i = 1, \dots, m$, then the level α is given by*

$$\alpha = \begin{cases} \dfrac{\tilde{u}_1 u_2}{\tilde{u}_1 u_2 U - 2}, & m = 2 \\[2ex] \dfrac{u_1(\tilde{\sigma} - \tilde{u}_1)}{u_1 U(\tilde{\sigma} - \tilde{u}_1) - m}, & m \geq 3. \end{cases} \tag{4.20}$$

The level is feasible, i.e., $\alpha \in I$ holds, where I is defined as specified in Theorem 4.5.

Proof. The representation (4.20) is directly derived from (4.17) and Lemma 4.10. It remains to show that α is feasible. As stated in (4.18), $\alpha > U^{-1}$ is satisfied which is the imposed lower bound. It remains to show that $\alpha \leq \frac{u_1}{u_1 U - m}$ for $u_1 > u_m$. Consider first $m \geq 3$. It holds that

$$\alpha = \frac{u_1(\tilde{\sigma} - \tilde{u}_1)}{u_1 U(\tilde{\sigma} - \tilde{u}_1) - m} = \frac{1}{U - \frac{m}{u_1(\tilde{\sigma} - \tilde{u}_1)}} \leq \frac{1}{U - \frac{m}{u_1}} = \frac{u_1}{u_1 U - m}$$

since $\tilde{\sigma} - \tilde{u}_1 \geq u_2 + u_3 - 2 \geq 2$ for $m = 3$. For $m = 2$,

$$\alpha = \frac{\tilde{u}_1 u_2}{\tilde{u}_1 u_2 U - 2} = \frac{1}{U - \frac{2}{\tilde{u}_1 u_2}} \leq \frac{1}{U - \frac{2}{u_1}} = \frac{u_1}{u_1 U - 2},$$

as $\tilde{u}_1 u_2 = (u_1 - 1)u_2 = u_1 u_2 - u_2 \geq 2u_1 - u_2 \geq u_1$. □

	$m = 2$	$m \geq 3$
α	$\dfrac{u_1 u_2 \tilde{u}_1}{(u_1 + u_2)\tilde{u}_1 - 2u_1}$	$\dfrac{u_1(\tilde{\sigma} - \tilde{u}_1)}{u_1 U(\tilde{\sigma} - \tilde{u}_1) - m}$
w_i	$\dfrac{u_1(u_2\tilde{u}_1 - u_i)}{u_i[(u_1 + u_2)\tilde{u}_1 - 2u_1]}$	$\dfrac{u_1(\tilde{\sigma} - \tilde{u}_1) - u_i}{u_i[u_1 U(\tilde{\sigma} - \tilde{u}_1) - m]}$
ρ	$\dfrac{u_1}{(u_1 + u_2)\tilde{u}_1 - 2u_1}$	$\dfrac{1}{u_1 U(\tilde{\sigma} - \tilde{u}_1) - m}$

Table 4.1: Parameters w and ρ and corresponding level α of an augmented weighted Tchebycheff norm such that $\|\tilde{u}\|_{w,\rho} = \|u^1\|_{w,\rho} = \cdots = \|u^m\|_{w,\rho}$ for two and more than two criteria

The value of α given in (4.20) is therefore feasible and can be used to compute parameters w and ρ such that the extreme points of a given box $B(u)$ as well as the point $\tilde{u} = u - e$ lie on a common contour of an augmented weighted Tchebycheff norm. A summary of the computed parameters is given in Table 4.1. Interestingly, for $m = 2$ the point $\tilde{u}$ is situated on another hyperplane than in the case $m \geq 3$. An illustration of this result is presented in the next example.

Example 4.12. *Consider first $m = 2$ and $u = (5, 3)^\top$, thus, $\tilde{u} = (4, 2)^\top$. An illustration of the resulting contour is shown in Figure 4.6. Using the formulas stated in Table 4.1, we compute $\alpha = \frac{30}{11}$, $w_1 = \frac{7}{22}$, $w_2 = \frac{15}{22}$ and $\rho = \frac{5}{22} \approx 0.227$. Moreover, $z^q \approx (4.19, 1.95)^\top$. Note that $z^q \neq \tilde{u}$ but that $\tilde{u}$ lies on the line for which $w_2 z_2 \geq w_1 z_1$*

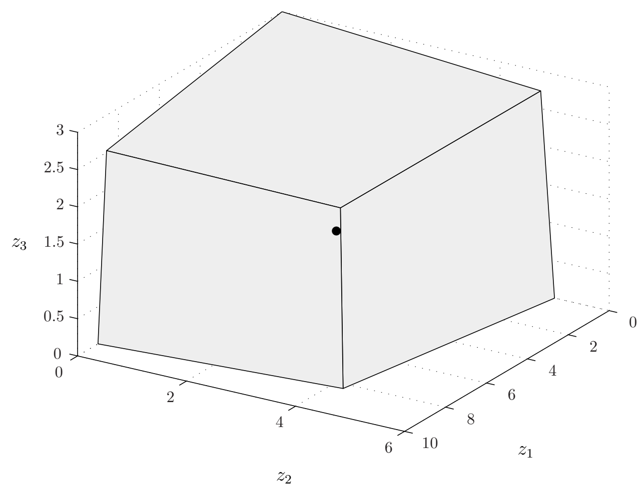


Figure 4.7: Contour of an augmented weighted Tchebycheff norm with $u = (9, 5, 4)^\top$. The point $\tilde{u} = (8, 4, 2)^\top$ is represented by a dot.

holds.

For an example with three criteria, consider $u = (9, 5, 3)^\top$*, hence* $\tilde{u} = (8, 4, 2)^\top$*. Then* $\alpha = \frac{90}{53}$*,* $w_1 = \frac{25}{159}$*,* $w_2 = \frac{49}{159}$*,* $w_3 = \frac{85}{159}$ *and* $\rho = \frac{5}{159} \approx 0.031$*. In Figure 4.7, the contour is depicted. The inflection point of the contour is* $z^q \approx (7.94, 4.05, 2.33)^\top$*. The point* $\tilde{u}$ *lies on the same hyperplane as* $u^1 = (9, 0, 0)^\top$*, i.e., on the hyperplane for which* $w_1 z_1 \geq w_2 z_2$ *and* $w_1 z_1 \geq w_3 z_3$ *hold.*

4.2.2 Optimal Parameter Choice

In the previous section we constructed an augmented weighted Tchebycheff norm such that the m points u^i, $i = 1, \ldots, m$, that represent extreme points of a box $B(u)$ and $\tilde{u} = (u_1 - 1, \ldots, u_m - 1)^\top$ lie on the same contour. However, as the point $\tilde{u}$ might be a feasible nondominated point itself, we must guarantee that it can be generated when searching for points in $B(u)$. Therefore, it must have a strictly smaller level than all u^i, $i = 1, \ldots, m$, i.e.,

$$\|\tilde{u}\|_{w,\rho} < \min_{i=1,\ldots,m} \{\|u^i\|_{w,\rho}\}$$

must hold. Therefore, we modify our setting slightly and replace u^i, $i = 1, \ldots, m$, by $u^i(\eta) := (0, \ldots, 0, u_i - \eta_i, 0, \ldots, 0)^\top$, respectively, with $\eta \in (0, 1)^m$. By constructing parameters w and ρ such that all $u^i(\eta)$, $i = 1, \ldots, m$, and $\tilde{u}$ lie on a common level curve, we guarantee that all u^i, $i = 1, \ldots, m$, have a strictly larger level than $\tilde{u}$ and, hence, $\tilde{u}$ can be found by an augmented weighted Tchebycheff method if it represents a nondominated point. Note that $\eta = 0$ corresponds to the formulas derived in the previous section.

Every choice of $\eta \in (0, 1)^m$ yields an augmented weighted Tchebycheff norm that satisfies our primary goal, i.e., the possible generation of every nondominated point. However, as stated in the literature (see, e.g. Ralphs et al. (2006)), a too small value of ρ might imply numerical problems. Therefore, our secondary goal is to choose ρ as large as possible. In this subsection we show that ρ as a function of η is strictly decreasing. Therefore, it is favorable to choose rather small values for the components of η.

In what follows, we first generalize all formulas of the previous section for a parameter $\eta \in [0, 1]^m$. This includes the case $\eta = 0$ treated so far as well as the case $\eta = e$ resulting in a weighted Tchebycheff norm. Then we investigate how ρ depends on η. Finally, we propose a concrete parameter choice for the practical implementation which theoretically guarantees the generation of every nondominated point.

Formulas Dependent on η

When replacing $u \in \mathbb{N}^m$, $u_i \geq 2$ for all $i = 1, \ldots, m$, by $u(\eta) \in \mathbb{R}^m, u_i(\eta) \geq 1$ for all $i = 1, \ldots, m, \eta \in [0, 1]^m$, we can basically repeat the steps of the previous section in order to determine the parameters of the augmented weighted Tchebycheff norm. Remember that we assumed without loss of generality that $u_1 \geq u_2 \geq \cdots \geq u_m$. Without loss of generality, we can also assume that the components of $u(\eta)$ are ordered decreasingly: If $u_i > u_{i+1}$ for some $i \in \{1, \ldots, m-1\}$, then $u_i(\eta) \geq u_{i+1}(\eta)$, as $\eta_k \leq 1$ for all $k = 1, \ldots, m$. If $u_i = u_{i+1}$ for some $i \in \{1, \ldots, m-1\}$, then $u_i(\eta)$ may be smaller than $u_{i+1}(\eta)$. However, in this case, we can interchange the components i and $i+1$. Thereby, u remains unchanged, hence the components of both vectors u and $u(\eta)$ are ordered decreasingly.

The recalculation of the respective formulas is straightforward and we omit the details. In analogy to Theorem 4.5 we compute the parameters w and ρ dependent on the level α and now, additionally, on $\eta \in [0, 1]^m$:

Theorem 4.13. *Let $\alpha \in \mathbb{R}_+$. The points $u^i(\eta), i = 1, \ldots, m$, lie on a common level curve of an augmented weighted Tchebycheff norm with level α, i.e., $\|u^i(\eta)\|_{w,\rho} = \alpha$ for all $i = 1, \ldots, m$, if and only if*

$$\rho(\alpha, \eta) = \frac{\alpha U(\eta) - 1}{m} \tag{4.21}$$

and

$$w_i(\alpha, \eta) = \frac{\alpha}{u_i(\eta)} - \rho(\alpha, \eta) \quad \forall\, i = 1, \ldots, m \tag{4.22}$$

with $U(\eta) := \sum_{j=1}^{m} \frac{1}{u_j(\eta)}$ and $\alpha \in I(\eta)$, where

(i) $I(\eta) := \left[(U(\eta))^{-1}, \infty\right)$, if $u_1(\eta) = u_m(\eta)$, and

(ii) $I(\eta) := \left[(U(\eta))^{-1}, \frac{u_1(\eta)}{u_1(\eta)U(\eta) - m}\right]$ otherwise, i.e., for $u_1(\eta) > u_m(\eta)$.

Proof. The proof is completely analogous to the proof of Theorem 4.5. □

In analogy to Lemma 4.8, the level α is uniquely determined if, additionally, the point $\tilde{u}$ lies on the contour. As in Lemma 4.8, the case $m = 2, u_1 = u_2 = 2$ and $\eta_1 = \eta_2 = 0$ must be excluded.

Lemma 4.14. *Let $m \geq 2$ and, for $m = 2$, let the case $u_1 = u_2 = 2$ and $\eta_1 = \eta_2 = 0$ be excluded. Let $\alpha \in \mathbb{R}_+$, and let $\tilde{\sigma} := \sum_{i=1}^{m} \tilde{u}_i$. If $\|\tilde{u}\|_{w,\rho} = \alpha$ and $\|u^i(\eta)\|_{w,\rho} = \alpha$ for all $i = 1, \ldots, m$ hold, then*

$$\alpha(\eta) = \frac{u_l(\eta)(\tilde{\sigma} - \tilde{u}_l)}{u_l(\eta)U(\eta)(\tilde{\sigma} - \tilde{u}_l) - m(1 - \eta_l)}, \tag{4.23}$$

where $l \in \{1, \ldots, m\}$ such that $w_l(\alpha, \eta)\tilde{u}_l \geq w_i(\alpha, \eta)\tilde{u}_i$ for all $i = 1, \ldots, m$.

Proof. The proof is analogous to the proof of Lemma 4.8. We use that $\|u^i(\eta)\|_{w,\rho} = \alpha$ for all $i = 1, \dots, m$ if and only if (4.21) and (4.22) hold. Again, we assume that $l \in \{1, \dots, m\}$ is the index for which

$$w_l(\alpha, \eta)\tilde{u}_l = \max_{i=1,\dots,m} \{w_i(\alpha, \eta)\tilde{u}_i\}$$

is satisfied which characterizes the hyperplane on which $\tilde{u}$ lies. Then

$$\|\tilde{u}\|_{w,\rho} = w_l(\alpha, \eta)\tilde{u}_l + \rho(\alpha, \eta)\sum_{i=1}^{m} \tilde{u}_i = \left(\frac{\alpha}{u_l(\eta)} - \frac{\alpha U(\eta) - 1}{m}\right) \cdot \tilde{u}_l + \frac{\alpha U(\eta) - 1}{m} \cdot \tilde{\sigma}$$

and, thus,

$$\begin{aligned}
\|\tilde{u}\|_{w,\rho} = \alpha \iff & \alpha\left(\frac{m - u_l(\eta)U(\eta)}{m u_l(\eta)} \cdot \tilde{u}_l\right) + \frac{1}{m}\tilde{u}_l + \alpha \cdot \frac{U(\eta)}{m} \cdot \tilde{\sigma} - \frac{1}{m} \cdot \tilde{\sigma} = \alpha \\
\iff & \alpha\left[\frac{[m - u_l(\eta)U(\eta)]\tilde{u}_l}{m u_l(\eta)} + \frac{U(\eta)}{m}\tilde{\sigma} - 1\right] = \frac{1}{m}\tilde{\sigma} - \frac{1}{m}\tilde{u}_l \\
\iff & \alpha\left[m\tilde{u}_l - u_l(\eta)U(\eta)\tilde{u}_l + U(\eta)u_l(\eta)\tilde{\sigma} - m u_l(\eta)\right] = u_l(\eta)(\tilde{\sigma} - \tilde{u}_l) \\
\iff & \alpha\,\underbrace{[u_l(\eta)U(\eta)(\tilde{\sigma} - \tilde{u}_l) - m(1 - \eta_l)]}_{=:A(\eta)} = u_l(\eta)(\tilde{\sigma} - \tilde{u}_l)
\end{aligned}$$

Consider $A(\eta)$: It holds that

$$u_l(\eta)U(\eta) = 1 + \sum_{\substack{i=1,\dots,m,\\ i\neq l}} \frac{u_l - \eta_l}{u_i - \eta_i} > 1.$$

We consider the following cases:

(1) There exists an index $j \in \{1, \dots, m\}\setminus\{l\}$ such that $u_j \geq 3$. Then

$$\tilde{\sigma} - \tilde{u}_l = \sum_{\substack{i=1,\dots,m,\\ i\neq l}} \tilde{u}_i \geq m,$$

thus, $A(\eta) > m - m(1 - \eta_l) = m \cdot \eta_l \geq 0$, i.e., $A(\eta) > 0$.

(2) For all $i = 1, \dots, m, i \neq l$ it holds that $u_i = 2$. Then

$$\tilde{\sigma} - \tilde{u}_l = \sum_{\substack{i=1,\dots,m,\\ i\neq l}} \tilde{u}_i = m - 1.$$

(2a) If $u_l > 2$, then, as $u_l - \eta_l \geq 2$ and $u_i(\eta) = 2 - \eta_i \leq 2$ hold for all $i \neq l$,

$$u_l(\eta)U(\eta) = 1 + \sum_{\substack{i=1,\dots,m,\\ i\neq l}} \frac{u_l - \eta_l}{2 - \eta_i} \geq 2,$$

with equality if and only if $u_l = 3, \eta_l = 1$ and $\eta_i = 0$ for all $i \neq l$. However, in this case, $A(\eta) \geq 2(m-1) - m(1-\eta_l) = m - 2 + m \cdot \eta_l > 0$. Otherwise, $A(\eta) > 2(m-1) - m(1-\eta_l) = m - 2 + m \cdot \eta_l \geq 0$, so in both cases $A(\eta) > 0$.

(2b) If $u_l = 2$, i.e., $u_i = 2$ for all $i = 1, \dots, m$, then

$$u_l(\eta)U(\eta) = 1 + \sum_{\substack{i=1,\dots,m,\\ i\neq l}} \frac{2-\eta_l}{2-\eta_i} \geq 1 + \frac{1}{2}(m-1)$$

as $1 \leq 2 - \eta_i \leq 2$ for all $i = 1, \dots, m$.
If $m \geq 3$, then $A(\eta) \geq 2(m-1) - m(1-\eta_l) = m - 2 + m \cdot \eta_l \geq 1$.
If $m = 2$, then, for $i \in \{1,2\} \backslash \{l\}$,

$$\begin{aligned} A(\eta) &= \left(1 + \frac{2-\eta_l}{2-\eta_i}\right) \cdot 1 - 2(1-\eta_l) = -1 + \frac{2 - \eta_l + 2\eta_l(2-\eta_i)}{2-\eta_i} \\ &= -1 + \frac{2 + \eta_l(4 - 2\eta_i - 1)}{2-\eta_i} \geq -1 + \frac{2+\eta_l}{2-\eta_i} \geq 0. \end{aligned}$$

Note that equality holds if and only if $\eta = 0$.

Hence, α is well defined except for the case $m = 2$, $u = (2,2)^\top$ and $\eta_1 = \eta_2 = 0$. □

In the following, we exclude this case from consideration. Again, we have to show that $\alpha(\eta)$ is not only formally well-defined but is also feasible, i.e., $\alpha \in I(\eta)$. We show equivalently that, for α specified as in (4.23), $\rho(\eta) \geq 0$, $w_i(\eta) \geq 0$ and $w_i(\eta) + \rho(\eta) > 0$ hold for all $i = 1, \dots, m$, i.e., that the parameters of the associated augmented weighted Tchebycheff norm are well-defined.

Lemma 4.15. *Let*

$$\alpha(\eta) = \frac{u_l(\eta)(\tilde{\sigma} - \tilde{u}_l)}{A(\eta)}$$

be defined as in (4.23) *with denominator* $A(\eta) := u_l(\eta)U(\eta)(\tilde{\sigma} - \tilde{u}_l) - m(1-\eta_l)$. *Then*

$$\rho(\eta) = \frac{1-\eta_l}{A(\eta)}, \quad w_i(\eta) = \frac{u_l(\eta)(\tilde{\sigma} - \tilde{u}_l) - u_i(\eta)(1-\eta_l)}{u_i(\eta)A(\eta)}, \ i = 1, \dots, m,$$

and $\rho(\eta) \geq 0$, $w_i(\eta) \geq 0$ *and* $w_i(\eta) + \rho(\eta) > 0$ *hold for all* $i = 1, \dots, m$.

Proof. Inserting $\alpha(\eta)$ into (4.21) yields

$$\rho(\eta) = \frac{u_l(\eta)(\tilde{\sigma} - \tilde{u}_l)U(\eta) - A(\eta)}{mA(\eta)} = \frac{1-\eta_l}{A(\eta)} \geq 0,$$

where equality holds if and only if $\eta_l = 1$. Inserting $\alpha(\eta)$ into formula (4.22) for all $i = 1, \ldots, m$ yields

$$w_i(\eta) = \frac{1}{u_i(\eta)} \cdot \frac{u_l(\eta)(\tilde{\sigma} - \tilde{u}_l)}{A(\eta)} - \rho(\eta) = \frac{u_l(\eta)(\tilde{\sigma} - \tilde{u}_l) - u_i(\eta)(1 - \eta_l)}{u_i(\eta)A(\eta)}.$$

As $u_i(\eta) > 0$ for all $i = 1, \ldots, m$ and $A(\eta) > 0$, it suffices to consider the numerator of $w_i(\eta)$ in order to show that it is well-defined. We distinguish the following cases: If $i = l$, then

$$u_l(\eta)(\tilde{\sigma} - \tilde{u}_l) - u_i(\eta)(1 - \eta_l) = u_l(\eta)(\tilde{\sigma} - \tilde{u}_l - 1 + \eta_l) \geq 0.$$

If $i \neq l$ and $m \geq 3$, then

$$\begin{aligned} u_l(\eta)(\tilde{\sigma} - \tilde{u}_l) - u_i(\eta)(1 - \eta_l) &\geq u_l(\eta)(\tilde{\sigma} - \tilde{u}_l) - u_i(\eta)u_l(\eta) \\ &= u_l(\eta)(\tilde{\sigma} - \tilde{u}_l - u_i(\eta)) \geq 0. \end{aligned}$$

If $i \neq l$ and $m = 2$, then

$$\begin{aligned} &u_l(\eta)(\tilde{\sigma} - \tilde{u}_l) - u_i(\eta)(1 - \eta_l) = u_l(\eta)\tilde{u}_i - u_i(\eta)(1 - \eta_l) \\ &= (u_l - \eta_l)(u_i - 1) - (u_i - \eta_i)(1 - \eta_l) = u_l(u_i - 1) - u_i + \eta_l(1 - \eta_i) + \eta_i \\ &\geq 2(u_i - 1) - u_i + \eta_l(1 - \eta_i) + \eta_i \geq u_i - 2 \geq 0. \end{aligned}$$

Finally, $w_i(\eta) + \rho(\eta) > 0$ holds for all $i = 1, \ldots, m$ as $\rho(\eta) = 0$ if and only if $\eta_l = 1$. But then the numerator of $w_i(\eta)$ is $u_l(\eta)(\tilde{\sigma} - \tilde{u}_l) > 1$, i.e., $w_i(\eta) > 0$ for all $i = 1, \ldots, m$. □

In the case $\eta = 0$ presented in the previous section, we proceeded by specifying the index l explicitly. Now the index l additionally depends on η. However, for our purpose, the determination of l for arbitrary η is not necessary, as the behavior of ρ as a function of η can be analyzed without specifying l. This will be presented in the following.

Maximality of ρ

In what follows, we analyze ρ as a function of η and show that $\rho(\eta)$ attains its maximum on $[0, 1]^m$ for $\eta = 0$. So far, we have computed $\rho(\eta)$ assuming that index $l \in \{1, \ldots, m\}$ is the index for which

$$w_l(\alpha, \eta)\tilde{u}_l = \max_{i=1,\ldots,m} \{w_i(\alpha, \eta)\tilde{u}_i\}$$

holds. This implies that ρ is defined piecewise on specific subsets of $[0, 1]^m$. For a given $l \in \{1, \ldots, m\}$ we define the set $\mathcal{U}_l$ by

$$\mathcal{U}_l := \{\eta \in [0, 1]^m : w_l(\alpha, \eta)\tilde{u}_l \geq w_i(\alpha, \eta)\tilde{u}_i \; \forall \, i = 1, \ldots, m\}.$$

Note that $\bigcup_{l=1}^{m} \mathcal{U}_l = [0,1]^m$ and

$$\begin{aligned}\mathcal{U}_i \cap \mathcal{U}_j = \{\eta \in [0,1]^m : & w_i(\alpha,\eta)\tilde{u}_i = w_j(\alpha,\eta)\tilde{u}_j, \\ & w_i(\alpha,\eta)\tilde{u}_i \geq w_k(\alpha,\eta)\tilde{u}_k \;\forall\, k \in \{1,\ldots,m\}\}\end{aligned}$$

for all $i \neq j$. Hence,

$$\rho(\eta) = \frac{1-\eta_l}{A(\eta)} \quad \text{for all } \eta \in \mathcal{U}_l$$

with $A(\eta) := u_l(\eta)U(\eta)(\tilde{\sigma}-\tilde{u}_l) - m(1-\eta_l)$. For every $l = 1,\ldots,m$, the function $\rho(\eta)$ is continuous on $\mathcal{U}_l$. As the intersection of two sets $\mathcal{U}_i$ and $\mathcal{U}_j, i \neq j$, is non-empty, we see that $\rho(\eta)$, defined piecewise on $\mathcal{U}_l$, is continuous on $[0,1]^m$. On every $\mathcal{U}_l$, the partial derivatives of $\rho(\eta)$ with respect to η_l and $\eta_i, i = 1,\ldots,m, i \neq l$, are

$$\begin{aligned}\frac{\partial\rho(\eta)}{\partial\eta_l} &= \frac{-A(\eta) - (1-\eta_l)\left[-(\tilde{\sigma}-\tilde{u}_l)\left(\sum_{i\neq l}\frac{1}{u_i(\eta)}\right) + m\right]}{A(\eta)^2} \\ &= \frac{-u_l(\eta)U(\eta)(\tilde{\sigma}-\tilde{u}_l) + (1-\eta_l)(\tilde{\sigma}-\tilde{u}_l)\left(\sum_{i\neq l}\frac{1}{u_i(\eta)}\right)}{A(\eta)^2} \\ &= \frac{(\tilde{\sigma}-\tilde{u}_l)}{A(\eta)^2}\left(-1 - \sum_{i\neq l}\frac{u_l-\eta_l}{u_i(\eta)} + \sum_{i\neq l}\frac{1-\eta_l}{u_i(\eta)}\right) \\ &= \frac{(\tilde{\sigma}-\tilde{u}_l)}{A(\eta)^2}\left(-1 - \sum_{i\neq l}\frac{u_l-1}{u_i(\eta)}\right) < 0\end{aligned}$$

since $\frac{u_l-1}{u_i(\eta)} > 0$ for all $i = 1,\ldots,m$ and $\tilde{\sigma}-\tilde{u}_l > 0$. For all $i = 1,\ldots,m, i \neq l$, we have

$$\frac{\partial\rho(\eta)}{\partial\eta_i} = \frac{-(1-\eta_l)(\tilde{\sigma}-\tilde{u}_l)u_l(\eta)}{(u_i(\eta))^2 A(\eta)^2} \leq 0,$$

with equality if and only if $\eta_l = 1$. Obviously, the partial derivatives of $\rho(\eta)$ are continuous and decreasing functions on $\mathcal{U}_l$ for all $l = 1,\ldots,m$. Furthermore, since $\bigcup_{l=1}^{m} \mathcal{U}_l = [0,1]^m$ we obtain

Lemma 4.16. *The maximal value for $\rho(\eta)$ on $[0,1]^m$ is attained for $\eta = 0$.*

Proof. The set $\mathcal{U} := [0,1]^m$ is compact and the function $\rho(\eta)$ is continuous on $\mathcal{U}$, thus, the maximum of $\rho(\eta)$ exists and is attained in $\mathcal{U}$. As all partial derivatives of $\rho(\eta)$ are negative on $\mathcal{U}$ with $\eta_l \neq 1$ and zero if and only if $\eta_l = 1$, the maximum of $\rho(\eta)$ is attained for $\eta = 0$. $\square$

Practical Parameter Choice

The foregoing lemma shows that the smaller the value η, the higher the value of the parameter ρ, which is aimed at for numerical purposes. However, as pointed out

	$m = 2$
$\alpha(\bar{\eta})$	$\dfrac{u_1(\bar{\eta})u_2(\bar{\eta})\tilde{u}_1}{(u_1(\bar{\eta}) + u_2(\bar{\eta}))\tilde{u}_1 - 2u_1(\bar{\eta})(1-\bar{\eta})}$
$w_i(\bar{\eta})$	$\dfrac{u_1(\bar{\eta})[u_2(\bar{\eta})\tilde{u}_1 - u_i(\bar{\eta})(1-\bar{\eta})]}{u_i(\bar{\eta})[(u_1(\bar{\eta}) + u_2(\bar{\eta}))\tilde{u}_1 - 2u_1(\bar{\eta})(1-\bar{\eta})]}$
$\rho(\bar{\eta})$	$\dfrac{u_1(\bar{\eta})(1-\bar{\eta})}{(u_1(\bar{\eta}) + u_2(\bar{\eta}))\tilde{u}_1 - 2u_1(\bar{\eta})(1-\bar{\eta})}$

Table 4.2: Practical choice of parameter values for an augmented weighted Tchebycheff norm dependent on $\bar{\eta} \in (0,1)$ for $m = 2$

before, the choice $\eta = 0$ is not feasible, as then $\|\tilde{u}\|_{w,\rho} < \min_{i=1,\dots,m}\{\|u^i\|_{w,\rho}\}$ is not satisfied. Therefore, we need to bound η away from zero. If we want to determine the parameters w and ρ for arbitrary $\eta \in (0,1)^m$, we need to specify l, which means that we need to analyze on which hyperplane $\tilde{u}$ lies. This investigation can be simplified if we restrict the choice of η to $\eta_1 = \eta_2 = \cdots = \eta_m = \bar{\eta}$ with $\bar{\eta} \in (0,1)$. In analogy to Lemma 4.10, we obtain

Lemma 4.17. *Let α be defined according to (4.23), where $l \in \{1,\dots,m\}$ such that $w_l(\eta)\tilde{u}_l \geq w_i(\eta)\tilde{u}_i$ for all $i = 1,\dots,m$, and let $\eta_1 = \eta_2 = \cdots = \eta_m = \bar{\eta}$ with $\bar{\eta} \in (0,1)$. Then it holds*

1. *for $m = 2$ that $w_2(\eta)\tilde{u}_2 \geq w_1(\eta)\tilde{u}_1$, i.e., $l = 2$, and*
2. *for $m \geq 3$ that $w_1(\eta)\tilde{u}_1 \geq w_i(\eta)\tilde{u}_i$ for all $i = 1,\dots,m$, i.e., $l = 1$.*

Proof. From Lemma 4.15 we see that for all $i = 1,\dots,m$

$$w_i(\eta)\tilde{u}_i = \frac{u_l(\eta)(\tilde{\sigma} - \tilde{u}_l) - u_i(\eta)(1-\eta_l)}{u_i(\eta)A(\eta)} \cdot \tilde{u}_i.$$

Therefore, using $u_i(\eta) > 0$ and $A(\eta) > 0$, it holds for all $i = 1,\dots,m, i \neq l$ that

$$\begin{aligned} w_l(\eta)\tilde{u}_l \geq w_i(\eta)\tilde{u}_i &\iff (\tilde{\sigma} - \tilde{u}_l)\left[\tilde{u}_l u_i(\eta) - \tilde{u}_i u_l(\eta)\right] \geq u_i(\eta)(1-\eta_l)(\tilde{u}_l - \tilde{u}_i) \\ &\iff (\tilde{\sigma} - \tilde{u}_l)\left[u_l(1-\eta_i) - u_i(1-\eta_l) + \eta_i - \eta_l\right] \geq u_i(\eta)(1-\eta_l)(u_l - u_i). \end{aligned}$$

If $\eta_1 = \eta_2 = \cdots = \eta_m = \bar{\eta}$ and $\bar{\eta} \in (0,1)$, this is equivalent to

$$(\tilde{\sigma} - \tilde{u}_l - u_i(\bar{\eta}))(u_l - u_i) \geq 0 \tag{4.24}$$

	$m \geq 3$
$\alpha(\bar{\eta})$	$\dfrac{u_1(\bar{\eta})(\tilde{\sigma}-\tilde{u}_1)}{u_1(\bar{\eta})U(\bar{\eta})(\tilde{\sigma}-\tilde{u}_1)-m(1-\bar{\eta})}$
$w_i(\bar{\eta})$	$\dfrac{u_1(\bar{\eta})(\tilde{\sigma}-\tilde{u}_1)-u_i(\bar{\eta})(1-\bar{\eta})}{u_i(\bar{\eta})[u_1(\bar{\eta})U(\bar{\eta})(\tilde{\sigma}-\tilde{u}_1)-m(1-\bar{\eta})]}$
$\rho(\bar{\eta})$	$\dfrac{1-\bar{\eta}}{u_1(\bar{\eta})U(\bar{\eta})(\tilde{\sigma}-\tilde{u}_1)-m(1-\bar{\eta})}$

Table 4.3: Practical choice of parameter values for an augmented weighted Tchebycheff norm dependent on $\bar{\eta} \in (0,1)$ for $m \geq 3$

For $m = 2$, it holds that $\tilde{\sigma} - \tilde{u}_l - u_i(\bar{\eta}) = \bar{\eta} - 1 < 0$ for any choice of l since $i \neq l$ is assumed. Thus, (4.24) holds for $m = 2$ if and only if $u_l \leq u_i$ for all $i \neq l$ which is satisfied for $l = 2$. If $m \geq 3$, then

$$\tilde{\sigma} - \tilde{u}_l - u_i(\bar{\eta}) = \sum_{j \neq l} \tilde{u}_j - u_i(\bar{\eta}) = \sum_{j \neq l,i} (u_j - 1) - 1 + \bar{\eta} \geq m - 3 + \bar{\eta} > 0.$$

Thus, (4.24) holds for $m = 3$ if and only if $u_l \geq u_i$ for all $i \neq l$ which is satisfied for $l = 1$. $\square$

Tables 4.2 and 4.3 show a summary of the formulas obtained for $\bar{\eta} \in (0,1)$.

4.3 Parameters of a Generalized Augmented Weighted Tchebycheff Norm

As we have seen in the previous section, the use of the classic augmented weighted Tchebycheff method for constructing a contour on which the points $u^i, i = 1, \dots, m$, and $\tilde{u}$ lie is possible but not necessarily intuitive as $\tilde{u}$ typically lies in the relative interior of one face of the contour. The construction would be much easier if we could choose $\tilde{u}$ as the inflection point z^q. Therefore, in this section we study a generalization of the augmented weighted Tchebycheff method, see, e.g., Kaliszewski (2000), which offers more flexibility. Let

$$\|z\|^G_{w,\rho} := \max_{i=1,\dots,m} \{w_i|z_i|\} + \sum_{j=1}^{m} \rho_j |z_j|, \tag{4.25}$$

where $w, \rho \in \mathbb{R}^m_+$, $\sum_{i=1}^m w_i = 1$ and $w_i + \rho_i > 0$ for all $i = 1, \dots, m$. We call $\|\cdot\|^G_{w,\rho}$ a *generalized augmented weighted Tchebycheff norm* in the following.

4.3.1 Feasible Parameter Choice

As in the previous section we determine the parameters w and ρ of (4.25) such that the m points $u^i, i = 1, \ldots, m$, and $\tilde{u}$ lie on a common contour. Now, additionally, $\tilde{u}$ equals the inflection point. In order to apply the next theorem not only to integer-valued (discrete) but also to continuous multicriteria optimization problems, we do not set $\tilde{u} := u - e$ but consider an arbitrary $\tilde{u} \in B(u)$. However, as we will see in the next theorem, in order to obtain a valid contour of a norm with a positive level α and well-defined parameters w and ρ, $\tilde{u}$ cannot be chosen freely in $B(u)$ but must satisfy the two technical conditions

$$\sum_{j=1}^{m} \frac{\tilde{u}_j}{u_j} > 1 \tag{4.26}$$

and

$$(m-1)\tilde{u}_i \geq \left(\sum_{j=1}^{m} \frac{\tilde{u}_j}{u_j} - 1 \right) u_i \tag{4.27}$$

for every $i = 1, \ldots, m$. Moreover, $\tilde{u}_i > 0$ for all $i = 1, \ldots, m$ must hold.

Theorem 4.18 (Parameters of the Generalized Augmented Weighted Tchebycheff Norm)**.** *Let $\alpha \in \mathbb{R}_+$, $u \in \mathbb{R}^m_+$, $u_i \geq 2$ for all $i = 1, \ldots, m$ as well as $\tilde{u} \in \mathbb{R}^m_+$ satisfying (4.26) and (4.27). It holds that $\|u^i\|^G_{w,\rho} = \alpha$ for all $i = 1, \ldots, m$, $\|\tilde{u}\|^G_{w,\rho} = \alpha$ and $w_i\tilde{u}_i = w_j\tilde{u}_j$ for all $i, j \in \{1, \ldots, m\}$ if and only if*

$$w_i = \left(\tilde{u}_i \sum_{j=1}^{m} \frac{1}{\tilde{u}_j} \right)^{-1} \quad \wedge \quad \rho_i = \frac{\alpha}{u_i} - w_i \quad \forall\, i = 1, \ldots, m,$$

where

$$\alpha = (m-1) \left[\left(\sum_{j=1}^{m} \frac{1}{\tilde{u}_j} \right) \left(\sum_{j=1}^{m} \frac{\tilde{u}_j}{u_j} - 1 \right) \right]^{-1}.$$

Proof. Assume that $w_i\tilde{u}_i = w_j\tilde{u}_j$ for all $i, j \in \{1, \ldots, m\}$ or, equivalently, that $w_1\tilde{u}_1 = w_j\tilde{u}_j$ for all $j \in \{1, \ldots, m\}$. As $\tilde{u}_j > 0$ for all $j = 1, \ldots, m$, it holds that

$$w_1\tilde{u}_1 = w_j\tilde{u}_j\ \forall\, j \in \{1, \ldots, m\} \iff w_j = \frac{w_1\tilde{u}_1}{\tilde{u}_j}\ \forall\, j \in \{1, \ldots, m\}.$$

Moreover, as $\sum_{j=1}^{m} w_j = 1$, it follows that $1 = w_1\tilde{u}_1 \sum_{j=1}^{m} \frac{1}{\tilde{u}_j}$ which is equivalent to

$$w_1 = \left(\tilde{u}_1 \sum_{j=1}^{m} \frac{1}{\tilde{u}_j} \right)^{-1}.$$

Thus, for all $i = 1, \dots, m$,

$$w_i = w_1 \frac{\tilde{u}_1}{\tilde{u}_i} = \left(\tilde{u}_i \sum_{j=1}^{m} \frac{1}{\tilde{u}_j} \right)^{-1}. \tag{4.28}$$

Note that $w_i > 0$ for all $i = 1, \dots, m$, as $\tilde{u}_i > 0$ for all $i = 1, \dots, m$. Furthermore, for all $i = 1, \dots, m$, using $u_i > 0$,

$$\|u^i\|^G_{w,\rho} = \alpha \iff (w_i + \rho_i) u_i = \alpha \iff \rho_i = \frac{\alpha}{u_i} - w_i$$

and, hence, using (4.28),

$$\begin{aligned}
\|\tilde{u}\|^G_{w,\rho} = \alpha \iff w_1 \tilde{u}_1 + \sum_{j=1}^{m} \rho_j \tilde{u}_j = \alpha &\iff w_1 \tilde{u}_1 + \sum_{j=1}^{m} \left(\frac{\alpha}{u_j} - w_j \right) \tilde{u}_j = \alpha \\
&\iff -(m-1) w_1 \tilde{u}_1 + \alpha \sum_{j=1}^{m} \frac{\tilde{u}_j}{u_j} = \alpha \\
&\iff \alpha \left(\sum_{j=1}^{m} \frac{\tilde{u}_j}{u_j} - 1 \right) = (m-1) \left(\sum_{j=1}^{m} \frac{1}{\tilde{u}_j} \right)^{-1} \\
&\overset{(4.26)}{\iff} \alpha = (m-1) \left[\left(\sum_{j=1}^{m} \frac{1}{\tilde{u}_j} \right) \left(\sum_{j=1}^{m} \frac{\tilde{u}_j}{u_j} - 1 \right) \right]^{-1}.
\end{aligned}$$

Then

$$\rho_i = \frac{\alpha}{u_i} - w_i = \frac{(m-1)\tilde{u}_i - \left(\sum_{j=1}^{m} \frac{\tilde{u}_j}{u_j} - 1 \right) u_i}{\tilde{u}_i u_i \left(\sum_{j=1}^{m} \frac{1}{\tilde{u}_j} \right) \left(\sum_{j=1}^{m} \frac{\tilde{u}_j}{u_j} - 1 \right)}$$

follows for all $i = 1, \dots, m$. Thereby, $\rho \in \mathbb{R}^m_+$ holds if, additionally to (4.26), (4.27) is valid for $i = 1, \dots, m$. The converse statement is obvious. □

In Theorem 4.18, the conditions (4.26) and (4.27) limit the possible choices of $\tilde{u}$. In the following we investigate two important choices for $\tilde{u}$ and study whether (4.26) and (4.27) are satisfied. First, we consider $\tilde{u} := u - \bar{\eta} e$ with $\bar{\eta} \in (0, 1]$. This case corresponds to the situation in which $\tilde{u}$ is placed with an absolute distance from u as treated in the previous section. However, different from the construction before, we do not perturb the values of u^i but choose $\tilde{u}$ slightly away from u. Note that this also guarantees that every integer-valued nondominated point in $B(u)$ has a strictly smaller level than the extreme points u^i, $i = 1, \dots, m$. Secondly, we investigate $\tilde{u} := \delta u$ for some $\delta \in (0, 1)$, i.e., the situation in which the inflection point is located with a relative distance from u. If δ is chosen sufficiently large, then also with this setting every integer-valued nondominated point in $B(u)$ can be generated.

Locating the Inflection Point with Absolute Distance to u

Let $\tilde{u} := u - \bar{\eta}e$, $\bar{\eta} \in (0,1]$. Then

$$\sum_{j=1}^{m} \frac{\tilde{u}_j}{u_j} = m - \sum_{j=1}^{m} \frac{\bar{\eta}}{u_j} \geq m\left(1 - \frac{\bar{\eta}}{2}\right) \geq m \cdot \frac{1}{2} \geq 1$$

holds with equality if and only if $m = 2$, $u_1 = u_2 = 2$ and $\bar{\eta} = 1$, thus, (4.26) is satisfied for all choices of u except the case $m = 2$, $u_1 = u_2 = 2$ and $\bar{\eta} = 1$. Recall that this case was also excluded in the previous section. For $i = 1, \ldots, m$, (4.27) can be reformulated as follows:

$$(m-1)\tilde{u}_i \geq \left(\sum_{j=1}^{m} \frac{\tilde{u}_j}{u_j} - 1\right) u_i \iff (m-1)(u_i - \bar{\eta}) \geq \left(m - \sum_{j=1}^{m} \frac{\bar{\eta}}{u_j} - 1\right) u_i$$

$$\iff -\bar{\eta}\,(m-1) \geq -\bar{\eta} \sum_{j=1}^{m} \frac{u_i}{u_j} \overset{\bar{\eta}>0}{\iff} \sum_{\substack{j=1,\ldots,m,\\ j\neq i}} \frac{u_i}{u_j} - (m-2) \geq 0 \tag{4.29}$$

For $m = 2$, (4.29) is satisfied for every $i = 1, 2$ and for any choice of u with $u_1, u_2 \geq 2$. However, for $m \geq 3$, not all choices of u are valid. Let, for example, $m = 3$ and $u = (20, 10, 5)^\top$. Then, for $i = 3$,

$$\sum_{j=1}^{2} \frac{u_3}{u_j} = \frac{5}{20} + \frac{5}{10} < 1,$$

which, in turn, implies $\rho_3 < 0$, thus, $\|\cdot\|_{w,\rho}^{G}$ does not represent a norm in this case. Note that for $m = 3$, (4.29) is satisfied for every $i = 1, 2$ and every $u \in \mathbb{R}^m_+$ for which, w.l.o.g., $u_1 \geq u_2 \geq u_3$ holds, as $u_i/u_3 \geq 1$ for every $i = 1, 2$ and $u_i/u_j > 0$ for every $i, j = 1, 2, i \neq j$. Therefore, when the components of u are ordered decreasingly, only $i = 3$ is critical and needs to be checked. An example in which a valid norm is obtained for $m = 3$ is given by $u = (10, 6, 4)^\top$. Condition (4.27) is satisfied for all $i = 1, 2, 3$ as

$$\sum_{j=1}^{2} \frac{u_3}{u_j} = \frac{4}{10} + \frac{4}{6} = \frac{16}{15} > 1.$$

We conclude that in the bicriteria case the generalized augmented weighted Tchebycheff norm (4.25) can be used to define a contour such that the points u^1, u^2 and $\tilde{u} := u - \bar{\eta}e$, $\bar{\eta} \in (0,1]$, lie on this contour and, additionally, $\tilde{u}$ equals the inflection point for every feasible choice of u and $\bar{\eta}$ besides $u_1 = u_2 = 2$ and $\bar{\eta} = 1$. For $m \geq 3$, this is only possible if u satisfies (4.29) for every $i = 1, \ldots, m$.

Locating the Inflection Point with Relative Distance to u

Let now $\tilde{u} := \delta u$ for some $\delta \in (0,1)$. Condition (4.26) yields

$$\sum_{j=1}^{m} \frac{\tilde{u}_j}{u_j} = \sum_{j=1}^{m} \frac{\delta u_j}{u_j} = m\delta > 1 \iff \delta > m^{-1},$$

i.e., the larger m is, the smaller δ can be chosen such that the resulting contour is valid. Condition (4.27) reads

$$(m-1)\delta u_i \geq (m\delta - 1)\, u_i \iff u_i(1-\delta) \geq 0,$$

which is valid for every $i = 1, \dots, m$, as $\delta \in (0,1)$. So the only limiting condition is $\delta > m^{-1}$. By setting $\delta := (\bar{u} - \bar{\eta})/\bar{u}$ with $\bar{u} := \max\{u_i : i = 1, \dots, m\}$ and $\bar{\eta} \in (0,1)$ a valid contour is defined for every $m \geq 2$ such that every integer-valued nondominated point in $B(u)$ has a strictly smaller level than all u^i, $i = 1, \dots, m$.

We conclude that for $m \geq 3$ even the generalized augmented weighted Tchebycheff norm does not provide enough flexibility to construct a contour for which the inflection point has the same small absolute distance from u with respect to every component. In particular, the point $\tilde{u} := u - e$ can not be chosen as inflection point for $m \geq 3$, in general. However, when the inflection point is located in a relative distance to u, the generalized augmented weighted Tchebycheff norm is suitable for arbitrary u under the rather mild condition that $\delta > m^{-1}$.

4.3.2 Relations Between Trade-Offs and Augmentation Parameters

In Section 4.1.2, we motivated the computation of the parameters of an augmented weighted Tchebycheff norm for integer-valued problems. The construction was based on the fact that due to the integrality of the nondominated points a part of the considered box $B(u)$ could be discarded because of being empty. If a continuous problem is given, we cannot exclude a part of the box $B(u)$ a priori, but a possible nondominated point might be located everywhere in $B(u)$. If we search for new nondominated points by solving a (generalized) augmented weighted Tchebycheff problem with $\rho > 0$, then only new nondominated points having a smaller level than the (nondominated) points used for defining the contour can be computed. Figure 4.8 depicts the 'reachable' and 'unreachable' area in the bicriteria case.

We can also say that only points having a certain trade-off can be generated. The fact that the parameter ρ of an augmented or modified weighted Tchebycheff scalarization provides a valuable source of trade-off information has been pointed out

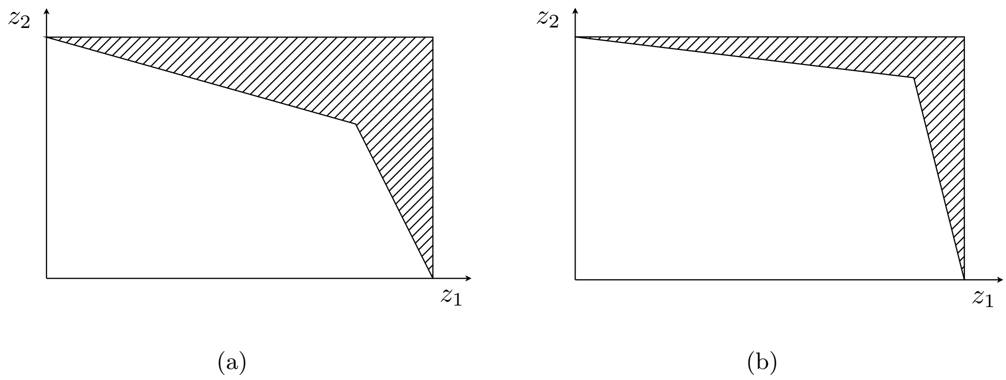


Figure 4.8: The 'reachable' (white) and 'unreachable' (shaded) area for different values of augmentation parameters

in, e.g., Kaliszewski (2000). The connection between trade-offs and the augmentation term(s) of different variants of Tchebycheff norms is studied in Kaliszewski and Michalowski (1997) and Kaliszewski (2000). Among others, it is shown in Kaliszewski (2000) that if a point $\bar{z}$ solves a generalized augmented weighted Tchebycheff problem

$$\min_{z \in Z} \max_{i=1,\dots,m} \left\{ w_i \left(z_i^U - z_i \right) \right\} + \sum_{j=1}^{m} \rho_j \left(z_j^U - z_j \right) \tag{4.30}$$

with $w, \rho \in \mathbb{R}^m_>$, then the trade-off as defined in (2.8) is bounded by

$$T_{ij}^G(\bar{z}) \leq \frac{w_j + \rho_j}{\rho_i}$$

for all $i, j = 1, \dots, m, i \neq j$. Note that Kaliszewski (2000) considers multicriteria optimization problems in maximization format. For minimization problems we analogously obtain

$$T_{ij}^G(\bar{z}) \leq \frac{\rho_i}{w_j + \rho_j} \tag{4.31}$$

for all $i, j = 1, \dots, m, i \neq j$. Conversely, it is demonstrated in Kaliszewski (2000) for $m = 2$ that specified trade-offs $T_{12}^G(\bar{z})$ and $T_{21}^G(\bar{z})$ can be translated into suitable values for ρ_1 and ρ_2 such that every (properly) nondominated point obtained as optimal solution of (4.30) satisfies the given trade-offs. The values of ρ are obtained by solving the linear system $T_{12}^G(\bar{z}) = (w_2 + \rho_2)\rho_1^{-1}$ and $T_{21}^G(\bar{z}) = (w_1 + \rho_1)\rho_2^{-1}$. Thereby, the weights w that determine the search direction can be chosen arbitrarily. However, as stated in Kaliszewski (2000), no straightforward generalization for $m \geq 3$ exists since the resulting system of $m(m-1)$ equations is not necessarily consistent.

In Podkopaev (2007) the problem of prescribing trade-offs for any $m \geq 2$ is studied. The author proposes to consider

$$\min_{z \in Z} \max_{i=1,\dots,m} \left\{ w_i \left(\left(z_i^U - z_i \right) + \sum_{\substack{j=1,\dots,m,\\ j \neq i}} \beta_{ij} \left(z_j^U - z_j \right) \right) \right\}, \tag{4.32}$$

where $(\beta_{ij})_{i,j=1,\dots,m}$ denotes a positive $(m \times m)$-matrix with $\beta_{ii} = 1$ for all $i = 1, \dots, m$ and $\beta_{ij}\beta_{jk} \leq \beta_{ik}$ for all $i \neq j$, $j \neq k$ which implies $\beta_{ij}\beta_{ji} \leq 1$ for $i = k$. It is shown that the trade-off of a point $\bar{z} \in Z$ that solves (4.32) is bounded by

$$T_{ij}^G(\bar{z}) \leq \frac{1}{\beta_{ji}}$$

for all $i, j = 1, \dots, m, i \neq j$. With the help of this formulation, a trade-off between every pair of objectives (for arbitrary many objectives) can be prescribed. The parameters β_{ij}, $i, j = 1, \dots, m$, represent augmentation parameters. The weights can be chosen arbitrarily.

Different from the approaches of Kaliszewski (2000) and Podkopaev (2007), which focus on the computation of the augmentation parameters and use arbitrarily chosen weights, Theorem 4.18 provides a formula for computing all parameters. Note that an adaptive selection of the weights is important for directing the search to a particular part of the search region. However, as pointed out above, only in the bicriteria case pairwise given trade-offs on all objectives can be transformed into suitable parameters. We conclude this section with a bicriteria example in which trade-offs are imposed.

Example 4.19. *Let $m = 2$ and let the two nondominated points $u^1 = (0, 6)^\top$ and $u^2 = (10, 0)^\top$ of an underlying minimization problem be given. First, assume that the problem is integer-valued and that parameters of (4.25) are to be constructed such that u^1, u^2 and $\tilde{u} := u - e = (9, 5)^\top$ lie on one contour. Alternatively, we can prescribe trade-offs $T_{12}^G(\tilde{u}) = T_{21}^G(u^1) \geq 1/9$ and $T_{21}^G(\tilde{u}) = T_{12}^G(u^2) \geq 1/5$. Then, using Theorem 4.18, we obtain*

$$w_1 = 5/14, \quad w_2 = 9/14, \quad \rho_1 = 25/308, \quad \rho_2 = 27/308.$$

For given weights we can also use (4.31) to compute ρ. Then, with $T_{12} := 1/9$ and $T_{21} := 1/5$ we obtain

$$\rho_1 = \frac{T_{12}(w_2 + w_1 T_{21})}{1 - T_{12}T_{21}}, \quad \rho_2 = \frac{T_{21}(w_1 + w_2 T_{12})}{1 - T_{12}T_{21}},$$

which yields the same values for ρ_1 and ρ_2 as computed above.

Now, assume that either a continuous problem is given or a representative subset that does not necessarily contain all nondominated points is to be computed. Let $\delta = {}^3\!/_5$ *and* $\tilde{u} := \delta u = (6, 3.6)^\top$ *or, alternatively,* $T^G_{12}(\tilde{u}) = T^G_{21}(u^1) \geq {}^2\!/_5$ *and* $T^G_{21}(\tilde{u}) = T^G_{12}(u^2) \geq {}^{10}\!/_9$*. As* $\delta > m^{-1}$ *we can apply the formulas derived in Theorem 4.18 and obtain*

$$w_1 = {}^3\!/_8, \quad w_2 = {}^5\!/_8, \quad \rho_1 = {}^3\!/_4, \quad \rho_2 = {}^5\!/_4.$$

Note that it is possible to prescribe a two-sided trade-off when using the generalized augmented weighted Tchebycheff problem for $m = 2$. If, instead, the classic augmented problem is used, only a one-sided trade-off, i.e., either $T^G_{12}(\tilde{u})$ or $T^G_{21}(\tilde{u})$ can be prescribed such that the resulting contour is uniquely determined. This is due to the fact that the inflection point can not be varied freely but lies on a curve, see Figure 4.6. Prescribing a one-sided trade-off is equivalent to fixing the slope of one of the lines of the contour. Thereby, the inflection point and, hence, the contour is fixed.

4.4 Parameters of Augmented ε-Constraint Scalarizations

The ideas presented in the foregoing sections for a (generalized) augmented weighted Tchebycheff method can be directly translated to an ε-constraint method with an augmenting l_1-term. While the notion of an augmented ε-constraint method has only recently been employed, see the approach of Mavrotas (2009) below, already the hybrid method can be interpreted as an augmented ε-constraint method. Recall from Section 2.3 that the hybrid method combines the ε-constraint method with a weighted sum objective.

A generalized ε-constraint method is given in Ruzika (2007), see also Ehrgott and Ruzika (2008). Therein, different improvements on the classic ε-constraint method are proposed. One improvement consists in adding slack variables to the constraints and supplementing the objective function by a weighted sum of these slack variables, which yields the problem

$$\begin{aligned} \min \quad & f_k(x) - \sum_{i \neq k} \mu_i s_i \\ \text{s.t.} \quad & f_i(x) + s_i \leq \varepsilon_i, \quad && i = 1, \ldots, m,\ i \neq k, \\ & s_i \geq 0, && i = 1, \ldots, m,\ i \neq k, \\ & x \in X, \end{aligned} \tag{4.33}$$

where $\varepsilon \in \mathbb{R}^m$ and $\mu \in \mathbb{R}^m_+$. The authors show that if (x^*, s^*) is an optimal solution of (4.33) with $\mu \in \mathbb{R}^m_>$, x^* is efficient. Furthermore, if $\mu \in \mathbb{R}^m_>$, every optimal solution of (4.33) satisfies $f_i(x^*) + s_i^* = \varepsilon_i$ for all $i \neq k$. Therefore, at optimality, (4.33) can be reformulated as

$$\begin{aligned} \min \quad & f_k(x) + \sum_{i \neq k} \rho_i f_i(x) \\ \text{s.t.} \quad & f_i(x) \leq \varepsilon_i, \qquad\qquad i = 1, \dots, m, \, i \neq k, \\ & x \in X \end{aligned} \tag{4.34}$$

with $\rho_i = \mu_i$ for all $i = 1, \dots, m$, since by setting $s_i := \varepsilon_i - f_i(x), i = 1, \dots, m$, the objective of (4.33) reads

$$f_k(x) - \sum_{i \neq k} \mu_i(\varepsilon_i - f_i(x)) = f_k(x) + \sum_{i \neq k} \mu_i f_i(x) - \sum_{i \neq k} \mu_i \varepsilon_i,$$

where $\sum_{i \neq k} \mu_i \varepsilon_i$ is a constant. The first constraint of (4.33) vanishes and the second becomes $f_i(x) \leq \varepsilon_i$ for all $i \neq k$. Therefore, (4.34) is a special case of the hybrid method with weights

$$\lambda = (\rho_1, \dots, \rho_{k-1}, 1, \rho_{k+1}, \dots, \rho_m)^\top.$$

In analogy to the previous section, we call (4.34) a *generalized augmented ε-constraint method.*

Mavrotas (2009) proposes a method called AUGMECON, an acronym for 'augmented ε-constraint method'. Thereby, the objective function of the classic ε-constraint method is augmented by a sum of slack variables. Analogously to the augmented weighted Tchebycheff method, one single augmentation parameter $\rho > 0$ is used. The considered problem is of the form

$$\begin{aligned} \min \quad & f_1(x) - \rho \sum_{i=2}^{m} \frac{s_i}{r_i} \\ \text{s.t.} \quad & f_i(x) + s_i \leq \varepsilon_i, \quad i = 2, \dots, m, \\ & s_i \geq 0, \quad i = 2, \dots, m, \\ & x \in X, \end{aligned} \tag{4.35}$$

where $\varepsilon \in \mathbb{R}^m$ and $r \in \mathbb{R}^m_>$ denotes the range of the objectives, obtained as difference between an estimate on the nadir point and the ideal point. Note that in order to ease comparison, we transformed the original maximization problem into minimization format. Moreover, note that (4.35) has the same structure as (4.33), hence, at optimality, we can transform it analogously into a problem of the form

$$\begin{aligned} \min \quad & f_k(x) + \rho \sum_{i \neq k} f_i(x) \\ \text{s.t.} \quad & f_i(x) \leq \varepsilon_i \quad \forall\, i \neq k, \\ & x \in X, \end{aligned} \tag{4.36}$$

with $k = 1$. Problem (4.36) is employed in Özpeynirci and Köksalan (2010a). The authors use this scalarization for integer-valued problems and indicate

$$\rho \in \left(0, \left(\sum_{i=1}^{m} z_i^M - z_i^I\right)^{-1}\right) \tag{4.37}$$

as an interval for a suitable choice of ρ where z^M denotes an upper bound on the nondominated set. Note that (4.36) corresponds to the hybrid method with

$$\lambda = (\rho, \ldots, \rho, 1, \rho, \ldots, \rho)^\top .$$

We call it *augmented ε-constraint method* in the following. Adaptive values of all parameters can be determined similar to the construction in the previous sections. In order to restrict the search for new nondominated points to the box $B(u)$, we set $\varepsilon_i := u_i$ for all $i \neq k$ in (4.36). For the computation of ρ we consider the contour of the objective function of (4.36), which is represented by an $(m-1)$-dimensional hyperplane of the form

$$H(\alpha) := \left\{ z \in \mathbb{R}^m : z_k + \rho \sum_{i \neq k} z_i = \alpha \right\}$$

with $\alpha > 0$. In order to obtain an appropriate value for ρ, we require that $\tilde{u} := u - e$ has a strictly smaller level α than the nondominated point defining u with respect to component k. Due to the integrality of the nondominated points, we assume without loss of generality that $u_i \geq 2$ holds for all $i = 1, \ldots, m$. The bicriteria case is illustrated in Figure 4.9 for $k = 2$. The contour is constructed such that $\tilde{u} := (u_1 - 1, u_2 - 1)^\top$ and $u^2(\bar{\eta}) := (0, u_2 - \bar{\eta})^\top$, $\bar{\eta} \in (0,1)$, both lie on $H(\alpha)$. In general, i.e., for arbitrary $m \geq 2$, we compute ρ and α such that $u^k(\bar{\eta}) \in H(\alpha)$ with $u^k(\bar{\eta}) := (0, \ldots, 0, u_k - \bar{\eta}, 0, \ldots, 0)^\top$, $\bar{\eta} \in (0,1)$, and $\tilde{u} := u - e \in H(\alpha)$. Hence,

$$u_k - \bar{\eta} = \alpha \quad \text{and} \quad u_k - 1 + \rho \sum_{i \neq k} (u_i - 1) = \alpha,$$

thus, $\rho \sum_{i \neq k} (u_i - 1) = 1 - \bar{\eta}$, which yields

$$\rho = \frac{1 - \bar{\eta}}{\sum_{i \neq k} (u_i - 1)}. \tag{4.38}$$

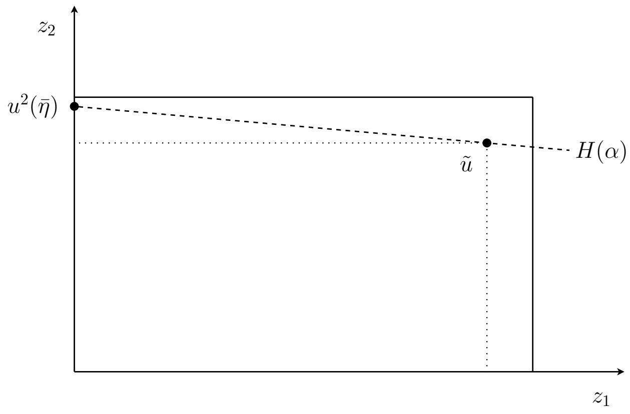


Figure 4.9: Contour line of an augmented ε-constraint problem for $m = 2$

Note that independent of m, the two points $u^k(\bar{\eta})$ and $\tilde{u}$ are sufficient to uniquely determine ρ and α. As in case of the augmented weighted Tchebycheff norm, the maximal value of ρ would be obtained for $\bar{\eta} = 0$. However, as discussed before, this choice is not possible as $u^k(\bar{\eta})$ must have a strictly smaller level than $\tilde{u}$.

4.5 Conclusion and Further Ideas

In this chapter, we studied the well-known augmented weighted Tchebycheff method and derived parameters for it such that every nondominated point of a discrete (integer-valued) multicriteria optimization problem with a finite nondominated set can be detected. In particular, we derived an upper bound on the parameter ρ such that for all choices of ρ smaller than this upper bound either a new nondominated point is generated or, otherwise, the considered box $B(u)$ can be discarded from further investigation.

We also studied a generalized augmented weighted Tchebycheff method that offers more flexibility. Moreover, based on the connection between trade-offs and augmentation parameters, we showed how trade-off information can be incorporated into our approach. Thereby, the proposed parameter scheme is not only applicable to discrete problems but also to continuous ones. Finally, we also stated an augmented ε-constraint method and showed how to choose the parameters in an analogous way.

In the future it would be interesting to study the approach of Podkopaev (2007) in more detail which allows the incorporation of pairwise trade-offs on all objectives for any number of criteria. A computation of all parameters would be useful such that all prescribed trade-offs can be respected *and* the investigation of a specified search region is possible.

5 A Parametric Algorithm with a New Bound on the Number of Subproblems

5.1 Introduction

In this chapter we present a new algorithm that generates a representation of the nondominated set by solving a series of scalarized problems whose parameters are varied in a systematic way. In particular, we focus on the generation of complete representations for discrete multicriteria optimization problems with a finite nondominated set. When generating complete representations, then, as already discussed in Section 3.1, common quality criteria like coverage, uniformity or cardinality are not meaningful. Instead, the main goal is to keep the number of subproblems as small as possible. Recall from the literature review in Section 3.2 that in the bicriteria case approaches are known which require the solution of at most $2|Z_N| - 1$ subproblems. Thereby, $|Z_N|$ subproblems are solved to generate all points in Z_N, and the additional $|Z_N| - 1$ subproblems are needed to ensure that no further nondominated points exist between the already generated ones. Corresponding algorithms have been proposed by Chalmet et al. (1986), who use a hybrid method as scalarization, and Ralphs et al. (2006), who employ Tchebycheff scalarizations. If the ε-constraint method is chosen as scalarization, the number of subproblems to be solved at maximum can be reduced to $|Z_N| + 1$ if the points are generated in a specific order, see, e.g., Laumanns et al. (2006). By indicating a corresponding algorithm for general $m \geq 2$, Laumanns et al. (2006) show that at most $(|Z_N| + 1)^{m-1}$ subproblems need to be solved to obtain a complete representation. While no better bound can be expected in the bicriteria case, the quadratic bound in the tricriteria case seemed not to be tight. Numerical experiments of Laumanns et al. (2006) for a knapsack problem with three objectives reveal that the number of subproblems that are solved in their algorithm is considerably smaller than $(|Z_N| + 1)^2$. Further algorithms for generating complete representations for discrete multicriteria optimization problems are proposed in Özlen and Azizoğlu (2009), Lokman and Köksalan (2013), Kirlik and Sayın (2014) and Ozlen et al. (2014). We refer to Section 3.3 for a detailed description of these algorithms. However, no better theoretical bound on the number of subproblems could be proven so far.

In the following, we propose a new parametric algorithm for which the number of subproblems solved to generate a complete representation of the nondominated set of a tricriteria optimization problem depends linearly on the number of nondominated points. More precisely, if $|Z_N| \geq 3$ and if the ideal point and an arbitrary upper bound on Z are given, at most $3|Z_N| - 2$ subproblems have to be solved. The linear bound is achieved by the definition of a new split criterion which allows to exclude redundant sets from the decomposition of the so-called *search region*. The latter describes a set in which further nondominated points might be contained. In our approach we describe the search region as the union of rectangular sets called *boxes*. All boxes have the ideal point as common lower vertex. Any upper bound on Z can be used as upper vertex of the initial search region. Whenever a new nondominated point is computed, this point and all points dominated by it are eliminated from the search region. This is achieved by splitting the search region according to some specific rule. In each iteration, a box is selected from the current decomposition of the search region and a corresponding subproblem is solved. To this end, any scalarization can be used with the help of which the selected box can be investigated. This means that a new nondominated point in the considered box has to be generated whenever there exists one, and the selected box has to be identified as empty otherwise. The weighted Tchebycheff method and the ε-constraint method can be used for this purpose if an appropriate parameter scheme as proposed, e.g., in Chapter 4 is applied. If the ε-constraint method is used, the upper bound on the number of subproblems can even be decreased to $2|Z_N| - 1$ due to particular properties of this scalarization. Note that the number of boxes of the decomposition which are obtained in the course of the algorithm equals the number of iterations of the algorithm, as every box is investigated exactly once. Moreover, since in every iteration one subproblem is solved, the number of iterations also equals the number of subproblems. Therefore, when addressing the upper bound on the number of subproblems, we can equivalently speak of an upper bound on the number of iterations or boxes.

The result that the search region of a tricriteria problem can be decomposed into a number of boxes (hypercubes) that depends linearly on the number of nondominated points goes in line with results from the field of computational geometry. In Boissonnat et al. (1998) it is shown that for a set of n points in $\mathbb{R}^m$ the maximum complexity of its Voronoi diagram under the l_∞-metric is $\mathcal{O}(n^{\lceil m/2 \rceil})$. Moreover, it is demonstrated that the same complexity holds for the union of n axis-parallel hypercubes in $\mathbb{R}^m$. If all hypercubes have the same size, the complexity can be improved to $\mathcal{O}(n^{\lfloor m/2 \rfloor})$ for $m \geq 2$. It remains $\mathcal{O}(n)$ for $m = 1$. While the boxes, into which we decompose the search region, are not of the same size, in general, they all share the vertex z^I. As shown in Bringmann (2013) an instance in which all boxes share one common vertex can be transformed into an instance in which all boxes have the same size. Therefore, the upper bound $\mathcal{O}(n^{\lfloor m/2 \rfloor})$ for $m \geq 2$ holds for a set of n nondominated points and yields $\mathcal{O}(n)$ for $m = 3$. However, no algorithm is indicated in Boissonnat et al. (1998) or Bringmann (2013).

A topic that is closely related to the description of the search region with respect to a set of nondominated points is the computation of the dominated hypervolume with respect to a set of points, see the definition in (2.19). Indeed, the dominated hypervolume can be seen as a complement to the search region in the sense that the union of both sets (with respect to the same set of points) yields the initial search region. Consequently, similar algorithms can be used to compute either the search region or the dominated hypervolume with respect to a given set of points. Beume et al. (2009) and Guerreiro et al. (2012) present algorithms for the determination of the dominated hypervolume with respect to a given set of n points for $m = 3$ and $m = 4$, respectively. They order the set of points with respect to one component beforehand and apply a so-called dimension sweep technique that makes use of the order. The overall complexity of the algorithms is shown to be $\mathcal{O}(n \log n)$ for $m = 3$ (Beume et al., 2009) and $\mathcal{O}(n^2)$ for $m = 4$ (Guerreiro et al., 2012). Note that this complexity refers to the overall algorithm and does not only count the number of hypercubes (subproblems) as the bounds stated above do.

The algorithm that is proposed in this chapter can also be used when a set of points is given and the search region potentially containing further nondominated points shall be generated. However, different from the approaches that apply a dimension-sweep technique, we insert the points one by one and update the search region directly after each insertion. An application of computing an initial search region with respect to a given set of points and updating it iteratively is given within a two phase method. In Przybylski et al. (2010a) all supported nondominated points are computed in a first stage. They define an initial search region. The nonsupported nondominated points, which are detected in the second phase, require an iterative update of the search region. In the approach of Przybylski et al. (2010a) basically the same decomposition is obtained as in our approach. However, no bound on the number of boxes, described by their respective upper vertices, is derived.

The remainder of this chapter is organized as follows. In Section 5.2 we present a decomposition of the search region based on nondominance and propose a general box algorithm for $m \geq 2$. We show that this algorithm may produce redundant boxes for problems with more than two criteria which makes the algorithm inefficient, in general. In Section 5.3 an improved split is presented for the tricriteria case. Under the technical assumption that all nondominated points differ pairwise in every component, we show how to construct a decomposition that only contains non-redundant boxes. As our main result, we prove that the number of boxes is bounded by $3|Z_N|-2$ for $|Z_N| \geq 3$. Finally, we show that the algorithm can also be applied if the nondominated points are in arbitrary (non-general) position, i.e., if every pair of points may have up to one equal component for $m=3$. The upper bound $3|Z_N|-2$ is also valid in this general case. In Section 5.4 the ε-constraint method is studied as a special scalarization for which the number of subproblems can be reduced further. In the tricriteria case we obtain a new upper bound of $2|Z_N|-1$. In Section 5.5 we propose a generalization of the algorithm presented in Section 5.3 that can be applied to problems with any number of criteria. Moreover, in the tricriteria case, this algorithm is supposed to reduce the number of boxes further if the nondominated points are in arbitrary (non-general) position. However, no theoretical upper bound on the number of subproblems analogous to the bound in Section 5.3 can be derived. Section 5.6 provides a conclusion and further ideas. Parts of this chapter have already been published as a technical report in Dächert and Klamroth (2013).

5.2 Split of the Search Region for Multicriteria Problems

Let B_0 denote an initial search region of the form

$$B_0 := \{z \in \mathbb{R}^m \,:\, l_j \leq z_j < u_j,\, j=1,\dots,m\}$$

with $l,u \in \mathbb{R}^m, l \leq u$. As lower and upper vertex of B_0 we choose a global lower and upper bound on the set of feasible outcomes, for example, $l := z^I$ and $u := z^M$ as defined in Section 2.1. Alternatively, explicit bounds on the search region as provided, for example, by a decision maker can be used to specify l and u.

If no special scalarization method is employed, the iterative reduction of the search region can solely be based on nondominance. Thereby, every generated nondominated point allows to restrict the search region, as for any $z^* \in Z_N$ the two sets

$$S_1(z^*) := \{z \in B_0 \,:\, z \leqq z^*\} \quad \text{and} \quad S_2(z^*) := \{z \in B_0 \,:\, z \geqq z^*\}$$

do not contain any nondominated points besides z^*, i.e., $S_1(z^*) \cap Z_N = \{z^*\}$ as well as $S_2(z^*) \cap Z_N = \{z^*\}$. Moreover, $S_1(z^*) \cap Z = \{z^*\}$, thus, $S_1(z^*) \backslash \{z^*\}$ contains no feasible points.

5.2.1 A Full-Dimensional Split

In the following, we decompose a given initial search region B_0 iteratively into subsets $B \subset B_0$ of the same form, i.e., into sets $B := \{x \in \mathbb{R}^m \,:\, l_j \leq x_j < u'_j,\, j=1,\dots,m\}$

with $u' \in \mathbb{R}^m, l \leq u' \leq u$. As the initial search region that potentially contains nondominated points of (2.1) as well as each subset B as defined above describe rectangular subsets of $\mathbb{R}^m$ with sides parallel to the coordinate axes, we call these sets *boxes* in the following. The search region is always represented as the union of certain boxes B. With the generation of every new nondominated point we replace some of the boxes of the current search region by appropriate new boxes such that the whole search region is covered. This property is called correctness in the following.

Definition 5.1 (Correct decomposition)**.** *Let B_0 denote the starting box, let $\mathcal{B}_s$ denote the set of boxes at the beginning of iteration $s \geq 1$, where $\mathcal{B}_1 := \{B_0\}$, and let $z^p \in Z_N$, $p = 1, \dots, s-1$, be already determined nondominated points. We call $\mathcal{B}_s$ correct with respect to $z^1, \dots, z^{s-1}$ if*

$$B_0 \setminus \left(\bigcup_{B \in \mathcal{B}_s} B \right) = \bigcup_{p=1,\dots,s-1} S_2(z^p) \tag{5.1}$$

holds, where $S_2(z^p) := \{z \in B_0 : z \geqq z^p\}$ denotes that subset of the box B_0 that is dominated by the point $z^p \in Z_N, p = 1, \dots, s-1$.

Any split presented in the following maintains a correct decomposition of the search region at any time. Under this basic condition, we try to generate as few boxes as possible, as for every generated box a scalarized subproblem needs to be solved. Our aim is to keep the number of subproblems low. The simplest split decomposes a box B which contains a new outcome $z^* \in (B \cap Z_N)$ into m subboxes (see also Tenfelde-Podehl (2003) or Dhaenens et al. (2010)).

Definition 5.2 (Full m-split)**.** *Let a nondominated point $z^* \in (B \cap Z_N)$ be given. We call the replacement of B by the m sets*

$$B_i := \{z \in B \,:\, z_i < z_i^*\} \quad \forall\, i = 1, \dots, m \tag{5.2}$$

a full m-split of B.

Recursively applying the full m-split to every box which contains the current nondominated point yields a correct decomposition, as the following lemma shows.

Lemma 5.3 (Correctness of the full m-split)**.** *Let $\mathcal{B}_s, s \geq 1$, with $\mathcal{B}_1 := \{B_0\}$ be a correct decomposition with respect to the nondominated points $z^1, \dots, z^{s-1}$, and let $z^s \in Z_N$. If a full m-split is applied to all boxes $B \in \mathcal{B}_s$ with $z^s \in B$, then the resulting decomposition is correct.*

Proof. By induction on s.
$\underline{s=1}$: Let $\mathcal{B}_1 := \{B_0\}$, and $z^1 \in Z_N$. Then, by definition of the full m-split, B_0 is replaced by m boxes. It holds that

$$B_0 \setminus \left(\bigcup_{B \in \mathcal{B}_2} B \right) = B_0 \setminus \left(\bigcup_{i=1,\dots,m} \{z \in B_0 \,:\, z_i < z_i^1\} \right) = S_2(z^1),$$

thus, $\mathcal{B}_2$ is correct.

$\underline{s \to s+1}$: Let $\mathcal{B}_s$ be correct, and let $z^s \in Z_N$. Let $\overline{\mathcal{B}}_s \subset \mathcal{B}_s$ denote the set of all boxes $B \in \mathcal{B}_s$ for which $z^s \in B$ holds. Let I be the index set of these boxes and let $Q := |\overline{\mathcal{B}}_s|$. Now, let a full m-split with respect to z^s be applied to all $B \in \overline{\mathcal{B}}_s$, i.e., each of the boxes $B^{I(q)}, q = 1, \dots, Q$, is replaced by m new boxes $B_1^{I(q)}, \dots, B_m^{I(q)}, q = 1, \dots, Q$ and

$$\bigcup_{\substack{i=1,\dots,m\\ q=1,\dots,Q}} B_i^{I(q)} = \bigcup_{B \in \overline{\mathcal{B}}_s} B \setminus S_2(z^s)$$

holds. Then

$$\begin{aligned}
B_0 \setminus \left(\bigcup_{B \in \mathcal{B}_{s+1}} B \right) &= B_0 \setminus \left(\left(\bigcup_{B \in \mathcal{B}_s \setminus \overline{\mathcal{B}}_s} B \right) \cup \left(\bigcup_{\substack{i=1,\dots,m\\ q=1,\dots,Q}} B_i^{I(q)} \right) \right) \\
&= B_0 \setminus \left(\left(\bigcup_{B \in \mathcal{B}_s \setminus \overline{\mathcal{B}}_s} B \right) \cup \left(\bigcup_{B \in \overline{\mathcal{B}}_s} B \setminus S_2(z^s) \right) \right) = B_0 \setminus \left(\left(\bigcup_{B \in \mathcal{B}_s} B \setminus S_2(z^s) \right) \right) \\
&= \left(B_0 \setminus \left(\bigcup_{B \in \mathcal{B}_s} B \right) \right) \cup S_2(z^s) = \bigcup_{p=1,\dots,s} S_2(z^p).
\end{aligned}$$

□

Note that all new boxes $B \in \mathcal{B}_{s+1}, s \geq 2$, obtained from boxes in $\overline{\mathcal{B}}_s$, are defined as sets with open upper boundary, as we need to exclude z^s from the search region in order to prevent it from further generation. In practical applications, it will often be useful to replace the boxes by closed subsets and exclude z^s by using, for example, appropriate scalarization approaches.

Also note that we describe the boxes by their upper vertex u only and that the lower vertex of all boxes is kept constant. This means that the decomposition of the search region contains the union of the sets $S_1(z^p) \setminus \{z^p\}$, $1 \leq p \leq s$, for all nondominated points $z^p \in Z_N$ which have already been generated by the algorithm, even if these sets do not contain any feasible points. However, the split operation is simplified by including these sets, since a box is never split into more than m new boxes.

A Generic Algorithm Based on the Full m-Split

Algorithm 1 shows a basic algorithm using the full m-split. Due to Lemma 5.3, the algorithm is correct as it does not exclude regions from the search region which might contain further nondominated points. A problem formulation is given as input, which is denoted by Z. Note that this does not mean that the set of feasible outcomes is known explicitly, but it is to be understood as a substitute for the objective functions and the constraints.

Algorithm 1 Algorithm with full m-split

Input: Image of the feasible set $Z \subset \mathbb{R}^m$, implicitly given by some problem formulation

$N := \emptyset$; $\delta > 0$;
InitStartingBox(Z, δ);
$s := 1$; // Initialize starting box
while $\mathcal{B}_s \neq \emptyset$ **do**
  Choose $B \in \mathcal{B}_s$;
  $z^s := opt(Z, u(B))$; // Solve scalarized subproblem
  if $z^s = \emptyset$ **then** // Subproblem infeasible
    $\mathcal{B}_{s+1} := \mathcal{B}_s \backslash \{B\}$; // Remove (empty) box
  else
    $N := N \cup \{z^s\}$; // Save nondominated point
    $\mathcal{B}_{s+1} := \mathcal{B}_s$; // Copy set of current boxes
    GenerateNewBoxes$(\mathcal{B}_s, z^s, z^I, \mathcal{B}_{s+1})$;
  end if
  $s := s + 1$;
end while

Output: Set of nondominated points N

procedure InitStartingBox(Z, δ)
  for $j = 1$ to m **do** // Compute bounds on Z
    $z_j^I := \min\{z_j \,:\, z \in Z\}$;
    $z_j^M := \max\{z_j \,:\, z \in Z\} + \delta$;
    $u_j(B_0) := z_j^M$;
  end for
  $\mathcal{B}_1 := \{B_0\}$; // Initialize set of boxes
  return $\mathcal{B}_1$
end procedure

25: **procedure** GenerateNewBoxes($\mathcal{B}_s, z^s, z^I, \mathcal{B}_{s+1}$)
26: **for all** $\hat{B} \in \mathcal{B}_s$ **do**
27: **if** $z^s < u(\hat{B})$ **then** // Point is contained in box
28: $\mathcal{B}_{s+1} := \mathcal{B}_{s+1} \setminus \{\hat{B}\}$; // Remove box
29: **for** $i = 1$ to m **do** // Apply full m-split
30: **if** $z_i^s > z_i^I$ **then**
31: $B' := \emptyset$; // Create new box
32: $u_i(B') := z_i^s$; // Update upper bound
33: $u_j(B') := u_j(\hat{B}) \; \forall\, j \neq i$;
34: $\mathcal{B}_{s+1} := \mathcal{B}_{s+1} \cup \{B'\}$; // Append new box
35: **end if**
36: **end for**
37: **end if**
38: **end for**
39: **return** $\mathcal{B}_{s+1}$
40: **end procedure**

As long as $\mathcal{B}_s$ contains unexplored boxes, a box B is selected according to some rule as specified, for example, by an error measure or by a decision maker. However, as we are interested in generating the entire nondominated set, no special rule is employed in the following, and we may, for example, always take the first box in the list $\mathcal{B}_s$. The upper bound vector $u(B)$ of the chosen box B is used to determine the parameters of the selected scalarization. Note that the scalarization method can be chosen freely as long as it is guaranteed that the method finds a nondominated point in B whenever there exists one. For example, the augmented weighted Tchebycheff scalarization with a parameter scheme as presented in Chapter 4 is an appropriate method. We do not specify a scalarization method in Algorithm 1, but express by $opt(\cdot)$ in Line 6 that any scalarization can be chosen that is suited for the discrete or non-convex case. The result of the subproblem is either a nondominated point z^s in the considered box or the detection of infeasibility which corresponds to the situation in which the considered box does not contain further nondominated points. In the latter case, B is removed from the list $\mathcal{B}_s$ and the iteration is finished. Otherwise, z^s is saved and all boxes $\hat{B} \in \mathcal{B}_s$ are identified that contain z^s. All these boxes are split with respect to all $i \in \{1, \ldots, m\}$ for which $z_i^s > z_i^I$ holds and replaced by the new boxes. The algorithm iterates until all boxes are explored. Then the entire nondominated set has been detected.

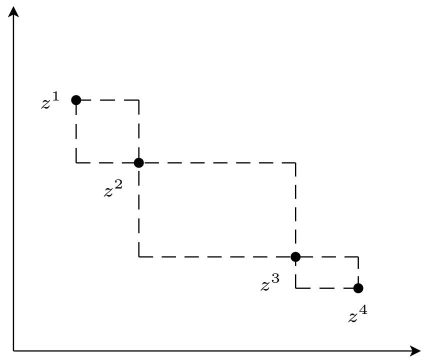


Figure 5.1: Decomposition of the search region for $m = 2$

The Bicriteria Case

For $m = 2$, Algorithm 1 is not only correct but also efficient, in the sense that the number of subproblems that need to be solved depends linearly on the number of nondominated points. As the decomposition does not contain redundant boxes, an upper bound on the number of boxes can easily be derived, which can be seen as follows. Let B_0 denote the starting box and let $z^1 \in B_0 \cap Z_N$ be the first generated point. Consider the two new boxes B_1, B_2 replacing B_0 in the first iteration. It holds that

$$\begin{aligned} B_1 \cap Z &= \{z \in B_0 \,:\, z_1 < z_1^1\} \cap Z = \Big(\{z \in B_0 \,:\, z_1 < z_1^1\} \cap Z\Big) \setminus S_1(z^1) \\ &= \{z \in B_0 \,:\, z_1 < z_1^1, z_2 > z_2^1\} \cap Z \end{aligned}$$

and, analogously,

$$B_2 \cap Z = \{z \in B_0 \,:\, z_2 < z_2^1\} \cap Z = \{z \in B_0 \,:\, z_2 < z_2^1, z_1 > z_1^1\} \cap Z,$$

thus, $(B_1 \cap Z) \cap (B_2 \cap Z) = \emptyset$. Therefore, the second generated point $z^2 \in Z_N$ is contained in exactly one of the two boxes B_1, B_2. This box is again split into two new boxes whose intersections with Z are disjoint among themselves as well as from the box (intersected with Z) which has not been changed in the current iteration. Repeating this argument, we see that for $m = 2$, no redundancy occurs. Therefore, we can easily indicate the number of iterations of Algorithm 1 in the bicriteria case based on the knowledge that a new nondominated point lies in exactly one box. In the initialization phase, z^I and z^M are computed in order to define B_0. In every iteration, either a (new) nondominated point is generated or a box is discarded from the search region. For every new nondominated point $z^s > z^I$, two new boxes replace

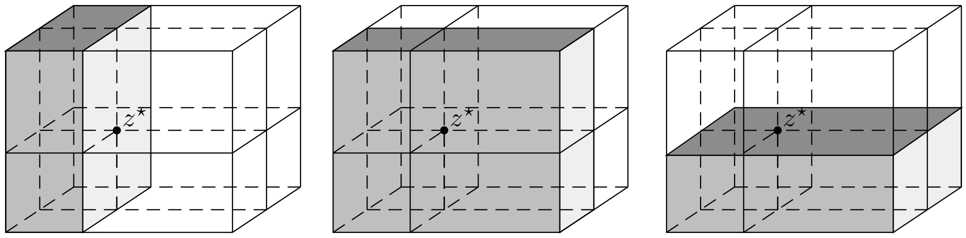


Figure 5.2: Boxes B_i, $i = 1, 2, 3$, obtained by a full 3-split of the initial search region with respect to $z^\star \in Z_N$

the currently investigated box, and for each of the two lexicographic optimal points (defining the ideal point) the current search box is replaced by one new box. So, the total number of iterations is $2|Z_N| - 1$, see Chalmet et al. (1986) or Ralphs et al. (2006).

In Figure 5.1, we illustrate the search region after four nondominated points z^1, z^2, z^3, z^4 have been generated. If we assume that these solutions build the entire nondominated set, Algorithm 1 terminates after seven iterations.

5.2.2 Redundancy for $m \geq 3$

When dealing with more than two criteria, the full m-split can also be applied, see Dhaenens et al. (2010) and Figure 5.2 for an illustration for $m = 3$. However, for $m \geq 3$, a nondominated point may lie in the intersection of multiple boxes. If we perform the full m-split in every box which contains the current nondominated point, we typically create nested and, thus, redundant subboxes. This is illustrated in the following example.

Example 5.4. *Let $m = 3$ and let the initial search region be given by*

$$B_0 := \{z \in Z \,:\, 0 \leq z_i \leq 5 \;\forall\, i = 1, 2, 3\}.$$

Assume that the first nondominated point that is generated is $z^1 = (2, 2, 2)^\top$. Performing a full 3-split in B_0 with respect to z^1 replaces the search region B_0 by the three sets

$$B_{1,i} := \{z \in B_0 \,:\, z_i < 2\},\; i = 1, 2, 3.$$

Let $z^2 = (1, 1, 4)^\top$ be the next nondominated point that is generated. It holds that $z^2 \in B_{11}$ as well as $z^2 \in B_{12}$, but $z^2 \notin B_{13}$. Performing a full 3-split in B_{11} yields

$$B_{21} := \{z \in B_0 \,:\, z_1 < 1\},$$
$$B_{22} := \{z \in B_0 \,:\, z_1 < 2, z_2 < 1\},$$

$$B_{23} := \{z \in B_0 \,:\, z_1 < 2, z_3 < 4\}.$$

Performing a full 3*-split in* B_{12} *yields*

$$\begin{aligned} B'_{21} &:= \{z \in B_0 \,:\, z_1 < 1, z_2 < 2\},\\ B'_{22} &:= \{z \in B_0 \,:\, z_2 < 1\},\\ B'_{23} &:= \{z \in B_0 \,:\, z_2 < 2, z_3 < 4\}. \end{aligned}$$

It holds that $B'_{21} \subset B_{21}$ *and* $B_{22} \subset B'_{22}$*, thus, the boxes* B_{22} *and* B'_{21} *are redundant in the decomposition of* B_0*.*

If redundant boxes are kept in the decomposition, this typically increases the running time of the algorithm, as additional, unnecessary scalarized subproblems are solved. Depending on the given problem, this may be time-consuming. Thus, redundant boxes should be detected and removed immediately. In the following, we will analyze under which conditions redundant boxes can occur. We first define our notion of non-redundancy:

Definition 5.5 (Non-redundant decomposition)**.** *Let* B_0 *denote the starting box and let* $\mathcal{B}_s$ *be a correct decomposition at the beginning of iteration* $s \geq 1$*. We call* $\mathcal{B}_s$ *(and every* $B \in \mathcal{B}_s$*) non-redundant if for every pair of boxes* $B, \tilde{B} \in \mathcal{B}_s, B \neq \tilde{B}$*, it holds that*

$$\exists\, i \in \{1, \ldots, m\} : u_i(B) < u_i(\tilde{B}) \text{ and } \exists\, j \in \{1, \ldots, m\} : u_j(B) > u_j(\tilde{B}).$$

In the case that $u(B) \leq u(\tilde{B})$ *we say that box* $\tilde{B}$ *dominates* B*.*

Note that the definition of a dominated box is somehow opposite to the definition of a dominated point. While $u \in \mathbb{R}^m$ is dominated by $u' \in \mathbb{R}^m$ if $u \geqq u'$, box B is dominated by B' if $u(B) \leqq u(B')$.

For simplicity, we make a technical assumption concerning the values of the nondominated points that will be removed later. Moreover, we define our general setting.

Assumption 5.6. *Let the following hold:*

1. *For all nondominated points* $z^p \in Z_N, p = 1, \ldots, s$*, generated up to iteration* $s \geq 1$*, it holds that* $z^p_j \neq z^q_j$ *for all* $j = 1, \ldots, m$ *and* $1 \leq q < p$*.*
2. *The starting box* B_0 *is non-empty, and* $\mathcal{B}_1 := \{B_0\}$ *denotes the initial decomposition of the search region.*
3. *For every* $1 \leq i \leq s$*,* $\mathcal{B}_i$ *is a correct, non-redundant decomposition of the search region. By* $\overline{\mathcal{B}}_s := \{B \in \mathcal{B}_s : z^s \in B\}$ *we denote the subset of boxes in iteration* s *containing* z^s*.*

Lemma 5.7 (Generation of redundant boxes)**.** *Let Assumption 5.6 be satisfied. If we apply a full m-split to every box $B \in \overline{\mathcal{B}}_s$, then redundancy can only occur among the 'descendants' of two different boxes which have been split with respect to the same component in the current iteration.*

Proof. We first show that no redundancy occurs between two boxes if at least one of the boxes has not been changed in the current iteration. Therefore, consider two arbitrary boxes $B, \tilde{B} \in \mathcal{B}_s$ where $\tilde{B} \in \mathcal{B}_s \backslash \overline{\mathcal{B}}_s$:

1. If $B \in \mathcal{B}_s \backslash \overline{\mathcal{B}}_s$, both boxes remain unchanged in the current iteration and, thus, due to Assumption 5.6 (3), both boxes are non-redundant.
2. If $B \in \overline{\mathcal{B}}_s$, none of the boxes obtained from a split in B can dominate $\tilde{B}$, as B does not dominate $\tilde{B}$ and the upper bound of B is only decreased by the split. Conversely, $\tilde{B}$ cannot dominate any of the boxes obtained from a split in B, as $\tilde{B} \in \mathcal{B}_s \backslash \overline{\mathcal{B}}_s$ implies that $u_j(\tilde{B}) \leq z_j^s$ for at least one $j \in \{1, \dots, m\}$. As $z^s < u(B)$, for every $B_i, i = 1, \dots, m$, resulting from a split of B it holds that $z_i^s = u_i(B_i)$ and $z_j^s < u_j(B_i)$ for all $j \neq i$. Thus, $\tilde{B}$ dominates B_i if and only if $z_i^s = u_i(\tilde{B})$ holds. This, however, is excluded by Assumption 5.6 (1).

Therefore, redundancy can only occur among newly generated boxes. Consider two boxes $B_i \neq \hat{B}_j$ obtained from $B, \hat{B} \in \overline{\mathcal{B}}_s$ (the case $B = \hat{B}$ is included) that are split with respect to components $i \neq j$. Then it holds that $u_i(B_i) = z_i^s < u_i(\hat{B}_j)$ and $u_j(B_i) > z_j^s = u_j(\hat{B}_j)$, thus, none of the boxes can dominate the other one. It follows that redundancy can only occur among the descendants of two different boxes that are split with respect to the same component. □

Corollary 5.8. *Let Assumption 5.6 hold. If only one box is split in some iteration, then all m resulting subboxes are non-redundant. In particular, the boxes obtained in the first iteration are always non-redundant.*

Corollary 5.9. *Let Assumption 5.6 hold. Let two boxes $B, \hat{B} \in \overline{\mathcal{B}}_s$ be split with respect to the same component $i = 1, \dots, m$. Then the resulting boxes $B_i, \hat{B}_i$ are non-redundant if and only if there exists an index $p \neq i$ such that $u_p(B_i) < u_p(\hat{B}_i)$ and there exists an index $q \neq i$ such that $u_q(B_i) > u_q(\hat{B}_i)$.*

These observations allow to detect redundant boxes by checking specific boxes of the current decomposition. Translating this to an algorithm for arbitrary $m \geq 3$, we apply, in every iteration, a full m-split to every box containing the current solution. If more than one box is split in one iteration, we compare the upper bounds of those new boxes that were generated with respect to the same component pairwise to detect redundancy. The respective boxes are then removed from the decomposition.

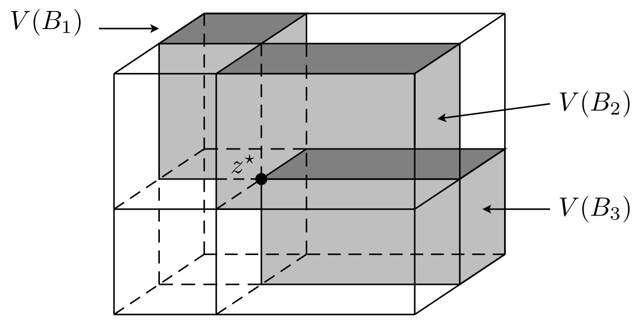


Figure 5.3: Individual subsets $V(B_i)$, $i = 1, 2, 3$, obtained by a full 3-split of the initial search region with respect to $z^\star \in Z_N$

A corresponding algorithm can be improved further if redundant boxes are already detected before their creation. In the next section we develop such an explicit criterion for tricriteria problems that indicates already before the split is performed whether the resulting box is redundant or not, and, thus, allows to maintain only non-redundant boxes in the decomposition. We prove that the number of non-redundant boxes or, equivalently, the number of subproblems to be solved in the course of an algorithm based on such an improved split operation depends linearly on the number of nondominated points.

5.3 An Improved Split in the Tricriteria Case

5.3.1 Individual Subsets and the v-Split

According to Definition 5.5 a non-redundant box can be characterized as follows. A box is non-redundant if and only if it contains a non-empty subset which is not part of any other box of the decomposition. These subsets are studied in the following.

Definition 5.10 (Individual subsets)**.** *Let $\mathcal{B}_s, s \geq 1$ be a non-redundant decomposition. For every $B \in \mathcal{B}_s$, the set*

$$V(B) := B \setminus \left(\bigcup_{\tilde{B} \in \mathcal{B}_s \setminus \{B\}} \tilde{B} \right) \tag{5.3}$$

is called individual subset of B.

Obviously, for every $B \in \mathcal{B}_s, s \geq 1$, it holds that $V(B) \subseteq B$ and $V(B) \cap V(\tilde{B}) = \emptyset$ for every $\tilde{B} \in \mathcal{B}_s, \tilde{B} \neq B$. Figure 5.3 shows the individual subsets of the three boxes $B_i, i = 1, 2, 3$, in $\mathbb{R}^3$ obtained by a full 3-split of the initial search box, which are depicted in Figure 5.2.

Now, maintaining only non-redundant boxes in the decomposition of the search region is equivalent to maintaining boxes with non-empty individual subsets. An explicit split criterion should indicate already before performing the split whether a given box will have a non-empty individual subset after having performed the split. To this end, we have to describe the individual subsets explicitly. For $m = 3$, we observe that the individual subset of a box is bounded by the neighbors of that box. After defining the neighbor of a box with respect to a certain component, we show its existence and indicate the respective neighboring boxes by a constructive proof.

Definition 5.11 (Neighbor of a box)**.** *Let $\mathcal{B}_s, s \geq 1$ be a non-redundant decomposition of the search region, and let $\underline{u}_i := \min\{u_i(B) : B \in \mathcal{B}_s\}$. Let any $\bar{B} \in \mathcal{B}_s$ be given. For every $i \in \{1, 2, 3\}$, for which $u_i(\bar{B}) > \underline{u}_i$, we call a box $\hat{B} \in \mathcal{B}_s \backslash \{\bar{B}\}$ that satisfies*

$$u_i(\hat{B}) < u_i(\bar{B}), \tag{5.4}$$

$$u_j(\hat{B}) > u_j(\bar{B}) \quad \text{for some } j \neq i, \tag{5.5}$$

$$u_k(\hat{B}) \geq u_k(\bar{B}) \quad \text{for } k \neq i, j \tag{5.6}$$

and

$$u_i(\hat{B}) = \max\{u_i(B) : B \in \mathcal{B}_s \backslash \{\bar{B}\}, u_i(B) < u_i(\bar{B})\} \tag{5.7}$$

the neighbor of $\bar{B}$ with respect to i at the beginning of iteration s, denoted by $B_i^s(\bar{B})$.

Example 5.12. *Consider Figure 5.2, which depicts the three boxes that are obtained in the first iteration. At the beginning of iteration $s = 2$, it holds that $B_1^2(B_2) = B_1$, since B_1 is the unique box satisfying (5.4)–(5.7) for $\bar{B} := B_2$. Analogously, $B_3^2(B_2) = B_3$ holds. A neighbor $B_2^2(B_2)$ is not defined as $u_2(B_2) = \underline{u}_2$.*

The following lemma shows that, under appropriate assumptions, for every box $\bar{B} \in \mathcal{B}_s$ and every component $i \in \{1, 2, 3\}$ for which $u_i(\bar{B}) > \underline{u}_i$ holds there exists a unique neighbor $B_i^s(\bar{B})$ satisfying (5.4)–(5.7) of Definition 5.11. These neighbors, which will be indicated with the help of a constructive proof, will turn out to be the boxes that define the individual subset of $\bar{B}$.

Assumption 5.13. *Let the following hold:*

1. *For all nondominated points $z^p \in Z_N, p = 1, \ldots, s$, generated up to iteration $s \geq 1$, it holds that $z_j^p \neq z_j^q$ for all $j \in \{1, 2, 3\}$ and $1 \leq q < p$.*
2. *The starting box B_0 is non-empty, and $\mathcal{B}_1 := \{B_0\}$ denotes the initial decomposition of the search region.*

3. *For every iteration* $1 \leq i \leq s$*, the set* $\mathcal{B}_{i+1}$ *is obtained from* $\mathcal{B}_i$ *by applying a full* 3*-split to every* $B \in \bar{\mathcal{B}}_i$*, where* $\bar{\mathcal{B}}_i := \{B \in \mathcal{B}_i : z^i \in B\}$*. All redundant boxes are removed from* $\mathcal{B}_{i+1}$ *at the end of the respective iteration* i*.*

Note that Assumption 5.13 substantiates Assumption 5.6 by specifying that the correct, non-redundant decompositions are obtained by iterative full 3-splits and that redundant boxes are removed.

As the proof of the following lemma is rather technical, we illustrate it with the help of two examples. In both examples $u(B_0) := (5,5,5)^\top$ is assumed, and $z^1 := (2,2,2)^\top$ is inserted as a first nondominated point. In the first example, depicted in Figure 5.4, $z^2 := (3,1,4)^\top$ is inserted as a second nondominated point. In the second example, depicted in Figure 5.5, $z^2 := (1,1,4)^\top$ represents the second nondominated point.

Lemma 5.14. *Let Assumption 5.13 be satisfied. Then, for every* $s \geq 2$*, every* $\bar{B} \in \mathcal{B}_s$ *and every* $i \in \{1,2,3\}$*, for which* $u_i(\bar{B}) > \underline{u}_i := \min\{u_i(B) : B \in \mathcal{B}_s\}$ *holds, there exists a unique neighbor* $B_i^s(\bar{B}) \in \mathcal{B}_s$ *satisfying* (5.4)–(5.7)*. Particularly,* $u_k(B_i^s(\bar{B})) = u_k(\bar{B})$ *holds, i.e.,* $B_i^s(\bar{B})$ *satisfies*

$$u_i(B_i^s(\bar{B})) < u_i(\bar{B}), \tag{5.8}$$

$$u_j(B_i^s(\bar{B})) > u_j(\bar{B}) \quad \textit{for some } j \neq i, \tag{5.9}$$

$$u_k(B_i^s(\bar{B})) = u_k(\bar{B}) \quad \textit{for } k \neq i,j \tag{5.10}$$

and

$$u_i(B_i^s(\bar{B})) = \max\{u_i(B) : B \in \mathcal{B}_s, u_i(B) < u_i(\bar{B})\}. \tag{5.11}$$

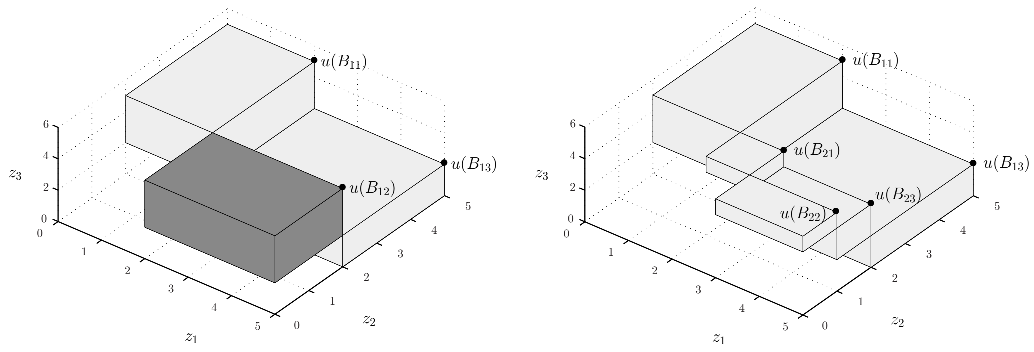
If $u_i(\bar{B}) = \underline{u}_i$*, we set* $B_i^s(\bar{B}) := \emptyset$*.*

Proof. By induction on s.

$\underline{s=2}$: $\mathcal{B}_1 = \{B_0\} = \overline{\mathcal{B}}_1$, as $z^1 < u(B_0) = z^M$. The starting box is split into three subboxes $\hat{B}_i := \{z \in B_0 : z_i < z_i^1\}, i \in \{1,2,3\}$. Due to Lemma 5.7, the new boxes are non-redundant, thus, $\mathcal{B}_2 = \{\hat{B}_1, \hat{B}_2, \hat{B}_3\}$. Consider $\hat{B}_i$ for fixed $i \in \{1,2,3\}$: Since $u_i(\hat{B}_i) = z_i^1 < u_i(\hat{B}_j)$ for every $j \neq i$, by definition, $B_i^2(\hat{B}_i) = \emptyset$ holds. Since $u_j(\hat{B}_i) > z_j^1 = \min\{u_j(B) : B \in \mathcal{B}_2\}$, for every $j \neq i$, a unique neighbor $B_j^2(\hat{B}_i)$ exists and, obviously, $B_j^2(\hat{B}_i) = \hat{B}_j$ holds.

$\underline{s \to s+1}$: We assume that unique neighbors satisfying (5.4)–(5.7) exist for all $B \in \mathcal{B}_s$ and that, additionally, (5.10) holds. We insert $z^s \in Z_N$. Due to the correctness of the full 3-split, there exists at least one box which is split, i.e., $|\overline{\mathcal{B}}_s| \geq 1$.

Case 1: $|\overline{\mathcal{B}}_s| = 1$, i.e., only one box is split. Let $\hat{B}$ be this box, and $\hat{B}_i, i \in \{1,2,3\}$, be the subboxes resulting from the split. Due to Lemma 5.7, the new boxes are



(a) Before the insertion of z^2

(b) After the insertion of z^2

Figure 5.4: Visualization of the upper bound vectors $u(B)$ in case 1 of the proof of Lemma 5.14: $z^2 = (3, 1, 4)^\top$ lies only in box B_{12} with $u(B_{12}) = (5, 2, 5)^\top$, i.e., $|\overline{\mathcal{B}}_2| = 1$. For a better illustration, the individual subsets $V(B)$ of all boxes are depicted.

non-redundant, thus, $\mathcal{B}_{s+1} = (\mathcal{B}_s \setminus \{\hat{B}\}) \cup \{\hat{B}_1, \hat{B}_2, \hat{B}_3\}$. The corresponding box in Figure 5.4 (a) is $\hat{B} = B_{12}$, which is replaced by $\hat{B}_1 = B_{21}$, $\hat{B}_2 = B_{22}$, $\hat{B}_3 = B_{23}$, see Figure 5.4 (b).

Consider an arbitrary box $\hat{B}_i, i \in \{1, 2, 3\}$. Then the following holds:

(i) $B_i^{s+1}(\hat{B}_i) = B_i^s(\hat{B})$:
$\hat{B}_i$ is the only new box B with $u_i(B) = z_i^s$ and there is no other new box B satisfying $u_i(B) < z_i^s$. Hence, $B_i^{s+1}(\hat{B}_i) \notin \{\hat{B}_1, \hat{B}_2, \hat{B}_3\}$. If $B_i^s(\hat{B}) = \emptyset$ then $B_i^{s+1}(\hat{B}_i) = \emptyset$. Otherwise, i.e., if $B_i^s(\hat{B})$ exists, $u_i(B_i^s(\hat{B})) < u_i(\hat{B})$, $u_j(B_i^s(\hat{B})) > u_j(\hat{B})$ for some $j \neq i$ and $u_k(B_i^s(\hat{B})) = u_k(\hat{B})$ for $k \neq i, j$ hold due to the induction hypothesis. Now $u_i(B_i^s(\hat{B})) \leq z_i^s$ must be satisfied, as otherwise $B_i^s(\hat{B}_i) \in \overline{\mathcal{B}}_s$ would hold, which would be a contradiction to the assumption that $\overline{\mathcal{B}}_s = \{\hat{B}\}$. Moreover, $u_i(B_i^s(\hat{B})) = z_i^s$ is excluded due to Assumption 5.13 (1). Therefore, (5.8)–(5.11) holds for $B_i^{s+1}(\hat{B}_i) = B_i^s(\hat{B})$. The uniqueness of $B_i^s(\hat{B})$ follows from the induction hypothesis.

In Figure 5.4 (b), an example of this case is given by $B_1^3(B_{21}) = B_1^2(B_{12}) = B_{11}$.

(ii) $B_j^{s+1}(\hat{B}_i) = \hat{B}_j$ for all $j \neq i$:
$\hat{B}_j$ is the only new box B with $u_j(B) = z_j^s$ and there is no other new box B satisfying $u_j(B) < z_j^s$. Furthermore, box $\hat{B}_j$ satisfies (5.8)–(5.10) since $u_j(\hat{B}_j) < u_j(\hat{B}_i)$, $u_i(\hat{B}_j) > u_i(\hat{B}_i)$ and $u_k(\hat{B}_j) = u_k(\hat{B}_i)$ for $k \neq i, j$ hold. Moreover, $u_j(\hat{B}_j)$ is maximal, as $u_j(\hat{B}_j) = z_j^s \geq u_j(B_j^s(\hat{B}))$ if $B_j^s(\hat{B}) \neq \emptyset$ and $u_j(B_j^s(\hat{B}))$ maximal due to the induction hypothesis. As $z_j^s = u_j(B_j^s(\hat{B}))$ is excluded due to Assumption 5.13 (1), the uniqueness of $B_j^{s+1}(\hat{B}_i)$ follows.

In Figure 5.4 (b), examples are given by $B_2^3(B_{21}) = B_{22}$ and $B_3^3(B_{21}) = B_{23}$.

Now consider an arbitrary box $B \neq \hat{B}$. Then the following holds:

(iii) If $B_i^s(B) \neq \hat{B}$ for some $i \in \{1,2,3\}$, then $B_i^{s+1}(B)$ remains unchanged:
Assume that $B_i^{s+1}(B)$ changes due to the split of box $\hat{B}$. Then the only candidate for $B_i^{s+1}(B)$ is $\hat{B}_i$ and only in case that $u_i(\hat{B}_i) < u_i(B) \leq u_i(\hat{B})$, as otherwise $B_i^s(B) = \hat{B}$ would have been valid. Now suppose that $B_i^{s+1}(B) = \hat{B}_i$. Since $u_j(\hat{B}_i) = u_j(\hat{B})$ for all $j \neq i$ and, by definition of $B_i^s(B)$, $u_j(\hat{B}_i) \geq u_j(B)$ for all $j \neq i$, we have that $u_j(\hat{B}) \geq u_j(B)$ for all $j \neq i$ and hence $u_l(\hat{B}) \geq u_l(B)$ for all $l \in \{1,2,3\}$, a contradiction to $\mathcal{B}_s$ being non-redundant. Thus, $B_i^{s+1}(B)$ remains unchanged.

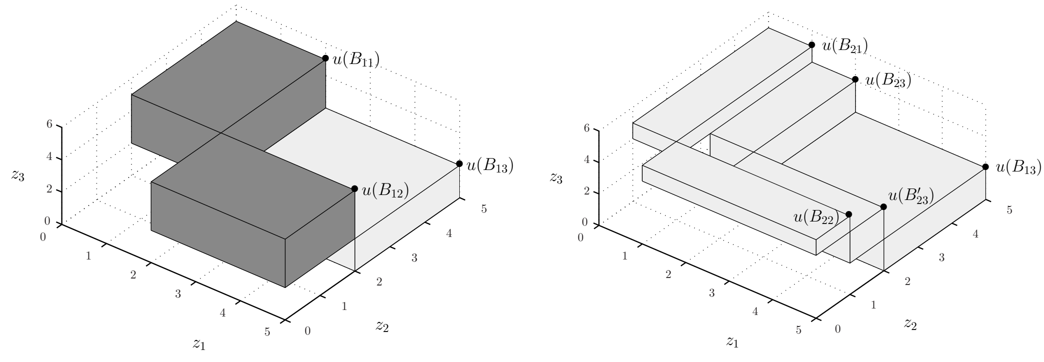
In the example depicted in Figure 5.4 (b), let $B = B_{13}$. As $B_1^2(B_{13}) = B_{11} \neq B_{12}$, the neighbor remains unchanged, thus, $B_1^3(B_{13}) = B_{11}$.

(iv) If $B_i^s(B) = \hat{B}$ for some $i = 1,\dots,m$, then $B_i^{s+1}(B) = \hat{B}_j$ with j being the unique index for which $u_j(\hat{B}) > u_j(B)$ holds:
By the induction hypothesis, $u_i(\hat{B}) < u_i(B)$, $u_j(\hat{B}) > u_j(B)$ for some $j \neq i$ and $u_k(\hat{B}) = u_k(B)$ for $k \neq i,j$. As $z_i^s = u_i(\hat{B}_i) < u_i(\hat{B})$ and $u_l(\hat{B}_i) = u_l(\hat{B})$ for all $l \neq i$, $\hat{B}_i$ is a candidate for $B_i^{s+1}(B)$. As $B \notin \overline{\mathcal{B}}_s$ and $z_l^s < u_l(B)$ for all $l \neq j$ it follows that $z_j^s \geq u_j(B)$, and, due to Assumption 5.13 (1), $z_j^s > u_j(B)$. Thus, $u_i(\hat{B}_j) = u_i(\hat{B}) < u_i(B)$, $u_j(\hat{B}_j) = z_j^s > u_j(B)$ and $u_k(\hat{B}_j) = u_k(\hat{B}) = u_k(B)$ hold. Therefore, $\hat{B}_j$ is the unique other candidate for $B_i^{s+1}(B)$ besides $\hat{B}_i$. As $u_i(\hat{B}_i) < u_i(\hat{B}_j) = u_i(\hat{B})$, $\hat{B}_j$ is the unique neighbor $B_i^{s+1}(B)$ after the split.

In the example depicted in Figure 5.4 (b), $B_2^2(B_{13}) = B_{12} = \hat{B}$ and $B_2^3(B_{13}) = B_{23}$. Box B_{22} is the unique other candidate for $B_2^3(B_{13})$, however, since $u_2(B_{22}) < u_2(B_{23})$ it holds that $B_2^3(B_{13}) = B_{23}$.

Case 2: $|\overline{\mathcal{B}}_s| > 1$. By definition of $\overline{\mathcal{B}}_s$, it holds that $z_i^s < u_i(B)$ for all $i \in \{1,2,3\}$ and $B \in \overline{\mathcal{B}}_s$, thus, $z_i^s < \min\{u_i(B) : B \in \overline{\mathcal{B}}_s\}$ for all $i \in \{1,2,3\}$. According to Lemma 5.7 and Corollary 5.9, redundancy occurs only for boxes $\hat{B}, \tilde{B} \in \overline{\mathcal{B}}_s$ which are split with respect to the same component $i \in \{1,2,3\}$ (i.e., $u_i(\hat{B}_i) = u_i(\tilde{B}_i) = z_i^s$) and for which $u_l(\hat{B}_i) \geq u_l(\tilde{B}_i)$ or $u_l(\hat{B}_i) \leq u_l(\tilde{B}_i)$ holds for all $l \neq i$. By assumption, those boxes are removed, i.e., $\mathcal{B}_{s+1}$ contains only non-redundant boxes. We illustrate this case by the example depicted in Figure 5.5.

Let $\overline{\mathcal{B}}_s = \{\hat{B}^1,\dots,\hat{B}^P\}$ with $P \in \mathbb{N}, P \geq 2$. The corresponding boxes in Figure 5.5 (a) are $\hat{B}^1 = B_{11}$ and $\hat{B}^2 = B_{12}$. For every $i \in \{1,2,3\}$, let $I_i \subseteq \{1,\dots,P\}$ be the index set of the boxes from $\overline{\mathcal{B}}_s$ whose split with respect to i yields a non-redundant box. Note that $I_i \neq \emptyset$ for every $i \in \{1,2,3\}$, which can be seen as follows:



(a) Before the insertion of z^2 (b) After the insertion of z^2

Figure 5.5: Visualization of the upper bound vectors $u(B)$ in case 2 of the proof of Lemma 5.14: $z^2 = (1, 1, 4)^\top$ lies in the two boxes B_{11} with $u(B_{11}) = (2, 5, 5)^\top$ and B_{12} with $u(B_{12}) = (5, 2, 5)^\top$, i.e., $|\overline{\mathcal{B}}_2| = 2$. For a better illustration, the individual subsets $V(B)$ of all boxes are depicted.

Consider an arbitrary box $B \in \overline{\mathcal{B}}_s$. Applying the full 3-split to B results in three new boxes. Now any of the resulting boxes is removed if and only if there exists another box that dominates it. According to Lemma 5.7 the dominating box must have been created by a split with respect to the same component as the dominated box. Therefore, $I_i \neq \emptyset$ for every $i \in \{1, 2, 3\}$. We set $Q_i := |I_i| \geq 1$ for every $i \in \{1, 2, 3\}$. Furthermore, let $\bar{u}_i := \max\{u_i(B), B \in \overline{\mathcal{B}}_s\}$ for all $i \in \{1, 2, 3\}$ in the following, which is well defined as $\overline{\mathcal{B}}_s \neq \emptyset$.

In the example depicted in Figure 5.5 (b), $Q_1 = Q_2 = 1$ and $Q_3 = 2$. Moreover, from Figure 5.5 (a) we see that $\bar{u} = (5, 5, 5)^\top$.

Consider now i arbitrary but fixed. Let $\hat{B}^{I_i(1)}, \dots, \hat{B}^{I_i(Q_i)}$ denote the boxes whose split with respect to component i yields a non-redundant box. As $u_i(\hat{B}_i^{I_i(q)}) = z_i^s$ holds for all $\hat{B}_i^{I_i(q)} \in \mathcal{B}_{s+1}, q = 1, \dots, Q_i$, as $m = 3$ and as we assume non-redundancy, we can order the boxes with respect to their upper bounds increasingly by some component $j \neq i$ and decreasingly by component $k \neq i, j$, i.e.,

$$z_j^s < u_j(\hat{B}_i^{I_i(1)}) < u_j(\hat{B}_i^{I_i(2)}) < \dots < u_j(\hat{B}_i^{I_i(Q_i)}), \tag{5.12}$$

$$u_k(\hat{B}_i^{I_i(1)}) > u_k(\hat{B}_i^{I_i(2)}) > \dots > u_k(\hat{B}_i^{I_i(Q_i)}) > z_k^s. \tag{5.13}$$

Thereby, $u_j(\hat{B}_i^{I_i(Q_i)}) = u_j(\hat{B}^{I_i(Q_i)}) = \bar{u}_j$ holds, since in the other case, i.e., if there was some $\tilde{B} \in \overline{\mathcal{B}}_s$ with $u_j(\tilde{B}) > u_j(\hat{B}_i^{I_i(Q_i)})$, either $\tilde{B}$ would have been the last box in (5.12) with index $I_i(Q_i)$ or $\hat{B}_i^{I_i(Q_i)}$ would have been dominated by $\tilde{B}$, both in contradiction to the construction. Analogously, $u_k(\hat{B}_i^{I_i(1)}) = u_k(\hat{B}^{I_i(1)}) = \bar{u}_k$ must hold.

In the example depicted in Figure 5.5 (a), consider $i = 3$ and, without loss of generality, let $\hat{B}^{I_3(1)} = B_{11}$ and $\hat{B}^{I_3(2)} = \hat{B}^{I_3(Q_3)} = B_{12}$. The upper bounds $u(B_{11}) = (2,5,5)^\top$ and $u(B_{12}) = (5,2,5)^\top$ can be ordered increasingly with respect to component $j = 1$ and decreasingly with respect to component $k = 2$. It holds that $u_1(\hat{B}_3^{I_3(Q_3)}) = 5 = \bar{u}_1$ and $u_2(\hat{B}_3^{I_3(1)}) = 5 = \bar{u}_2$.

If $u_i(\hat{B}^{I_i(1)}) = \max\{u_i(B) : B \in \mathcal{B}_s, u_k(B) = \bar{u}_k\} =: \bar{u}_{i,k}$ holds, then the split of $\hat{B}^{I_i(1)}$ with respect to j generates a non-redundant box, too, and, depending on the chosen enumeration, $I_i(1)$ either equals $I_j(1)$ or $I_j(Q_j)$. W.l.o.g. we can set $I_i(1) = I_j(1)$. Otherwise, i.e., if $u_i(\hat{B}^{I_i(1)}) < \bar{u}_{i,k}$ holds, then $\hat{B}_j^{I_i(1)}$ is dominated by a unique box $\tilde{B} \in \overline{\mathcal{B}}_s$ with $u_k(\tilde{B}) = \bar{u}_k$ and $u_i(\tilde{B}) = \bar{u}_{i,k}$. Then $\tilde{B} = \hat{B}^{I_j(1)}$ holds.

Analogously, if $u_i(\hat{B}^{I_i(Q_i)}) = \max\{u_i(B) : B \in \mathcal{B}_s, u_j(B) = \bar{u}_j\} =: \bar{u}_{i,j}$ holds, then the split of $\hat{B}^{I_i(Q_i)}$ with respect to k generates a non-redundant box, too, and, w.l.o.g., we can identify $I_i(Q_i) = I_k(Q_k)$. Otherwise, i.e., if $u_i(\hat{B}^{I_i(Q_i)}) < \bar{u}_{i,j}$ holds, $\hat{B}_k^{I_i(Q_i)}$ is dominated by a unique box $\tilde{B} \in \overline{\mathcal{B}}_s$ with $u_j(\tilde{B}) = \bar{u}_j$ and $u_i(\tilde{B}) = \bar{u}_{i,j}$. Then $\tilde{B} = \hat{B}^{I_k(Q_k)}$ holds.

Note that if $Q_i = 1$, then $\hat{B}^{I_i(1)} = \hat{B}^{I_i(Q_i)} =: \hat{B}$ and $u_j(\hat{B}) = \bar{u}_j$ as well as $u_k(\hat{B}) = \bar{u}_k$ hold. In this case, $u_i(\hat{B}) < \bar{u}_i$ must be satisfied, as otherwise $\hat{B}$ would dominate any other box in $\overline{\mathcal{B}}_s$, a contradiction to $|\overline{\mathcal{B}}_s| > 1$ and $\mathcal{B}_s$ being non-redundant.

In the example depicted in Figure 5.5, consider $i = 3$ and $k = 2$. It holds that $u_3(\hat{B}^{I_3(1)}) = 5 = \max\{u_3(B) : B \in \mathcal{B}_s, u_2(B) = 5\}$. The split of $\hat{B}^{I_3(1)}$ with respect to $j = 1$ generates the non-redundant box B_{21}. If we consider $i = 1$, then $u_1(\hat{B}^{I_1(1)}) = u_1(B_{11}) = 2 < 5 = \max\{u_1(B) : B \in \mathcal{B}_s, u_3(B) = 5\}$, hence, the split of B_{11} with respect to component $j = 2$ must be redundant, and, indeed, the resulting box is dominated by B_{22}.

Analogously to Case 1, we will now indicate the neighbor boxes explicitly. Therefore, consider $\hat{B}_i^{I_i(q)} \in \mathcal{B}_{s+1}$ for fixed $i \in \{1,2,3\}, q \in \{1,\dots,Q_i\}$. It holds that

(i) $B_i^{s+1}(\hat{B}_i^{I_i(q)}) = B_i^s(\hat{B}^{I_i(q)})$:
Assume that $B_i^s(\hat{B}^{I_i(q)}) \in \overline{\mathcal{B}}_s$. By definition of B_i^s, $u_i(B_i^s(\hat{B}^{I_i(q)})) < u_i(\hat{B}^{I_i(q)})$ and $u_l(B_i^s(\hat{B}^{I_i(q)})) \geq u_l(\hat{B}^{I_i(q)})$ for all $l \neq i$ hold. But then, by an i-split of $\hat{B}^{I_i(q)}$ and $B_i^s(\hat{B}^{I_i(q)})$, the box $\hat{B}_i^{I_i(q)}$ would be redundant. So, $B_i^s(\hat{B}^{I_i(q)}) \notin \overline{\mathcal{B}}_s$ must hold. Analogously to Case 1(i), we obtain $B_i^{s+1}(\hat{B}_i^{I_i(q)}) = B_i^s(\hat{B}^{I_i(q)})$.

In the example depicted in Figure 5.5, it holds that $B_3^3(B_{23}) = B_3^2(B_{11}) = B_{13}$ and $B_3^3(B'_{23}) = B_3^2(B_{12}) = B_{13}$.

(ii) Determination of $B_j^{s+1}(\hat{B}_i^{I_i(q)})$ and $B_k^{s+1}(\hat{B}_i^{I_i(q)})$ for $j, k \neq i$:
Consider all $\hat{B}_i^{I_i(q)} \in \mathcal{B}_{s+1}, q = 1,\dots,Q_i$, ordered as in (5.12) and (5.13): It

holds that

$$B_j^{s+1}(\hat{B}_i^{I_i(q)}) = \hat{B}_i^{I_i(q-1)} \quad \text{for all } q = 2, \dots, Q_i,$$

since for all other boxes $\hat{B}_i^{I_i(p)}$, $p \neq q-1$, it holds that either $u_j(\hat{B}_i^{I_i(p)}) < u_j(\hat{B}_i^{I_i(q-1)})$ or $u_j(\hat{B}_i^{I_i(p)}) > u_j(\hat{B}_i^{I_i(q)})$. Moreover, all new boxes split with respect to j have component u_j smaller than $u_j(\hat{B}_i^{I_i(q-1)})$ and all new boxes split with respect to k have component u_k smaller than $u_k(\hat{B}_i^{I_i(q)})$, and, thus, do not satisfy (5.10). For all boxes $B \notin \overline{\mathcal{B}}_s$, it holds that $u_l(B) < \min\{u_l(B) : B \in \overline{\mathcal{B}}_s\}$ for some l, so either $u_j(B) < u_j(\hat{B}_i^{I_i(q-1)})$ or (5.10) is not satisfied.

Next, we determine $B_j^{s+1}(\hat{B}_i^{I_i(1)})$: Since $u_j(\hat{B}_i^{I_i(q)}) > u_j(\hat{B}_i^{I_i(1)})$ holds for all $q = 2, \dots, Q_i$, no box that is split with respect to i can be the neighbor $B_j^{s+1}(\hat{B}_i^{I_i(1)})$. Furthermore, as $u_k(\hat{B}_i^{I_i(1)}) > z_k^s$, $B_j^{s+1}(\hat{B}_i^{I_i(1)})$ cannot be found among the new boxes split with respect to k. Therefore, $B_j^{s+1}(\hat{B}_i^{I_i(1)})$ can only be found among the boxes split with respect to component j. Now, as shown above, $u_k(\hat{B}^{I_i(1)}) = \bar{u}_k$ holds, which implies that $u_k(B_j^{s+1}(\hat{B}_i^{I_i(1)})) = \bar{u}_k$ must be satisfied. Therefore, the unique candidate for $B_j^{s+1}(\hat{B}_i^{I_i(1)})$ is $\hat{B}_j^{I_j(1)}$, which, as explained above, either equals the box obtained from $\hat{B}^{I_i(1)}$ by a split with respect to j or the unique box dominating it.

Analogously, it can be shown that

$$B_k^{s+1}(\hat{B}_i^{I_i(q)}) = \hat{B}_i^{I_i(q+1)} \quad \text{for all } q = 1, \dots, Q_i - 1,$$

and

$$B_k^{s+1}(\hat{B}_i^{I_i(Q_i)}) = \hat{B}_k^{I_k(Q_k)},$$

where $\hat{B}_k^{I_k(Q_k)}$ either equals the box obtained from $\hat{B}^{I_i(Q_i)}$ by a split with respect to k (then $I_i(Q_i) = I_k(Q_k)$) or the unique box dominating it.

In the example depicted in Figure 5.5, $B_1^3(B'_{23}) = B_{23}$ and $B_1^3(B_{23}) = B_{21}$ hold.

Finally, for all $B \notin \overline{\mathcal{B}}_s$ we obtain the following results which are equivalent to Case 1:

(iii) If $B_i^s(B) \notin \overline{\mathcal{B}}_s$ for some $i \in \{1, 2, 3\}$, then $B_i^{s+1}(B)$ remains unchanged. As in the example depicted in Figure 5.5 box B_{13} is the unique box which is not split and all of its neighbors are split, this case does not occur.

(iv) If $B_i^s(B) =: \hat{B} \in \overline{\mathcal{B}}_s$ for some $i \in \{1, 2, 3\}$, then, following the same argumentation as in Case 1(iv), $z_j^s > u_j(B)$ for one unique index $j \neq i$ and, thus, the correct candidate for $B_i^{s+1}(B)$ would be $\hat{B}_j$. It remains to show that $\hat{B}_j$ exists and that $u_i(\hat{B}_j) = \max\{u_i(\tilde{B}) : \tilde{B} \in \mathcal{B}_{s+1}, u_i(\tilde{B}) < u_i(B)\}$.

Assume that $\hat{B}_j$ does not exist, i.e., it is redundant in $\mathcal{B}_{s+1}$. Then there exists $\bar{B} \in \overline{\mathcal{B}}_s$ with $u_i(\bar{B}) \geq u_i(\hat{B})$ and $u_k(\bar{B}) \geq u_k(\hat{B})$. As $\bar{B}, \hat{B} \in \mathcal{B}_s$ and $\mathcal{B}_s$, by induction, is non-redundant, $u_j(\bar{B}) < u_j(\hat{B})$ must hold. As $\bar{B} \in \overline{\mathcal{B}}_s$ it follows that $z_j^s < u_j(\bar{B})$, so $u_j(B) < z_j^s < u_j(\bar{B})$ and $u_k(B) = u_k(\hat{B}) \leq u_k(\bar{B})$ hold. If $u_i(\bar{B}) \geq u_i(B)$, B would have been redundant in $\mathcal{B}_s$. Thus, $u_i(\bar{B}) < u_i(B)$ must hold. However, as $B_i^s(B) = \hat{B}$, the induction hypothesis then implies that $u_i(\bar{B}) < u_i(\hat{B})$, a contradiction to the assumption on $\bar{B}$. Thus, $\hat{B}_j$ is non-redundant and $u_i(\hat{B}_j) = u_i(\hat{B}) = \max\{u_i(\tilde{B}) : \tilde{B} \in \mathcal{B}_{s+1}, u_i(\tilde{B}) < u_i(B)\}$ holds.

In the example depicted in Figure 5.5, consider $B_1^2(B_{13}) = B_{11}$. Since $j = 3$ is the unique index $\neq 1$ such that $z_j^2 > u_j(B_{13})$, it holds that $B_1^3(B_{13}) = B_{23}$, i.e., the new neighbor is the box which results from B_{11} by a split with respect to j.

□

In the next corollary we summarize the properties of the neighbors of all new boxes obtained in the constructive proof of Lemma 5.14.

Corollary 5.15. *Let Assumption 5.13 be satisfied. Besides, for every $i \in \{1, 2, 3\}$, let $I_i \subseteq \{1, \ldots, P\}$, $I_i \neq \emptyset$, $P \in \mathbb{N}$, $|I_i| = Q_i$, be the index set of the boxes of $\overline{\mathcal{B}}_s$ whose split with respect to $i \in \{1, 2, 3\}$ yields a non-redundant box. Then for all new boxes $\hat{B}_i^{I_i(q)}$, $q = 1, \ldots, Q_i$, it holds that*

$$B_i^{s+1}(\hat{B}_i^{I_i(q)}) = B_i^s(\hat{B}^{I_i(q)}) \quad \forall\, q = 1, \ldots, Q_i, \tag{5.14}$$

$$B_j^{s+1}(\hat{B}_i^{I_i(q)}) = \begin{cases} \hat{B}_j^{I_j(1)} & q = 1, \\ \hat{B}_i^{I_i(q-1)} & \forall\, q = 2, \ldots, Q_i, \end{cases} \tag{5.15}$$

$$B_k^{s+1}(\hat{B}_i^{I_i(q)}) = \begin{cases} \hat{B}_i^{I_i(q+1)} & \forall\, q = 1, \ldots, Q_i - 1, \\ \hat{B}_k^{I_k(Q_k)} & q = Q_i, \end{cases} \tag{5.16}$$

where the indices j and k are chosen as in (5.12) *and* (5.13), *which means that the boxes $\hat{B}_i^{I_i(q)}, q = 1, \ldots, Q_i$, are ordered with respect to their upper bounds increasingly by component $j \neq i$ and decreasingly by component $k \neq i, j$. Moreover, $I_j(1)$ and $I_k(Q_k)$ are chosen such that $\hat{B}_j^{I_j(1)}$ and $\hat{B}_k^{I_k(Q_k)}$ either equal $\hat{B}_j^{I_i(1)}$ and $\hat{B}_k^{I_i(Q_i)}$, respectively, or the unique box dominating it.*

Using Lemma 5.14 we can derive an explicit formulation of the individual subsets $V(B)$ for $m = 3$:

Lemma 5.16. *Let Assumption 5.13 hold. Then, for $m = 3$, the individual subsets $V(B), B \in \mathcal{B}_s$, which are introduced in Definition 5.10, can be represented as*

$$V(B) = \{z \in B_0 : v(B) \leqq z < u(B)\}$$

with

$$v_i(B) := \begin{cases} u_i(B_i^s(B)), & \text{if } B_i^s(B) \neq \emptyset \\ z_i^I, & \text{otherwise} \end{cases}, \quad i \in \{1, 2, 3\}. \tag{5.17}$$

Proof. For $\bar{B} \in \mathcal{B}_s, s \geq 1$, by definition,

$$V(\bar{B}) := \bar{B} \setminus \left(\bigcup_{\tilde{B} \in \mathcal{B}_s \setminus \{\bar{B}\}} \tilde{B} \right)$$

holds. We consider the sets $\mathcal{B}_{s,i} := \{B \in \mathcal{B}_s : u_i(B) < u_i(\bar{B})\}$ for $i = 1, 2, 3$. For fixed $i \in \{1, 2, 3\}$, the following two cases can occur: If $\mathcal{B}_{s,i} \neq \emptyset$, then, as shown in Lemma 5.14, $B_i^s(\bar{B}) \neq \emptyset$ and $B_i^s(\bar{B}) \in \mathcal{B}_{s,i}$, where $u_i(B_i^s(\bar{B})) = \max\{u_i(B) : B \in \mathcal{B}_{s,i}\}$. Furthermore, as $u_l(B_i^s(\bar{B})) \geq u_l(\bar{B})$ for all $l \neq i$,

$$\bar{B} \setminus \left(\bigcup_{\tilde{B} \in \mathcal{B}_{s,i}} \tilde{B} \right) = \{z \in \bar{B} : z_i \geq u_i(B_i^s(\bar{B}))\}.$$

Otherwise, i.e., if $\mathcal{B}_{s,i} = \emptyset$, then, obviously, $\bar{B} \setminus \left(\bigcup_{\tilde{B} \in \mathcal{B}_{s,i}} \tilde{B} \right) = \bar{B}$. So, in both cases, it holds that

$$\bar{B} \setminus \left(\bigcup_{\tilde{B} \in \mathcal{B}_{s,i}} \tilde{B} \right) = \{z \in \bar{B} : z_i \geq v_i(\bar{B})\}$$

with

$$v_i(\bar{B}) := \begin{cases} u_i(B_i^s(\bar{B})), & \text{if } B_i^s(\bar{B}) \neq \emptyset \\ z_i^I, & \text{otherwise.} \end{cases}$$

As every box $B \in \mathcal{B}_s \setminus \{\bar{B}\}$ belongs, due to the assumption of non-redundancy, to at least one set $\mathcal{B}_{s,i}, i \in \{1, 2, 3\}$, there does not exist any other box which can reduce $V(\bar{B})$ further. Thus, we obtain the desired representation. □

Lemma 5.16 shows that for $m = 3$ the individual subset of a box can be represented as a box itself. As the upper bound of $V(B)$ and B are the same, $V(B)$ can be described by its lower bound $v(B) \in \mathbb{R}^m$ only. Next we show, using Corollary 5.15, how the lower bounds $v(B)$ can be updated in an iterative algorithm.

Lemma 5.17. *Let Assumption 5.13 be satisfied. We use the notation of Corollary 5.15. Let $\hat{B}_i^{I_i(q)}, q = 1, \ldots, Q_i$, be the non-redundant boxes that are obtained* *from* $\hat{B}^{I_i(q)} \in \overline{\mathcal{B}}_s$ *by a split with respect to* $i \in \{1,2,3\}$*. Then the lower bound vectors* $v(B) \in \mathbb{R}^m$ *of these new boxes in* $\mathcal{B}_{s+1}$ *are determined by*

$$\begin{aligned}
v_i(\hat{B}_i^{I_i(q)}) &= v_i(\hat{B}^{I_i(q)}) && \forall\, q = 1,\dots,Q_i,\\
v_j(\hat{B}_i^{I_i(q)}) &= \begin{cases} u_j(\hat{B}_j^{I_j(1)}) = z_j^s & q = 1,\\ u_j(\hat{B}_i^{I_i(q-1)}) & \forall\, q = 2,\dots,Q_i, \end{cases}\\
v_k(\hat{B}_i^{I_i(q)}) &= \begin{cases} u_k(\hat{B}_i^{I_i(q+1)}) & \forall\, q = 1,\dots,Q_i - 1,\\ u_k(\hat{B}_k^{I_k(Q_k)}) = z_k^s & q = Q_i. \end{cases}
\end{aligned}$$

All individual subsets $V(B)$ *of all* $B \notin \overline{\mathcal{B}}_s$ *remain unchanged.*

Proof. The update of $v(\hat{B}_i^{I_i(q)})$ of all new boxes $\hat{B}_i^{I_i(q)}, q = 1,\dots,Q_i$, for some fixed $i \in \{1,2,3\}$ is derived directly from Corollary 5.15. The individual subsets of all boxes which are not split in the current iteration do not change, as, according to the proof of Lemma 5.14, either $B_i^{s+1}(B)$ remains unchanged (Case (iii)) or $B_i^{s+1}(B) = \hat{B}_j$ (Case (iv)), i.e., $u_i(B)$ remains unchanged. □

Recall that we want to split a box $B \in \overline{\mathcal{B}}_s$ with respect to a component $i \in \{1,2,3\}$ if and only if the individual subset $V(B_i)$ of the resulting box B_i is non-empty, which is equivalent to B_i being non-redundant. With the vector $v(B) \in \mathbb{R}^m$ at hand, this can be easily checked as the following lemma shows.

Lemma 5.18. *Let Assumption 5.13 hold up to iteration* $s-1$ *for* $s \geq 2$*, i.e., let* $\mathcal{B}_s$ *be a correct, non-redundant decomposition of the search region obtained by iterative* 3*-splits. Let* $z^s \in Z_N$ *satisfy Assumption 5.13 (1), and let* B_i *be the box obtained from* $B \in \overline{\mathcal{B}}_s$ *by a split with respect to component* $i \in \{1,2,3\}$*. Then* B_i *is non-redundant if and only if* $z_i^s > v_i(B)$ *holds.*

Proof. Consider a fixed $i \in \{1,2,3\}$.

"$\Rightarrow$": Let B_i be non-redundant and assume that $z_i^s < v_i(B)$ holds. Note that the case $z_i^s = v_i(B)$ does not occur due to Assumption 5.13 (1). Then $v_i(B) > z_i^I$, and, thus, $v_i(B) = u_i(B_i^s(B))$ with $B_i^s(B) \neq \emptyset$. As $u_l(B_i^s(B)) \geq u_l(B)$ for all $l \neq i$, $z^s \in B_i^s(B)$ must hold. But then, B_i would be redundant as it would be dominated by the box obtained from $B_i^s(B)$ by a split with respect to i, a contradiction to the assumption of non-redundancy.

"$\Leftarrow$": Let $z_i^s > v_i(B)$. A split of B with respect to i yields $B_i = \{z \in B : z_i < z_i^s\}$. Assume that there exists $\tilde{B}_i \neq B_i$ which dominates B_i. As $\mathcal{B}_s$ is non-redundant and due to Lemma 5.7, $\tilde{B}_i$ must result from a split with respect to i from some box $\tilde{B} \in \overline{\mathcal{B}}_s$, i.e., $z^s \in \tilde{B}$ must hold. As B and $\tilde{B}$ are split with respect to i,

$u_i(B_i) = u_i(\tilde{B}_i) = z_i^s$ holds, and, due to the assumption that $\tilde{B}_i$ dominates B_i, $u_l(\tilde{B}) \geq u_l(B)$ for all $l \neq i$. Now $B, \tilde{B} \in \mathcal{B}_s$ and $\mathcal{B}_s$ being non-redundant imply that $u_i(\tilde{B}) < u_i(B)$. This in turn means that $v_i(B) \geq u_i(\tilde{B})$. But then $z_i^s > u_i(\tilde{B})$, a contradiction to $z^s \in \tilde{B}$. It follows that B_i is non-redundant.

□

Lemma 5.18 provides a tool for defining a split operation for tricriteria problems which generates all boxes that are necessary for maintaining the correctness of a decomposition, but avoids the generation of redundant boxes. We call the split based on the individual subsets $V(B)$ a v-split in the following.

Definition 5.19 (v-split)**.** *Let Assumption 5.13 hold up to iteration $s-1$ for $s \geq 2$, i.e., let $\mathcal{B}_s$ be a correct, non-redundant decomposition of the search region obtained by iterative 3-splits, and let $z^s \in Z_N$. We call the split of a box $B \in \overline{\mathcal{B}}_s$ with respect to components $i \in \{1, 2, 3\}$ for which*

$$z_i^s \geq v_i(B) \tag{5.18}$$

holds a v-split of B.

Note that equality in (5.18) does not occur due to Assumption 5.13 (1). However, as Assumption 5.13 (1) will be removed in Section 5.3.4, we present the v-split already at this point in this general form.

Lemma 5.20. *Let Assumption 5.13 (1),(2) hold. Then the iterative application of a v-split to every $B \in \overline{\mathcal{B}}_s$ in every iteration $s \geq 1$ yields a correct, non-redundant decomposition.*

Proof. Due to Assumption 5.13 (1), $z_i^s \geq v_i(B)$ is equivalent to $z_i^s > v_i(B)$. According to Lemma 5.18, the v-split avoids exactly the generation of redundant boxes and, therefore, yields a correct, non-redundant decomposition. □

5.3.2 The v-Split Algorithm

Algorithm 2 shows the implementation of the v-split. As in Algorithm 1, an initial box B_0 is computed, which is represented by its upper bound vector $u(B_0)$. Additionally, for B_0 as well as for all other boxes B which are generated in the course of the algorithm, the lower bound vector of the individual subset $v(B)$ is saved. Analogously to Algorithm 1, as long as the decomposition contains unexplored boxes, a box is selected and a subproblem is solved. If the box is empty, which corresponds

Algorithm 2 Algorithm with v-split for $m = 3$

Input: Image of the feasible set $Z \subset \mathbb{R}^m$, implicitly given by some problem formulation

$N := \emptyset; \delta > 0;$
InitStartingBoxVsplit(Z, δ);
$s := 1$;
while $\mathcal{B}_s \neq \emptyset$ **do**
  Choose $\bar{B} \in \mathcal{B}_s$;
  $z^s :=$ *opt*$(Z, u(\bar{B}))$; // Solve scalarized subproblem
  if $z^s = \emptyset$ **then** // No nondominated point found
    $\mathcal{B}_{s+1} := \mathcal{B}_s \backslash \{\bar{B}\}$;
  else
    $N := N \cup \{z^s\}$; // Add point to nondominated set
    $\mathcal{B}_{s+1} := \mathcal{B}_s$; // Copy set of current boxes
    GenerateNewBoxesVsplit$(\mathcal{B}_s, z^s, z^I, \mathcal{B}_{s+1})$;
    UpdateIndividualSubsets$(\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{B}_{s+1})$;
  end if
  $s := s + 1$;
end while
return Set of nondominated points N

to $z^s = \emptyset$, the selected box is deleted from the list of unexplored boxes. Otherwise, the nondominated point z^s is saved and all boxes are determined that contain z^s.

Now, different from Algorithm 1, z^s is compared componentwise to $v(B)$ for every $B \in \overline{\mathcal{B}}_s$. A split with respect to component i is performed if and only if $z_i^s \geq v_i(B)$ and $z_i^s > z_i^I$ hold. If $v_i(B) > z_i^s$ for all $i \in \{1, 2, 3\}$, then B is deleted. Finally, the vectors v of all new boxes are updated according to Lemma 5.17 and a new iteration

procedure InitStartingBoxVsplit(Z, δ)
  for $j = 1$ to 3 **do**
    $z_j^I := \min\{z_j \,:\, z \in Z\}$;
    $z_j^M := \max\{z_j \,:\, z \in Z\} + \delta$;
    $v_j(B_0) := z_j^I; u_j(B_0) := z_j^M$;
  end for
  $\mathcal{B}_1 := \{B_0\}$;
  return $\mathcal{B}_1$
end procedure

procedure GenerateNewBoxesVsplit($\mathcal{B}_s, z^s, z^I, \mathcal{B}_{s+1}$)
    $\mathcal{S}_i := \emptyset \; \forall i = 1, 2, 3$; // Initialize set for each component $i \in \{1, 2, 3\}$
    for all $B \in \mathcal{B}_s$ **do**
        if $z^s < u(B)$ **then** // Point is contained in box
            for $i = 1$ to 3 **do** // Apply v-split
                if $z_i^s \geq v_i(B)$ and $z_i^s > z_i^I$ **then**
                    $B' := \emptyset$; // Create new box
                    $u(B') := u(B); v(B') := v(B)$; // Copy bounds
                    $u_i(B') := z_i^s$; // Update upper bound
                    $\mathcal{S}_i := \mathcal{S}_i \cup \{B'\}$; // Save new box in respective set $\mathcal{S}_i$
                end if
            end for
            $\mathcal{B}_{s+1} := \mathcal{B}_{s+1} \backslash \{B\}$; // Remove B
        end if
    end for
    return $\mathcal{B}_{s+1}, \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$;
end procedure

procedure UpdateIndividualSubsets($\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{B}_{s+1}$)
    for $i = 1$ to 3 **do**
        $Q := |\mathcal{S}_i|$;
        Sort all boxes $B_i^{I_i(q)}, q = 1, \dots, Q_i$, in $\mathcal{S}_i$ such that for $j, k \neq i$
        $u_j(B_i^{I_i(1)}) \leq u_j(B_i^{I_i(2)}) \leq \dots \leq u_j(B_i^{I_i(Q_i)})$ and
        $u_k(B_i^{I_i(1)}) \geq u_k(B_i^{I_i(2)}) \geq \dots \geq u_k(B_i^{I_i(Q_i)})$;
        if $u(B_i^{I_i(q)}) = u(B_i^{I_i(q+1)})$ for some $q = 1, \dots, Q_i - 1$ **then**
            (re)sort $B_i^{I_i(q)}$ and $B_i^{I_i(q+1)}$ such that
            $v_j(B_i^{I_i(q)}) \leq v_j(B_i^{I_i(q+1)})$ and $v_k(B_i^{I_i(q)}) \geq v_k(B_i^{I_i(q+1)})$;
        end if
        Set $v_j(B_i^{I_i(1)}) := z_j^s; v_k(B_i^{I_i(Q_i)}) := z_k^s$; // Update v
        for $q = 2$ to Q_i **do**
            $v_j(B_i^{I_i(q)}) := u_j(B_i^{I_i(q-1)}); v_k(B_i^{I_i(q-1)}) := u_k(B_i^{I_i(q)})$;
        end for
        $\mathcal{B}_{s+1} := \mathcal{B}_{s+1} \cup \mathcal{S}_i$; // Append new boxes
    end for
    return $\mathcal{B}_{s+1}$
end procedure

starts. According to the proof of Lemma 5.14 we can uniquely order all newly generated, non-redundant boxes resulting from a split with respect to component i such that their upper bound values u are increasing in one component $j \neq i$ and decreasing in the remaining component $k \neq i, j$. Hence, in Line 47 in the procedure UpdateIndividualSubsets, strict inequalities hold between each pair of upper bounds. However, in order to make the algorithm also applicable when Assumption 5.13 (1) is removed, see Section 5.3.4 below, we formulate Algorithm 2 already in a general form. Therefore, in Line 47, the strict inequalities are replaced by inequalities and in Lines 48 to 50 the case that the upper bound vectors of two boxes are equal is handled. Note that this case does not occur under Assumption 5.13 (1).

Example 5.21 (Application of Algorithm 2)**.** *Consider again the tricriteria problem of Example 5.4 with initial search region*

$$B_0 := \{z \in Z \,:\, 0 \leq z_i \leq 5 \;\forall\, i = 1,2,3\}$$

and $V(B_0) = B_0$*, thus,* $v(B_0) = (0,0,0)^\top$*. Consider* $z^1 = (2,2,2)^\top$*. The* v*-split applied to the initial box equals a full* 3*-split and, thus, results in*

$$B_{1,i} := \{z \in B_0 \,:\, z_i < 2\},\; i = 1,2,3.$$

The corresponding individual subsets are

$$V(B_{1,i}) := \{z \in B_{1,i} \,:\, z_j \geq 2 \;\forall\, j \neq i\},\; i = 1,2,3,$$

thus, $v(B_{11}) = (0,2,2)^\top$*,* $v(B_{12}) = (2,0,2)^\top$ *and* $v(B_{13}) = (2,2,0)^\top$*. Consider* $z^2 = (1,1,4)^\top$ *as second point. It holds that* $z^2 \in B_{11}$ *as well as* $z^2 \in B_{12}$*, but* $z^2 \notin B_{13}$*. Consider first the* v*-split in* B_{11}*: As* $z_1^2 \geq v_1(B_{11})$*,* $z_2^2 \ngeq v_2(B_{11})$ *and* $z_3^2 \geq v_3(B_{11})$*,* B_{11} *is split with respect to the first and third component into*

$$B_{21} := \{z \in B_{11} \,:\, z_1 < 1\} \quad \text{and} \quad B_{23} := \{z \in B_{11} \,:\, z_3 < 4\}.$$

We save $\mathcal{S}_1 = \{B_{21}\}$*,* $\mathcal{S}_2 = \emptyset$ *and* $\mathcal{S}_3 = \{B_{23}\}$*. Applying the* v*-split to* B_{12} *results in a split with respect to the second and third component into*

$$B_{22} := \{z \in B_{12} \,:\, z_2 < 1\} \quad \text{and} \quad B'_{23} := \{z \in B_{12} \,:\, z_3 < 4\}$$

and $\mathcal{S}_1 = \{B_{21}\}$*,* $\mathcal{S}_2 = \{B_{22}\}$ *and* $\mathcal{S}_3 = \{B_{23}, B'_{23}\}$*. Note that the redundant boxes which were obtained with the full* 3*-split in Example 5.4 are not generated by the* v*-split.*

Finally, the individual subsets of the new boxes of each set $\mathcal{S}_i, i \in \{1,2,3\}$*, are updated: Box* B_{21} *is the only box generated for* $i = 1$*, box* B_{22} *the only one for* $i = 2$*.*

Therefore, $v(B_{21}) = (v_1(B_{11}), z_2^2, z_3^2)^\top = (0,1,4)^\top$ *and* $v(B_{22}) = (z_1^2, v_2(B_{12}), z_3^2)^\top = (1,0,4)^\top$. *Boxes* B_{23} *and* B'_{23} *are both generated by a split with respect to* $i = 3$. *We can order the upper bounds of the boxes* $u(B_{23}) = (2,5,4)^\top$ *and* $u(B'_{23}) = (5,2,4)^\top$ *increasingly with respect to component* $j = 1$ *and, at the same time, decreasingly with respect to* $k = 2$, *thus,* $B_3^{I(1)} := B_{23}$ *and* $B_3^{I(2)} := B'_{23}$. *Therefore,*

$$\begin{aligned} v_1(B_{23}) &= z_1^2 = 1, & v_2(B'_{23}) &= z_2^2 = 1, \\ v_2(B_{23}) &= u_2(B'_{23}) = 2, & v_1(B'_{23}) &= u_1(B_{23}) = 2. \end{aligned}$$

The third component is not changed, so $v(B_{23}) = (1,2,2)^\top$ *and* $v(B'_{23}) = (2,1,2)^\top$.

According to Lemma 5.20, Algorithm 2 maintains a correct, non-redundant decomposition in each iteration. In this sense, the non-redundant representation of the search region based on the upper bounds $u(B)$ is equivalent to the construction presented in Przybylski et al. (2010a). While the algorithm of Przybylski et al. (2010a) filters out 'dominated' upper bound vectors by performing pair-wise comparisons with all other upper bound vectors, no pair-wise comparisons are needed in Algorithm 2 since all information is captured in the vectors $v(B)$, respectively.

5.3.3 A Linear Bound on the Number of Subproblems

In the following, we will bound the number of boxes generated in the course of the algorithm with the help of the v-split. If a box $B \in \mathcal{B}_s$ contains the current point z^s, i.e., if $B \in \overline{\mathcal{B}}_s$, we can make the following assertion concerning the neighbors of B.

Lemma 5.22. *Let Assumption 5.13 hold. Let any* $B \in \overline{\mathcal{B}}_s$ *be given. We denote by* $J_B \subseteq \{1,2,3\}$ *the index set of all components with respect to which* B *is split, and by* $\bar{J}_B := \{1,2,3\} \backslash J_B$ *the complement of* J_B. *Then the following holds:*

1. *If* $\bar{J}_B \neq \emptyset$, *then for every* $j \in \bar{J}_B$, *the neighbor* $B_j^s(B)$ *exists and contains* z^s, *i.e.,* $B_j^s(B) \neq \emptyset$ *and* $B_j^s(B) \in \overline{\mathcal{B}}_s$ *holds for every* $j \in \bar{J}_B$.

2. *If* $\bar{J}_B = \emptyset$, *then* $\overline{\mathcal{B}}_s = \{B\}$ *holds.*

Proof. Let $B \in \overline{\mathcal{B}}_s$. By definition of the v-split, it holds that $z^s < u(B), z_j^s \geq v_j(B)$ for every $j \in J_B$ and $z_j^s < v_j(B)$ for every $j \in \bar{J}_B$. Thus, $v_j(B) > z_j^I$ holds for every $j \in \bar{J}_B$. This, however, implies that $B_j^s(B) \neq \emptyset$ for every $j \in \bar{J}_B$, see the update of v in (5.17).

First, let $\bar{J}_B \neq \emptyset$. Then, for fixed $j \in \bar{J}_B$ and according to Definition 5.11, it holds that $u_j(B_j^s(B)) \leq u_j(B)$ and $u_l(B_j^s(B)) \geq u_l(B)$ for all $l \neq j$. As $u_j(B_j^s(B)) = v_j(B) > z_j^s$, $B_j^s(B) \in \overline{\mathcal{B}}_s$ holds.

Now, consider the case $\bar{J}_B = \emptyset$. Due to Lemma 5.14, every box $\tilde{B} \in \mathcal{B}_s \backslash \{B\}$ has an upper bound vector for which $u_l(\tilde{B}) \leq v_l(B)$ for at least one $l \in \{1, 2, 3\}$. This implies that $z^s \notin \tilde{B}$ for any $\tilde{B} \in \mathcal{B}_s \backslash \{B\}$, thus, $\overline{\mathcal{B}}_s = \{B\}$. □

With the help of Lemma 5.22, we can bound the number of new boxes that are generated in each iteration of Algorithm 2.

Lemma 5.23. *Let Assumption 5.13 hold. Then, in every iteration $s \geq 1$ in which a new nondominated point z^s is found, the number of boxes in the decomposition increases by at most two.*

Proof. If there exists a box $B \in \overline{\mathcal{B}}_s$ which is split with respect to all three components, then, using Lemma 5.22, $\overline{\mathcal{B}}_s = \{B\}$ holds, thus, $|\overline{\mathcal{B}}_s| = 1$. In this case, the box B is removed and replaced by three new boxes in the decomposition, and, thus, the number of boxes in the decomposition increases by two.

It follows that if $|\overline{\mathcal{B}}_s| > 1$, then every $B \in \overline{\mathcal{B}}_s$ is split with respect to at most two components. Let $|\overline{\mathcal{B}}_s| > 1$ and let $B \in \overline{\mathcal{B}}_s$ be split with respect to two components $i, j \in \{1, 2, 3\}$, $j \neq i$. Then, for all other boxes $\tilde{B} \in \mathcal{B}_s \backslash \{B\}$ it holds that $u_l(\tilde{B}) \leq v_l(B)$ for some $l \in \{1, 2, 3\}$. If $l = i$, then $u_i(\tilde{B}) \leq v_i(B) \leq z_i^s$, thus the box is not split with respect to i. Analogously, if $l = j$, then $u_j(\tilde{B}) \leq v_j(B) \leq z_j^s$, thus, the box is not split with respect to j. If $l = k$ (with $k \neq i, j$), then, for any $\tilde{B}$ satisfying $u_k(\tilde{B}) \leq v_k(B)$ it holds that $v_i(\tilde{B}) \geq u_i(B)$ or $v_j(\tilde{B}) \geq u_j(B)$, thus, $\tilde{B}$ cannot be split with respect to both components i and j.

Therefore, if two boxes are split with respect to two components, then these components must differ in one component. This implies that in one iteration, at most three boxes are split with respect to two components. Any other boxes in $\overline{\mathcal{B}}_s$ are split with respect to at most one component.

In the case that three boxes are split with respect to two components, six new boxes would replace three old ones, thus, the number of boxes would increase by three. So it remains to show that in this case, at least one box B in $\overline{\mathcal{B}}_s$ is removed without being split, i.e., $v(B) > z^s$ holds for at least one $B \in \overline{\mathcal{B}}_s$. In other words, we have to prove the existence of a '0-box', i.e., a box, which is contained in $\overline{\mathcal{B}}_s$, but is not split with respect to any component.

To this end, we assume to the contrary that $\overline{\mathcal{B}}_s$ contains three boxes which are split with respect to two components ('2-boxes'), respectively, but that no '0-box' exists. From Lemma 5.22 we see that a '2-box' has exactly one neighbor in $\overline{\mathcal{B}}_s$, as $\bar{J}_B$ contains exactly one element. A '1-box' has exactly two neighbors in $\overline{\mathcal{B}}_s$, while all three neighbors are contained in $\overline{\mathcal{B}}_s$ in case of a '0-box'. Now, starting from a '2-box', one uniquely defined neighbor of it must be in $\overline{\mathcal{B}}_s$. If that box is also a '2-box' (see

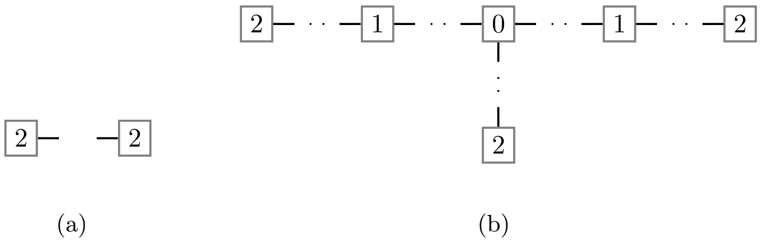


Figure 5.6: Possible neighborhood structures of boxes in $\overline{\mathcal{B}}_s$: (a) $\overline{\mathcal{B}}_s$ contains two '2-boxes'; (b) $\overline{\mathcal{B}}_s$ contains three '2-boxes'.

Figure 5.6 (a)), no neighbor of the latter box is in $\overline{\mathcal{B}}_s$. The third '2-box' would require a neighbor in $\overline{\mathcal{B}}_s$, but only '1-boxes' are available, which require a second neighbor in turn. Thus, a fourth '2-box' would be needed, which, however, does not exist. Therefore, the three '2-boxes' must all be connected by one structure of neighbors. But this implies that there exists exactly one '0-box' connecting the three branches emerging from each '2-box' (see Figure 5.6 (b)). □

Theorem 5.24. *For a finite set of nondominated points and a given appropriate starting box which includes all nondominated points and has the ideal point as lower bound vector, Algorithm 2 requires the solution of at most* $3|Z_N| - 2$ *subproblems in order to generate the entire nondominated set.*

Proof. In every iteration of Algorithm 2, one scalarized subproblem is solved, and, thus, the number of subproblems to be solved equals the number of iterations. When a nondominated point is generated, then the number of boxes increases by at most two according to Lemma 5.23. As every nondominated point is generated exactly once, and since every empty box is investigated exactly once to verify that no further nondominated points are contained, at most $3|Z_N|$ boxes are explored in the course of the algorithm. Together with the initial search box, at most $3|Z_N| + 1$ boxes are explored, which corresponds to the number of subproblems to be solved for a given appropriate initial search box containing all nondominated points.

Since we additionally assume that the ideal point is given, we can reduce this bound further: In every iteration in which the current nondominated point equals the ideal point in at least one component, one box per component equal to the ideal point can be directly discarded. For each component $i \in \{1, 2, 3\}$, there must exist at least one nondominated point whose i-th component equals z_i^I. Therefore, the total number of subproblems to be solved is at most $3|Z_N| - 2$. □

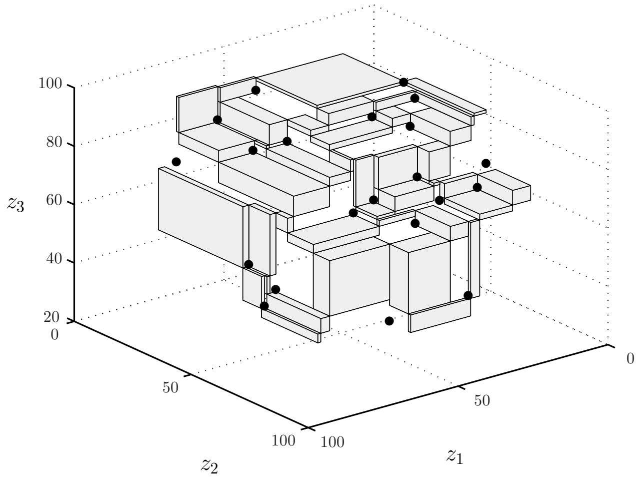


Figure 5.7: Individual subsets of the final decomposition of an example with 21 nondominated points from the nadir point perspective

Figure 5.7 shows an example with 21 nondominated points. After having determined the initial search box, $3|Z_N| - 2 = 61$ subproblems are solved until the termination criterion of Algorithm 2 is reached, i.e., the upper bound derived in Theorem 5.24 is sharp.

5.3.4 Quasi Non-Redundancy

Until now we assumed that no pair of nondominated points has an identical value in any component, i.e., that all values are pairwise different (Assumption 5.13 (1)). Under this assumption the individual subsets of all boxes which are necessary and sufficient to describe the search region can be represented as boxes themselves and are non-empty. Conversely, all boxes with empty individual subset are redundant and not maintained in the decomposition.

In practice, Assumption 5.13 (1) is typically not satisfied, as arbitrary nondominated points may coincide in up to $m-2$ components, i.e., in one component for $m = 3$. In this case additional redundant boxes may occur as the following example shows.

Example 5.25. *Let an initial decomposition be given by* $\mathcal{B}_1 := \{B_0\}$ *with*

$$B_0 := \{z \in Z \,:\, 0 \leq z_i \leq 5 \;\forall\, i = 1, 2, 3\}.$$

Let $z^1 = (3,1,4)^\top$ *and* $z^2 = (3,2,1)^\top$ *be two nondominated points. By inserting* z^1 *into* B_0 *we obtain* $\mathcal{B}_2 = \{B_{11}, B_{12}, B_{13}\}$*, where* $B_{1,i} := \{z \in B_0 \,:\, z_i < z^1_i\},\; i = 1, 2, 3,$

and

$$u(B_{11}) = (3,5,5)^\top, u(B_{12}) = (5,1,5)^\top, u(B_{13}) = (5,5,4)^\top.$$

The second point $z^2 = (3,2,1)^\top$ *is only contained in* B_{13}*, thus,* B_{13} *is replaced by the three subboxes* $B_{2,i} := \{z \in B_{13} \,:\, z_i < z_i^2\}$*,* $i = 1,2,3$*, with respective upper bounds*

$$u(B_{21}) = (3,5,4)^\top, u(B_{22}) = (5,2,4)^\top, u(B_{23}) = (5,5,1)^\top.$$

It holds that $B_{21} \subseteq B_{11}$*, thus,* B_{21} *is redundant.*

Note that under Assumption 5.13 (1) no redundancy appears if $|\overline{\mathcal{B}}_s| = 1$ which is, as shown in Example 5.25, no longer true if the nondominated points are in arbitrary position. If the redundant box B_{21} is removed from the decomposition, i.e., if we set $\mathcal{B}_3 := \{B_{11}, B_{12}, B_{22}, B_{23}\}$, then the individual subset $V(B_{11})$ does not have the structure of a box anymore, as

$$\begin{aligned} V(B_{11}) &= B_{11} \setminus \left(\bigcup_{\tilde{B} \in \mathcal{B}_3 \setminus \{B_{11}\}} \tilde{B} \right) \\ &= \{z \in B_{11} : z \geqq (0,2,1)^\top\} \cup \{z \in B_{11} : z \geqq (0,1,4)^\top\}. \end{aligned}$$

In Section 5.5 we will present an algorithm that removes all redundant boxes, including the ones that appear due to nondominated points having equal components. Alternatively, we can maintain the redundant box B_{21} in the decomposition and apply Algorithm 2 as usual. In Example 5.25 this implies to maintain B_{11} and B_{21} with $v(B_{11}) = (0,1,4)^\top$ and $v(B_{21}) = (3,2,1)^\top$. As $B_{21} \subseteq B_{11}$, the individual subset

$$V(B_{21}) = \{z \in B_0 : (3,2,1)^\top \leqq z < (3,5,4)^\top\}$$

is empty. Nevertheless, in the following iterations, the v-split can be applied regularly, as v is compared to the current nondominated point z^s component-wise. Thus, it is irrelevant for the v-split if $V(B)$ for some $B \in \mathcal{B}_s$ is empty or not. Clearly, in Example 5.25, box B_{21} cannot be split with respect to the first component, but it can be split with respect to the second and the third component like a 'regular' non-redundant box.

In order to distinguish the redundant boxes that appear when two nondominated points coincide in (at least) one component from the redundant boxes treated so far, we denote the former as *quasi non-redundant* boxes in the following.

Definition 5.26 (Quasi non-redundant boxes)**.** *Let a box* B *be given, defined by its upper bound vector* $u(B) \in \mathbb{R}^m$*. Let* $V(B)$ *be the individual subset of* B*, and let the vector* $v(B) \in \mathbb{R}^m$ *describe the lower bound of* $V(B)$*. We call* B *quasi non-redundant (with respect to* i*) if* $v_i(B) = u_i(B)$ *for some* $i \in \{1,2,3\}$ *holds.*

In Example 5.25, B_{21} is quasi non-redundant with respect to the first component. As discussed above, the occurrence of quasi non-redundant boxes does not pose any difficulties, but Algorithm 2 can be applied also when Assumption 5.13 (1) does not hold. In the following we show two special situations that might occur for quasi non-redundant boxes. The first example shows that a box might be quasi non-redundant with respect to two components, which implies that the respective box is contained in two different boxes of the decomposition. The second example demonstrates that two quasi non-redundant boxes might be contained in each other, which results in two boxes of the decomposition that have identical upper bound vectors.

Example 5.27. *As in Example 5.25, let* $\mathcal{B}_1 := \{B_0\}$ *with*

$$B_0 := \{z \in Z \,:\, 0 \leq z_i \leq 5 \;\forall\, i = 1, 2, 3\}$$

be a given initial decomposition, and let $z^1 = (3,1,4)^\top$ *and* $z^2 = (3,2,1)^\top$ *be two nondominated points. Then, the decomposition of the search region (including the quasi non-redundant box* B_{21}*) at the beginning of the third iteration is* $\mathcal{B}_3 := \{B_{11}, B_{12}, B_{21}, B_{22}, B_{23}\}$ *with*

$$\begin{aligned}
u(B_{11}) &= (3,5,5)^\top, & v(B_{11}) &= (0,1,4)^\top,\\
u(B_{12}) &= (5,1,5)^\top, & v(B_{12}) &= (3,0,4)^\top,\\
u(B_{21}) &= (3,5,4)^\top, & v(B_{21}) &= (3,2,1)^\top,\\
u(B_{22}) &= (5,2,4)^\top, & v(B_{22}) &= (3,1,1)^\top,\\
u(B_{23}) &= (5,5,1)^\top, & v(B_{23}) &= (3,2,0)^\top.
\end{aligned}$$

The corresponding individual subsets are depicted in Figure 5.8 (a). Note that the empty individual subset of the quasi non-redundant box B_{21} *is illustrated as the two-dimensional face*

$$\{z \in B_0 : v(B_{21}) \leqq z \leqq u(B_{21})\}\,.$$

Let now as third nondominated point $z^3 = (2,2,2)^\top$ *be given. As* z^3 *is contained in* B_{11} *and* B_{21}*, we consider* $v(B_{11}) = (0,1,4)^\top$ *and* $v(B_{21}) = (3,2,1)^\top$ *for the* v*-split. Comparing* z^3 *with these two vectors reveals that* B_{11} *is split with respect to the first and the second component, and* B_{21} *is split with respect to the second and the third component, which yields*

$$u(B_{31}) = (2,5,5)^\top, u(B_{32}) = (3,2,5)^\top, u(B'_{32}) = (3,2,4)^\top, u(B_{33}) = (3,5,2)^\top.$$

Hence, $\mathcal{B}_4 := \{B_{12}, B_{22}, B_{23}, B_{31}, B_{32}, B'_{32}, B_{33}\}$*. As* B_{31} *is the only box obtained by a split with respect to the first component, we obtain* $v(B_{31}) = (0,2,2)^\top$*. Analogously,*

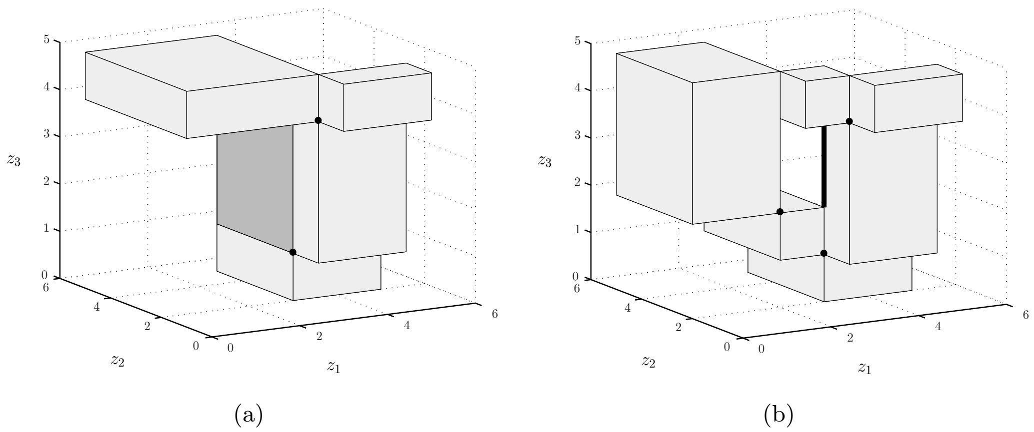


Figure 5.8: Illustration of the sets $V(B)$ in Example 5.27; In (a) the individual subset of the occurring quasi non-redundant box (which is actually empty) is represented as a two-dimensional face, in (b) as a one-dimensional face, i.e., a line.

$v(B_{33}) = (2,2,1)^\top$. *For the update of* $v(B_{32})$ *and* $v(B_{32})'$, *the upper bound vectors* $u(B_{32})$ *and* $u(B'_{32})$ *are ordered increasingly with respect to one component* $j \neq 2$ *and decreasingly with respect to the remaining component* $k \neq j, k \neq 2$, *e.g.,* $j = 1$ *and* $k = 3$. *As* $u_1(B_{32}) = u_1(B'_{32})$ *and* $u_3(B_{32}) > u_3(B'_{32})$, *we can order the boxes strictly decreasingly with respect to* $k = 3$. *Therefore,*

$$v(B_{32}) = (2,1,4)^\top \quad \text{and} \quad v(B'_{32}) = (3,2,2)^\top$$

is obtained. As $v_1(B'_{32}) = u_1(B'_{32})$ *and* $v_2(B'_{32}) = u_2(B'_{32})$, *box* B'_{32} *is quasi non-redundant with respect to the first and second component. This implies that* B'_{32} *is completely contained in two other boxes of the decomposition. Indeed,* $B'_{32} \subset B_{22}$ *and* $B'_{32} \subset B_{32}$, *as* $(3,2,4)^\top \leqq (5,2,4)^\top$ *and* $(3,2,4)^\top \leqq (3,2,5)^\top$, *respectively.*

The individual subsets of all $B \in \mathcal{B}_4$ *are depicted in Figure 5.8 (b). As the individual subset* $V(B'_{32})$ *is empty, we depict the set*

$$\{z \in B_0 : v(B'_{32}) \leqq z \leqq u(B'_{32})\} = \{z \in B_0 : (3,2,2)^\top \leqq z \leqq (3,2,4)^\top\}$$

instead, which describes a one-dimensional face. It is represented as a black line in Figure 5.8 (b).

Example 5.28. *Consider again Example 5.25, i.e.,* $\mathcal{B}_3 := \{B_{11}, B_{12}, B_{21}, B_{22}, B_{23}\}$ *is the decomposition obtained by inserting* z^1 *and* z^2 *as described above. Let now* $z^3 = (1,3,4)^\top$ *be given. This point is only contained in* B_{11}, *which is, therefore, split into three new boxes* B_{31}, B_{32} *and* B_{33} *with*

$$u(B_{31}) = (1,5,5)^\top, \quad v(B_{31}) = (0,3,4)^\top,$$

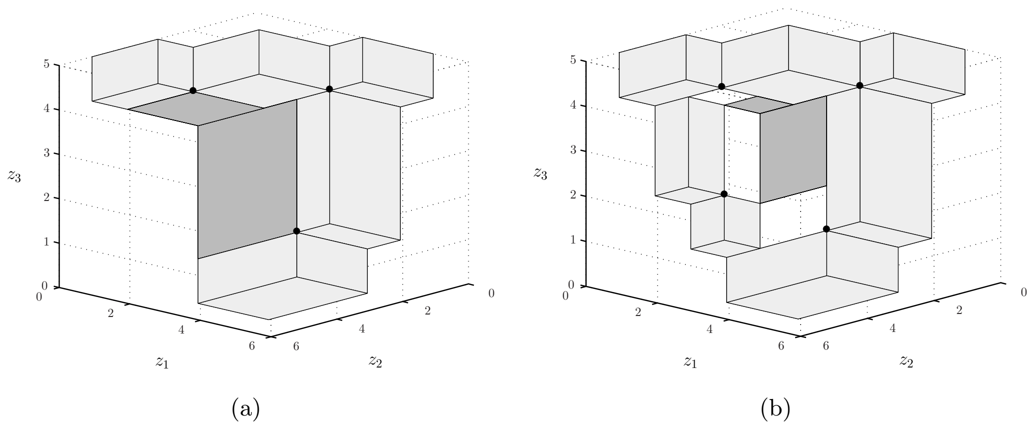


Figure 5.9: Illustration of the sets $V(B)$ in Example 5.28; the individual subsets of the occurring quasi non-redundant boxes (which are actually empty) are represented as two-dimensional faces

$$u(B_{32}) = (3,3,5)^\top, \quad v(B_{32}) = (1,1,4)^\top,$$
$$u(B_{33}) = (3,5,4)^\top, \quad v(B_{33}) = (1,3,4)^\top.$$

The resulting decomposition is $\mathcal{B}_4 := \{B_{12}, B_{21}, B_{22}, B_{23}, B_{31}, B_{32}, B_{33}\}$. Box B_{33} is quasi non-redundant with respect to the third component. Moreover, $u(B_{33}) = u(B_{21})$, thus, the two quasi non-redundant boxes B_{21} and B_{33} are equal (as the boxes are characterized by their upper bounds). However, they differ in their associated vectors v. Figure 5.9 (a) shows the respective sets $V(B)$, where, as above, the sets

$$\{z \in B_0 : v(B) \leqq z \leqq u(B)\}$$

are drawn to represent also boxes with empty individual subsets.

Let, finally, $z^4 = (2,4,2)^\top$ be given, which is contained in the two quasi non-redundant boxes B_{33} and B_{21}. The box B_{33} is split with respect to the first and second component, B_{21} is split with respect to the second and third component. Hence, the new decomposition is $\mathcal{B}_4 := \{B_{12}, B_{22}, B_{23}, B_{31}, B_{32}, B_{41}, B_{42}, B'_{42}, B_{43}\}$ with

$$u(B_{41}) = (2,5,4)^\top, u(B_{42}) = (3,4,4)^\top, u(B'_{42}) = (3,4,4)^\top, u(B_{43}) = (3,5,2)^\top.$$

As only B_{33} is split with respect to the first component, $v(B_{41}) = (1,4,2)^\top$. Analogously, $v(B_{43}) = (2,4,1)^\top$. As both boxes are split with respect to the second component, B_{42} and B'_{42} must be ordered appropriately. As the corresponding upper bound vectors are equal, we sort the boxes according to the temporary values of $v(B_{42})$ and $v(B_{42})'$ (see Lines 48 to 50 in Algorithm 2), which are identical to $v(B_{33})$ and $v(B_{21})$, *respectively. As* $v_1(B_{33}) < v_1(B_{21})$*, the boxes are ordered increasingly with respect to the first and decreasingly with respect to the third component. Hence,*

$$v(B_{42}) = (2, 3, 4)^\top \quad \textit{and} \quad v(B'_{42}) = (3, 2, 2)^\top$$

are uniquely determined. The corresponding individual subsets are depicted in Figure 5.9 (b).

Example 5.28 illustrates that the correct order of the boxes, that is required for the update of v, cannot always be determined solely based on the upper bound vectors in the quasi non-redundant case. It might be necessary to take the corresponding (temporary) values of v into account, with the help of which a unique ordering is obtained.

The Notion of Neighbors in the Quasi Non-Redundant Case

As we have seen, we can apply Algorithm 2 also when Assumption 5.13 (1) does not hold. The algorithmic procedure does not change when quasi non-redundant boxes occur. It might only be necessary to establish the required ordering of the upper bounds of the new boxes with the help of the vectors v, as illustrated in Example 5.28.

However, in the quasi non-redundant case, the notion of a neighbor of a box changes. This can be seen from Example 5.25. Box B_{21} would not have a neighbor with respect to the first component according to Definition 5.11, as $u_1(B_{21}) = \min\{u_1(B) : B \in \mathcal{B}_3\}$. This, in turn, would imply $v_1(B_{21}) = z_1^I = 0$ according to Lemma 5.16. However, $v_1(B_{21}) = 3 = u_1(B_{11})$, thus, $B_1^2(B_{21}) = B_{11}$ must hold. Moreover, even if neighbors according to Definition 5.11 exist, they might not be unique, which can also be seen from Example 5.25. Boxes B_{11} and B_{21} are both neighbors of B_{23} with respect to the first component according to Definition 5.11.

The notion of a neighbor is not relevant for the application of Algorithm 2, as the neighbors are not determined explicitly therein. However, the existence of a unique neighbor is required for the proofs of Lemma 5.22, Lemma 5.23 and Theorem 5.24, respectively. Fortunately, the neighborhood structure in the non-redundant case can be transferred to the quasi non-redundant case. However, the neighbors are no longer determined according to Definition 5.11, but are recursively defined as it is implicitly done in Algorithm 2. In the following, we indicate this recursive definition of a neighbor explicitly. Note that this definition coincides with Definition 5.11 when Assumption 5.13 (1) holds. As we will see, this explicit recursive definition is not only the correct notion of a neighbor in the quasi non-redundant case, but can also be used to design an improved variant of Algorithm 2 in the general case, i.e., also when Assumption 5.13 (1) holds.

Assumption 5.29. *Let the starting box be denoted by B_0 and defined by its upper bound vector $u(B_0) := z^M$. Furthermore, let $v(B_0) := z^I$. The initial decomposition of the search region is denoted by $\mathcal{B}_1 := \{B_0\}$. The neighbors of the initial box are defined to be empty, i.e., $B_i^1(B_0) := \emptyset$ for all $i \in \{1, 2, 3\}$.*

Definition 5.30 (Recursive definition of neighbors)**.** *Let Assumption 5.29 hold. For every $s \geq 1$, let $z^s \in Z_N$ denote the nondominated point that is generated in iteration s. Furthermore, let $\overline{\mathcal{B}}_s := \{B \in \mathcal{B}_s : z^s \in B\}$.*

Then, for every $s \geq 1$, unique neighbors of all boxes $B \in \mathcal{B}_{s+1}$ are defined recursively in the following way:

1. *Determination of boxes to be split in iteration s:*
   For every $i \in \{1, 2, 3\}$, let $S_i := \{B \in \overline{\mathcal{B}}_s : v_i(B) \leq z_i^s\}$. Let $Q_i := |S_i|$ for all $i \in \{1, 2, 3\}$ and let $I_i \subset \mathbb{N}$, $I_i \neq \emptyset$, denote the index set of the boxes contained in S_i, i.e., $S_i := \{\hat{B}^{I_i(1)}, \ldots, \hat{B}^{I_i(Q_i)}\}$. Furthermore, let these boxes be ordered such that

$$z_j^s < u_j(\hat{B}^{I_i(1)}) \leq u_j(\hat{B}^{I_i(2)}) \leq \cdots \leq u_j(\hat{B}^{I_i(Q_i)}), \tag{5.19}$$
$$u_k(\hat{B}^{I_i(1)}) \geq u_k(\hat{B}^{I_i(2)}) \geq \cdots \geq u_k(\hat{B}^{I_i(Q_i)}) > z_k^s, \tag{5.20}$$

   holds for some $j \neq i$ and $k \neq i, j$. If there exists $p \in \{1, \ldots, Q_i - 1\}$ with $u_j(\hat{B}^{I_i(p)}) = u_j(\hat{B}^{I_i(p+1)})$ and $u_k(\hat{B}^{I_i(p)}) = u_k(\hat{B}^{I_i(p+1)})$, then order $\hat{B}^{I_i(p)}$ and $\hat{B}^{I_i(p+1)}$ such that $v_j(\hat{B}^{I_i(p)}) \leq v_j(\hat{B}^{I_i(p+1)})$ and $v_k(\hat{B}^{I_i(p)}) \geq v_k(\hat{B}^{I_i(p+1)})$ hold.

2. *Determination of neighbors:*
   a) *For all new boxes $\hat{B}_i^{I_i(q)} \in \mathcal{B}_{s+1}$, $q = 1, \ldots, Q_i$, obtained from $\hat{B}^{I_i(q)}$ by a split with respect to i, we set:*

$$B_i^{s+1}(\hat{B}_i^{I_i(q)}) := B_i^s(\hat{B}^{I_i(q)}) \quad \forall\, q = 1, \ldots, Q_i, \tag{5.21}$$
$$B_j^{s+1}(\hat{B}_i^{I_i(q)}) := \begin{cases} \hat{B}_j^{I_j(1)} & q = 1, \\ \hat{B}_i^{I_i(q-1)} & \forall\, q = 2, \ldots, Q_i, \end{cases} \tag{5.22}$$
$$B_k^{s+1}(\hat{B}_i^{I_i(q)}) := \begin{cases} \hat{B}_i^{I_i(q+1)} & \forall\, q = 1, \ldots, Q_i - 1, \\ \hat{B}_k^{I_k(Q_k)} & q = Q_i, \end{cases} \tag{5.23}$$

   where I_j is assumed to be sorted decreasingly with respect to k and I_k is assumed to be sorted increasingly with respect to j.

b) The neighbors of all $B \in \mathcal{B}_s, B \notin \overline{\mathcal{B}}_s, B_i^s(B) \notin \overline{\mathcal{B}}_s$ *for all* $i \in \{1,2,3\}$ *remain unchanged, i.e.,*

$$B_i^{s+1}(B) = B_i^s(B)\ \forall\, i \in \{1,2,3\}. \tag{5.24}$$

c) The neighbors of all boxes $B \in \mathcal{B}_s, B \notin \overline{\mathcal{B}}_s$*, for which* $B_j^s(B) = \hat{B} \in \overline{\mathcal{B}}_s$ *for some* $j \in \{1,2,3\}$*, are defined as follows:*

$$B_j^{s+1}(B) := \hat{B}_i, \tag{5.25}$$

where $i \neq j$ *is the unique index for which* $z_i^s \geq u_i(B)$ *holds.*

Note that (5.19) corresponds to (5.12) and (5.20) corresponds to (5.13) in the proof of Lemma 5.14. Moreover, (5.21), (5.22) and (5.23) correspond to (5.14), (5.15) and (5.16) in Corollary 5.15. Finally, (5.24) and (5.25) relate to cases (iii) and (iv) in the proof of Lemma 5.14.

Once having specified unique neighbors for each box, the lower bounds $v(B)$ are updated as usual. As in the non-redundant case, in every iteration s, the lower bound vectors of all $B \in \mathcal{B}_s$ are

$$v_i(B) := \begin{cases} u_i(B_i^s(B)), & \text{if } B_i^s(B) \neq \emptyset \\ z_i^I, & \text{otherwise} \end{cases}, \quad i \in \{1,2,3\}.$$

Due to Definition 5.30 every box has a unique neighbor with respect to every component. Based on this neighborhood structure, Lemma 5.22, Lemma 5.23 and Theorem 5.24 also hold in the quasi non-redundant case.

A Variant of Algorithm 2 with Explicit Update of the Neighbors

As already mentioned above, the explicit recursive update of the neighbors is not only needed in the quasi non-redundant case, but also yields a variant of Algorithm 2 with improved performance. This variant is briefly described in the following. Recall that in Algorithm 2 in every iteration, in which a new nondominated point is found, the entire list of boxes is scanned in order to find the boxes that contain the current nondominated point. Moreover, in the update of the individual subsets, the boxes that are obtained by the split have to be ordered such that the correct new values of v can be determined.

Both aspects can be improved if the neighbors are saved explicitly. Due to Lemma 5.22 and the proof of Lemma 5.23 we know that if $|\overline{\mathcal{B}}_s| > 1$, i.e., if more than one box contains the nondominated point in the current iteration, then all boxes $B \in \overline{\mathcal{B}}_s$ are connected by a chain of neighbors. We start from the box B, in which the scalarization has been solved, and which, therefore, contains the current point z^s. By investigating $v(B)$, we can identify the components, with respect to which B is split as well as the components, with respect to which a neighbor exists which is contained in $\overline{\mathcal{B}}_s$. If, e.g., $v(B) \leqq z^s$, then we do not have to search for further boxes in $\overline{\mathcal{B}}_s$, as then $|\overline{\mathcal{B}}_s| = 1$ must hold. If, however, $v_i(B) > z_i^s$ for some component $i = 1, 2, 3$, then $B_i^s(B) \in \overline{\mathcal{B}}_s$ must hold, as $u_i(B_i^s(B)) = v_i(B) > z_i^s$ and $u_j(B_i^s(B)) \geq u_j(B) > z_j^s$ for all $j \neq i$. By subsequently investigating neighbor boxes, $\overline{\mathcal{B}}_s$ can be obtained without testing every box of the current decomposition. Moreover, by using the neighborhood structure, the sorting of the boxes in lines 47 to 50 in Algorithm 2 can be simplified. While detecting $\overline{\mathcal{B}}_s$, the sets $\mathcal{S}_i, i = 1, 2, 3$, can be sorted immediately.

5.4 The ε-Constraint Method in Combination with the v-Split

Algorithm 2 presented in Section 5.3 is formulated independently of a specific scalarization. In every iteration only points that are dominated by the current nondominated point are eliminated. In this section we show that we can reduce the search region and, thereby, the number of subproblems further if we use the ε-constraint method. In the tricriteria case the number of subproblems can be bounded by $2|Z_N| - 1$.

The algorithm of Laumanns et al. (2006) makes implicitly use of this property of the ε-constraint method by projecting the set of feasible outcomes to an $(m-1)$-dimensional subset. Thereby, in the bicriteria case, Laumanns et al. (2006) can generate a complete representation by solving $|Z_N| + 1$ subproblems, see Section 3.2 for a detailed description of the corresponding algorithm. Also Hamacher et al. (2007) exploit properties of the ε-constraint method, by which the area of the rectangles considered in their bicriteria method can be diminished further. Details can be found in Section 3.2. However, as to the best of our knowledge we present the first algorithm for tricriteria problems whose number of subproblems depends linearly on the number of nondominated points, the bound which is derived in this section is new as well.

The reduction of the search region stems from the following property of the ε-constraint method, which holds for any number of criteria. First, recall that for every $z^* \in Z_N \cap B$, by definition of nondominance, we can exclude the two sets

$$S_1(z^*) := \{z \in B \,:\, z \leqq z^*\} \quad \text{and} \quad S_2(z^*) := \{z \in B \,:\, z \geqq z^*\}$$

from a given box $B := \{z \in \mathbb{R}^m : z < u(B)\}$. If the point z^* has been obtained as

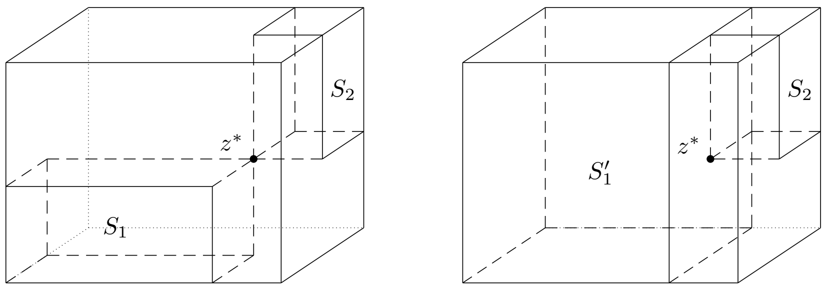


Figure 5.10: Reduction of the search region for $m = 3$: Solely based on nondominance of z^* (left) and when taking into account that z^* is obtained as optimal point of a corresponding ε-constraint method (right)

an optimal solution of an ε-constraint problem of the form

$$\begin{aligned} \min \quad & z_1 \\ \text{s.t.} \quad & z_i < u_i(\bar{B}) \quad \forall\, i = 2, \ldots, m, \end{aligned} \tag{5.26}$$

where $\bar{B}$ is a box of the current decomposition, then, additionally to $S_1(z^*)$ and $S_2(z^*)$, the set

$$S_1'(z^*) := \{z \in \bar{B} \,:\, z_1 < z_1^*\} = \{z \in \mathbb{R}^m \,:\, z_1 < z_1^*,\; z_i < u_i(\bar{B})\; \forall\, i = 2, \ldots, m\}$$

cannot contain any further nondominated points, as this would contradict the optimality of z^* in (5.26). Thereby, $S_1'(z^*)$ depends on the chosen box $\bar{B}$ as well as on the component i with respect to which the ε-constraint problem is minimized. Throughout this section we choose $i = 1$ without loss of generality. In Figure 5.10, an example of the sets S_1,S_1' and S_2 is depicted. Note that in order to guarantee that the outcome z^* is nondominated, a two-stage or an augmented ε-constraint method should be employed.

We consider now the implications of this additional reduction of the search region in the tricriteria case, i.e., in combination with the v-split algorithm. Let box $\bar{B} \in \mathcal{B}_s$ be the currently selected box, and let z^s denote the nondominated point obtained in iteration s. Then, the set $S_1'(z^s) := \{z \in \bar{B} \,:\, z_1 < z_1^s\}$ corresponds to the box obtained by a split of $\bar{B}$ with respect to the first component. Since $S_1'(z^s)$ is empty, $\bar{B}$ does not need to be split with respect to the first component. A split with respect to all other components $i \in \{2, 3\}$ is performed according to the v-split criterion, i.e., if and only if $z_i^s \geq v_i(\bar{B})$ holds for $i \in \{2, 3\}$. For all other boxes $B \in \overline{\mathcal{B}}_s \backslash \{\bar{B}\}$ the usual v-split criterion is employed with respect to all components. In particular, a box $B \in \overline{\mathcal{B}}_s \backslash \{\bar{B}\}$ must be split with respect to the first component whenever $z_1^s \geq v_1(B)$ holds.

In order to benefit from the fact that the set $S_1'(z^s)$ can be excluded additionally from the search region, we must guarantee that the box resulting from a split of $\bar{B}$ with respect to the first component would have been part of the decomposition. According to the definition of the v-split, this is the case if $z_1^s \geq v_1(\bar{B})$ holds. A sufficient criterion to guarantee that $z_1^s \geq v_1(\bar{B})$ holds is to select a box $\bar{B}$ that does not have a neighbor in $\mathcal{B}_s$ with respect to $i = 1$, i.e., $B_i^s(\bar{B}) = \emptyset$. Equivalently, we might select a box $\bar{B}$ which satisfies $v_1(\bar{B}) = \min\{v_1(B) : B \in \mathcal{B}_s\}$. This means that we replace Line 5 in Algorithm 2 by 'choose $\bar{B} \in \mathcal{B}_s$ such that $v_1(\bar{B}) = \min\{v_1(B) : B \in \mathcal{B}_s\}$'. If a box with minimal value v_1 is selected according to this rule at the beginning of each iteration, then one box is saved in each iteration in which a new nondominated point is generated that does not equal the ideal point in the first component. Therefore, we obtain $2|Z_N| - 1$ as new upper bound on the number of subproblems to be solved in the tricriteria case.

Example 5.31 (Application of Algorithm 2 with ε-constraint scalarization)**.** *Consider a tricriteria problem with $Z_N = \{(2,2,4)^\top, (4,1,2)^\top, (3,4,1)^\top\}$, i.e., the nondominated set consists of three points. Let the initial search region*

$$B_0 := \{z \in Z \,:\, 0 \leq z_i \leq 5 \;\forall\, i = 1,2,3\}$$

be given. Solving (5.26) *in B_0 yields the nondominated point with the smallest value in the first component, thus, $z^1 = (2,2,4)^\top$. The v-split with respect to z^1 applied to B_0 results in three boxes. As, due to* (5.26)*, the box resulting from B_0 by a split with respect to the first component is known to be empty, this box is not generated, and we obtain $\mathcal{B}_2 = \{B_{12}, B_{13}\}$ with*

$$\begin{aligned} u(B_{12}) &:= (5,2,5)^\top, \quad v(B_{12}) := (2,0,4)^\top, \\ u(B_{13}) &:= (5,5,4)^\top, \quad v(B_{13}) := (2,2,0)^\top. \end{aligned}$$

In the next iteration, we may choose any of the two boxes in $\mathcal{B}_2$ as $v_1(B_{12}) = v_1(B_{13})$ holds. W.l.o.g., we select box B_{12}. Solving (5.26) *with right-hand side $u(B_{12})$ yields $z^2 = (4,1,2)^\top$, as $(3,4,1)^\top \notin B_{12}$. Since $z^2 \in B_{13}$, a v-split with respect to z^2 is applied to both boxes B_{12} and B_{13}. As z^2 has been generated in B_{12}, no box with respect to the first component is derived from B_{12}. As $z_2^2 \geq v_2(B_{12})$ but $z_3^2 < v_3(B_{12})$, box B_{12} is only split with respect to its second component. Comparing all three components of $v(B_{13})$ with z^2 yields that B_{13} is split with respect to its first and third component. Thus, $\mathcal{B}_3 = \{B_{21}, B_{22}, B_{23}\}$, where*

$$\begin{aligned} u(B_{21}) &:= (4,5,4)^\top, \quad v(B_{21}) := (2,1,2)^\top, \\ u(B_{22}) &:= (5,1,5)^\top, \quad v(B_{22}) := (4,0,2)^\top, \end{aligned}$$

$$u(B_{23}) := (5,5,2)^\top, \quad v(B_{23}) := (4,1,0)^\top.$$

In the next iteration we choose B_{21}*, as* $v_1(B_{21}) < \min\{v_1(B_{22}), v_1(B_{23})\}$ *holds. Solving* (5.26) *with right-hand side* $u(B_{21})$ *yields* $z^3 = (3,4,1)^\top$*. It holds that* $z^3 \in B_{21}$*,* $z^3 \notin B_{22}$ *and* $z^3 \in B_{23}$*. Again, box* B_{21}*, that has been selected in the current iteration, is not split with respect to the first component. As* $v_2(B_{21}) < z_2^3$*,* $v_3(B_{21}) > z_3^3$*,* $v_1(B_{23}) > z_1^3$*,* $v_2(B_{23}) < z_2^3$ *and* $v_3(B_{23}) < z_3^3$*,* B_{21} *and* B_{23} *are both split with respect to the second component,* B_{23} *is split with respect to the third component and no box is split with respect to the first component. In total, as in the previous iterations, one additional box is obtained. Note that if we had chosen* B_{23} *instead of* B_{21} *for solving the third subproblem, then, as* $v_1(B_{23}) > z_1^3$*, box* B_{23} *would not have been split with respect to the first component regardless of the scalarization used. Therefore, two new boxes would have been generated in this case.*

The presented improved method based on Algorithm 2 in combination with the ε-constraint scalarization requires only $(2|Z_N| - 1)/(3|Z_N| - 2) \approx 2/3$ of the subproblems needed in the general case, i.e., when no specific scalarization is applied. However, in order to benefit from this saving of subproblems, the boxes cannot be solved in an arbitrary order, but the next box must be selected from the boxes with minimal value v_1. Note that this restriction is irrelevant whenever a complete representation is sought, as the order in which nondominated points are generated does not matter then. However, with regard to an incomplete representation, it might be unfavorable to select the next box only from a very restricted subset of all boxes.

5.5 Generalization of the v-Split Algorithm for $m \geq 3$

In this section we present an improvement on Algorithm 2 in the sense that the generation of quasi non-redundant boxes is avoided. At the same time, the resulting algorithm generalizes the v-split algorithm to any number of criteria.

In Section 5.3 we showed that each box of the decomposition has a unique neighbor with respect to every component $i \in \{1,2,3\}$. The respective components of the upper bound vectors of these three uniquely determined boxes defined the lower bound of the individual subset of the considered box. The components of this lower bound vector were used to decide whether the considered box is split with respect to this component or not. We obtained a quasi non-redundant box from $\bar{B} \in \overline{\mathcal{B}}_s$ with respect to component $i \in \{1,2,3\}$ if and only if $v_i(\bar{B}) = z_i^s$. Therefore, as a direct consequence, we can avoid quasi non-redundant boxes if we split a box $\bar{B} \in \overline{\mathcal{B}}_s$ with respect to component $i \in \{1,2,3\}$ if and only if $z_i^s > v_i(\bar{B})$ holds. However, then, as

Algorithm 3 A v-split algorithm for $m \geq 3$

Input: Image of the feasible set $Z \subset \mathbb{R}^m$, implicitly given by some problem formulation

$N := \emptyset; \delta > 0;$
for $j = 1$ to m **do**
    $z_j^I := \min\{z_j \,:\, z \in Z\};$
    $z_j^M := \max\{z_j \,:\, z \in Z\} + \delta;$
end for
$U := \{z^M\};$ // Boxes represented by upper bound vectors only
while $U \neq \emptyset$ **do**
    Choose $\bar{u} \in U;$
    $z^s := \mathit{opt}(Z, \bar{u});$ // Solve scalarized subproblem
    if $z^s = \emptyset$ **then** // No nondominated point found
        $U := U \backslash \{\bar{u}\};$
    else
        $N := N \cup \{z^s\};$ // Add point to nondominated set
        Determine $T := \{u \in U : z < u\};$ // Upper bounds that ‘contain’ z^s
        for $i = 1$ to m **do**
            $U^i := T;$ // Copy set T
            for all $\tilde{u} \in U^i$ **do**
                Determine $N(\tilde{u}) := \{u \in U : u_i < \tilde{u}_i, u_j \geq \tilde{u}_j \;\forall\, j \neq i\};$
                if $N(\tilde{u}) = \emptyset$ **then**
                    $v := z_i^I;$
                else
                    $v := \max\{u_i : u \in N(\tilde{u})\};$ // ‘Closest’ upper bound
                end if
                if $v < z_i$ **then**
                    $\tilde{u}_i = z_i;$ // Change bound
                else
                    $U^i := U^i \backslash \{\tilde{u}\};$ // Remove upper bound from temporary set
                end if
            end for
        end for
        $U := (U \backslash T) \cup \left(\bigcup_{i=1}^m U^i\right);$ // Remove old and add new upper bounds to U
    end if
end while

Output: Set of nondominated points N

discussed in Section 5.3.4, the sets

$$\left\{z \in \mathbb{R}^3 : v(\bar{B}) \leqq z < u(\bar{B})\right\} \tag{5.27}$$

do no longer describe the individual subset $V(\bar{B})$, i.e., the set that is contained in $\bar{B}$ but in no other box $B \neq \bar{B}$ of the current decomposition. While the concept of individual subsets served as motivation for constructing the v-split, it is not necessary for the algorithm, i.e., Algorithm 2 does not rely on whether (5.27) describes the individual subset of $\bar{B}$ or not. As the current nondominated point is compared component-wise to $v(\bar{B})$, it is sufficient that the components of $v(\bar{B})$ equal the upper bound values of the respecting neighbor boxes.

If no quasi non-redundant boxes occur, then, according to Definition 5.11, a neighbor $\hat{B} \in \mathcal{B}_s$ of a box $\bar{B} \in \mathcal{B}_s$ with respect to $i \in \{1, 2, 3\}$ satisfies

$$\begin{aligned} &u_i(\hat{B}) < u_i(\bar{B}), \\ &u_j(\hat{B}) > u_j(\bar{B}) \quad \text{for some } j \neq i, \\ &u_k(\hat{B}) \geq u_k(\bar{B}) \quad \text{for } k \neq i, j \end{aligned} \tag{5.28}$$

and

$$u_i(\hat{B}) = \max\{u_i(B) : B \in \mathcal{B}_s, u_i(B) < u_i(\bar{B})\}.$$

This definition of a neighbor can be generalized to arbitrary $m \geq 3$ as follows. Box $\hat{B} \in \mathcal{B}_s$ is defined to be a neighbor of box $\bar{B} \in \mathcal{B}_s$ with respect to $i = 1, \ldots, m$ if

$$\begin{aligned} &u_i(\hat{B}) < u_i(\bar{B}), \\ &u_j(\hat{B}) \geq u_j(\bar{B}) \quad \text{for all } j \neq i \end{aligned} \tag{5.29}$$

and

$$u_i(\hat{B}) = \max\{u_i(B) : B \in \mathcal{B}_s, u_i(B) < u_i(\bar{B})\}.$$

Obviously, if no quasi non-redundant boxes occur, (5.28) is a special case of (5.29). By using (5.29) we can formulate an algorithm for any number of criteria, i.e., $m \geq 3$. Algorithm 3 shows the pseudocode of a corresponding algorithm. Note that, different from Algorithm 2, we do not save the vectors $v(B)$ for each box B explicitly, but determine the values required to decide whether a box is split each time from scratch. Therefore, instead of using box structures as in Algorithm 2, it is sufficient to save a list of upper bound vectors U. As discussed in Section 5.3.4, (5.28) and, thus, (5.29) are only valid if no quasi non-redundant boxes occur. However, quasi non-redundant boxes can be easily suppressed as explained above. This is realized in Line 24 in Algorithm 3.

While the correctness of Algorithm 3 is immediately clear for $m = 3$ and easy to see for $m \geq 4$, no theoretical bound on the number of subproblems can be derived from this algorithm, in general. Recall that the results of Section 5.3.3, in which the linear bound in the tricriteria case was proven, relied on the representation of the individual subsets, which is unknown for Algorithm 3. However, as Algorithm 3 is very similar to Algorithm 2 in the tricriteria case, and as quasi non-redundant boxes are additionally suppressed, we can expect that less than $3|Z_N| - 2$ subproblems are solved in the tricriteria case. Numerical experiments in Section 6.3 for tricriteria problems confirm that, in general, less subproblems are solved with Algorithm 3 compared to Algorithm 2, but that no longer a constant number of new boxes is created per iteration. Moreover, different from Algorithm 2, the order, in which the nondominated points are generated, influences the number of subproblems. This is illustrated in the next example.

Example 5.32. *Let*

$$Z_N = \left\{ \begin{pmatrix} 1 \\ 1 \\ 9 \end{pmatrix}, \begin{pmatrix} 1 \\ 4 \\ 8 \end{pmatrix}, \begin{pmatrix} 1 \\ 6 \\ 7 \end{pmatrix}, \begin{pmatrix} 4 \\ 6 \\ 6 \end{pmatrix}, \begin{pmatrix} 5 \\ 6 \\ 5 \end{pmatrix}, \begin{pmatrix} 7 \\ 1 \\ 6 \end{pmatrix}, \begin{pmatrix} 9 \\ 1 \\ 2 \end{pmatrix} \right\}$$

be the nondominated set of a given problem, which is a slight modification of an example of Fonseca (2013). We apply Algorithm 3 using an ε-constraint scalarization in a two-stage (TS) and an augmented formulation (A). Recall from Section 5.4 that in case that the ε-constraint method is used and we want to make use of the additional reduction of the search region, the box at the beginning of each iteration can not be chosen freely, but must be a box with minimal v_1-value. As Algorithm 3 does not save the vectors v for each box B explicitly, we determine a box instead that does not have a neighbor with respect to the first component.

Table 5.2 shows the upper bounds U and the solution of the subproblem z of each iteration, which either corresponds to a new nondominated point or the empty set.

We see that depending on the formulation of the scalarization, the nondominated points are computed in a different order. In this example, the two-stage variant requires one iteration less than the augmented variant. Moreover, the number of additional boxes generated in each iteration varies. If we apply Algorithm 2, then, independent of a two-stage or an augmented formulation $2|Z_N| - 1 - 2 = 11$ subproblems are solved. Note that two additional subproblems are saved as there are three nondominated points which equal the ideal point in the second or third component.

It	TS						A					
	U			z			U			z		
1	10	10	10	1	6	7	10	10	10	1	1	9
2	10	6	10	1	4	8	10	10	9	1	4	8
	10	10	7									
3	10	10	7	4	6	6	10	4	9	7	1	6
	10	4	10				10	10	8			
	10	6	8									
4	10	4	10	1	1	9	7	10	8	1	6	7
	10	6	8				10	10	6			
	10	10	6									
5	10	6	8	7	1	6	10	10	6			
	10	10	6				7	6	8		$\emptyset$	
	10	4	9				7	10	7			
6	10	10	6	5	6	5	10	10	6			
	7	4	9				7	10	7	4	6	6
7	7	4	9		$\emptyset$		10	10	6	5	6	5
	10	6	6				7	6	7			
	10	10	5									
8	10	6	6	9	1	2	7	6	7		$\emptyset$	
	10	10	5				10	6	6			
							10	10	5			
9	9	10	5		$\emptyset$		10	6	6	9	1	2
							10	10	5			
10		–			–		9	10	5		$\emptyset$	

Table 5.2: Upper bound vectors U and nondominated points z obtained in Example 5.32; due to different formulations of the ε-constraint method the nondominated points are generated in different orders, which results in a different number of iterations of Algorithm 3

5.6 Conclusion and Further Ideas

In this chapter we have developed a new parametric algorithm. We particularly focused on the application to discrete problems with a finite nondominated set, for which a complete representation can be computed. In this situation it is of particular interest to limit the number of subproblems that need to be solved. While a linear bound was known in the bicriteria case, the best known bound so far in the tricriteria case had a quadratic dependence on the number of nondominated points.

We demonstrated by indicating a new parametric algorithm that the maximal number of subproblems to be solved depends linearly on the number of nondominated points in the tricriteria case. More precisely, an explicit upper bound is given by $3|Z_N| - 2$. The linear upper bound is achieved by a new split criterion, the so-called v-split, which identifies redundant boxes and does not insert them into the decomposition of the search region. As scalarization any method can be chosen by which a given box can be investigated for further nondominated points. Possible choices are, e.g., a weighted Tchebycheff method or an ε-constraint method. This flexibility in choosing a scalarization is very appealing and makes the parametric algorithm applicable to a multitude of problems.

If the ε-constraint method is applied and the box to be investigated next is chosen according to a specific rule as specified in Section 5.4, the number of subproblems can be bounded by $2|Z_N| - 1$.

We have also presented a generalization of the v-split algorithm for an arbitrary number of criteria. This algorithm slightly differs from the algorithm, for which a linear bound on the number of subproblems was shown. So far we have not derived an upper bound on the number of subproblems for the generalized algorithm.

According to the results of Boissonnat et al. (1998), mentioned in Section 5.1, and a counterexample given in Fonseca et al. (2011), no linear bound exists for $m \geq 4$. In general, a bound $\mathcal{O}(|Z_N|^{\lfloor m/2 \rfloor})$ is expected (Fonseca, 2013), which would match the complexity stated in Boissonnat et al. (1998). However, no algorithm realizing the expected complexity for $m \geq 4$ is known so far. Hence, future research should address the question whether an upper bound of $\mathcal{O}(|Z_N|^{\lfloor m/2 \rfloor})$ can be derived for Algorithm 3. Thereby, ideas from the field of computational geometry as well as algorithms that determine the dominated hypervolume in more than three dimensions could be useful.

Part II

Practical Application of Adaptive Parametric Algorithms

6 Generation of Complete Representations for Discrete Test Problems

6.1 Introduction

In this chapter, we apply the theoretical results of Part I to discrete problems. Thereby, Chapter 4 provides the formulas for the determination of adaptive parameters for each subproblem and Chapter 5 yields the general parametric algorithm, by which a representation of the nondominated set is generated subsequently. If the nondominated set is finite, a complete representation can be computed. Throughout this chapter we assume that a discrete multicriteria optimization problem with a finite nondominated set is given. In particular, we generate complete representations for bi- and tricriteria problems. The topic of generating incomplete representations will be treated in Chapter 7 below.

For the bicriteria case, we focus on testing the parameters of the augmented weighted Tchebycheff method derived in Chapter 4. In particular, we want to verify that with our adaptive parameter scheme every nondominated point can be generated, while this is not necessarily the case when employing the (classic) parameter scheme with fixed augmentation parameters. We also study the implications of the chosen reference point with respect to the values of the augmentation terms as well as computational time.

In the tricriteria case, we do not only validate the adaptive parameter scheme of Chapter 4, but also test the new parametric algorithm for tricriteria problems that is proposed in Chapter 5. In particular, we want to confirm the new linear upper bound on the number of subproblems. As scalarization, we use a two-stage and an augmented weighted Tchebycheff method, for which an upper bound of $3|Z_N| - 2$ should be realized according to the theoretical results of Chapter 5. As described in Section 5.4, the upper bound on the number of subproblems can be improved to $2|Z_N| - 1$ when an ε-constraint method is used. We also validate this bound by numerical results for a two-stage and an augmented ε-constraint method. Besides the number of subproblems solved, also the achieved computational times are of interest. Finally, we compare our new algorithm to three state of the art approaches to test its performance in comparison to existing methods.

In all tests, i.e., in the bicriteria and tricriteria case, we additionally study the performance of the augmented weighted Tchebycheff method in comparison to a two-stage method. In Miettinen et al. (2006), an experimental comparison of methods with and without augmentation term for continuous problems revealed that methods with augmentation term significantly outperform equivalent methods without such a term with respect to computational costs. We are interested whether we can observe the same effect in the discrete case.

The remainder of this chapter is organized as follows. Section 6.2 contains a numerical study for the bicriteria case. In Section 6.3 the tricriteria case is treated. A conclusion together with further ideas is presented in Section 6.4.

6.2 Bicriteria Problems

As already discussed in Section 5.2, the bicriteria case is special for several reasons. First, if the nondominated points are maintained in a list that is sorted increasingly with respect to one component, then, due to the definition of nondominance, the points are automatically sorted decreasingly with respect to the other component. Secondly, the upper bound $u \in \mathbb{R}^2$ of each box is defined by two subsequent nondominated points in the list. Therefore, all information can be easily retrieved from the sorted list of points and there is no need to save the upper bounds explicitly. Finally, a new nondominated point is always located in exactly one box. Therefore, the regions that contain further nondominated points do not overlap and no redundancy occurs. Consequently, no specific split criterion is required, and the general full m-split algorithm presented in Section 5.2 can be applied efficiently.

This section is organized as follows. In Subsection 6.2.1 we present a particular full 2-split algorithm which makes use of the simplifications in the bicriteria case mentioned above. In Subsection 6.2.2, we discuss the use of local ideal points as reference points. Our computational setup is presented in Subsection 6.2.3. Finally, in Subsection 6.2.4, numerical results comparing different scalarization variants are presented and discussed. Parts of this section have already been published in Dächert et al. (2012).

6.2.1 Implementation of the Full 2-Split Algorithm

Algorithm 1 presented in Section 5.2 is a general framework of a parametric algorithm that relies on the full m-split. In every iteration, in which a new nondominated point is generated, every box that contains the current nondominated point is decomposed into m new boxes. Algorithm 4 presented below can be seen as a special case of Algorithm 1 for bicriteria problems. Moreover, it is similar to existing algorithms that generate complete representations for bicriteria problems, see, e.g., Aneja and Nair (1979), Chalmet et al. (1986), Solanki (1991) or Ralphs et al. (2006). However, different from these algorithms, no particular scalarization is prescribed in Algorithm 4.

Algorithm 4 The full m-split algorithm in the bicriteria case

---

Input: Image of the feasible set $Z \subset \mathbb{R}^2$, implicitly given by some problem formulation

$z^1 = (z^1_1, z^1_2) := \text{lexmin}\{(z_1, z_2) : z \in Z\}$; // Compute the two
$z^2 = (z^2_1, z^2_2) := \text{lexmin}\{(z_2, z_1) : z \in Z\}$; // lexicographic minima
if $z^1 = z^2$ **then**
  $\mathcal{N} := \{z^1\}$;
else
  $\mathcal{N} := \{z^1, z^2\}$; // sorted list of nondominated points
  $l(z^1, z^2) = 1$; // Label
  while $I := \{i \in \{1, \dots, |\mathcal{N}| - 1\} : l(z^i, z^{i+1}) = 1\} \neq \emptyset$ **do**
    Choose $i \in I$ according to some predefined rule
    Compute parameter set $\mathcal{P}$ wrt. z^i,z^{i+1};
    $z^* := opt(Z, \mathcal{P})$; // Solve scalarized subproblem
    if $z^* = z^i$ or $z^* = z^{i+1}$ **then**
      $l(z^i, z^{i+1}) := 0$; // Label as investigated
    else
      Insert z^* between z^i and z^{i+1} in ND;
      $l(z^i, z^*) := 1$, $l(z^*, z^{i+1}) := 1$;
    end if
  end while
end if

Output: Set of nondominated points $\mathcal{N}$ (points ordered increasingly w.r.t. z_1)

The algorithm starts by computing the two lexicographic minima, which are the first two entries of the list $\mathcal{N}$. All elements which are consecutively added to the list are kept sorted with respect to their first coordinate in ascending order. Furthermore, a label is assigned to each pair of subsequent elements to indicate whether the pair has already been investigated. If yes, the corresponding pair is labeled permanently by zero, otherwise it is labeled temporarily by one. In each iteration, a temporarily labeled pair of points is selected according to some specified rule. Note that whenever a complete representation is sought, we might choose any temporarily labeled pair of points, e.g., always the first in the list. After having selected an unlabeled pair of points, the parameters of the selected scalarization are computed with respect to this pair of points. Thereby, the parameters are chosen such that either a new nondominated point is obtained 'between' these points or the pair of points can be labeled as investigated, since no further nondominated points exist 'between' them. We use the notation $z^* := opt(Z, \mathcal{P})$ in Line 10 of Algorithm 4 to indicate that the selected scalarization is solved for a particular parameter choice ($\mathcal{P}$) and using the problem formulation (Z). Note that, as discussed in Chapter 5, a scalarization must be chosen that is suitable for discrete problems, i.e., for the case that nonsupported points exist.

In our numerical study, either a Tchebycheff-type method with parameters as specified in Section 4.2 or an augmented ε-constraint approach with parameters chosen as in Section 4.4 is used for solving the subproblems. If the lexicographic (two-stage) weighted Tchebycheff method is used, the values of the parameters w_1 and w_2 are chosen as given in (4.9). The same formula is used to determine the weights for the classic augmented weighted Tchebycheff method with fixed augmentation term. Note that if this method is applied, the value of ρ is specified at the beginning of the algorithm and remains unchanged for all subproblems. If, in contrast, the adaptive augmented weighted Tchebycheff method is applied, then w_1, w_2 and ρ are dynamically updated according to the formulas given in Section 4.2, Table 4.2 for some $\bar{\eta} \in (0, 1)$.

The augmented ε-constraint method has been additionally implemented in order to study whether and how the structure of the respective scalarization influences computational time. We use formulation (4.36) with $k = 2$. Furthermore, the parameter ρ is set adaptively according to (4.38) and dependent on the same parameter $\bar{\eta} \in (0, 1)$ as it is used for the augmented weighted Tchebycheff method. Note that we do not make use of the reduction of subproblems that is possible when the complete nondominated set is generated with the help of the ε-constraint method as described in Section 5.4, since, in this numerical study, we want to evaluate the scalarizations and not the number of subproblems.

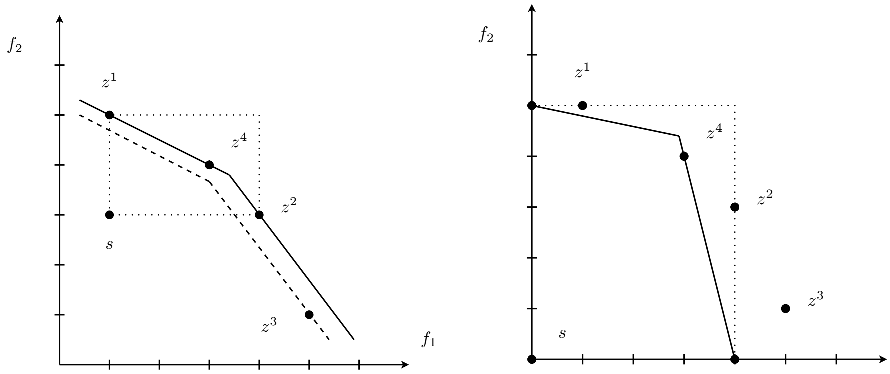
Due to the particular construction of the parameters as presented in Chapter 4, a subproblem is never infeasible, but always computes a nondominated point. Let z^* denote the nondominated point which corresponds to the optimal solution of the current subproblem. Two cases may occur: If $z^* = z^i$ or $z^* = z^{i+1}$, the results of Chapter 4 imply that the box $B(u)$ with $u := (z_1^{i+1}, z_2^i)^\top$ (that is not explicitly computed here) contains no further nondominated points and, hence, (z^i, z^{i+1}) can be labeled as investigated. This case corresponds to the situation that the subproblem is infeasible, see Line 7 in Algorithm 1. Otherwise, i.e., if z^* does not equal z^i or z^{i+1}, then the nondominated point z^* implicitly induces two new boxes that may contain further nondominated points. In this case, z^* is inserted into $\mathcal{N}$ at the respective position to maintain the desired ordering of the elements, and the two new pairs of subsequent points are labeled temporarily.

The procedure of selecting and solving subproblems is repeated until every pair of consecutive points in $\mathcal{N}$ is labeled permanently. Due to the correctness of Algorithm 4, that follows from the correctness of Algorithm 1, the complete nondominated set has been generated, then. As explained in Section 5.2, the overall number of subproblems, including the computation of the two lexicographical minima at the beginning, is given by $2|Z_N| - 1$, see also Chalmet et al. (1986), Ralphs et al. (2006). However, note that the complexity of Algorithm 4 does not only depend on the number of subproblems, but also on the complexity required to solve one subproblem. Since the original bicriteria optimization problem might be intractable, the time needed to generate Z_N may grow exponentially with the size of the considered bicriteria problem.

6.2.2 Local Ideal Points as Reference Points

In the classic formulation of the (augmented) weighted Tchebycheff method, see (2.26) and (2.30), a utopian or the ideal point is chosen as reference point. In the following, we discuss the use of local ideal points as reference points. A variation of the reference point is interesting since the parameter values derived in Chapter 4 depend on the size of the considered box, hence, in particular on the reference point that constitutes the lower bound of the respective box. If local ideal points are chosen as reference points, then, in general, larger values of ρ can be obtained. Therefore, the use of local ideal points as reference points seems to be advantageous and is included in our numerical study. Recall from Section 2.1 that a local ideal point is defined with respect to a subset of Z. In this section, it will always be defined with respect to a pair of consecutive points in the sorted list $\mathcal{N}$, i.e., for two points $z^1, z^2 \in \mathcal{N}$ the local ideal point is given by $s = (z_1^1, z_2^2)^\top$.

In order to use local ideal points as reference points in the augmented weighted



(a) Local ideal point as reference point (b) (Global) ideal point as reference point

Figure 6.1: Example in which a nondominated point z^4 cannot be generated with the augmented weighted Tchebycheff method using a local ideal point with respect to z^1 and z^2, since z^3 has a smaller level than z^4 (a); in contrast, z^4 can be generated when the (global) ideal point is used (b); the solid and dashed lines represent parts of the contour of an augmented weighted Tchebycheff norm with the origin translated to the local ideal point

Tchebycheff method, we consider the formulation

$$\min_{x\in X} \max_{i=1,\dots,m} \{w_i \, |f_i(x) - s_i|\} + \rho \sum_{j=1}^{m} |f_j(x) - s_j| \tag{6.1}$$

with $s \in \mathbb{R}^m$ a given reference point.

We call (6.1) an augmented weighted Tchebycheff problem with reference point $s \in \mathbb{R}^m$, even if it does not meet the definition in its original form for $s \neq z^U$ or $s \neq z^I$. Note that the objective function of (6.1) can be seen as a special achievement (scalarizing) function (Wierzbicki, 1980). Of course, the theoretical properties of the augmented weighted Tchebycheff method do not necessarily hold if we choose an arbitrary reference point. As stated in Steuer (1986), see also Section 2.3, the absolute values in the weighted Tchebycheff problem (2.26) and the augmented weighted Tchebycheff problem (2.30) can be dropped if the ideal or a utopian point is taken as reference point. A feasible outcome of problem (2.26) or (2.30) with the absolute values dropped is not only proven to be (weakly) nondominated, but automatically lies in the considered box. Consequently, we can discard a box from the search space when no new outcome is obtained. When a local ideal point is used as reference point and we drop the absolute values in (6.1), then a feasible outcome is also (weakly) nondominated, see, e.g., Luque et al. (2012). However, the outcome does not necessarily lie in the considered box. This is illustrated in Figure 6.1(a), where a local ideal point with respect to z^1 and z^2 is used for constructing the contour of an augmented

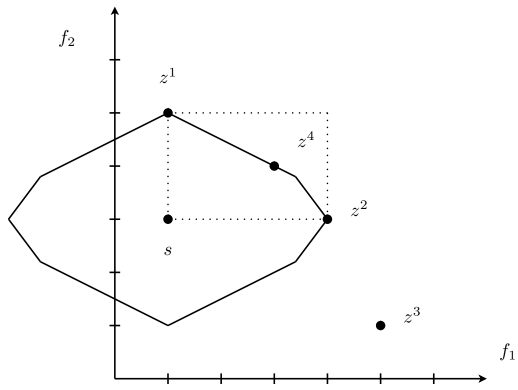


Figure 6.2: Contour of an augmented weighted Tchebycheff norm with the origin translated to the local ideal point

weighted Tchebycheff norm with the absolute values omitted. The point $z^3 \notin B(u)$ with $u := (z_1^2, z_2^1)^\top$ has a strictly smaller level than $z^4 \in B(u)$ and, thus, would be obtained when searching for new outcomes in $B(u)$. In this case, it may happen that a box $B(u)$ would be selected over and over again in Algorithm 4 without being able to generate z^4. Moreover, if no stopping criterion is active, Algorithm 4 would not terminate. This situation does not occur when the ideal point is chosen as a fixed reference point, see Figure 6.1(b), as then every feasible outcome must lie in the considered box by construction of the contour.

We apply three different approaches to enable the use of local ideal points as reference points of an augmented weighted Tchebycheff problem. One possibility consists in modeling the absolute values in (6.1) by introducing two additional variables $\mu_i \in \mathbb{R}$, $i = 1, 2$, and four additional constraints. Thereby, we obtain the linear program

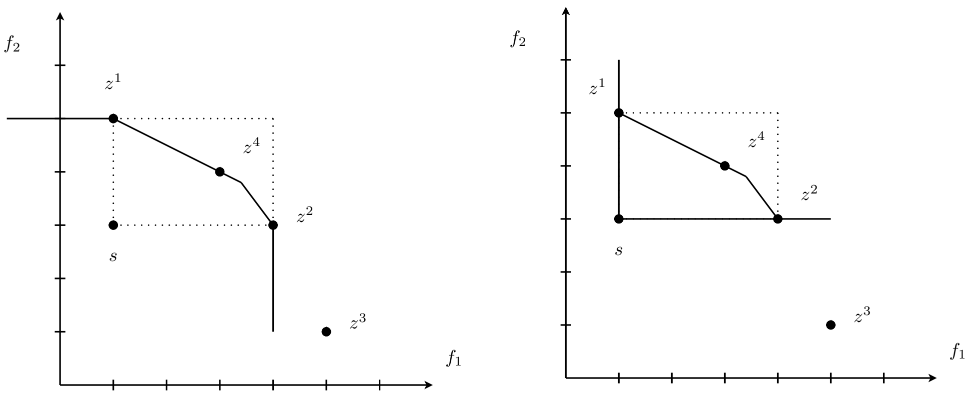
$$\begin{array}{lllll} \min & \lambda + \rho\,(\mu_1 + \mu_2) & & & \\ \text{s.t.} & \lambda & \geq & w_i \mu_i, & i = 1, 2, \\ & \mu_i & \geq & z_i - s_i, & i = 1, 2, \\ & \mu_i & \geq & -(z_i - s_i), & i = 1, 2, \\ & z & \in & Z. & \end{array} \tag{6.2}$$

An alternative approach consists in dropping the absolute values, but restricting the search to the desired box by adding either $m = 2$ constraints from above, i.e.,

$$z_1 \leq z_1^2 \quad \text{and} \quad z_2 \leq z_2^1 \tag{6.3}$$

or from below, i.e.,

$$z_1 \geq z_1^1 \quad \text{and} \quad z_2 \geq z_2^1 \tag{6.4}$$



(a) P_{ubc} : Upper bound constraints

(b) P_{lbc} : Lower bound constraints

Figure 6.3: Adding constraints such that the nondominated point z^4 'between' the two nondominated points z^1 and z^2 can be generated with the augmented weighted Tchebycheff method using the local ideal point with respect to z^1 and z^2 as reference point

to the problem

$$\begin{array}{lrcl} \min & \lambda + \rho \sum_{j=1}^{2} (z_j - s_j) & & \\ \text{s.t.} & \lambda & \geq & w_i (z_i - s_i), \quad i = 1, 2, \\ & z & \in & Z. \end{array} \tag{6.5}$$

A visualization of these three variants is given in Figures 6.2 and 6.3.

6.2.3 Computational Setup

For our computational tests we consider a bicriteria knapsack problem as a well-studied example of a discrete bicriteria optimization problem with non-negative objective values. To be consistent with problem formulation (2.1), we formulate the problem as a minimization problem with objective coefficients (costs) $C \in \mathbb{Z}_+^{2\times n}$, constraint vector (profit) $a \in \mathbb{Z}_+^n$ and minimum profit requirement $b \in \mathbb{Z}_+$:

$$\begin{array}{lrcl} \min & f(x) = Cx & & \\ \text{s.t.} & a^\top x & \geq & b, \\ & x_l & \in & \{0,1\}, \quad l = 1, \dots, n. \end{array} \tag{6.6}$$

Note that the bicriteria knapsack problem has also been used as a test problem, e.g., in Sayın and Kouvelis (2005) and Ralphs et al. (2006). Similar to these references, we do not aim at outperforming specialized algorithms for the determination of the nondominated set of the bicriteria knapsack problem with respect to computational times. Algorithm 4 is rather used as a general framework to compare the performance of different variants of scalarizations. For these reasons, a standard integer linear programming solver is sufficient for the solution of the respective subproblems.

The test problems are generated using the approach and the code of Sayın and Kouvelis (2005). In this approach, the elements of $C \in \mathbb{Z}_+^{2\times n}$ and $a \in \mathbb{Z}_+^n$ are randomly drawn from the interval $[1, 1000]$, following a uniform distribution. The minimum profit $b \in \mathbb{Z}_+$ of a feasible knapsack is set to $b := 0.5 \cdot \sum_{i=1}^n a_i$, rounded to the nearest integer. We consider five different problem sizes $n \in \{50, 75, 100, 125, 150\}$, where n denotes the total number of available items. For every problem size n, 30 different instances are evaluated. Note that the instances belonging to one problem size typically all have a different number of nondominated points. As already stated above, the parameters of the Tchebycheff scalarizations are set according to the formulas given in Chapter 4. The scalar $\bar{\eta}$ in the formulas of Table 4.2 is set to 0.1 by default. Moreover, if not stated otherwise, the ideal point is taken as reference point.

Two series of experiments are presented. In a first test series, we compare the augmented weighted Tchebycheff method with adaptively chosen parameters (AAWT) to the augmented weighted Tchebycheff method with a priori chosen values of ρ, where we set ρ to $10^{-2}, 10^{-3}$ and 10^{-4}, respectively, as proposed by Steuer (1986). We denote the resulting problems by $P_{0.01}$, $P_{0.001}$ and $P_{0.0001}$, respectively. Furthermore we compare the adaptive augmented weighted Tchebycheff method to an adaptive augmented ε-constraint method (AEC) of the form (4.36) with $k = 1$. The parameter ρ is set according to (4.38) with $\bar{\eta} = 0.1$.

The two-stage or lexicographic weighted Tchebycheff method (TS) is used as reference method, i.e., we compare all methods with TS in terms of the number of determined nondominated points and computational time. Note that we exactly compute the complete nondominated set with the help of the two-stage approach. We confirmed this additionally with the help of a dynamic programming algorithm, see Klamroth and Wiecek (2000). We also verified that whenever a lower number of points was found by the respective method, it is always a strict subset of the nondominated set.

In a second test series, we investigate the impact of using local ideal points as reference points in the augmented weighted Tchebycheff method. In a first step, we compare the different reformulations of the absolute values described in Section 6.2.2 with respect to CPU time. In a second step, we test an improved approach that still uses local ideal points but replaces the absolute values only when needed, i.e., whenever nondominated points outside the current box may potentially be generated. Dependent on the parameter $\bar{\eta}$, we compare CPU time and average values of ρ.

For better comparison we always state absolute values for TS and relative devi-

n	TS	AAWT	$P_{0.01}$	$P_{0.001}$	$P_{0.0001}$	AEC
50	46.43	0	-0.0622	-0.0066	0	0
75	91.57	0	-0.0695	-0.0055	0	0
100	150.60	0	-0.0748	-0.0051	-0.0004	0
125	225.23	0	-0.0770	-0.0056	-0.0005	0
150	341.77	0	-0.0695	-0.0054	-0.0004	-0.0001

Table 6.1: Relative deviation of the number of nondominated points found by different augmented weighted Tchebycheff methods with respect to TS (absolute values)

ations from the respective value of TS for all other methods. All figures represent averages over $K = 30$ instances, where the average relative deviations are computed as follows: Let $v_{n,k,M}$ be a considered value (e.g., CPU time) of problem size n, instance k and method M different from TS. Then

$$V_{n,M} := \frac{1}{K}\sum_{k=1}^{K}\frac{v_{n,k,M} - v_{n,k,TS}}{v_{n,k,TS}}.$$

We additionally indicate standard deviations whenever useful in order to help to understand the relevance of the observed differences in CPU time or the values of ρ. The standard deviations are calculated as

$$\sigma_{n,M} := \sqrt{\frac{1}{K}\sum_{k=1}^{K}\left(\frac{v_{n,k,M} - v_{n,k,TS}}{v_{n,k,TS}} - V_{n,M}\right)^2}$$

for all problem sizes n and methods M.

The computational platform for our study is a compute server with 4x Intel Xeon E7540 CPUs (2.0 GHz) and 128 GB of memory. Algorithm 4 is implemented in C++, and as subproblem solver we use CPLEX 11.2.0. We turned off the option of CPLEX to parallelize due to much longer computational times for the small instances.

6.2.4 Computational Results

A direct comparison of the different scalarization approaches is given in Tables 6.1 and 6.2. Thereby, Table 6.1 shows the average number of nondominated points found, and Table 6.2 shows the average CPU time needed by the respective method.

From Table 6.1 we see that the adaptive augmented weighted Tchebycheff method reliably generates the complete nondominated set (deviation of 0% from TS). However, this does not hold for the methods in which ρ is fixed a priori: For $\rho = 0.01$ and $\rho = 0.001$, for each problem size, a certain percentage of nondominated points is missed. For example, for $\rho = 0.01$ and $n = 50$, on average 6.2% of the nondominated

n		TS	AAWT	$P_{0.01}$	$P_{0.001}$	$P_{0.0001}$	ε-constr
50	CPU	1.668	**-0.433**	-0.458	-0.435	**-0.435**	**-0.450**
	σ	0.865	0.036	0.045	0.037	0.037	0.054
75	CPU	5.651	**-0.471**	-0.506	-0.473	**-0.472**	**-0.475**
	σ	3.278	0.027	0.024	0.027	0.027	0.048
100	CPU	13.889	**-0.490**	-0.526	-0.491	-0.491	**-0.488**
	σ	6.182	0.025	0.030	0.025	0.026	0.054
125	CPU	30.734	**-0.497**	-0.537	-0.498	-0.497	**-0.485**
	σ	15.880	0.024	0.031	0.025	0.024	0.064
150	CPU	67.227	**-0.496**	-0.539	-0.499	-0.499	-0.511
	σ	26.807	0.024	0.028	0.024	0.024	0.048

Table 6.2: Absolute CPU time of TS and relative deviation of CPU time (with respect to TS) of different augmented weighted Tchebycheff methods. The second row contains the respective standard deviations. Highlighted CPU times correspond to the case in which the complete nondominated set has been generated.

points are missed. When setting $\rho = 0.0001$, for problem sizes $n = 50$ and $n = 75$, the entire set of nondominated points is found, whereas for larger problem sizes also some nondominated points are missed. This shows the difficulty when using a fixed value for the parameter ρ, as already discussed in Section 4.1: Depending on the data of the given problem, a fixed value of ρ may not be appropriate (i.e., too large) for the generation of all nondominated points. In contrast, the complete nondominated set is computed with an adaptive calculation of ρ based on the given problem data.

We note for the augmented ε-constraint method, that, for problem size $n = 150$, 0.01% of the nondominated points is missed, which corresponds to one missing solution in one test instance. As the ε-constraint method uses an adaptive parameter scheme, this solution should theoretically have been found and is probably missed for numerical reasons. It could be generated when an ε-constraint was set on the first objective. This, in turn, led to much higher computational times of this method.

In Table 6.2 we show the respective average CPU times and standard deviations. For example, for $n = 50$, TS requires on average 1.668 seconds with a standard deviation of 0.865 seconds. The adaptive augmented weighted Tchebycheff method requires on average 43.3% less CPU time than the two-stage approach, the respective standard deviation amounts to 3.6%. Analyzing computational time in Table 6.2 in general, it can be observed that for all problem sizes the methods with augmentation term require at least 43% less CPU time than the two-stage method. This observation reflects the fact that in TS, two optimization problems have to be solved for each

n	TS	σ	P_{avc}	σ	P_{ubc}	σ	P_{lbc}	σ
50	1.668	0.8654	0.073	0.1216	-0.281	0.0455	-0.048	0.1268
75	5.651	3.2778	0.458	0.2000	-0.246	0.0476	0.301	0.1724
100	13.889	6.1824	0.747	0.1939	-0.225	0.0421	0.568	0.1520
125	30.734	15.8798	0.951	0.2039	-0.210	0.0334	0.754	0.1707
150	67.227	26.8068	1.089	0.1226	-0.177	0.0270	0.850	0.1100

Table 6.3: Absolute CPU time of TS and relative deviation of CPU time (with respect to TS) of different augmented weighted Tchebycheff methods. Additionally, the respective standard deviations are given.

feasible point, resulting in considerably longer computational times. Note that we did not pass the solution from the first stage as a starting solution to the second stage as it took more time to solve the specific problem in practice.

Comparing computational times among the different methods with augmentation term is only meaningful for those methods that find all nondominated points. The corresponding entries are highlighted in Table 6.2. We observe that the solution times of all augmented methods, which generate the entire nondominated set, differ only slightly. Hence, the adaptive update of the parameters does not have a negative impact on the computational time. Comparing the augmented weighted Tchebycheff method with the augmented ε-constraint method, we state that no method outperforms the other. Since only one additional constraint is introduced in the augmented ε-constraint method in comparison to two additional constraints in the case of the augmented weighted Tchebycheff method, the solution of the augmented ε-constraint method might take less computational time. However, we could not observe such an effect in this numerical study.

In the second test series we address the question whether we can obtain better (i.e., larger) values for ρ for AAWT without impairing the CPU times recorded in Table 6.2. Therefore, we replace the ideal point by local ideal points, individually chosen for each subproblem. As explained in Section 6.2.2, this implies the necessity to take the absolute values in problem (6.1) into account. In the method P_{avc} we include the absolute values explicitly by introducing additional variables and constraints as specified in (6.2). In the methods P_{ubc} and P_{lbc} we omit the absolute values but add upper box constraints (6.3) and lower box constraints (6.4), respectively. Since all three methods are equivalent in the sense that they reliably generate all nondominated points, we only report average CPU times in Table 6.3 and omit the (equal) numbers of determined nondominated points. As before, we show the deviations in CPU time of the three formulations as compared to the two-stage ap-

			$\bar{\eta} = 0.1$				$\bar{\eta} = 0.9$			
n	TS	σ	AAWT	σ	P_{comb}	σ	AAWT	σ	P_{comb}	σ
50	1.668	0.865	-0.433	0.036	-0.416	0.036	-0.432	0.041	-0.446	0.039
75	5.651	3.278	-0.471	0.027	-0.430	0.035	-0.471	0.027	-0.479	0.026
100	13.889	6.182	-0.490	0.025	-0.423	0.031	-0.490	0.026	-0.494	0.025
125	30.734	15.880	-0.497	0.024	-0.427	0.028	-0.496	0.023	-0.501	0.024
150	67.227	26.807	-0.496	0.024	-0.408	0.027	-0.498	0.025	-0.495	0.024

Table 6.4: Absolute CPU time of TS and relative deviation of CPU time (with respect to TS) of different augmented weighted Tchebycheff methods with different scaling of ρ. Additionally, the respective standard deviations are given.

proach. Only method P_{ubc} performs better than TS for all problem sizes. The other two methods even consumed more CPU time than TS for all problem sizes except P_{lbc} for $n = 50$. But also the CPU times of P_{ubc}, compared to the CPU times of AAWT in Table 6.2, show a clear impairment (gain with respect to TS is less than 30%). We conclude that the additional variables and constraints seem to make the augmented problem more difficult to solve.

In order to avoid as much as possible of the additional computational burden induced by the linearizations of the absolute values in (6.1), we additionally implemented an alternative approach, where these reformulations are only used when necessary. Indeed, the absolute values in (6.1) are only needed if there exists a nondominated point that is located outside the considered box and that minimizes the augmented weighted Tchebycheff scalarization with absolute values omitted (cf. Figure 6.1(a)). Instead of introducing additional variables and/or constraints in every subproblem, we may also omit the absolute values, i.e., solve problem (6.5), and check afterwards whether the solution satisfies the box constraints. If the solution lies in the considered box, we can proceed as usual and turn to the next subproblem. If, however, the solution lies outside the box selected box, we insert the solution (if it is not already contained in $\mathcal{N}$). Then we repeat the search for nondominated points in the same box, but this time by solving problem P_{ubc}, i.e., by explicitly including the box constraints in the problem formulation.

The average CPU times for this alternative approach, denoted by P_{comb}, are given in Table 6.4 in relation to the two-stage approach. For better comparison, Table 6.4 also contains the data of method AAWT using the global ideal point. For TS, the global ideal point is used. Additionally to the default setting $\bar{\eta} = 0.1$, we test $\bar{\eta} = 0.9$.

For $\bar{\eta} = 0.1$, AAWT clearly outperforms P_{comb} for all problem sizes. However, for $\bar{\eta} = 0.9$, P_{comb} performs as good as AAWT. This can be explained as follows: The

	$\bar{\eta} = 0.1$				$\bar{\eta} = 0.9$			
n	AAWT	σ	P_{comb}	σ	AAWT	σ	P_{comb}	σ
50	**0.00054**	0.00011	0.01040	0.00396	0.00006	0.00001	**0.00112**	0.00039
75	**0.00039**	0.00007	0.01398	0.00394	0.00004	0.00001	**0.00150**	0.00040
100	**0.00030**	0.00004	0.01810	0.00410	0.00003	0.00001	**0.00192**	0.00039
125	**0.00024**	0.00004	0.02033	0.00379	0.00003	0.00001	**0.00217**	0.00037
150	**0.00019**	0.00003	0.02401	0.00360	0.00002	0.00000	**0.00254**	0.00035

Table 6.5: Comparison of average values of ρ (with standard deviation σ)

larger $\bar{\eta}$ is, the less likely it is that solutions of (6.5) lie outside the current box, since then the contour of the corresponding augmented weighted Tchebycheff norm is only slightly lifted as compared to the weighted Tchebycheff norm without the augmentation term. Having fewer solutions outside of the considered box implies that fewer of the (computationally more expensive) problems with additional box constraints have to be solved. In the limit, i.e., if no solution lies outside the corresponding box, a computational time similar to AAWT is obtained.

The advantage when using local ideal points instead of the (global) ideal point is that we can expect that, on average, larger values of ρ in the augmented weighted Tchebycheff subproblems are obtained. This can be observed in Table 6.5, where values of ρ (averaged over all instances and over all subproblems for each problem size) for AAWT and P_{comb} with $\bar{\eta} = 0.1$ and $\bar{\eta} = 0.9$ are given. It is interesting to note that even for P_{comb} with $\bar{\eta} = 0.9$, larger average values for ρ are obtained than for AAWT with $\bar{\eta} = 0.1$ (the respective columns are highlighted).

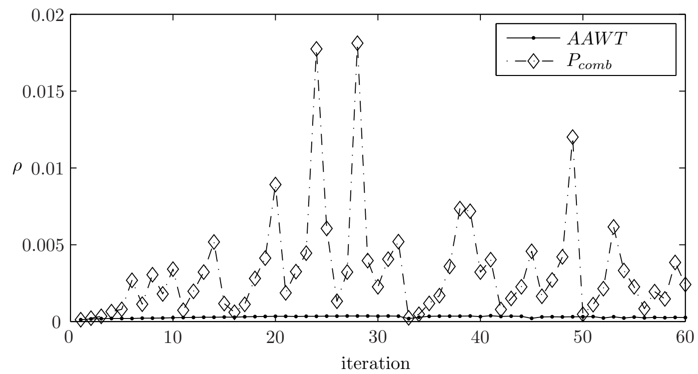


Figure 6.4: Exemplary development of ρ for augmented weighted Tchebycheff method with local and global ideal point and the same parameter $\bar{\eta} = 0.1$ (instance $n = 50$, $|Z_N| = 32$)

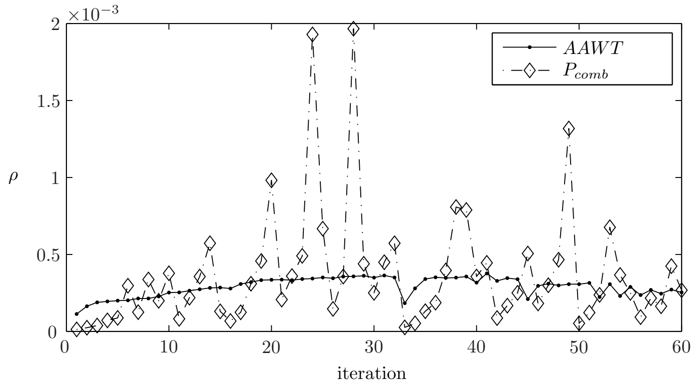


Figure 6.5: Exemplary development of ρ for augmented weighted Tchebycheff method with local and global ideal point and different parameters $\bar{\eta}$: we set $\bar{\eta} = 0.1$ for (AAWT) and $\bar{\eta} = 0.9$ for (P_{comb}) (instance $n = 50$, $|Z_N| = 32$)

Taking into account that the computational times are nearly the same for both variants (see Table 6.4), this indicates that larger average values for ρ, and, thus, a probably numerically more stable method, are obtained with local ideal points.

Figures 6.4 and 6.5 show an exemplary development of ρ for one selected instance of problem size $n = 50$. The value of ρ is plotted for the consecutively solved subproblems. In Figure 6.4 we see that, for all subproblems, larger values of ρ can be achieved when using local ideal points (P_{comb}) instead of global ideal points (AAWT) and the same choice of $\bar{\eta}$ ($= 0.1$) for both methods. However, the larger values of ρ imply higher computational times, see Table 6.4. For both methods shown in Figure 6.5, computational times are nearly equal. The values of ρ achieved with the local version (P_{comb}) with $\bar{\eta} = 0.9$ are partially higher, partially lower than those of the global version (AAWT) with small $\bar{\eta} = 0.1$, but on average higher, see Table 6.5.

6.3 Tricriteria Problems

In this section, we generate complete representations of the nondominated set of discrete tricriteria optimization problems. We apply Algorithm 2, introduced in Section 5.3, that is based on the v-split. Recall that by this split the generation of redundant boxes is avoided. Only so-called quasi non-redundant boxes occur in the situation in which nondominated points with equal components exist. Besides Algorithm 2 we also evaluate Algorithm 3, see Section 5.5, in which additionally the generation of quasi non-redundant boxes is suppressed.

The aim of our numerical study is twofold. On the one hand, we validate the formulas derived in Chapter 4 for an adaptive parameter scheme, in particular for the augmented weighted Tchebycheff method. On the other hand, we verify the upper bounds on the number of subproblems derived in Chapter 5, i.e., $3|Z_N|-2$ in the case of the (augmented) weighted Tchebycheff method and $2|Z_N|-1$ in the case of the ε-constraint method, in which, as discussed in Section 5.4, one box per iteration can be discarded immediately. Note that the (augmented) weighted Tchebycheff method is used as an example of an arbitrary scalarization which is suitable for discrete problems.

6.3.1 Computational Setup

For our tests we use five instances of a tricriteria multidimensional knapsack problem, i.e., a zero-one knapsack problem with three objectives and three constraints. The considered instances have already been employed for numerical experiments in Laumanns et al. (2006), Laumanns et al. (2005) and Ozlen et al. (2014), wherefore we regard them as a good benchmark. The five instances correspond to five different numbers of (knapsack) items $n = 10, 20, 30, 40, 50$. The respective cardinality of the nondominated set is $9, 61, 195, 389$ and 1048, as reported in Laumanns et al. (2005) and Ozlen et al. (2014) and verified by our algorithms. Note that we generated and saved the nondominated set of every instance once. For all methods presented in the following we always compare the respective generated representation with the saved nondominated set in order to verify that the complete nondominated set is computed correctly.

For all algorithms we use the adaptive parameter scheme from Chapter 4. As the parameters are constructed such that a specified box is investigated for new nondominated points, no additional constraints need to be set in the scalarizations which might force the outcome to lie in the considered box. Instead, we check after the solution of the scalarization (in case of feasiblity) whether the computed point lies in the considered box, i.e., whether all of its components are strictly smaller than the upper bound vector of the current box. In this case the generated point must be a new (weakly) nondominated point. Note that when the two-stage formulation is used, then a second-stage problem is solved in the following such that the generation of a nondominated point is assured. Otherwise, i.e., if the generated point does not lie in the considered box, we define the subproblem to be infeasible and remove the corresponding box from the decomposition.

In the first test, we compare Algorithm 2 and Algorithm 3, each of which in combination with a weighted Tchebycheff and an ε-constraint method. Thereby, the weighted Tchebycheff and the ε-constraint method are both tested in an augmented and a two-stage formulation. The parameters of the augmented weighted Tchebycheff method are computed according to the formulas given in Section 4.2, Table 4.3 with $\bar{\eta} = 0.1$. The two-stage weighted Tchebycheff method employs Lemma 4.3 for the computation of the weights. If the ε-constraint method is used, we set the right-hand side vector ε to the upper bound vector of the considered box. Note that throughout this section we minimize all ε-constraint variants with respect to the first component, without loss of generality. Therefore, no constraint on the first objective is required. The parameter ρ in the augmented ε-constraint method as described in Section 4.4 is set according to formula (4.38) with $k = 1$ and $\bar{\eta} = 0.1$.

In the second test, we compare our algorithms to three recent algorithms for generating complete representations for discrete multicriteria optimization problems with finite nondominated set, see Section 3.3 for a detailed description. These comprise the second algorithm stated in Lokman and Köksalan (2013), the approach of Kirlik and Sayın (2014) and the method of Ozlen et al. (2014). All three methods employ an ε-constraint scalarization, however, in three different variants: Lokman and Köksalan (2013) use an augmented, Kirlik and Sayın (2014) a two-stage and Ozlen et al. (2014) a lexicographic ε-constraint method. Note that Lokman and Köksalan (2013) state that the augmentation parameter has to be chosen sufficiently small, but do not specify how they select the parameter in their numerical study.

In Section 3.3 further methods to compute complete representations of discrete multicriteria optimization problems were presented. The method of Sylva and Crema (2004) is not evaluated as it was shown to be outperformed by the approaches of Lokman and Köksalan (2013) and Kirlik and Sayın (2014). We neither incorporate the algorithm of Tenfelde-Podehl (2003), as it was shown to be outperformed by the method of Laumanns et al. (2005) in the computational experiments of Przybylski et al. (2009) due to the large memory requirement in the second phase. We neither test the algorithm of Laumanns et al. (2005) as it was shown to be outperformed considerably by the methods of Kirlik and Sayın (2014) and Ozlen et al. (2014) in their respective numerical studies. However, so far, no comparative numerical study between the methods of Ozlen et al. (2014), Lokman and Köksalan (2013) and Kirlik and Sayın (2014) has been undertaken. Finally, we neither include two phase methods as proposed in Przybylski et al. (2010a) into our experiments, as, different from the general methods considered here, the latter explicitly exploit the particular combinatorial structure of the underlying problem.

As observed in the numerical study in the bicriteria case, it makes a difference for computational time whether an augmented or a two-stage approach is used. More precisely, the results in Section 6.2.3 show that the augmented weighted Tchebycheff method requires roughly half of the computational time that is needed by the two-stage weighted Tchebycheff method. This is caused by the fact that in the latter two integer problems are solved in every subproblem in which the first stage yields a feasible solution. In contrast, when an augmented method is used, only one integer problem per subproblem is solved. In order to make the comparison in our numerical study as fair as possible, all methods should use the same scalarization type with the same parameter scheme to solve the subproblems. Hence, we implement and test all algorithms with both, a two-stage and an augmented formulation, where the augmentation parameter ρ is set according to (4.38) with $\bar{\eta} = 0.1$ in all compared algorithms. Note, however, that thereby we modify the original algorithms of Lokman and Köksalan (2013), Kirlik and Sayın (2014) and Ozlen et al. (2014).

All algorithms are (re)implemented in MATLAB R2013a and call IBM ILOG CPLEX Optimization Studio Version 12.5 to solve the subproblems. Note that we chose an implementation in MATLAB for compatibility with the implementation for continuous problems, see Chapter 7. However, one should keep in mind that an implementation in C probably yields much better results concerning (absolute) CPU time.

6.3.2 Computational Results

Comparison of the v-Split Algorithm and an Algorithmic Variant

First, we test Algorithm 2 and Algorithm 3 both in combination with a weighted Tchebycheff method (WT) and an ε-constraint method (EC). In particular, we are interested in the question whether the upper bounds on the number of subproblems $3|Z_N|-2$ (WT) and $2|Z_N|-1$ (EC), which were derived in Chapter 5, can be validated numerically.

In Table 6.6, the results of the four algorithmic variants are reported. The CPU times (in seconds) are averaged over three independent runs. The corresponding standard deviations are given in Table 6.7. Since the number of subproblems solved is the same in all of the three independent runs, no standard deviations are given with respect to the number of subproblems. Note that, as in the bicriteria study, a subproblem refers to the solution of the selected scalarization. In the two-stage approach, a subproblem comprises both stages, i.e., if the solution of the first stage is feasible, then the second stage problem is solved without being counted separately. To ease the comparison of the number of subproblems with the values $3|Z_N| - 2$ and $2|Z_N| - 1$, respectively, we indicate these values in parentheses in the second column of Table 6.6.

Consider first the number of subproblems solved in Algorithm 2 in combination with a weighted Tchebycheff scalarization (WT) and an ε-constraint scalarization (EC). From Table 6.6 we see that Algorithm 2 (WT) requires exactly $3|Z_N| - 2$

n	$\lvert Z_N \rvert$		Algo 2 (WT)		Algo 3 (WT)		Algo 2 (EC)		Algo 3 (EC)	
			CPU	#SP	CPU	#SP	CPU	#SP	CPU	#SP
10	9	TS	10.03	25	9.73	25	7.97	17	7.96	17
	(25/17)*	A	7.81		7.76		6.09		6.08	
20	61	TS	56.42	181	55.33	177	43.29	121	42.36	117
	(181/121)*	A	42.72		41.87		30.02		29.20	
30	195	TS	213.31	583	209.02	568	163.15	389	159.13	371
	(583/389)*	A	163.29		159.02		114.39		110.35	372
40	389	TS	464.47	1165	453.70	1127	361.74	777	355.50	742
	(1165/777)*	A	361.01		349.53		257.64		251.62	
50	1048	TS	1552.56	3142	1498.61	2985	1369.89	2095	1340.13	1924
	(3142/2095)*	A	1174.90		1122.74	2980	1012.15		998.60	1925

Table 6.6: Average CPU times (in seconds) and number of subproblems solved by Algorithm 2 and Algorithm 3 both in combination with a weighted Tchebycheff method (WT) and an ε-constraint method (EC). Each scalarization is evaluated in a two-stage (TS) and an augmented (A) formulation, respectively. In the second column, additional to $|Z_N|$, the theoretical upper bounds $3|Z_N| - 2$ (WT) and $2|Z_N| - 1$ (EC) are given in parentheses $()^\star$ for better comparison.

subproblems and Algorithm 2 (EC) exactly $2|Z_N| - 1$ subproblems for all problem sizes and for both formulations, i.e., for a two-stage (TS) and an augmented (A) formulation. Hence, the predicted upper bound on the number of subproblems is met precisely.

Regarding the number of subproblems for Algorithm 3 we can expect that less subproblems are solved in comparison to Algorithm 2, since Algorithm 3 suppresses the generation of quasi non-redundant boxes. Indeed, in all instances besides $n = 10$, less subproblems are solved by Algorithm 3 compared to Algorithm 2 (with the same scalarization used). Note that the nondominated points of instance $n = 10$ satisfy Assumption 5.6 (1), i.e., are in general position, wherefore no quasi non-redundant boxes occur and, consequently, the number of subproblems solved with Algorithm 2 and Algorithm 3 is the same. However, in all other instances, a saving with respect to the number of subproblems solved is observed for Algorithm 3. Moreover, we notice that the number of subproblems in Algorithm 3 differs in some instances for the two-stage and the augmented formulation, see $n = 50$ for Algorithm 3 (WT) and $n = 30$ and $n = 50$ for Algorithm 3 (EC). Indeed, as demonstrated in Example 5.32, the number of iterations of Algorithm 3 might vary due to the order in which nondominated points are computed and inserted. Since typically the nondominated points are obtained in a different order by a two-stage or augmented formulation, the number of subproblems can vary slightly depending on the formulation used. In contrast, the number of subproblems that are solved in Algorithm 2 does not depend on the order in which the nondominated points are inserted.

While Algorithm 3 requires less subproblems than Algorithm 2 in all instances of our numerical tests, no theoretical upper bound on the number of subproblems is available for Algorithm 3. More precisely, the fact that the number of additional boxes, that are created in each iteration, is constantly two as in Algorithm 2 (WT) or constantly one as in Algorithm 2 (EC) does not hold for Algorithm 3. As an example, consider Algorithm 3 (EC) in variant (TS) for $n = 50$. According to Table 6.6, 1924 subproblems are solved in total. Besides the 1048 subproblems that contribute a nondominated point to the representation, 876 additional subproblems are solved. We recorded the additional boxes created in each of the 1048 iteration, in which a new nondominated point is found. In 188 of these iterations the number of boxes remains the same. In 833 iterations one additional box and in 23 iterations two additional boxes are created. In four iterations, the number of boxes decreases by one. Hence, together with the initial box, we obtain 876 additional boxes. In comparison, Algorithm 2 (EC) generates constantly one additional box in each of the 1048 iterations, in which a new nondominated point is found that does not equal the ideal point in the second or third component. As two such points exist for the considered instance $n = 50$, Algorithm 2 (EC) creates one additional box in 1046 iterations. From this example we see that also Lemma 5.23 (and the corresponding result when the ε-constraint method is used) is confirmed numerically for Algorithm 2, but no corresponding result can be expected for Algorithm 3.

Regarding computational times in Table 6.6 we observe that Algorithm 3 requires less CPU time than Algorithm 2 (with the same scalarization used) in all instances

n	$\lvert Z_N \rvert$		Algo 2 (WT)	Algo 3 (WT)	Algo 2 (EC)	Algo 3 (EC)
10	9	TS	0.05	0.04	0.04	0.04
		A	0.03	0.05	0.06	0.06
20	61	TS	0.50	0.34	0.27	0.28
		A	0.18	0.25	0.17	0.22
30	195	TS	1.39	1.50	1.11	1.00
		A	0.94	0.92	0.72	0.61
40	389	TS	2.60	2.37	1.72	1.67
		A	1.62	1.68	1.18	1.16
50	1048	TS	7.31	6.98	4.80	5.96
		A	5.16	5.08	4.23	4.23

Table 6.7: Standard deviations of CPU times stated in Table 6.6, which are averaged over three independent runs

in which less subproblems are solved. Moreover, all variants using (EC) are considerably faster than the variants using (WT) as the former solve about one third less subproblems compared to the latter. However, the savings with respect to computational time are not proportional to the savings with respect to the number of subproblems, in general. For example, for $n = 50$, Algorithm 3 (EC) solves 1924 subproblems in the two-stage formulation. In comparison to Algorithm 3 (WT), which requires 2985 subproblems, a saving of 36% is obtained. However, the corresponding computational times amount to 1340.13 and 1498.61 seconds, respectively. Hence, the saving of variant (EC) with respect to CPU time is only approximately 10%. The explanation for this observation can be found when the algorithmic differences as discussed in Section 5.4 are taken into account. Recall that in order to achieve a saving with respect to the number of subproblems when an ε-constraint method is used, the box at the beginning of each iteration cannot be selected arbitrarily, but a box which has no neighbor with respect to the first component in the current decomposition must be identified. This causes an additional computational effort in comparison to all variants employing (WT). Summarizing, Algorithm 3 (EC) performs best among all four variants compared in Table 6.6. Algorithm 2 (EC) consumes only slightly more computational time, followed by Algorithm 3 (WT) and Algorithm 2 (WT).

Finally, we analyze the computational time of the two-stage versus the augmented scalarizations in all variants in Table 6.6. We observe that in all instances smaller CPU times are recorded with the augmented scalarization. More precisely, with the augmented formulation, between 20% and 31%, on average 26% less computational time is required than with the two-stage formulation. Recall that in the two-stage method, a second stage problem is solved for every feasible point in the first stage, which increases the CPU time compared to the augmented approach. We conclude that for the tested instances, Algorithm 3 in combination with an augmented ε-constraint method performs best with respect to computational time.

Comparison of Three Recent Algorithms to the New Algorithm

In a second study, we compare Algorithm 3 (EC) with reimplementations of the recent methods of Lokman and Köksalan (LK), Kirlik and Sayın (KS) and Ozlen, Burton and MacRae (OBM). Note that the algorithm of Lokman and Köksalan (2013) is the only of the tested algorithms that is formulated for problems in maximization format. For the sake of simplicity, we implement it for minimization problems. Moreover, as recommended in Lokman and Köksalan (2013), we keep the list of current nondominated points sorted, as, thereby, better computational times are obtained. As described in Section 3.3, the methods of Lokman and Köksalan (2013) and Ozlen

n	$\lvert Z_N \rvert$		LK		KS		OBM		Algo 3 (EC)	
			CPU	#SP	CPU	#SP	CPU	#SP	CPU	#SP
10	9	TS	9.48	20	8.50	17	8.85	19	7.88	17
	$(17)^\star$	A	6.67		6.07		6.46		6.05	
20	61	TS	53.04	127	50.08	115	48.50	117	42.06	117
	$(121)^\star$	A	31.76	128	30.26		28.83		28.94	
30	195	TS	267.88	468	242.42	373	197.05	375	158.04	371
	$(389)^\star$	A	159.12	464	155.89	372	110.33	374	109.39	372
40	389	TS	657.58	852	701.95	739	430.84	741	351.63	742
	$(777)^\star$	A	445.07		516.15	738	246.68	740	248.79	
50	1048	TS	4772.89	2193	4174.48	1913	1533.93	1915	1326.12	1924
	$(2095)^\star$	A	4129.47	2200	3603.67	1914	945.35	1916	987.52	1925

Table 6.8: Average CPU times (in seconds) and number of subproblems solved by three state of the art algorithms and Algorithm 3 (EC). Each scalarization is evaluated in a two-stage (TS) and an augmented (A) formulation, respectively. In the second column, additional to $|Z_N|$, the value of $2|Z_N| - 1$ is given in parenthesis $()^\star$ for better comparison.

et al. (2014) both save the bounds, i.e., the right-hand side vectors, of previously solved subproblems as well as the corresponding results, i.e., a (nondominated) point or a value indicating infeasibility. Before solving a subproblem, the list of bounds is scanned to find a so-called relaxation. If a relaxed problem exists and it is either infeasible or the saved point is feasible for the current subproblem, then the current subproblem does not need to be solved since the solution of the relaxation is also valid for the considered subproblem. In this case, the bounds of the current subproblem should not be saved, as they do not contribute new information and, clearly, the shorter the list of bounds is, the better computational times can be expected. In the implementation of (KS) we change a detail with respect to the pseudocode given in Kirlik and Sayın (2014). When a new nondominated point is generated, all cells of the decomposition are checked twice in Kirlik and Sayın (2014): first, to identify the cells to be split, secondly, to remove cells that can not contain further nondominated points. We combine both checks, which are performed within two independent procedures in the original pseudocode, into one by removing cells that can not contain further nondominated points immediately after or during the split. In our implementation, this slight modification led to a huge saving of computational time.

The CPU times and the number of subproblems solved by all methods and for all instances are given in Table 6.8. Again, the given CPU times are averaged over three independent runs. The corresponding standard deviations are given in Table 6.9. For a better comparison the value of $2|Z_N| - 1$ is indicated in parentheses in the second column of Table 6.8. Note that, as in the previous study, all methods generate the complete nondominated set correctly.

Regarding the number of subproblems solved, we observe that (KS) generates a complete representation within the lowest number of subproblems among all compared methods in almost all instances. Only for $n = 30$ and the two-stage formulation, Algorithm 3 (EC) requires two subproblems less than (KS). Methods (OBM) and Algorithm 3 (EC) solve both only very few additional subproblems in comparison to (KS). While (OBM) requires exactly two subproblems more than (KS) in all tested instances and for all tested formulations, Algorithm 3 (EC) requires 11 subproblems more for $n = 50$, but, on the other hand, performs as good as (KS) or even better for $n = 10$ and $n = 30$. Method (LK) requires the largest number of subproblems in all instances. While methods (OBM), (KS) and Algorithm 3 (EC), except (OBM) for $n = 10$, solve at most $2|Z_N| - 1$ subproblems, (LK) exceeds this bound in all instances. These results go in line with the results of Lokman and Köksalan (2013), who state that they solved on average 2.08 subproblems per nondominated point in their numerical study for a classic (one-dimensional) tricriteria knapsack problem. Our results also coincide with the results of Kirlik and Sayın (2014), who state that they required on average 1.97 and at most 1.99 subproblems per nondominated point with their algorithm when it was applied to a classic tricriteria knapsack problem. In our study, (KS) even performs better. In the worst case ($n = 30$) less than 1.92 subproblems per nondominated point are solved. The results of (OBM) can be compared directly with the results reported in Ozlen et al. (2014), as they solve the same problem with the same instances. In their numerical study, $46, 333, 1204, 2357$ and 6001 subproblems are solved for $n = 10, \ldots, 50$, respectively. Interestingly, we obtain a considerably smaller number of subproblems with our reimplementation in all instances. A possible reason for this mismatch might be the scalarization used. While we apply (OBM) in combination with a two-stage and an augmented scalarization, a lexicographic ε-constraint scalarization is used in Ozlen et al. (2014), for which the subproblems might have been counted in a different way.

Considering CPU times, we obtain a slightly different picture. For the small instance $n = 10$, the CPU times of all methods are quite close, with Algorithm 3 (EC) slightly leading. For all other problem sizes, the best CPU times are clearly obtained by Algorithm 3 (EC) and (OBM). Thereby, when the augmented formulation is used, in all instances besides $n = 50$ both methods perform almost equally well. For $n = 50$, (OBM) consumes less CPU time than Algorithm 3 (EC). When the two-stage formulation is used, Algorithm 3 (EC) outperforms (OBM) for all problem sizes.

Both other methods, i.e., (LK) and (KS) require considerably more CPU time than

n	$\lvert Z_N \rvert$		LK	KS	OBM	Algo 3 (EC)
10	9	TS	0.02	0.02	0.02	0.01
		A	0.01	0.01	0.01	0.01
20	61	TS	0.06	0.06	0.26	0.07
		A	0.04	0.04	0.04	0.03
30	195	TS	0.18	0.27	0.29	0.18
		A	0.18	0.28	0.13	0.12
40	389	TS	0.75	1.17	0.53	0.44
		A	0.50	0.97	0.28	0.26
50	1048	TS	9.13	12.71	1.99	1.04
		A	10.45	12.93	1.04	0.71

Table 6.9: Standard deviations of CPU times stated in Table 6.8, which are averaged over three independent runs

(OBM) and Algorithm 3 (EC) for $n = 20, 30, 40, 50$. Besides $n = 40$, (LK) performs worst. As (LK) solves more subproblems than all other methods, this result is not surprising. In contrast, the rather bad performance of (KS) is not expected with regard to the fact that (KS) solves the lowest number of subproblems in basically all instances. The reason lies in the huge number of cells, which are maintained in (KS) and which are scanned several times during each iteration. This computational effort is reflected in the CPU times.

Finally, we compare the two-stage scalarization to the augmented scalarization for all methods in Table 6.8. In all instances and for all methods, a considerable saving in computational time is reported when the augmented scalarization is used. In particular, with the augmented formulation, between 13% and 44%, on average 32% less computational time is required than with the two-stage formulation.

6.4 Conclusion and Further Ideas

In this chapter, we numerically validated the theoretical results of Chapters 4 and 5. In the bicriteria study, we compared different scalarizations and parameter schemes embedded in a full 2-split algorithm by applying them to a classic knapsack problem. Our numerical results show that the augmented weighted Tchebycheff method with adaptively chosen parameters reliably finds all nondominated points. This does not hold for the classic method, in which the augmentation parameter ρ is chosen fixed. Indeed, in our numerical study, a certain percentage of nondominated points is missed when the augmentation parameter is chosen a priori from $\{10^{-2}, 10^{-3}, 10^{-4}\}$, which is common in the literature. Moreover, in our tests considerably better computational times are recorded for the augmented weighted Tchebycheff method in comparison to a two-stage approach. We also tested a variant of the adaptive augmented weighted Tchebycheff method that uses local ideal points. Depending on the slope of the level curve we observed a trade-off between CPU time and the average value of ρ. When a rather small inclination of the slopes of the level curve is used, similar CPU times as compared to the method using the global ideal point are achieved while getting a larger average value for ρ.

In the tricriteria case, we applied Algorithm 2 and its variant Algorithm 3 from Chapter 5 as well as three state of the art methods from the literature to a multi-dimensional knapsack problem. All methods are evaluated with the same two-stage and augmented scalarization, where the parameters are computed according to the formulas obtained in Chapter 4.

We observed that all methods generate the complete nondominated set reliably in all instances. This validates the parameter scheme derived in Chapter 4, by which all parameters and, in particular, the augmentation parameter can be chosen such that no nondominated point is missed when a discrete multicriteria optimization problem is given.

For Algorithm 2 that is based on the v-split, the predicted upper bound on the number of subproblems that was derived in Chapter 5 is met in all instances. In our tests, exactly $3|Z_N|-2$ subproblems are solved when the (augmented) weighted Tchebycheff method is used as scalarization. When the (augmented) ε-constraint method is applied, exactly $2|Z_N|-1$ subproblems are solved. Algorithm 3, that does not generate quasi non-redundant boxes, performs slightly better than Algorithm 2, i.e., in general, less subproblems are solved in comparison to Algorithm 2. However, for Algorithm 3, no theoretical upper bound on the number of subproblems is known so far.

Our numerical study reveals that in case that a complete representation is sought, the use of the ε-constraint method is favorable, since approximately one third less subproblems have to be solved compared to an arbitrary scalarization method, e.g., the weighted Tchebycheff method. The reduction of the number of subproblems yields a considerable saving of computational time. Our new parametric algorithm in combination with an ε-constraint method can compete with state of the art methods. It outperforms the method of Lokman and Köksalan (2013) with respect to CPU time and the number of subproblems solved. While the method of Kirlik and Sayın (2014) often generates slightly fewer subproblems than our approach, it is outperformed with respect to CPU time. The method of Ozlen et al. (2014) turns out to be comparable to our approach with respect to the number of subproblems solved and CPU time. Depending on the instance, one method yields a better result than the other.

The scalarizations used in all algorithms in the tricriteria case were tested in a two-stage and an augmented formulation, respectively. We observe that in all instances, the augmented scalarization yields better computational times than the two-stage scalarization. In particular, savings of approximately 25% are achieved, on average. This confirms the importance of suitable parameters for methods with augmentation terms as proposed in Chapter 4 of this thesis.

The new parametric algorithm can compete with state of the art algorithms from the literature. However, while the latter rely on the ε-constraint scalarization and are explicitly designed to generate complete representations, our approach is much more general. It is not only possible to use other scalarizations as, e.g., a weighted Tchebycheff method, but a further advantage consists in the fact that our algorithm gives a description of the remaining search region, i.e., the region which might contain further nondominated points, at any time. Note that this is not possible with any of the three algorithms from the literature presented in this chapter.

So far, the total number of subproblems has attracted the attention of most authors. However, the performance of the algorithms is not necessarily mainly driven by the time needed to solve the subproblems. As the numerical results show, the approach of Kirlik and Sayın (2014) which solves the fewest number of subproblems does not perform best. Therefore, in the future, the overall complexity of the algorithms should be additionally studied. Besides, it would be interesting if an improved worst-case bound can be given for the approaches of Kirlik and Sayın (2014) and Ozlen et al. (2014), which showed a very competitive behavior with respect to the number of subproblems solved in this numerical study.

In this chapter we only considered bi- and tricriteria problems. While Algorithm 2 can only be applied to tricriteria problems, Algorithm 3 is applicable to problems with an arbitrary number of criteria. As the algorithms of Lokman and Köksalan (2013), Kirlik and Sayın (2014) and Ozlen et al. (2014) can also be applied to problems with any number of criteria, a comparison would be interesting.

Another idea consists in using our new approach within a two phase method which generates all supported nondominated points in a first phase and searches for unsupported nondominated points in a second phase. For example, it would be interesting to study whether the method of Przybylski et al. (2010a) can benefit from the v-split criterion with respect to computational time.

7 Generation of Incomplete Representations for Continuous Test Problems

7.1 Introduction

In this chapter we demonstrate that the adaptive parametric algorithm which was developed in Part I is not only applicable for generating a complete representation of the nondominated set of a discrete multicriteria optimization problem but can also be used when an incomplete representation of the nondominated set of a discrete or continuous, convex or non-convex multicriteria optimization problem is sought. Throughout this chapter, we consider continuous multicriteria optimization problems.

When an incomplete representation is sought, quality criteria as discussed in Section 2.2 are used to evaluate the corresponding representation. Moreover, in this numerical study we are particularly interested in the question whether less infeasible and/or redundant subproblems are obtained by an adaptive in comparison to an a priori parameter scheme. Recall from Section 2.4 that the parameters of a priori approaches are fixed in the initialization phase. Therefore, in this case, typically a certain quantity of infeasible and/or redundant subproblems which do not contribute points to the representation is produced. Since adaptive approaches take all previously obtained nondominated points into account, we expect them to perform better in this regard.

In the remainder of this section we describe the general setting of our numerical study. In Section 7.2 bicriteria problems are studied. Tricriteria problems are considered in Section 7.3. Section 7.4 contains the conclusion and future directions of research.

Parametric Algorithms

Analogously to the discrete case, we use Algorithm 4 for bicriteria and Algorithm 2 for tricriteria problems in combination with an adaptive parameter scheme. For the a priori parameter scheme, no particular algorithmic framework is required. After having computed the parameter grid, the corresponding subproblems are solved. In the bicriteria case, we additionally apply the sensitivity-based approach of Eichfelder (2006), see Section 3.2 for a detailed description.

As scalarization for the a priori approach we select the ε-constraint method in the formulation

$$\min \{f_m(x) : f_j(x) \leq \varepsilon_j \; \forall \, 1 \leq j \leq m-1, x \in X\} \,.$$

Note that, different from Section 5.4 and Section 6.3, we do not minimize with respect to the first component but with respect to component m here. This is due to the fourth tricriteria test case, in which component m is selected in the literature. For simplicity, we keep this choice fixed for all test cases in this chapter. In order to avoid weakly nondominated outcomes, a second stage problem of the form

$$\min \left\{ \sum_{i=1}^{m} f_i(x) : f_j(x) \leq f_j(x^*), j = 1, \ldots, m, \, x \in X \right\}$$

is solved with x^* being the solution obtained in the first stage. We do not employ an augmented formulation, as this would require prescribed trade-off information in the continuous case which is not given. However, if trade-off information was given, we could easily translate this information into suitable parameters, see Section 4.3.

All adaptive approaches employ a two-stage weighted Tchebycheff method, if not stated otherwise. Again, the reason is that no trade-off information is given. However, in order to validate the parameters given in Section 4.3, we also test one variant that uses a generalized augmented weighted Tchebycheff method in the tricriteria case. Note that we do not incorporate an adaptive ε-constraint method, as this would require to specify a certain distance of the right-hand side parameters ε from the upper bounds of the respective box, in order to make sure that a nondominated point different from the previous ones is computed, see, e.g., Hamacher et al. (2007). A disadvantage of this artificial reduction of the current box is that the resulting subproblem might be infeasible even if the considered box contains additional nondominated points. Moreover, the choice of the distance of ε from the upper bound of the box strongly influences the points that are obtained. Therefore, a Tchebycheff scalarization seems to be more appropriate, and we do not use an ε-constraint scalarization as adaptive method. In the method of Eichfelder (2006) we also choose a weighted Tchebycheff scalarization, see Problem (3.5).

All algorithms are implemented in MATLAB R2011a. We use the non-linear function `fmincon` to solve the scalarizations. As single-criterion solver the SQP-method is selected. The use of gradient information is turned on.

Refinement of the Representation and Termination Criterion

The adaptive algorithms are slight modifications of Algorithm 4 in the bicriteria and Algorithm 2 in the tricriteria case. They only differ in two details. First, since the nondominated set is not finite, a suitable termination criterion is required. Secondly, a rule is to be formulated which box shall be selected at the beginning of each iteration, i.e., where the representation is to be refined next. Note that neither a termination criterion nor a particular selection rule is needed when a complete representation is to be generated.

As termination criterion, we will always set a bound on the number of subproblems to be solved. For the refinement of the representation, we implement and test two different rules, yielding two different algorithmic variants. The first variant, which we call *volume-based* in the following, saves the volume that contains possibly further nondominated points of each box in the decomposition. In each iteration, the box having the largest volume is selected. In the bicriteria case, this criterion equals the one used in Hamacher et al. (2007). In the second variant, called *hypervolume-based* in the following, we select the next box according to the highest contribution to the dominated hypervolume that is realized by the outcome computed in this box. Recall from Section 2.2 that the dominated hypervolume, defined in (2.19), describes the set that is dominated by all points of the current representation. In order to be able to insert the point with the highest contribution to the dominated hypervolume, we must know the outcome obtained in each box before selecting a box. Therefore, in this variant, we solve a subproblem for each new box directly after the split. The resulting point is not inserted directly into the search region but is saved with the box in which it was generated. Moreover, the theoretical contribution of this point to the dominated hypervolume is computed and saved. In each iteration, that box is selected whose associated nondominated point contributes most to the dominated hypervolume. Note that the hypervolume-based variant bares similarity to the gauge algorithm of Klamroth et al. (2002) that constructs a piece-wise linear inner or outer approximation of the nondominated set, see Section 3.3. Instead of the contribution to the dominated hypervolume, the latter uses the gauge of the current point as selection criterion where to refine next. Thereby, in the general non-convex case, the gauge corresponds to the scaled level of a weighted Tchebycheff norm. As we do not construct an approximation of the nondominated set, we do not include the gauge algorithm into our numerical study. One drawback of the hypervolume-based variant is the fact that more solutions are computed than inserted. However, it is possible to insert all computed points into the final representation at the end of the algorithm.

Quality Criteria

In Section 2.2, we presented common quality criteria for (discrete) representations according to Zitzler and Thiele (1998), Sayın (2000) and Eichfelder (2006), which are used for the evaluation in this numerical study. The criteria cardinality and uniformity are defined according to Sayın (2000). Considering coverage, we use (2.18), thus, we follow the concept of Eichfelder (2006). The author argues that if we can assume that the representation covers all parts of the nondominated set sufficiently well, the computation of the coverage error, which is a measure for the worst-represented point of the nondominated set, can be approximated by half of the largest distance among ‘adjacent’ points. For simplicity, we omit the division by the constant factor one half and state the largest distance as coverage error in the following. While the definition of adjacent points is obvious for bicriteria problems, it becomes involved for three or more criteria. Therefore, we evaluate this criterion only in the bicriteria case. Moreover, as also discussed in Eichfelder (2006) and Section 2.2, the coverage error is only defined in a meaningful way among representing points from the same connected component of the nondominated set. In this study we assume that we know the connected components. In general, information about connectedness might be retrieved from the scalarization, see, e.g., Eichfelder (2006). As fourth quality criterion, we use the dominated hypervolume (2.19) introduced in Zitzler and Thiele (1998) and relate it to the volume of the initial search region. Summarizing, we evaluate the four quality criteria

(i) cardinality, given by $|\mathcal{R}|$,

(ii) uniformity, computed as

$$d_U(\mathcal{R}) := \min_{y\in\mathcal{R}, q\in\mathcal{R}\setminus\{y\}} \|y-q\|_2,$$

(iii) coverage, given by

$$d_C(\mathcal{R}) := \max_{y\in\mathcal{R}} \min_{q\in\mathcal{N}(y)} \|y-q\|_2,$$

where $\mathcal{N}(y) \subseteq \mathcal{R}$ denotes the set of neighbors of $y \in \mathcal{R}$, see Section 2.2, and

(iv) relative dominated hypervolume, given by

$$r_H(\mathcal{R}) := \frac{V\left(\bigcup_{y\in\mathcal{R}}\{z \in B_0 : z \geqq y\}\right)}{V(B_0)},$$

where $B_0 := \{z \in \mathbb{R}^m : z^I \leqq z \leqq r\}$ with $r \in \mathbb{R}^m$ a given reference point, and $V(S)$ denotes the volume of $S \subset \mathbb{R}^m$. Throughout this study, we set $r := z^N$ for $m = 2$ and $r := z^M$ for $m = 3$.

Before presenting the test cases and results, we state particularities of the implementation in the bi- and tricriteria case, respectively.

7.2 Bicriteria Problems

7.2.1 Computational Setup

Initial Bounds on the Nondominated Set

All tested methods start by computing the lexicographically minimal points that define the ideal point z^I as well as the nadir point z^N. Recall that it is a speciality of bicriteria problems that the nadir point can be computed from the lexicographically minimal points.

Termination Criterion

As already mentioned in Section 7.1, the number of subproblems solved serves as termination criterion. Since the method of Eichfelder (2006) does not deal with the number of subproblems as input data but with a given prescribed spread between the points to be computed, we first evaluate this algorithm for a selected accuracy α, i.e., a prescribed l_2-distance among all points of the representation, and count the required number of subproblems $s \in \mathbb{N}$. The latter serves as termination criterion for all other methods. Note that the subproblems solved to obtain the lexicographic minima at the beginning of each method are not counted.

Tested Variants and their Parameters

A summary of all tested methods is stated in Table 7.1. Based on the number of iterations $s \in \mathbb{N}$ and the bounds z^I and z^N, the ε-constraint method (EC) computes a one-dimensional grid of parameter values $\varepsilon_j \in \mathbb{R}$, $j \in \{1, \ldots, s\}$, where

$$\varepsilon_j = z_1^I + j \cdot \frac{z_1^N - z_1^I}{s+1} \quad \text{for } j = 1, \ldots, s.$$

Note that the parameters are set such that the lexicographic minima, which have already been generated beforehand, are not necessarily recomputed. In the sensitivity-based method of Eichfelder (SB) we use the direction $d := (1,1)^\top$ and define the hyperplane, from which the reference points are taken, by $b := (1,1)^\top$ and $\beta := 0$.

EC	ε-constraint method with uniformly chosen parameters (a priori)
SB	Sensitivity-based approach of Eichfelder (2006)
LV	Algorithm 4 with local ideal points as reference points and a volume-based selection of the next box
GV	Algorithm 4 with (global) ideal point as reference point and a volume-based selection of the next box
LH	Algorithm 4 with local ideal points as reference points and a hypervolume-based selection of the next box
GH	Algorithm 4 with (global) ideal point as reference point and a hypervolume-based selection of the next box

Table 7.1: Summary of tested methods in the bicriteria case

The parameters of all other methods using a two-stage weighted Tchebycheff scalarization are computed according to (4.9). Thereby, the reference point is either set to the (global) ideal point (GV, GH) or to a local ideal point (LV, LH).

Refinement of the Representation

In Section 7.1, we specified the volume-based and the hypervolume-based selection rule according to which we select the box for the next subproblem. In the bicriteria case, the calculation of both, the volume of a box containing possibly further nondominated points as well as the contribution of a point to the dominated hypervolume is simple. For every pair of adjacent points that implicitly define a box, the volume is computed as the area of the rectangle whose lower and upper vertices are given by the local ideal and the local nadir point with respect to the two adjacent points. The contribution of a new point to the dominated hypervolume is determined as the area of the rectangle defined by the new point and the upper bound of the box to which the point is associated. We test two variants that use a volume-based selection of the next box (LV, GV) as well as two variants employing a hypervolume-based choice (LH, GH).

7.2.2 Test Problems

Test Problem 1

The first test case is taken from Eichfelder (2006) and given by

$$\begin{aligned}
\min \quad & \begin{pmatrix} \sqrt{1+x_1^2} \\ x_1^2 - 4x_1 + x_2 + 5 \end{pmatrix} \\
\text{s.t.} \quad & x_1^2 - 4x_1 + x_2 + 5 \leq 3.5, \\
& x_1, x_2 \geq 0, \\
& x \in \mathbb{R}^2.
\end{aligned} \tag{7.1}$$

The efficient set of (7.1) is

$$X_E = \left\{ x \in \mathbb{R}^2 : x_1 \in \left[2 - \sqrt{2.5}, 2\right], x_2 = 0 \right\}.$$

This example with $z^I \approx (1.08, 1)^\top$ and $z^N \approx (2.24, 3.5)^\top$ has a simple $\mathbb{R}^2_+$-convex nondominated set. We apply the algorithm of Eichfelder (2006) with a given l_2-distance of $\alpha = 0.5$ and $\alpha = 0.2$ among the representing points, which yields $s = 6$ and $s = 15$ subproblems, respectively. The results of the tested six methods with $s = 6$ and $s = 15$ are stated in Table 7.2. The respective final representations of the nondominated set are depicted in Figures 7.1 and 7.2.

Consider first the cardinality of the representations in Table 7.2. Including the two lexicographically minimal points, methods EC, LV and GV generate 8 and 17 points, respectively, i.e., every of the 6 and 15 subproblems solved yields a point of the final representation. Method SB generates one point less, which is due to the fact that the second lexicographic minimum that is computed at the beginning is recomputed within the algorithm. The final representations of methods LH and GH comprise

#SP		EC	SB	LV	GV	LH	GH
6 (+2)	$\lvert\mathcal{R}\rvert$	8	7	8	8	6	6
	d_U	0.17	0.44	0.27	0.18	0.27	0.20
	d_C	0.95	0.50	0.60	0.67	0.96	1.15
	r_H	64.67%	64.00%	65.22%	64.28%	62.07%	61.35%
15 (+2)	$\lvert\mathcal{R}\rvert$	17	16	17	17	10	10
	d_U	0.07	0.14	0.13	0.05	0.14	0.11
	d_C	0.49	0.21	0.27	0.40	0.52	0.67
	r_H	69.18%	69.35%	69.71%	68.92%	67.20%	66.25%

Table 7.2: Cardinality, uniformity, coverage and relative dominated hypervolume for test case 1

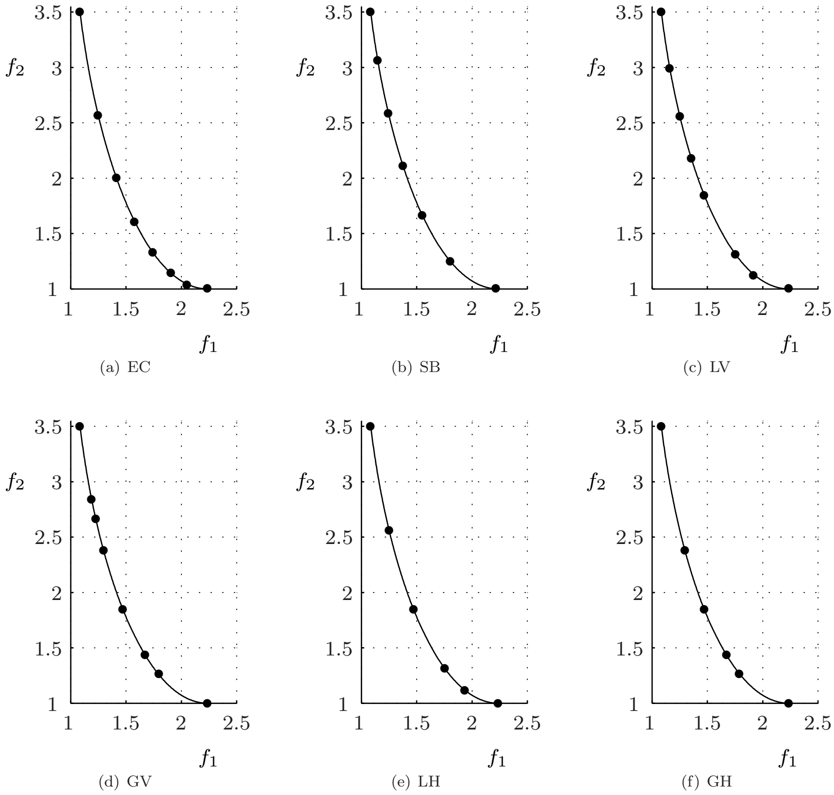


Figure 7.1: Representations of the nondominated set computed by different parametric algorithms for test case 1 and 6 subproblems

less points, respectively, as in each iteration (besides the first one) two subproblems are solved, but only one point is inserted into the representation. Therefore, when the termination criterion is reached, LH and GH both have inserted 6 and 10 points, respectively. Hence, no infeasibility or redundancy occurs for any method in this test case.

Considering uniformity and coverage, given as maximal distance between adjacent points, we observe that SB achieves the best result in the sense that the distances between adjacent points vary the fewest, namely in the interval $[d_U, d_C] = [0.44, 0.5]$ for $s = 6$ and $[d_U, d_C] = [0.14, 0.21]$ for $s = 15$. Remember that SB is particularly designed for generating equidistant representations, and that we chose $\alpha = 0.5$ and $\alpha = 0.2$, respectively. As the second best method with respect to the criterion ‘maximal uniformity and minimal coverage’ we identify LV with $[d_U, d_C] = [0.27, 0.6]$ for $s = 6$ and $[d_U, d_C] = [0.13, 0.27]$ for $s = 15$. In comparison, the representation computed by EC is much less uniform, since $[d_U, d_C] = [0.17, 0.95]$ for $s = 6$ and

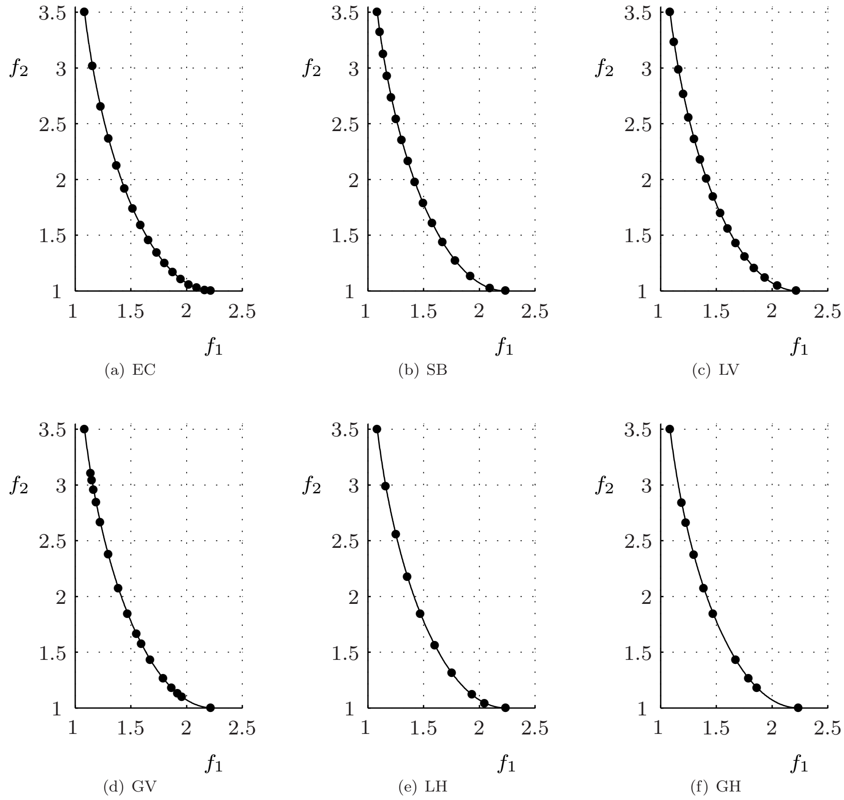


Figure 7.2: Representations of the nondominated set computed by different parametric algorithms for test case 1 and 15 subproblems

$[d_U, d_C] = [0.07, 0.49]$ for $s = 15$. Also GV, LH and GH yield worse results than LV.

When comparing the relative dominated hypervolumes of the final representations, the differences among the methods EC, SB, LV and GV are small with LV slightly leading. The results of LH and GH are worse. However, we have to take into account that the cardinality of the representations obtained by LH and GH is smaller compared to LV.

The observations derived from Table 7.2 are also confirmed by the visualizations in Figures 7.1 and 7.2. Because of EC not having the possibility to adjust to the shape of the nondominated set, more points are computed where the curve is rather flat and less where it is steep. Method SB computes points with nearly equal distance in every part of the nondominated set. Also LV obtains a nearly equidistant representation. In contrast, method GV tends to build clusters of points. Also method GH computes a non-uniform representation. Method LH produces an approximately equidistant representation for $s = 15$, however containing less points compared to LV.

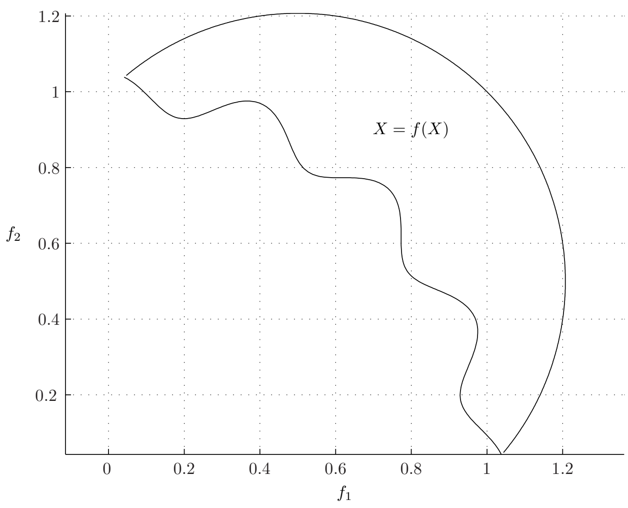


Figure 7.3: Feasible set of test case 2

In general, comparing the global to the local variants, i.e., GV to LV and GH to LH, we observe that the local variants result in more uniform representations than the global ones. Hence, the use of the (global) ideal point as fixed reference point seems to be disadvantageous. The local variants can adapt better to the shape of the nondominated set.

Concluding, we state that no infeasible or redundant subproblems occur for any of the tested methods. Method SB generates an equidistant representation of the nondominated set. Among the adaptive methods particularly LV performs well and produces a more uniform representation than the a priori method.

Test Problem 2

The second test case is taken from Tanaka et al. (1995) and has also been used, e.g., in Eichfelder (2006). Its formulation is

$$\begin{aligned}
\min \quad & \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \\
\text{s.t.} \quad & x_1^2 + x_2^2 - 1 - 0.1 \cos\left(16 \arctan\left(\tfrac{x_1}{x_2}\right)\right) \geq 0, \\
& (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5, \\
& x_1, x_2 \in (0, \pi], \\
& x \in \mathbb{R}^2.
\end{aligned} \tag{7.2}$$

# SP		EC	SB	LV	GV	LH	GH
8 (+2)	$\lvert\mathcal{R}\rvert$	8	7	10	10	7	7
	d_U	0.05	0.12	0.08	0.04	0.14	0.13
	d_C	0.28	0.23	0.22	0.19	0.34	0.19
	r_H	23.68%	26.02%	27.12%	22.25%	21.47%	21.30%
16 (+2)	$\lvert\mathcal{R}\rvert$	13	15	18	10	11	11
	d_U	0.04	0.02	0.03	0.05	0.07	0.05
	d_C	0.23	0.12	0.14	0.19	0.22	0.19
	r_H	26.82%	28.97%	28.36%	27.68%	27.61%	22.73%

Table 7.3: Cardinality, uniformity, coverage and relative dominated hypervolume for test case 2

The efficient set equals the nondominated set and consists of three unconnected parts. Numerically, we obtained $z^I \approx (0.04, 0.04)^\top$ and $z^N \approx (1.04, 1.04)^\top$. The feasible set of (7.2) is depicted in Figure 7.3. Again, we first evaluate SB, here with a given l_2-distance of $\alpha = 0.2$ and $\alpha = 0.1$, which results in $s = 8$ and $s = 16$ subproblems, respectively. The results for all methods are presented in Table 7.3, the final representations as well as the $\mathbb{R}^2_+$-non-convex, disconnected nondominated set are depicted in Figures 7.4 and 7.5.

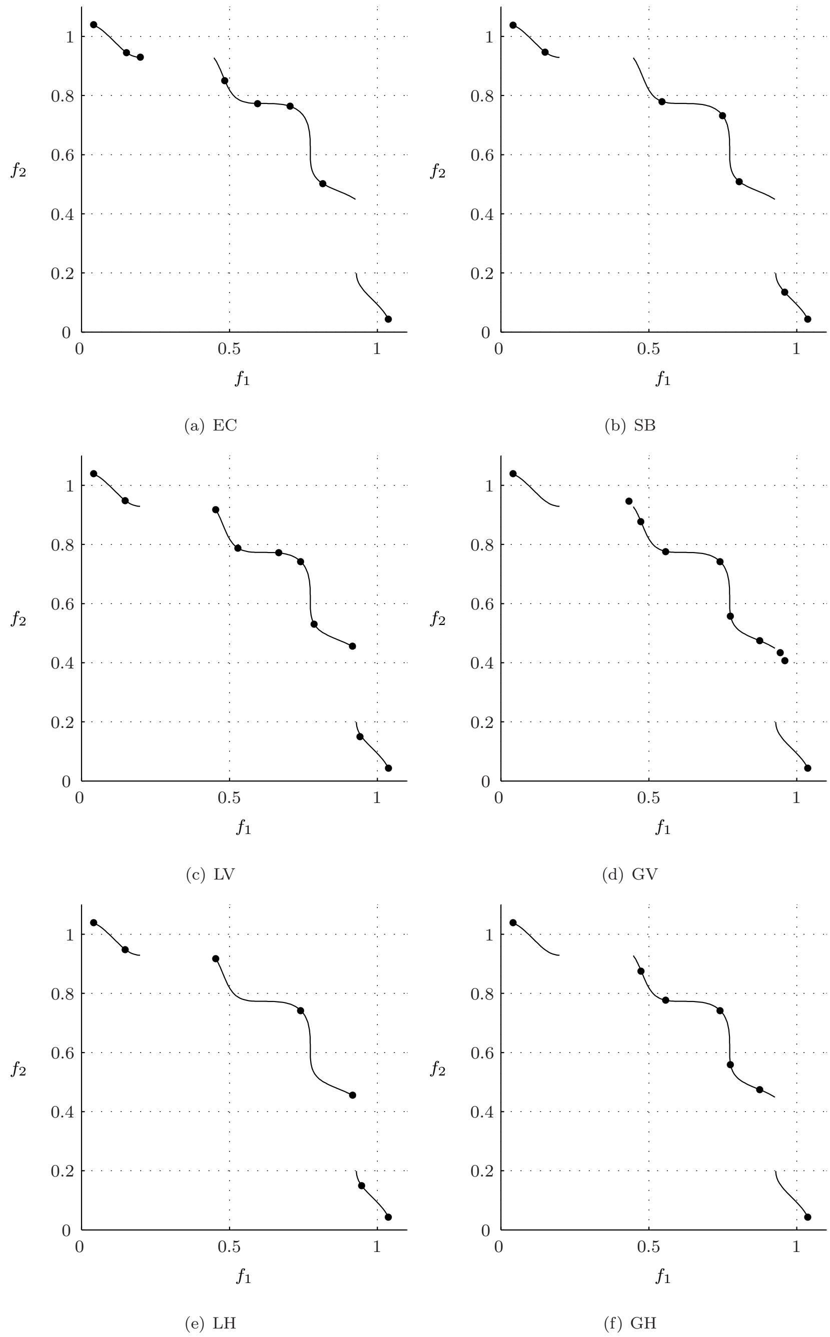
Due to the non-convexity, dominated outcomes might be obtained which typically correspond to local minima of the scalarized problem. For example, in Figure 7.4 (d), three points of the final representation are actually dominated. Therefore, Eichfelder (2006) proposes to apply global single-criterion solvers in the non-convex case. However, as our results with SQP are rather good and the observed dominated outcomes lie quite close to the nondominated set, we do not change the solver and accept the occurrence of some dominated points.

Consider first the cardinality of all representations stated in Table 7.3. While for methods LV, LH and GH in every iteration a point of the final representation is computed, this is not the case for the other methods. In EC, only eight of ten ($s = 8$) and 13 of 18 ($s = 16$) subproblems contribute a point to the final representation. The small cardinality of method GV is caused by dominated points. Note that if in the course of the algorithm a dominating point is computed, all dominated points are removed.

Considering uniformity and coverage, the results are less clear than in the $\mathbb{R}^2_+$-convex case. Due to the unconnectedness of the nondominated set, SB can not compute all points with the desired accuracy of 0.2 and 0.1, respectively, which results particularly for $s = 16$ in a rather small uniformity level of 0.02. Nevertheless, the results $[d_U, d_C] = [0.12, 0.23]$ for $s = 8$ and $[d_U, d_C] = [0.02, 0.12]$ for $s = 16$ are satisfying. A quite similar interval is obtained by LV. The results of EC are inferior, even if for $s = 16$ a slightly higher uniformity level compared to LV is obtained. However, coverage is considerably worse in both runs. Interestingly, for $s = 8$, methods GV and GH achieve a better result with respect to uniformity and coverage than LV. Considering the relative dominated hypervolume we observe that SB and LV achieve the best result and GH the worst.

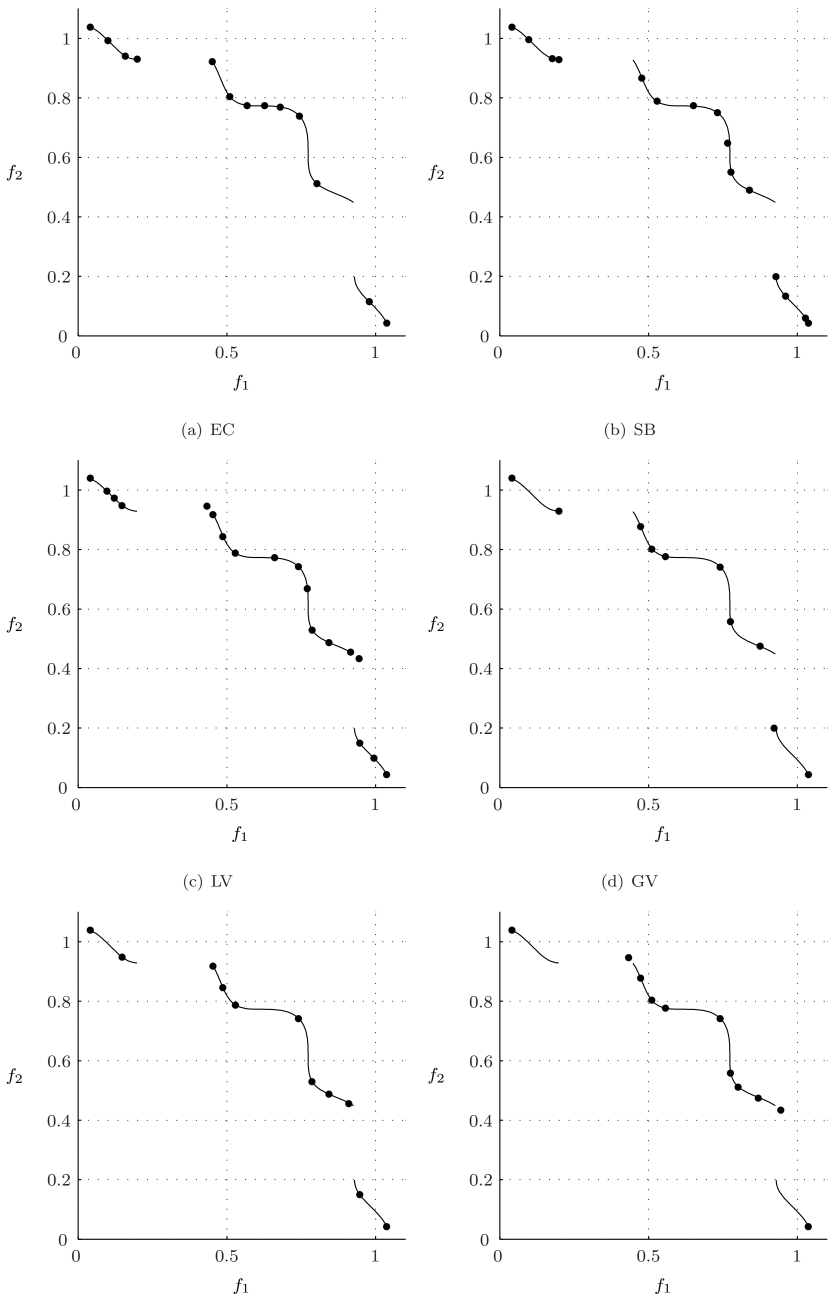
A view on the graphics explains the surprisingly good values of the global variants particularly with respect to coverage. Two of the three connected parts are, besides for GV and $s = 16$, only covered by one single point. As coverage is computed only among points of the same connected component, two of the three parts are not included into the computation of the coverage error, yielding to misleading results. This shows that replacing the computation of coverage by the maximal distance between neighboring points of the final representation must be done with care, as it is not always assured that all parts of the nondominated set are covered well. The representations of all other methods cover all three parts of the disconnected nondominated set. Note that while the nondominated set is symmetric, the representation obtained by EC is not. This is due to the fact that constraints are only set on the first objective. This effect does not occur for all other methods as both components of the objective function are treated equally in the weighted Tchebycheff method.

We conclude that in every iteration a point of the final representation is computed by the adaptive methods LV, LH and GH but not by EC. Moreover, the adaptive methods generate more uniform representations than the a priori method, in general. Besides SB, particularly LV performs well in this example.



(a) EC (b) SB (c) LV (d) GV (e) LH (f) GH

Figure 7.4: Representations of the nondominated set computed by different parametric algorithms for test case 2 and 8 subproblems



(a) EC (b) SB (c) LV (d) GV (e) LH (f) GH

Figure 7.5: Representations of the nondominated set computed by different parametric algorithms for test case 2 and 16 subproblems

7.3 Tricriteria Problems

In this section, we consider continuous tricriteria optimization problems. As stated above, we are mainly interested in the question whether an algorithm with adaptive parameter selection as derived in Part I performs better than an algorithm with a priori parameter choice, in the sense that less infeasible and/or redundant subproblems occur. Before presenting the test cases, we discuss particularities of the implementation in the tricriteria case.

7.3.1 Computational Setup

Initial Bounds on the Nondominated Set

As in the bicriteria case, all algorithmic variants start by computing the lexicographically minimal points, which define the ideal point. However, as explained in Section 2.1, only an estimate on the nadir point can be obtained based on these points for more than two criteria. In order to have a valid upper bound at hand, we additionally compute the individual maxima on $f(X)$ that define z^M, which serves as the upper bound to start with. Note that we explicitly deal with the general situation in which the nadir point is not known.

Decomposition of the Search Region

In the bicriteria case the split of the search region into boxes is obvious. Every new nondominated point lies in exactly one box and evokes a split of this box into two new boxes. In three dimensions, principally two possibilities for splitting the search region exist. The first relies on the v-split criterion as proposed in Chapter 5, with the help of which the creation of redundant boxes can be avoided. Since at most two new boxes are created in each iteration in total, see Lemma 5.23, the search region is decomposed into at most $2n + 1$ boxes, where n denotes the number of points inserted so far. The main feature of Algorithm 2, that relies in the fact that no redundant boxes are generated and, hence, no unnecessary subproblems are solved, is also advantageous in the continuous case. Note that quasi non-redundant boxes as discussed in Section 5.3.4 typically do not occur in a continuous setting, hence, there is no advantage in considering Algorithm 3 instead of Algorithm 2. The v-split algorithm is especially favorable when the dominated hypervolume is used as selection criterion of the next box. In this variant, as explained in Section 7.1, a subproblem is solved for every new box that is created. In each iteration, only the nondominated point that contributes most to the dominated hypervolume is inserted. As we use the number of subproblems as termination criterion, each new

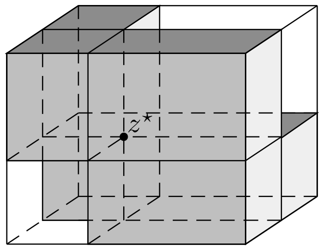


Figure 7.6: Decomposition of a box into six disjoint boxes

box implicitly increases the subproblem counter. The less new boxes are created, the less subproblems have to be solved and the more (nondominated) points can be inserted before the algorithm terminates.

The second possibility of decomposing the search region is given by a split into disjoint boxes, see Figure 7.6. Clearly, a higher number of boxes has to be handled in this split, wherefore we have not used it in the situation where a complete representation is sought. However, an advantage consists in the fact that a local ideal point can be easily associated with each box, as it equals the lower bound of the respective box. Consequently, the volume of each box can be easily computed based on its lower and upper bound. This is particularly favorable for the variant in which the next box is selected as the one with the largest volume.

Refinement of the Representation

As in the bicriteria case, we select the next box either according to its volume or to the contribution to the dominated hypervolume, which is achieved by inserting the point generated in the respective box. However, the computation of both, the volume and the dominated hypervolume, becomes more involved than in the bicriteria case.

First, consider the computation of the volume. If we apply the v-split, the part of a box containing possibly further nondominated points is a subset of this box and is given by the union of several disjoint subboxes. Hence, the volume of a box is determined as the sum of the volumes of all disjoint subboxes that are contained in this box. Consider Figure 7.7 for an illustration. The volume of each of the three boxes consists of the volumes of three disjoint subboxes, respectively. To ease the computation of the volumes, we compute and save the upper and lower vertices of these disjoint boxes additionally to the bounds of the boxes needed for the v-split.

Also the computation of the contribution to the dominated hypervolume of a newly generated point becomes more involved in comparison to the bicriteria case, since a nondominated point might lie in more than one box. Hence, all boxes that contain

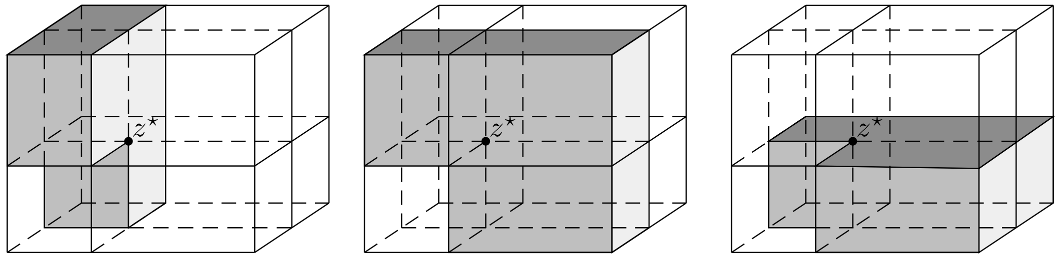


Figure 7.7: Volumes of boxes decomposed by the v-split

the current point have to be taken into account when computing the contribution to the dominated hypervolume of this point. In our implementation, we associate points with the boxes in which they were generated. We also associate the contribution to the dominated hypervolume of some point with the box, in which the point has been generated. Therefore, the contribution to the dominated hypervolume obtained by a point in a certain box might change even if the corresponding box has not been split in the current iteration. Consequently, in our implementation, we always recompute the contribution to the dominated hypervolume for every box of the decomposition. Thereby, the contribution to the dominated hypervolume is determined as difference between the dominated hypervolumes before inserting and after having inserted the considered point. We use the algorithm of Beume et al. (2009) for this purpose, which can be briefly summarized as follows. First, the given n points are ordered with respect to one component. The dominated hypervolume is decomposed into layers. Two-dimensional projections of the points build the basis of the hypervolume, which can be computed with a complexity of $\mathcal{O}(n \log n)$. As reference point we use z^M.

Quality Criteria

In the tricriteria case, we only evaluate cardinality and the relative dominated hypervolume. As in the bicriteria case, cardinality measures the number of subproblems that contribute points to the final representation, hence, implicitly reflects infeasibility or redundancy of subproblems. The relative dominated hypervolume indicates the percentage of the portion of the initial search region that is discarded by the computed points. We use the algorithm of Beume et al. (2009) for its determination with z^M as reference point. Note that, if z^M overestimates z^N considerably, the hypervolume of two representations might be nearly the same as the nondominated set lies in a relatively small portion of the initial search region.

The computation of the coverage error requires the knowledge of neighboring

EC	ε-constraint method with uniformly chosen parameters (a priori)
GV	Algorithm 2 with (global) ideal point as reference point and a volume-based selection of the next box
GH	Algorithm 2 with (global) ideal point as reference point and a hypervolume-based selection of the next box
GH2	like GH, but all computed points are inserted when the termination criterion is reached
GHA	like GH, but a generalized augmented weighted Tchebycheff norm with $\delta = 0.7$ is used
LV	Decomposition of the search region into disjoint boxes, use of local ideal points as reference points and a volume-based selection of the next box

Table 7.4: Summary of tested methods in the tricriteria case

points, which are not as easy to obtain as in the bicriteria case. Therefore, we do not evaluate coverage and uniformity for the tricriteria case.

Tested Variants and their Parameters

All methods that are compared in the tricriteria case are summarized in Table 7.4. After having computed z^I and z^M, the a priori approach installs a two-dimensional grid of N^2 parameter values $(\varepsilon_{1,k}, \varepsilon_{2,l})^\top \in \mathbb{R}^2$, $k, l \in \{1, \ldots, N\}$, where, for $i = 1, 2$,

$$\varepsilon_{i,j} = z_i^I + (j-1) \cdot \frac{z_i^M - z_i^I}{N-1} \quad \text{for } j = 1, \ldots, N.$$

For simplicity, we use the same number of grid points $N = 5, 7, 10$ in each dimension, resulting in $s = 25, 49, 100$ subproblems, respectively. Note that we define ε slightly different from the bicriteria case such that points that coincide with z^I or z^M in one component can also be computed, resulting in a possibly better coverage of the boundary of the nondominated set.

Again, the parameters of all adaptive variants are computed according to the formulas provided in Chapter 4. We also include a generalized augmented weighted Tchebycheff method (GHA) as scalarization, where the inflection point is set to the upper bound of the respective box multiplied by $\delta = 0.7$, see the formulas in Theorem 4.18 in Section 4.3.

In all variants in which we apply the v-split, we use the ideal point as reference point. Therefore, in analogy to $m = 2$, we call these variants *global* variants (GV, GH, GH2 and GHA in Table 7.4). Note that also other reference points could be chosen, but that we do not additionally vary the reference point in this study.

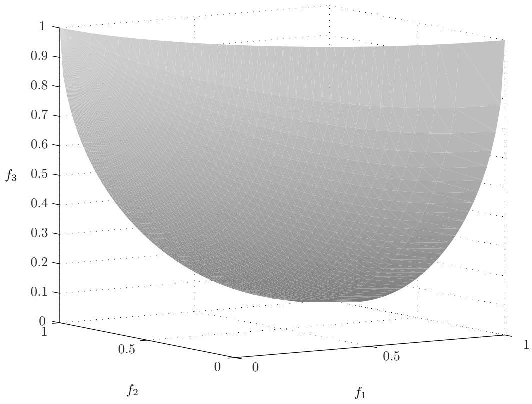


Figure 7.8: Nondominated set of test problem 1

7.3.2 Test Problems

Test Problem 1

The first test case is given by

$$\begin{array}{ll} \min & \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \\ \text{s.t.} & (x_1-1)^2+(x_2-1)^2+(x_3-1)^2 \leq 1, \\ & x \in \mathbb{R}^3. \end{array} \tag{7.3}$$

The feasible set X of (7.3) is the unit ball centered at $(1,1,1)^\top$. The efficient set equals the nondominated set and is given by

$$X_E = Z_N = \{x \in [0,1]^3 : (x_1-1)^2+(x_2-1)^2+(x_3-1)^2 = 1\},$$

i.e., it equals the lower left part of the sphere, see Figure 7.8 for an illustration. The ideal point is $z^I = (0,0,0)^\top$, and the local nadir point which is defined by the lexicographic minima is $(1,1,1)^\top$. It equals the true nadir point. The upper bound on $X = f(X)$ is $z^M = (2,2,2)^\top$. Despite its simplicity, already this simple test case can be an issue for parametric algorithms if the initial search region is not chosen appropriately.

As the nondominated set is $\mathbb{R}^3_+$-convex, the weighted sum method can be used to generate every nondominated point in theory. Specific methods for $\mathbb{R}^m_+$-convex problems exist, see, e.g., Klamroth et al. (2002) or Rennen et al. (2009). However, as pointed out above, our approach does not assume convexity, but is intended for any problem type. Therefore, we do not incorporate specific approaches, but only compare our new adaptive parametric algorithm to an a priori algorithm using the ε-constraint scalarization. A visualization of the respective final representations is given in Figures 7.10 to 7.12. Cardinality and relative dominated hypervolume of the corresponding representations are stated in Table 7.5. Note that, as in the bicriteria case, the lexicographic minima, that are generated for each method at the beginning, are counted separately.

Different from the bicriteria case, the cardinality of the representations obtained by the adaptive methods and the a priori approach varies considerably. This is due to the fact that the initial parameter grid for EC is no longer computed with respect to z^N, but with respect to z^M. Typically, in addition to certain infeasible subproblems, a high percentage of redundant subproblems occurs. In test case 1, only about one fourth of the subproblems solved in EC yield a nondominated point. The adaptive methods yield a significantly better cardinality than EC. Indeed, for method GV, with every subproblem solved a point is generated that contributes to the final representation. In methods GH2 and LV, not all but still many subproblems contribute points to the representation. Methods GH and GHA generate considerably less points. This is due to the fact that in both algorithms only around half of the points that are obtained from the subproblems are inserted into the representation. However, even these variants contribute more points than EC.

The stated relative dominated hypervolumes draw basically the same picture. However, there, the differences are less sharp, which is on the one hand due to the fact that z^M overestimates z^N, and on the other hand caused by the fact that some of the adaptive methods produce clustered points which contribute few to the dominated hypervolume. This is confirmed by the graphical illustration in Figures 7.10 to 7.12. Particularly method GV produces clusters at the boundary of Z_N. Also LV generates most of its points at the boundary of Z_N. This is caused by boxes that have an upper bound equal to the artificial initial bound z^M in at least one com-

# SP		EC	GV	GH	GH2	GHA	LV
25 (+3)	$\lvert\mathcal{R}\rvert$	6	28	11	24	11	25
	r_H	66.43%	80.96%	77.02%	80.77%	77.13%	80.48%
49 (+3)	$\lvert\mathcal{R}\rvert$	11	52	19	45	19	40
	r_H	73.39%	82.69%	80.06%	82.67%	80.29%	82.28%
100 (+3)	$\lvert\mathcal{R}\rvert$	26	103	29	71	29	76
	r_H	77.75%	83.05%	81.69%	83.42%	81.84%	83.66%

Table 7.5: Cardinality of $\mathcal{R}$ and relative dominated hypervolume for test case 1

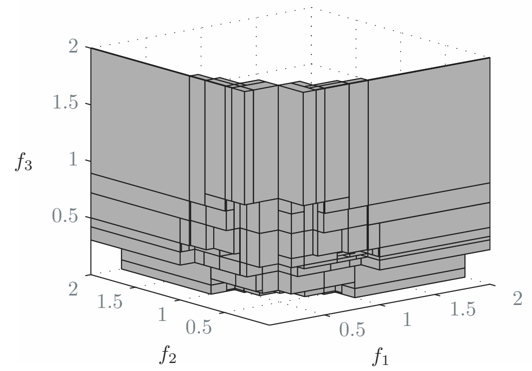
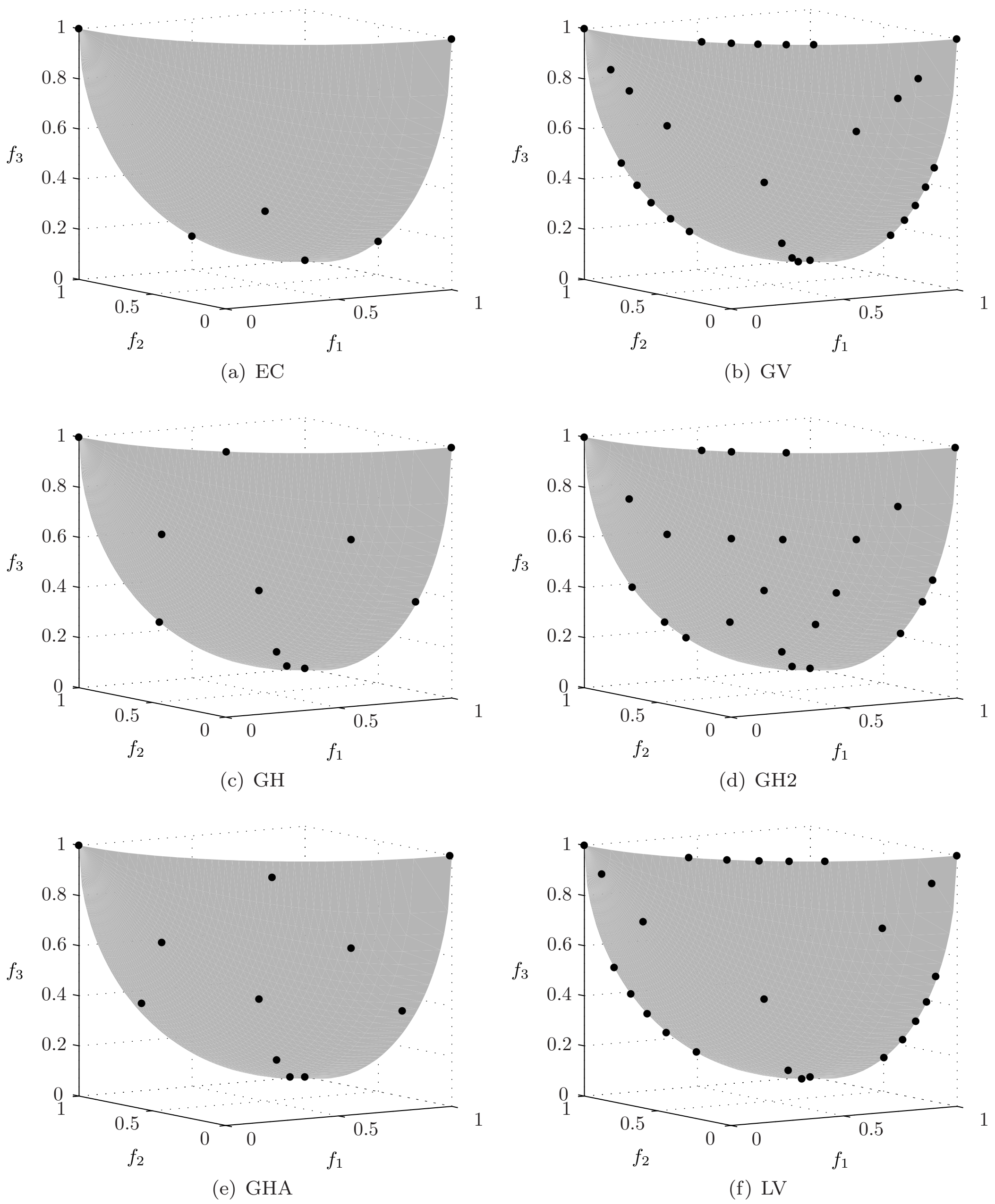


Figure 7.9: Decomposition of the search region obtained by LV after 20 iterations

ponent. These boxes have a rather large volume and are, thus, selected preferably. Figure 7.9 shows an exemplary decomposition of the search region obtained by LV after 20 iterations. The decomposition contains many boxes with a vertex equal to two in at least one component. While the clustering of GV is unwanted, the covering of the boundary obtained by LV may be advantageous as it provides a good idea of the shape of the nondominated set. Particularly if only a coarse initial representation shall be generated, then LV in combination with a modest number of subproblems provides a good result. The variants GH, GH2 and GHA, that all select the next box according to the contribution to the dominated hypervolume, also perform well. Particularly GH is of interest as it does not generate as many points as GV or LV, but provides at least optically a well-spread representation.

Summarizing the results for the sphere problem we see that all adaptive methods generate fewer infeasible or redundant problems than the a priori method with uniformly chosen parameters. The latter is outperformed by all adaptive methods with respect to cardinality and dominated hypervolume.

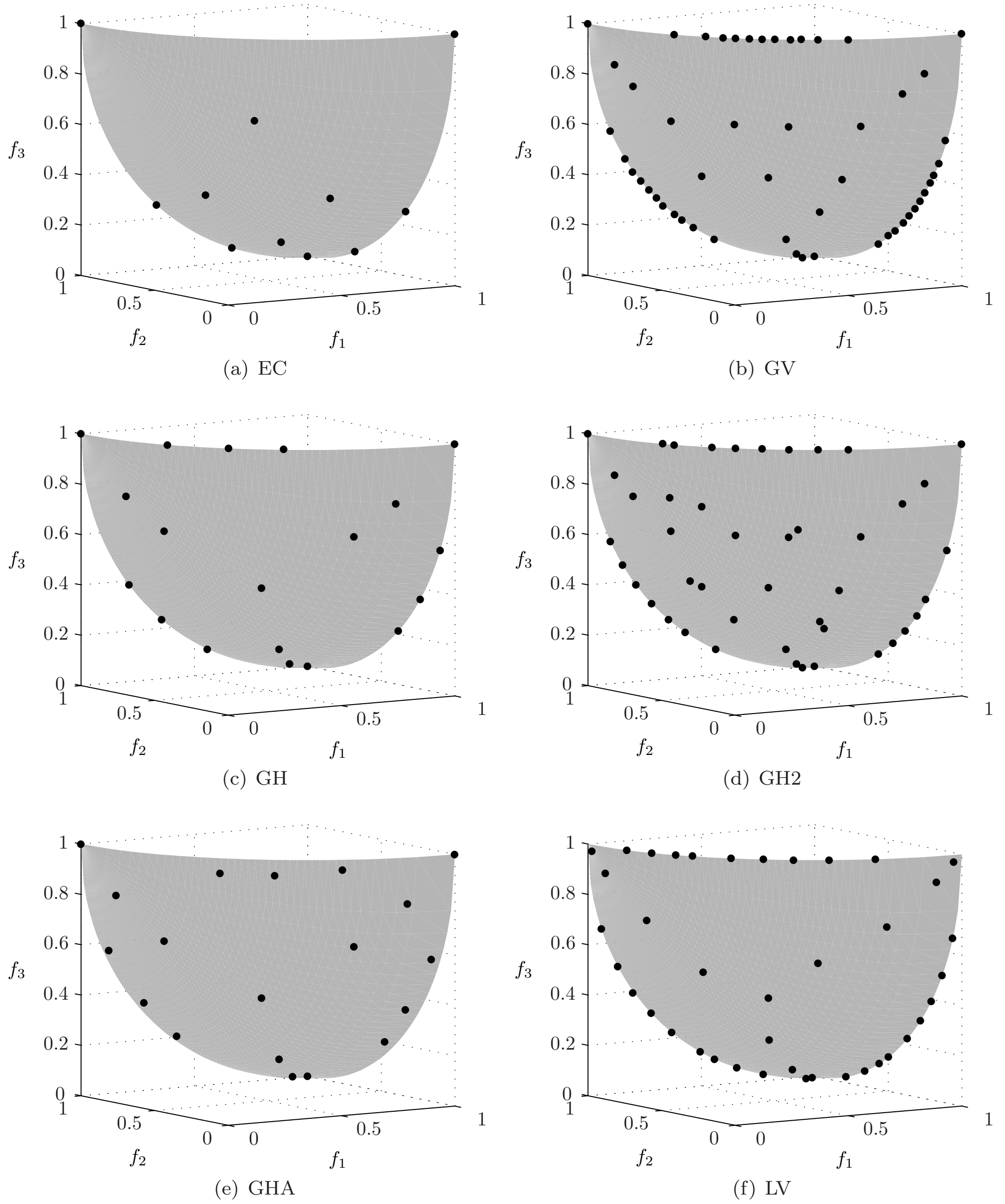


(a) EC (b) GV

(c) GH (d) GH2

(e) GHA (f) LV

Figure 7.10: Representations of the nondominated set computed by different parametric algorithms for test case 1 and 25 subproblems



(a) EC (b) GV (c) GH (d) GH2 (e) GHA (f) LV

Figure 7.11: Representations of the nondominated set computed by different parametric algorithms for test case 1 and 49 subproblems

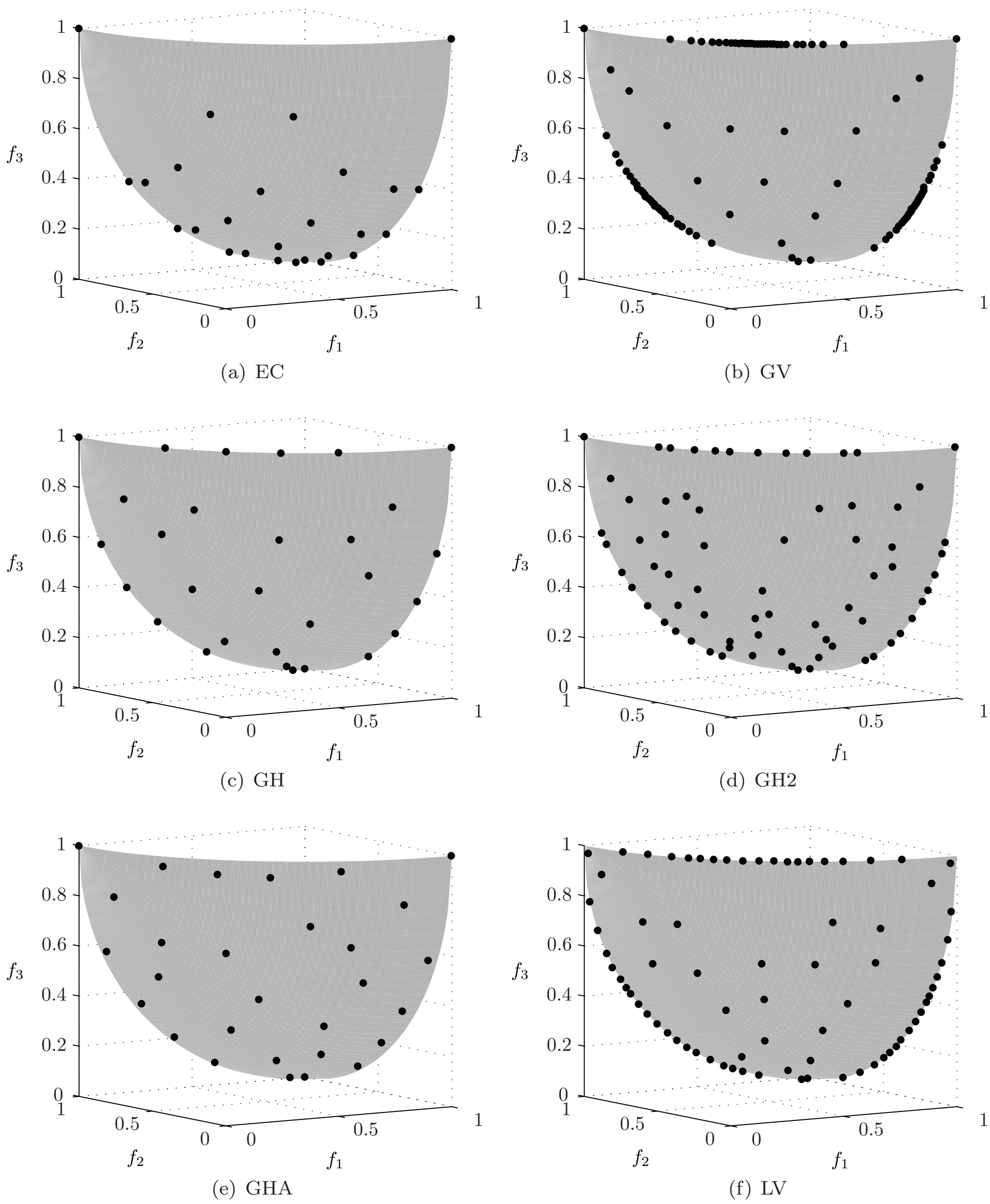


Figure 7.12: Representations of the nondominated set computed by different parametric algorithms for test case 1 and 100 subproblems

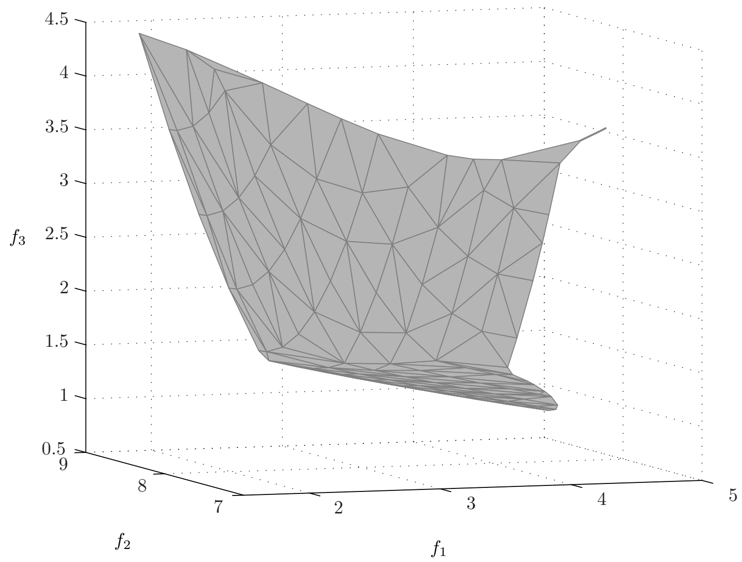


Figure 7.13: Approximate nondominated set of test problem 2

Test Problem 2

The second test problem is taken from Rennen et al. (2009) and is given by

$$\begin{array}{ll} \min & \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \\ \text{s.t.} & \begin{array}{lcl} x_1 & \geq & (x_2-9)^2+(x_3-3)^2, \\ x_2 & \geq & (x_1-4)^2+(x_3-3)^2, \\ x_3 & \geq & (x_1-4)^2+(x_2-9)^2, \\ x \in \mathbb{R}^3 & & \end{array} \end{array} \tag{7.4}$$

As in the first test problem, the efficient set equals the nondominated set and is $\mathbb{R}^3_+$-convex. However, no explicit description of it is available, wherefore we use a (piece-wise linear) approximation of the nondominated set for the visualization in Figures 7.13 to 7.16. As ideal point we obtain $z^I \approx (1.91, 7.08, 0.79)^\top$, as vector of individual maxima $z^M \approx (6.24, 10.92, 5.41)^\top$. The local nadir point is approximately $(4.88, 9, 4.38)^\top$ and equals the true nadir point. As in test case 1, z^M overestimates z^N in all components.

The results obtained are presented in Figures 7.14 to 7.16 and Table 7.6. Note that the visualization suggests a cluster of points at the lower right boundary, which, however, is caused by the perspective and could not be avoided without impairing

# SP		EC	GV	GH	GH2	GHA	LV
25 (+3)	$\lvert\mathcal{R}\rvert$	8	28	11	24	11	25
	r_H	58.86%	72.09%	67.85%	71.94%	67.93%	71.55%
49 (+3)	$\lvert\mathcal{R}\rvert$	14	52	18	42	19	45
	r_H	65.58%	73.51%	71.09%	74.10%	71.54%	74.24%
100 (+3)	$\lvert\mathcal{R}\rvert$	31	77	29	76	30	79
	r_H	70.27%	73.55%	73.22%	75.64%	73.50%	75.76%

Table 7.6: Cardinality of $\mathcal{R}$ and relative dominated hypervolume for test case 2

the visualization at another part of the nondominated set. In contrast, the cluster of points of GV observed at the left boundary is not caused by a visualization problem.

Comparing all methods, we can basically draw the same conclusions as for test case 1. However, the number of points obtained by EC increases to about one third of the subproblems solved, because of z^M overestimating z^N less than in test case 1. Consequently, the differences with respect to cardinality and dominated hypervolume between EC and the adaptive methods decrease. For $s = 100$, the final representation of EC contains slightly more points than that of GH and GHA. However, with respect to the relative dominated hypervolume all adaptive methods outperform EC.

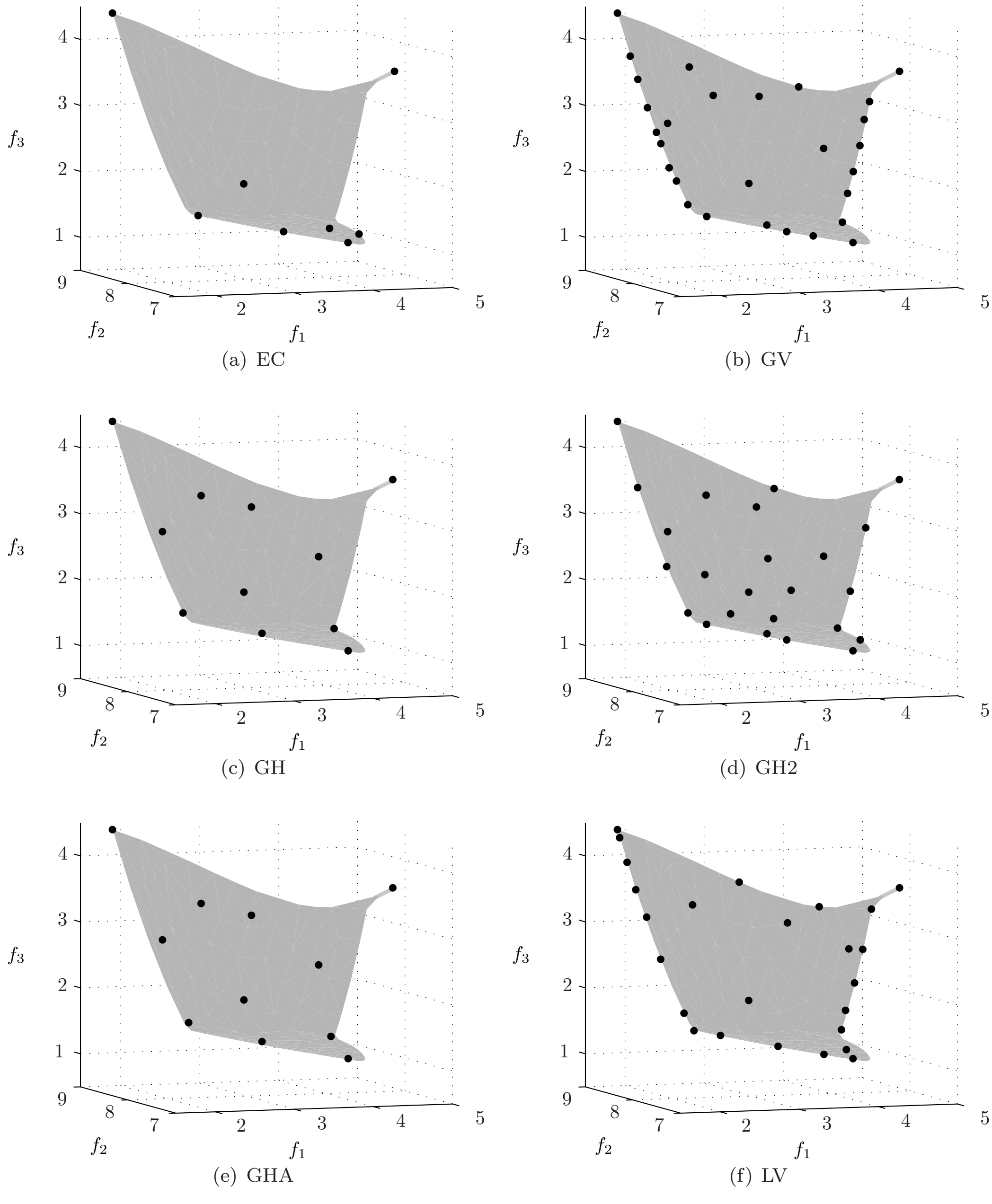
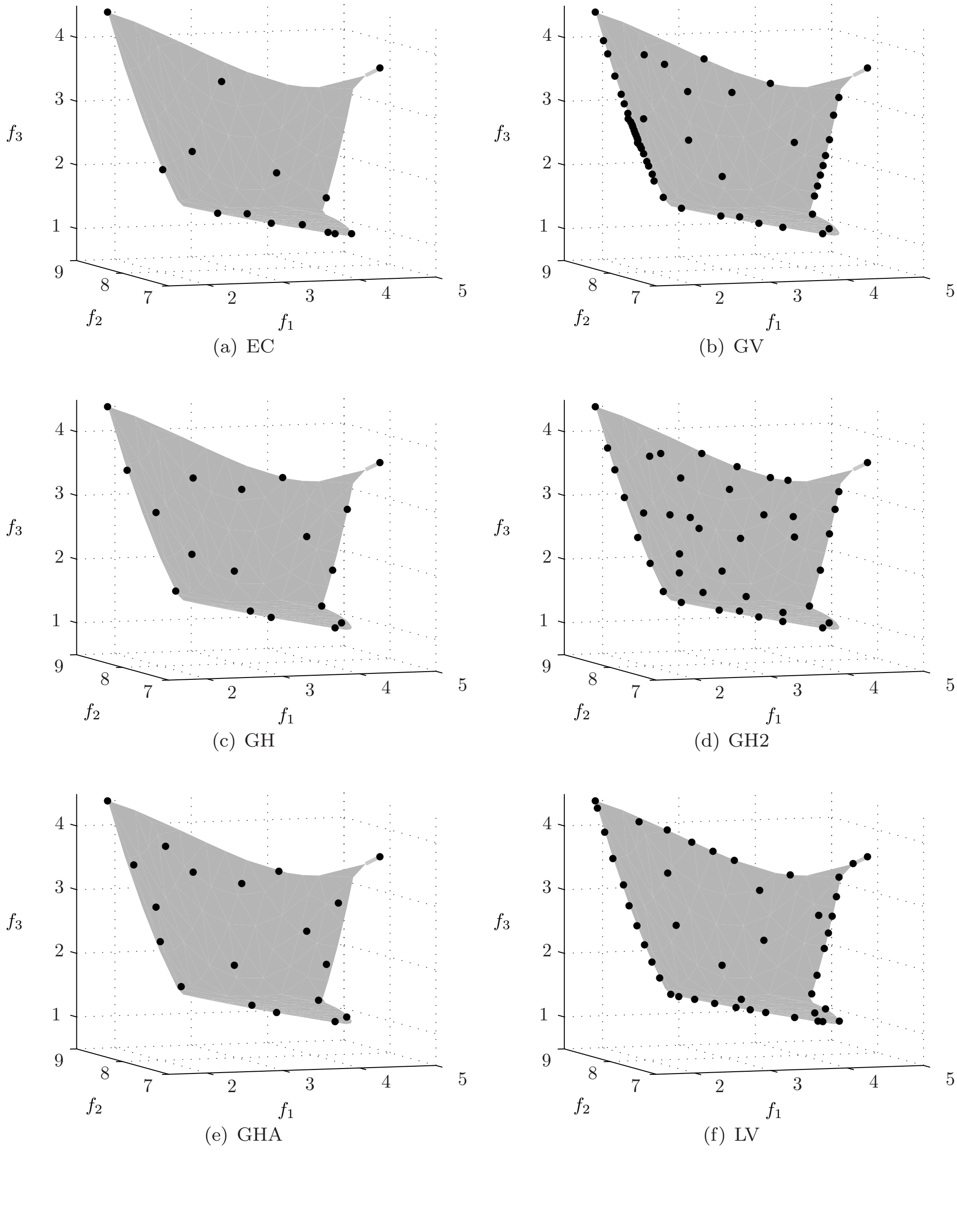


Figure 7.14: Representations of the nondominated set computed by different parametric algorithms for test case 2 and 25 subproblems



(a) EC (b) GV (c) GH (d) GH2 (e) GHA (f) LV

Figure 7.15: Representations of the nondominated set computed by different parametric algorithms for test case 2 and 49 subproblems

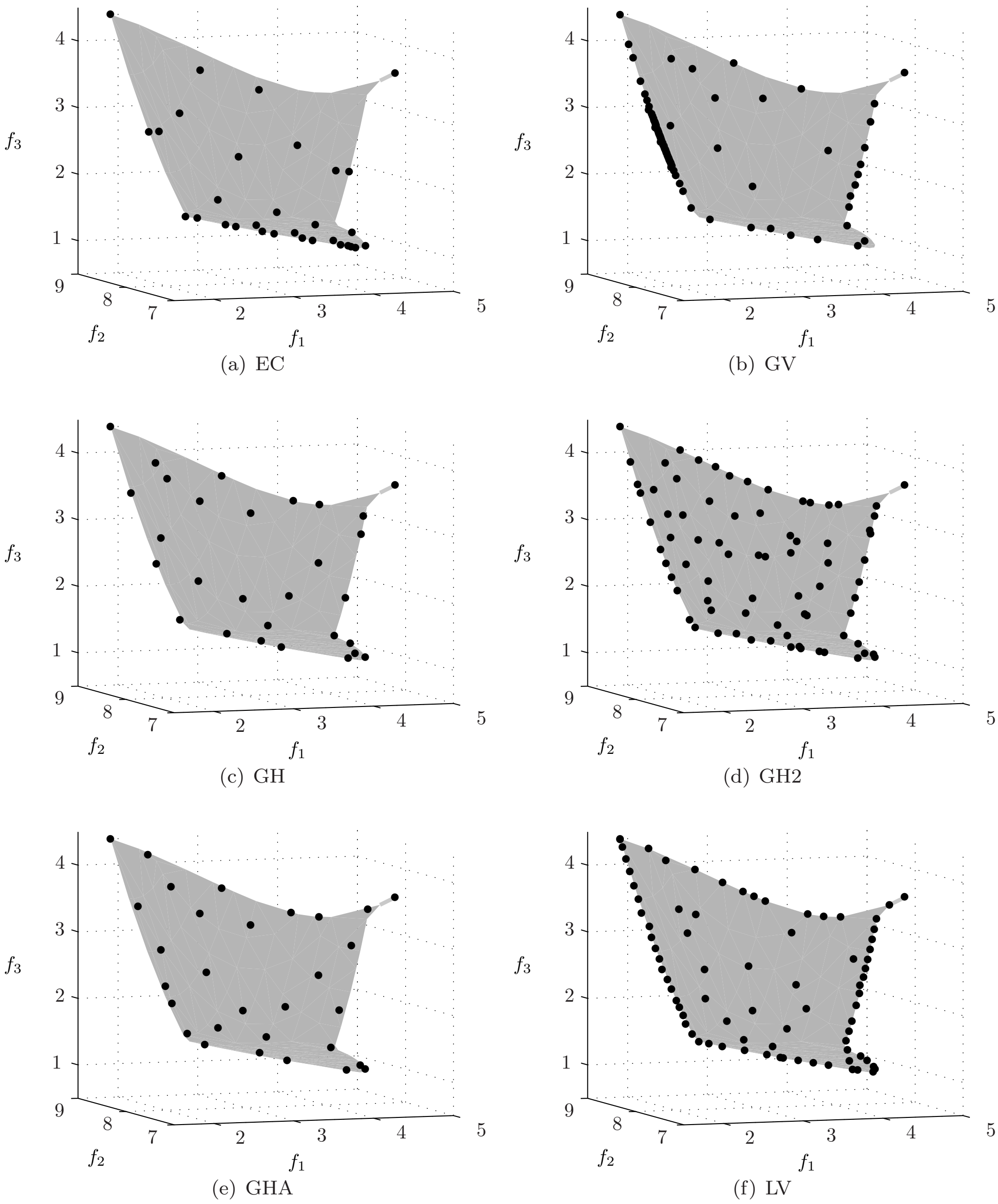


Figure 7.16: Representations of the nondominated set computed by different parametric algorithms for test case 2 and 100 subproblems

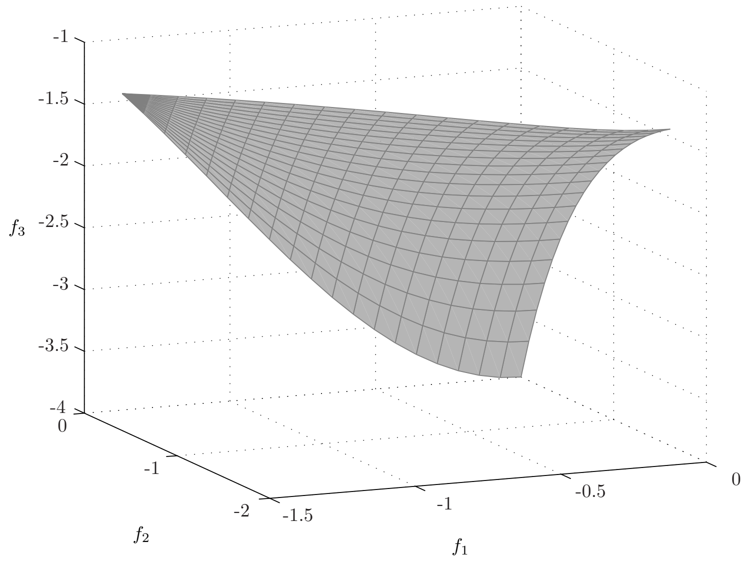


Figure 7.17: Nondominated set of test problem 3

Test Problem 3

This problem with an $\mathbb{R}^3_+$-non-convex nondominated set is taken from Eichfelder (2006), see also Eichfelder (2009a). It has also been used as a test case in Gourion and Luc (2010) and it is a slight modification of one of the test cases proposed in Kim and de Weck (2006). It is given by

$$\begin{aligned} \min \quad & -\begin{pmatrix} x_1 \\ x_2 \\ (x_3)^2 \end{pmatrix} \\ \text{s.t.} \quad & -\cos(x_1) - \exp(-x_2) + x_3 \leq 0, \\ & 0 \leq x_1 \leq \pi,\ 0 \leq x_2,\ 1.2 \leq x_3, \\ & x \in \mathbb{R}^3. \end{aligned} \tag{7.5}$$

The corresponding efficient set is

$$\begin{aligned} X_E = \{x \in \mathbb{R}^3 : 0 \leq x_1 \leq \arccos(0.2), 0 \leq x_2 \leq -\ln(1.2 - \cos(x_1)), \\ 1.2 \leq x_3 \leq \cos(x_1) + \exp(-x_2)\}, \end{aligned}$$

see Gourion and Luc (2010). The corresponding nondominated set is depicted in Figure 7.17. The ideal point is $z^I \approx (-1.37, -1.61, -4)^\top$ and the local nadir point equals the vector of individual maxima $z^M = (0, 0, -1.44)^\top$, thus, $z^M = z^N$.

The results are stated in Table 7.7 and in Figures 7.18 to 7.20. In this test case, EC performs quite well. About half of the subproblems solved by EC contribute

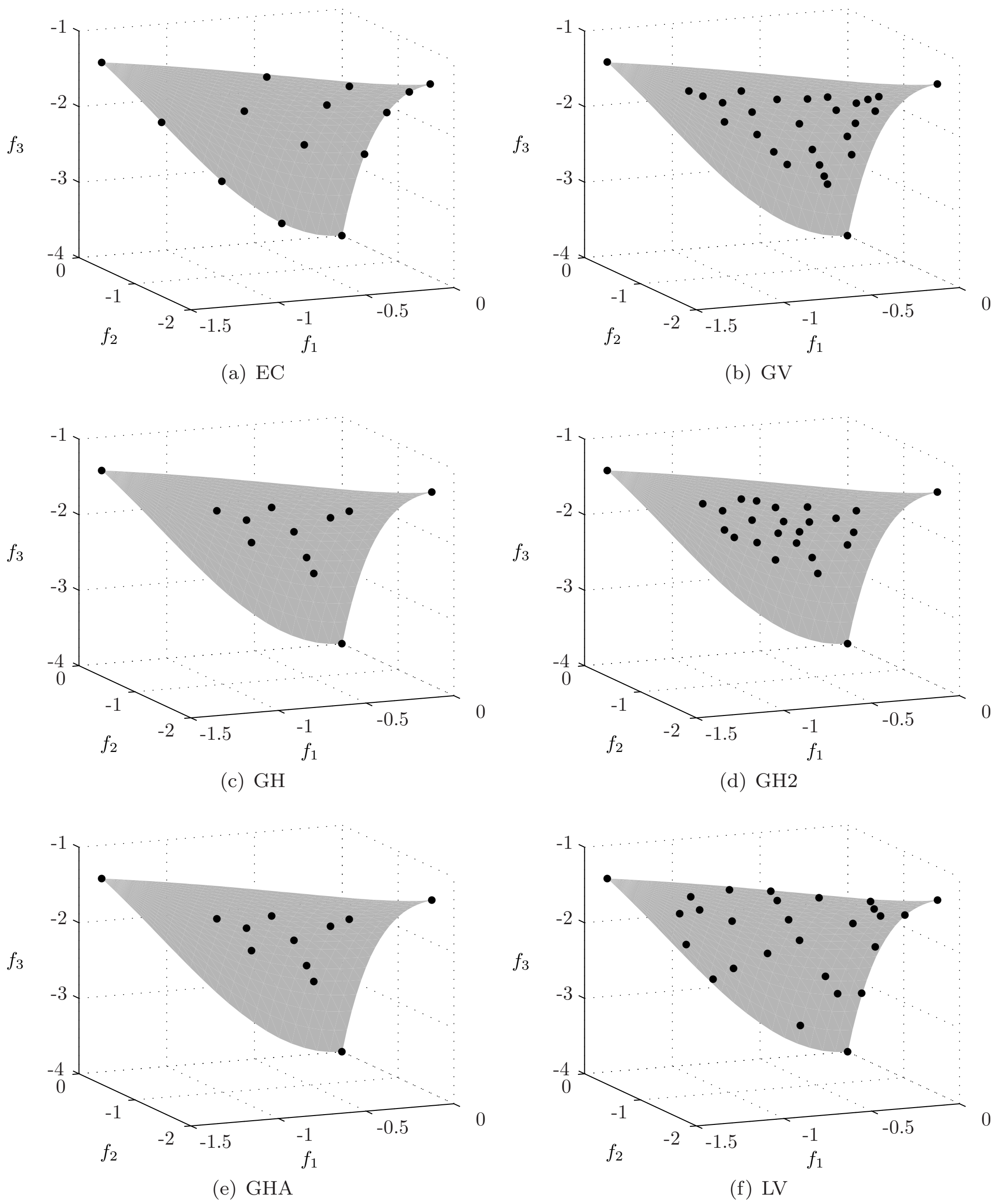
# SP		EC	GV	GH	GH2	GHA	LV
25 (+3)	$\lvert\mathcal{R}\rvert$	14	28	12	25	12	27
	r_H	6.43%	10.06%	8.05%	9.52%	8.05%	9.62%
49 (+3)	$\lvert\mathcal{R}\rvert$	26	52	18	45	19	49
	r_H	9.10%	11.18%	9.21%	10.89%	9.30%	11.25%
100 (+3)	$\lvert\mathcal{R}\rvert$	52	103	32	87	32	100
	r_H	11.14%	12.11%	10.56%	11.97%	10.30%	12.56%

Table 7.7: Cardinality of $\mathcal{R}$ and relative dominated hypervolume for test case 3

a point to the final representation. This is due to the fact that the nadir point is implicitly known, i.e., considerably less subproblems with redundant solutions are obtained. Comparing cardinality, we observe that EC generates less points than GV, GH2 and LV, but more points than GH and GHA for $s = 25, 49, 100$. However, when comparing EC to GH and GHA, a better dominated hypervolume is only obtained for $s = 100$. All other methods, i.e., GV, LV and GH2, yield a better dominated hypervolume than EC.

The representations depicted in Figures 7.18 to 7.20 reveal that all hypervolume-based methods generate basically all points at the interior of Z_N. This is caused by the fact that the reference point equals z^N and that all points at the boundary share one component with z^N. Consequently, these points are never inserted since their contribution to the dominated hypervolume would be zero. Interestingly, also GV represents the boundary of Z_N rather badly. Only LV provides a good cover of the boundary as well as the interior of Z_N. It generates twice as many points as EC while providing approximately the same quality for the same number of points (compare, e.g., the cardinality of LV for $s = 25$ to the one of EC for $s = 49$).

We conclude that the variant using the dominated hypervolume as refinement criterion does not cover the entire nondominated set, in particular not its boundary, whenever the true nadir point is known and is used as reference point for the computation of the dominated hypervolume. In this situation it might be advantageous to shift the reference point slightly. In contrast, the a priori method works quite well when the nadir point is known. Nevertheless, even in this situation, the adaptive volume-based variant LV produces better representations in terms of cardinality and relative dominated hypervolume.



(a) EC (b) GV (c) GH (d) GH2 (e) GHA (f) LV

Figure 7.18: Representations of the nondominated set computed by different parametric algorithms for test case 3 and 25 subproblems

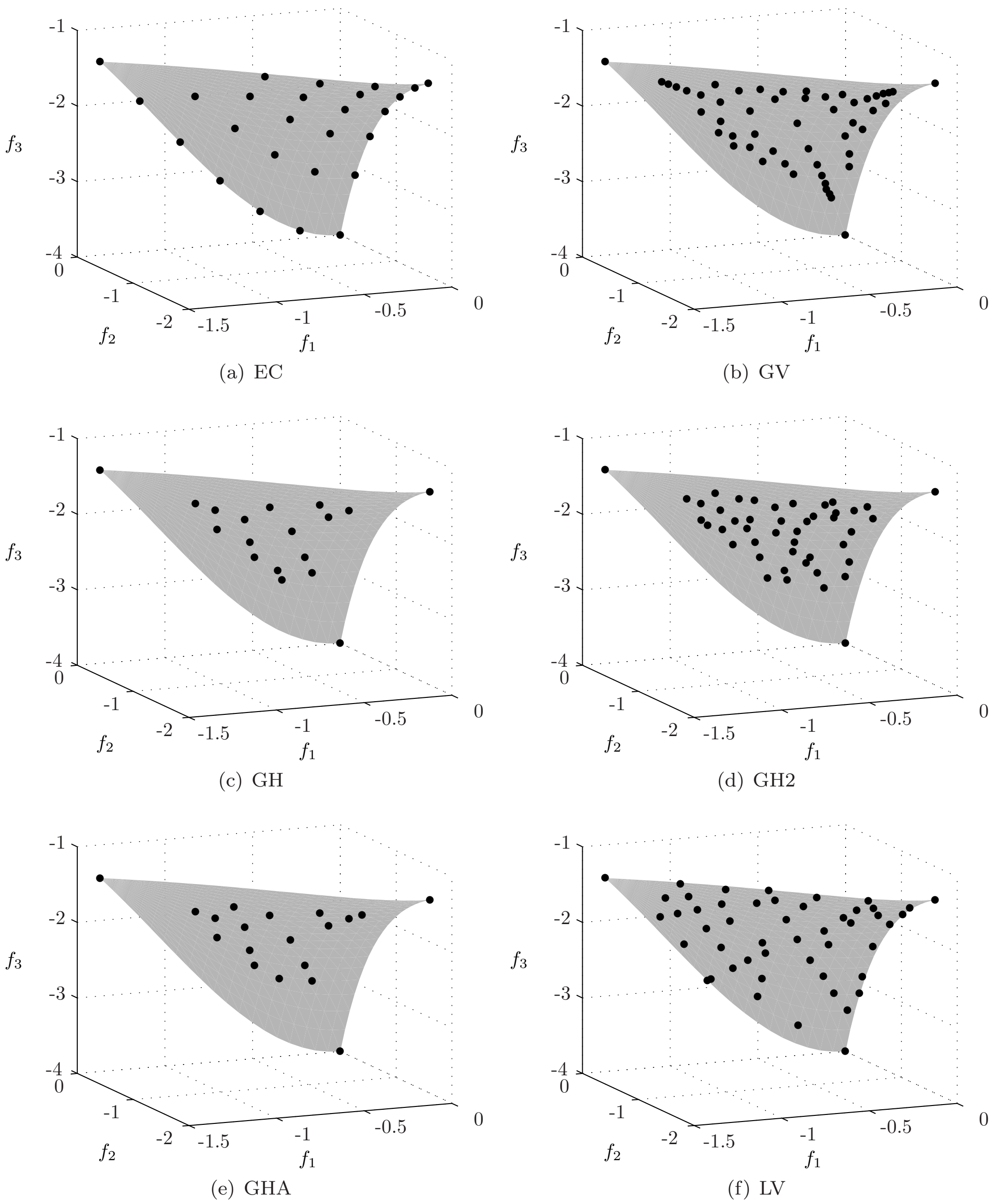
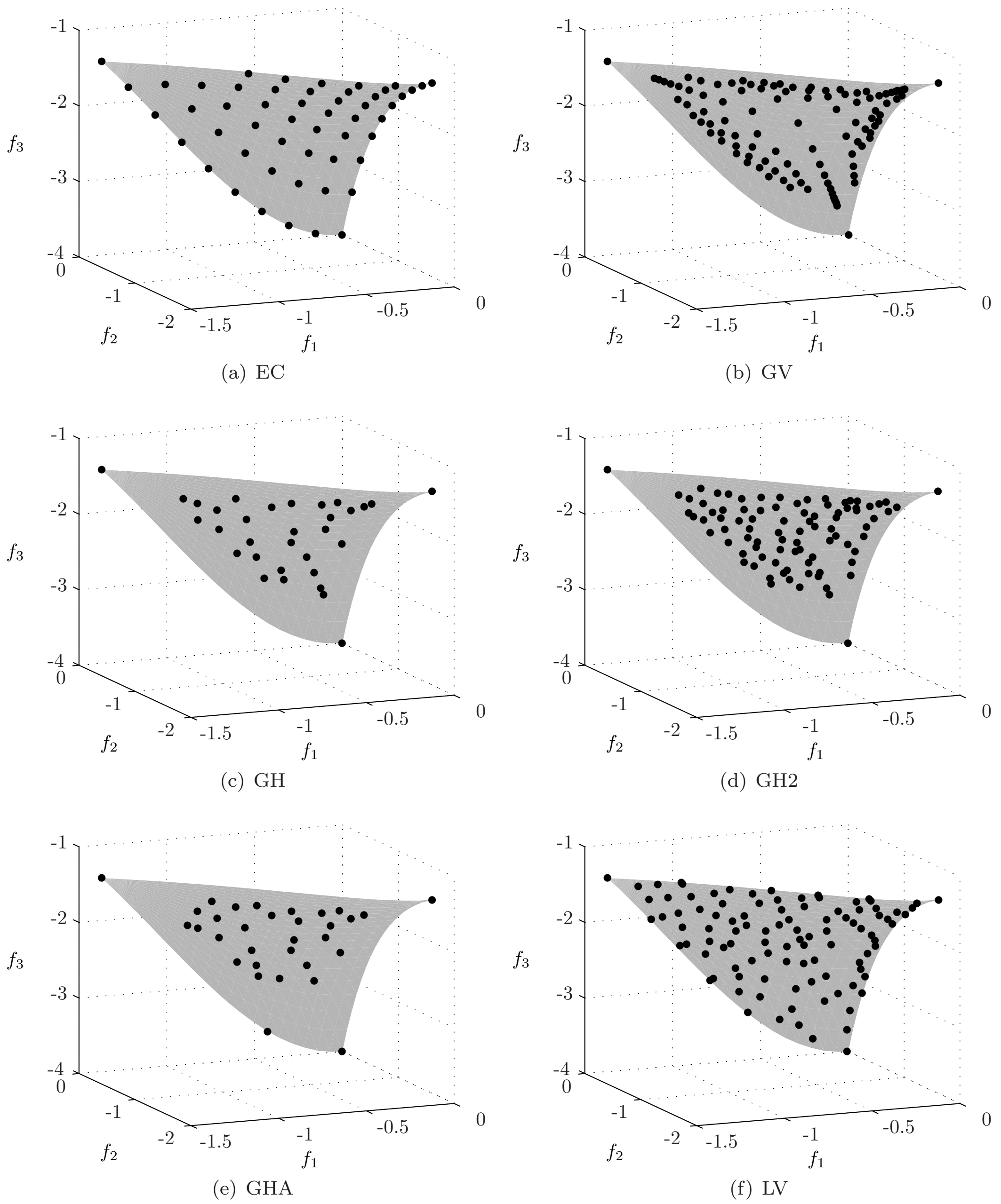


Figure 7.19: Representations of the nondominated set computed by different parametric algorithms for test case 3 and 49 subproblems



(a) EC (b) GV (c) GH (d) GH2 (e) GHA (f) LV

Figure 7.20: Representations of the nondominated set computed by different parametric algorithms for test case 3 and 100 subproblems

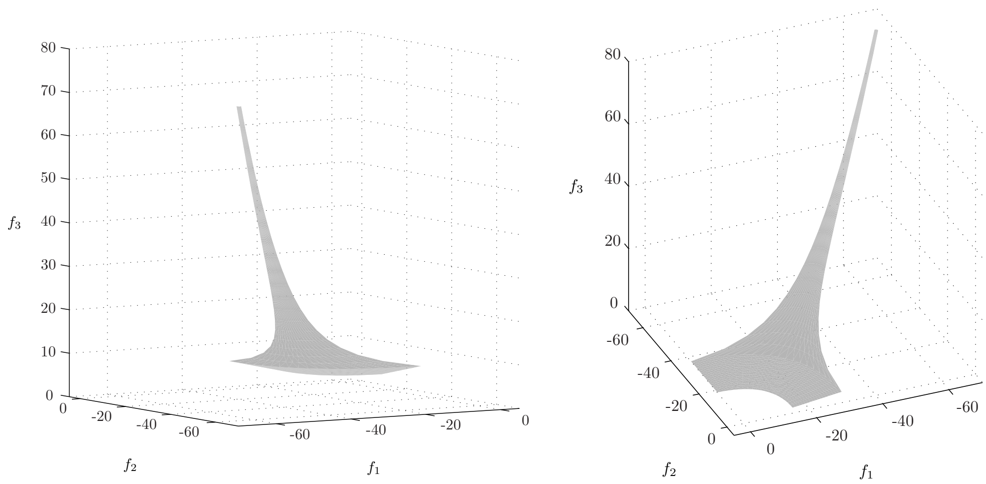


Figure 7.21: Nondominated set of test problem 4 from different perspectives

Test Problem 4

This tricriteria test problem from Deb et al. (2001), see also Deb et al. (2005), has been designed originally for testing evolutionary methods. It also served as a test case in Eichfelder (2006), see also Eichfelder (2009a). It is given by

$$\begin{array}{ll} \min & \begin{pmatrix} (1+x_3)(x_1^3x_2^2-10x_1-4x_2) \\ (1+x_3)(x_1^3x_2^2-10x_1+4x_2) \\ 3(1+x_3)x_1^2 \end{pmatrix} \\ \text{s.t.} & \begin{array}{rcccl} 1 & \leq & x_1 & \leq & 3.5, \\ -2 & \leq & x_2 & \leq & 2, \\ 0 & \leq & x_3 & \leq & 1. \end{array} \end{array} \tag{7.6}$$

The name of the problem stems from the form of the nondominated set that resembles a comet. In Deb et al. (2001) it is stated that the efficient set is given by

$$S := \{x \in \mathbb{R}^3 : 1 \leq x_1 \leq 3.5,\ -2 \leq x_2x_1^3 \leq 2,\ x_3 = 0\}.$$

This is, however, not correct. Consider, e.g., $\bar{x} := (3.5, 0, 0)^\top$. Then $\bar{x} \in S$ and $f(\bar{x}) = (-35, -35, 36.75)^\top$. Let $\tilde{x} := (2, 0, 1)^\top$. This point is feasible for (7.6) and $\tilde{x} \notin S$ holds. However, $f(\tilde{x}) = (-40, -40, 24)^\top \leqq f(\bar{x})$. Hence, S does not represent the efficient set of (7.6). Without proof, but confirmed by the numerical results, we claim that $f(S')$ with

$$\begin{aligned} S' :=& \{x \in \mathbb{R}^3 : 1 \leq x_1 \leq 3.5,\ -2 \leq x_2x_1^3 \leq 2,\ x_3 = 1\} \\ &\cup \{x \in \mathbb{R}^3 : x_1 = 1,\ -2 \leq x_2 \leq 2,\ 0 \leq x_3 \leq 1\} \end{aligned} \tag{7.7}$$

# SP		EC	GV	GH	GH2	GHA	LV
25 (+4)	$\|\mathcal{R}\|$	5	29	12	21	12	21
	r_H	71.08%	86.89%	86.21%	86.82%	86.24%	87.18%
49 (+4)	$\|\mathcal{R}\|$	8	53	17	39	17	37
	r_H	71.09%	87.01%	86.83%	87.42%	86.90%	87.96%
100 (+4)	$\|\mathcal{R}\|$	6	78	32	82	29	73
	r_H	77.81%	87.15%	87.76%	88.19%	87.27%	88.44%

Table 7.8: Cardinality of $\mathcal{R}$ and relative dominated hypervolume for test case 4

represents the nondominated set of (7.6). An illustration of $f(S')$ is given in Figure 7.21. Note that the same shape of the nondominated set has been obtained in the numerical study of Eichfelder (2006). The ideal point is $z^I \approx (-70.19, -70.19, 3)^\top$, the vector of individual maxima is $z^M = (289, 289, 73.5)^\top$. The local nadir point is given by $(2, 2, 73.5)^\top$, the true nadir point is $z^N = (4, 4, 73.5)^\top$.

According to Deb et al. (2001), the comet problem is difficult to solve by classic generating methods as the ε-constraint method. When choosing the third objective as objective of the ε-constraint method, Deb et al. (2001) estimate that 88% of the subproblems are redundant if a uniform set of ε-vectors is chosen.

The results for all tested variants are given in Table 7.8. The corresponding representations are depicted in Figures 7.22 to 7.24. Let us consider the graphics first. Already for $s = 25$, LV provides a nice coarse initial representation comprising 21 points. In contrast, for EC, only 5 points are generated. If we take into account that the given problem has four lexicographic minima which are generated beforehand, only one subproblem of EC contributes a point to the final representation for $s = 25$. This rather extreme result is also observed for $s = 49$ and $s = 100$. While only few points are computed by EC, the adaptive methods contribute many points to the final representation. Particularly LV provides a very good (graphical) coverage of the nondominated set. Note that only LV approximates the two points, by which the true nadir point is defined, well. Also GH yields a good representation of the nondominated set. Method GV produces a cluster of points, hence performs worse. Table 7.8 confirms the graphical observations.

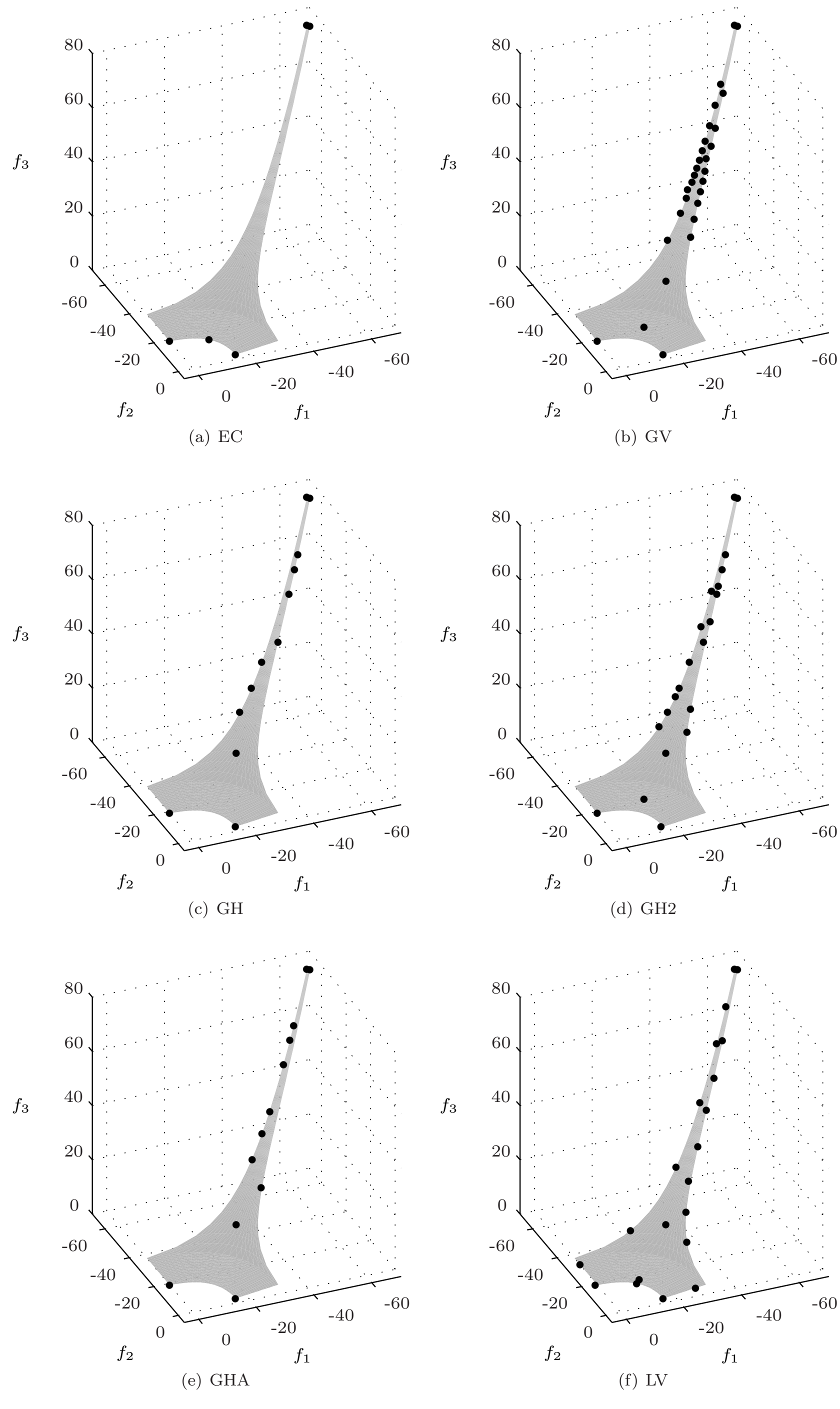


Figure 7.22: Representations of the nondominated set computed by different parametric algorithms for test case 4 and 25 subproblems

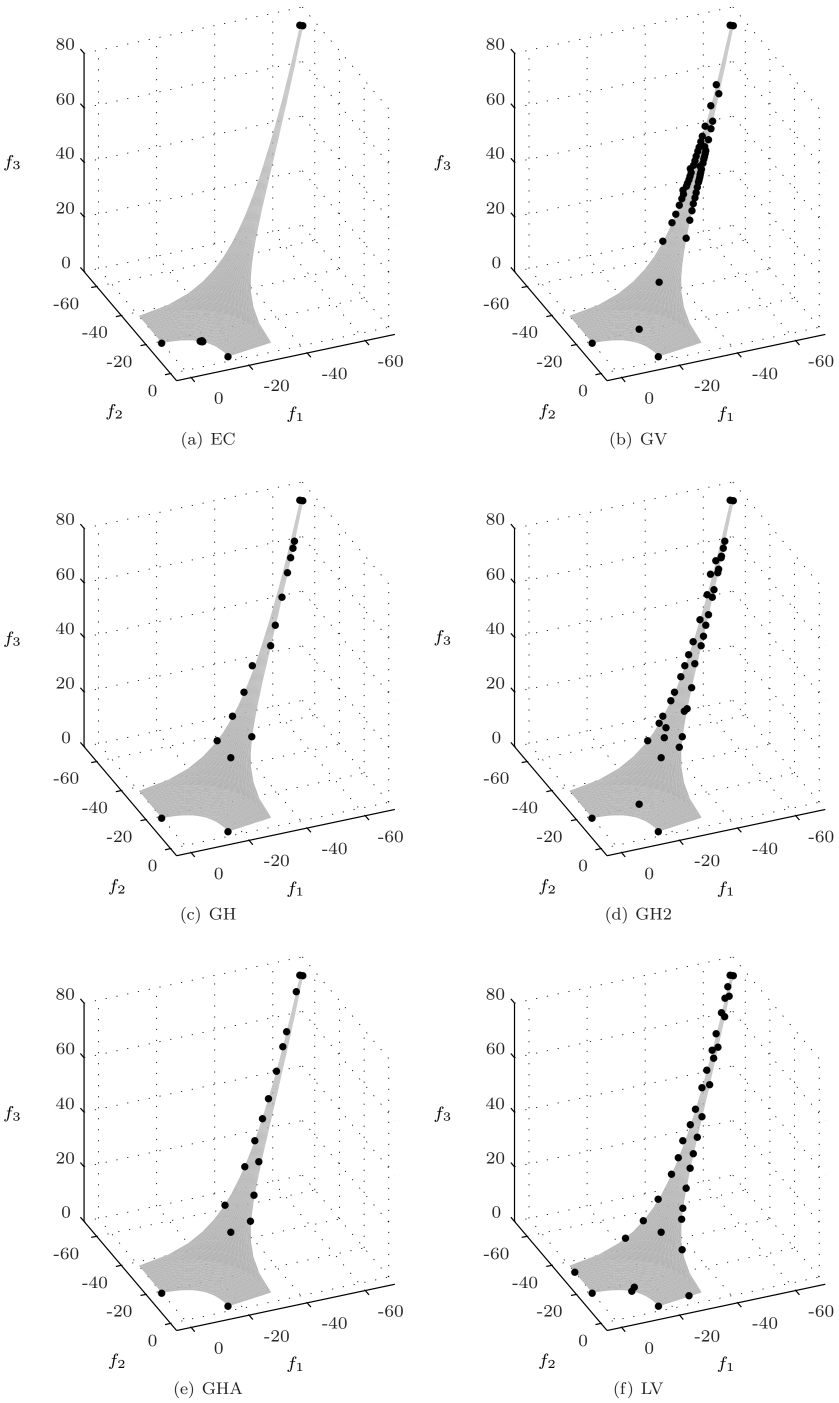


Figure 7.23: Representations of the nondominated set computed by different parametric algorithms for test case 4 and 49 subproblems

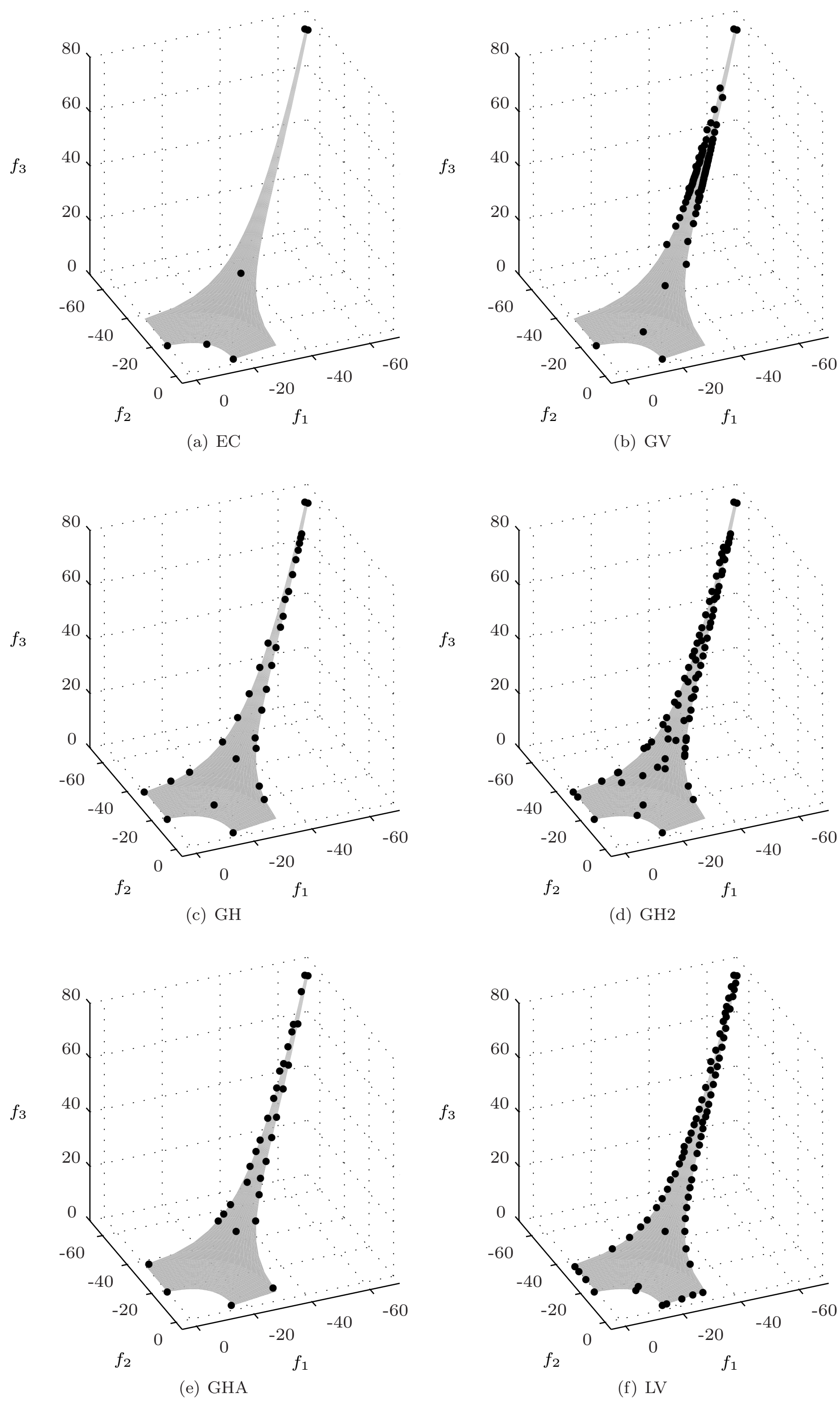


Figure 7.24: Representations of the nondominated set computed by different parametric algorithms for test case 4 and 100 subproblems

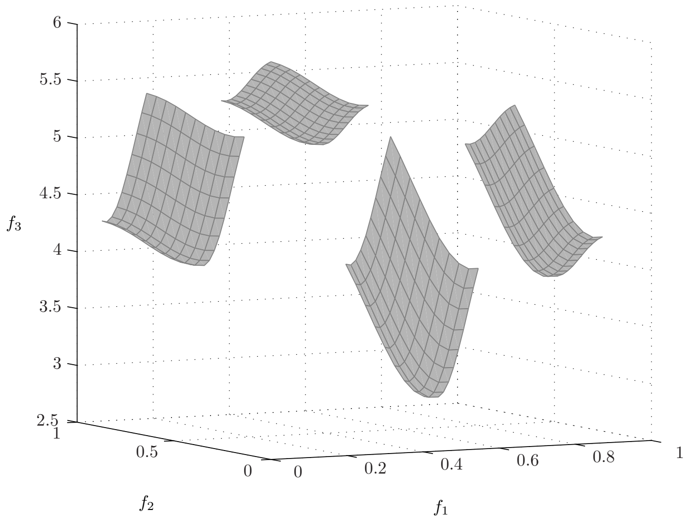


Figure 7.25: Nondominated set of test problem 5

Test Problem 5

This test problem is a modification of problem DTLZ7, see Deb et al. (2001) and Deb et al. (2005). The problem is designed for an arbitrary number of objectives and $n = m - 1 + k$ variables, where $k \in \mathbb{N}$ is a parameter. The authors suggest $k = 20$. The resulting problem formulation in the tricriteria case, which is also considered in Eichfelder (2006), is

$$\begin{aligned} \min \quad & \begin{pmatrix} x_1 \\ x_2 \\ g(x) \cdot \left(3 - \sum_{i=1}^{2} \left(\frac{x_i}{g(x)} \left(1 + \sin(3\pi x_i)\right)\right)\right) \end{pmatrix} \\ \text{s.t.} \quad & x_i \in [0,1] \quad \forall\, i = 1, \ldots, 2+k, \\ & x \in \mathbb{R}^{2+k} \end{aligned} \tag{7.8}$$

with

$$g(x) = 2 + \frac{9}{k} \sum_{i=3}^{2+k} x_i.$$

For $k = 20$, a nonlinear solver might experience problems when solving the subproblems. As our focus does not lie on improving existing nonlinear single objective

# SP		EC	GV	GH	GH2	GHA	LV
25 (+4)	$\lvert\mathcal{R}\rvert$	22	27	11	20	13	25
	r_H	36.48%	33.28%	29.77%	31.43%	32.77%	35.20%
49 (+4)	$\lvert\mathcal{R}\rvert$	41	47	19	40	18	44
	r_H	36.24%	36.53%	32.49%	34.14%	34.80%	38.25%
100 (+4)	$\lvert\mathcal{R}\rvert$	54	82	31	74	29	89
	r_H	38.82%	37.46%	35.32%	36.59%	35.71%	39.61%

Table 7.9: Cardinality of $\mathcal{R}$ and relative dominated hypervolume for test case 5

solvers, we consider the modified problem

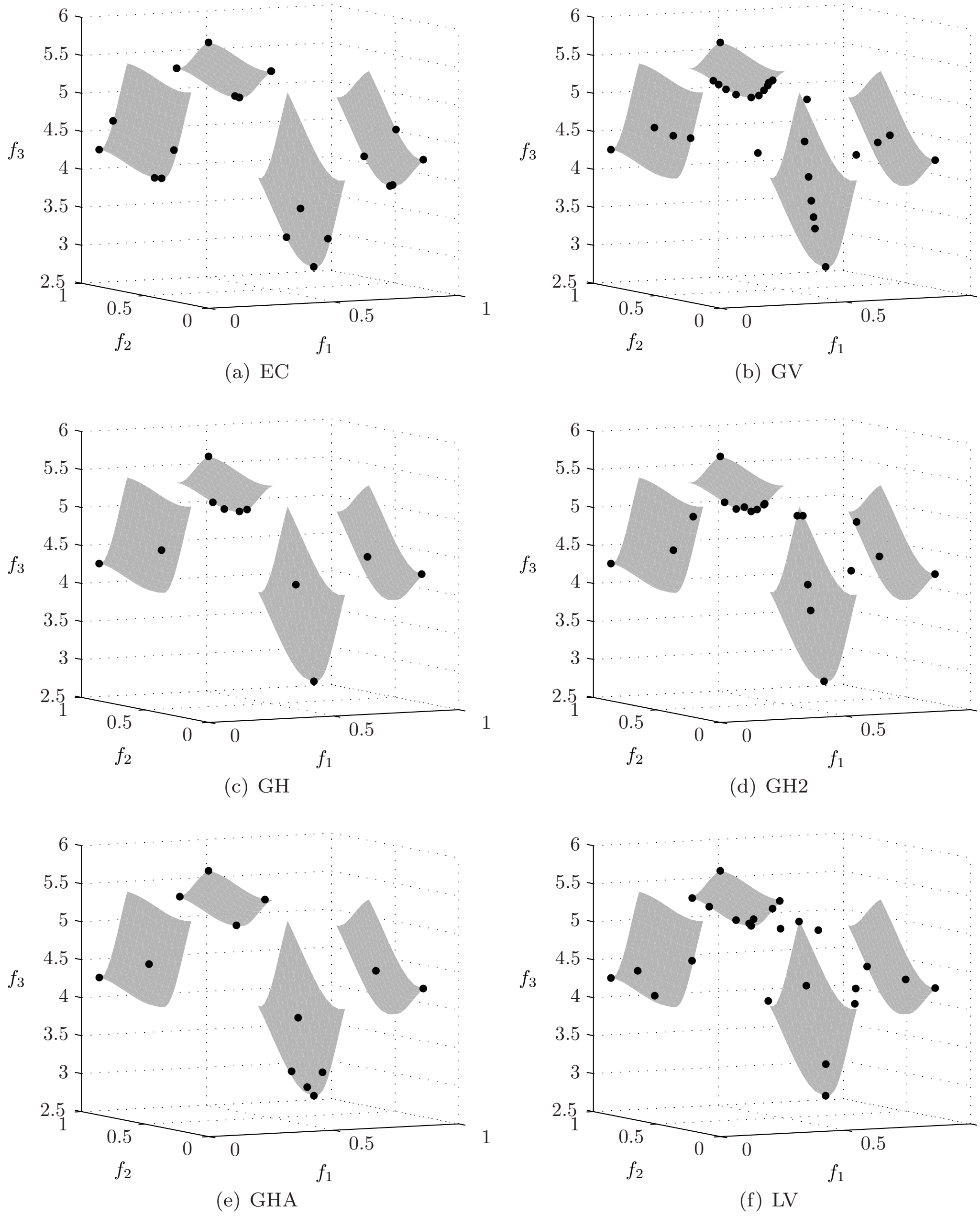
$$\begin{aligned} \min \quad & \begin{pmatrix} x_1 \\ x_2 \\ 6 - \sum_{i=1}^{2} \left(x_i \left(1 + \sin(3\pi x_i)\right)\right) \end{pmatrix} \\ \text{s.t.} \quad & x_i \in [0,1] \quad \forall\, i = 1,2, \\ & x \in \mathbb{R}^2 \end{aligned} \tag{7.9}$$

with only two variables. Note that (7.9) is not a special case of (7.8), as $g(x)$ is not defined for $k = 0$. However, the nondominated set of (7.9) is the same as of (7.8) for arbitrary $k \in \mathbb{N}_0$, which can be easily seen. The third objective of (7.8) can be reformulated as

$$f_3(x) = 3g(x) - \sum_{i=1}^{2} x_i \left(1 + \sin(3\pi x_i)\right).$$

Since $g(x)$ does not depend on x_1 and x_2, and, at the same time, the variables $x_3, \ldots, x_{k+2}$ only occur in $g(x)$, which only occurs in the third objective function, we can eliminate the variables $x_3, \ldots, x_{k+2}$ from (7.8). Indeed, for any nondominated point of (7.8), $x_3 = \cdots = x_{k+2} = 0$ must hold (see also Deb et al. (2001), Deb et al. (2005), Eichfelder (2006)). This implies $g(x) = 2$. As $f(x) = (x_1, x_2, f_3(x))^\top$ only depends on x_1 and x_2, the nondominated sets of (7.8) and (7.9) are the same. An illustration of the $\mathbb{R}^3_+$-non-convex nondominated set of (7.9) is provided in Figure 7.25. It consists of four disconnected parts. The ideal point is $z^I = (0, 0, 2.61)^\top$, the vector of individual maxima is $z^M = (1, 1, 6)^\top$ and the local nadir point is given by $(0.86, 0.86, 6)^\top$, which equals z^N.

The results are given in Table 7.9 and Figures 7.26 to 7.28. Also in this test case, EC performs pretty well, because z^N is well estimated by z^M. Moreover, the minimization of the third objective is advantageous for this problem due to the shape of the nondominated set, which is symmetric with respect to the first and second component. For $s = 25$, EC even provides the highest relative dominated hypervolume of all tested variants, hence outperforms the other methods. Observing the figures we see that only LV can compete with EC in this test case. Indeed, LV provides a higher cardinality and, for $s = 49$ and $s = 100$, a higher relative dominated hypervolume. Note that in this test case, the single-criterion solver seems to compute local minima, as the representations computed by all adaptive variants besides GH contain dominated points.



(a) EC (b) GV (c) GH (d) GH2 (e) GHA (f) LV

Figure 7.26: Representations of the nondominated set computed by different parametric algorithms for test case 5 and 25 subproblems

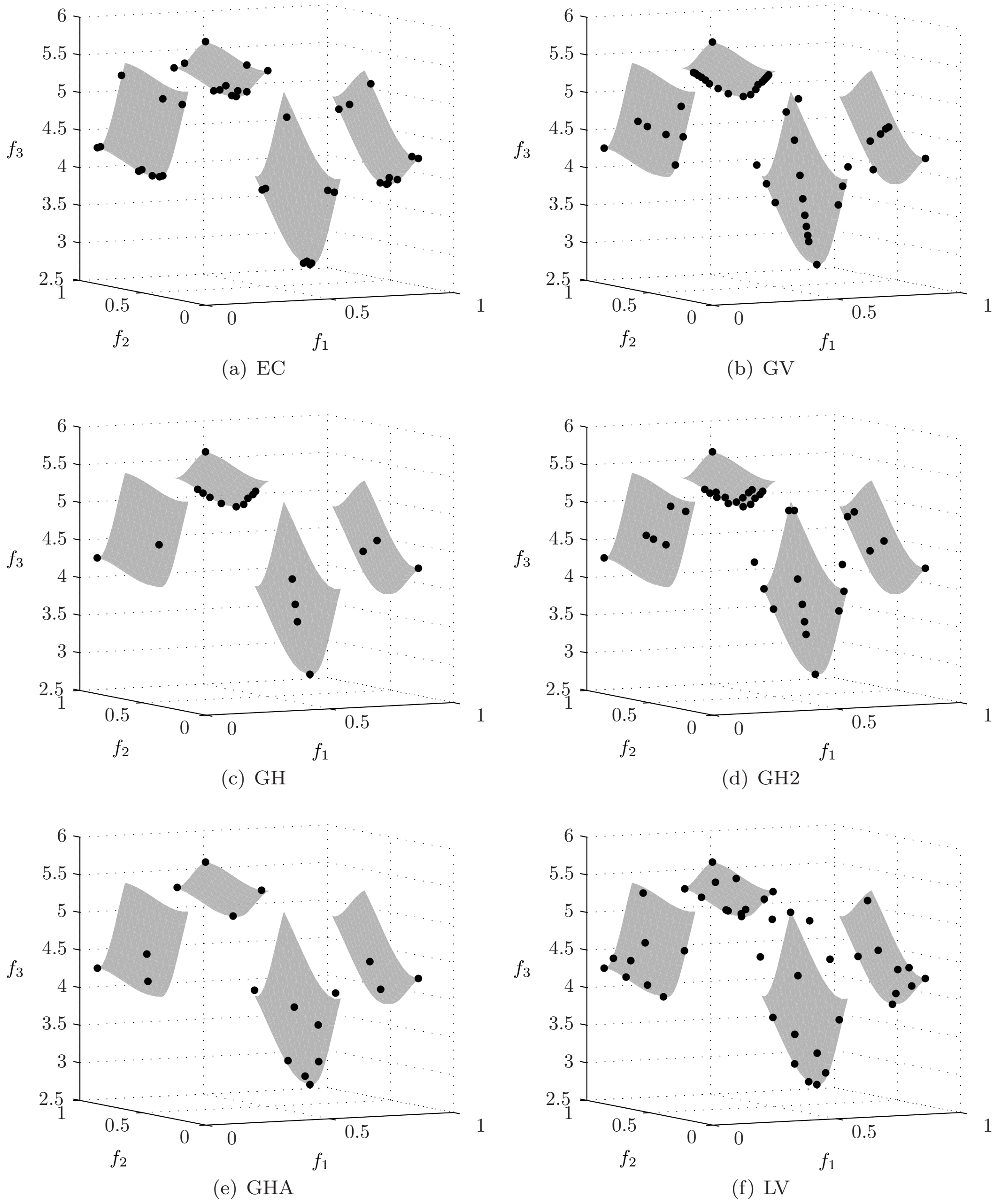


Figure 7.27: Representations of the nondominated set computed by different parametric algorithms for test case 5 and 49 subproblems

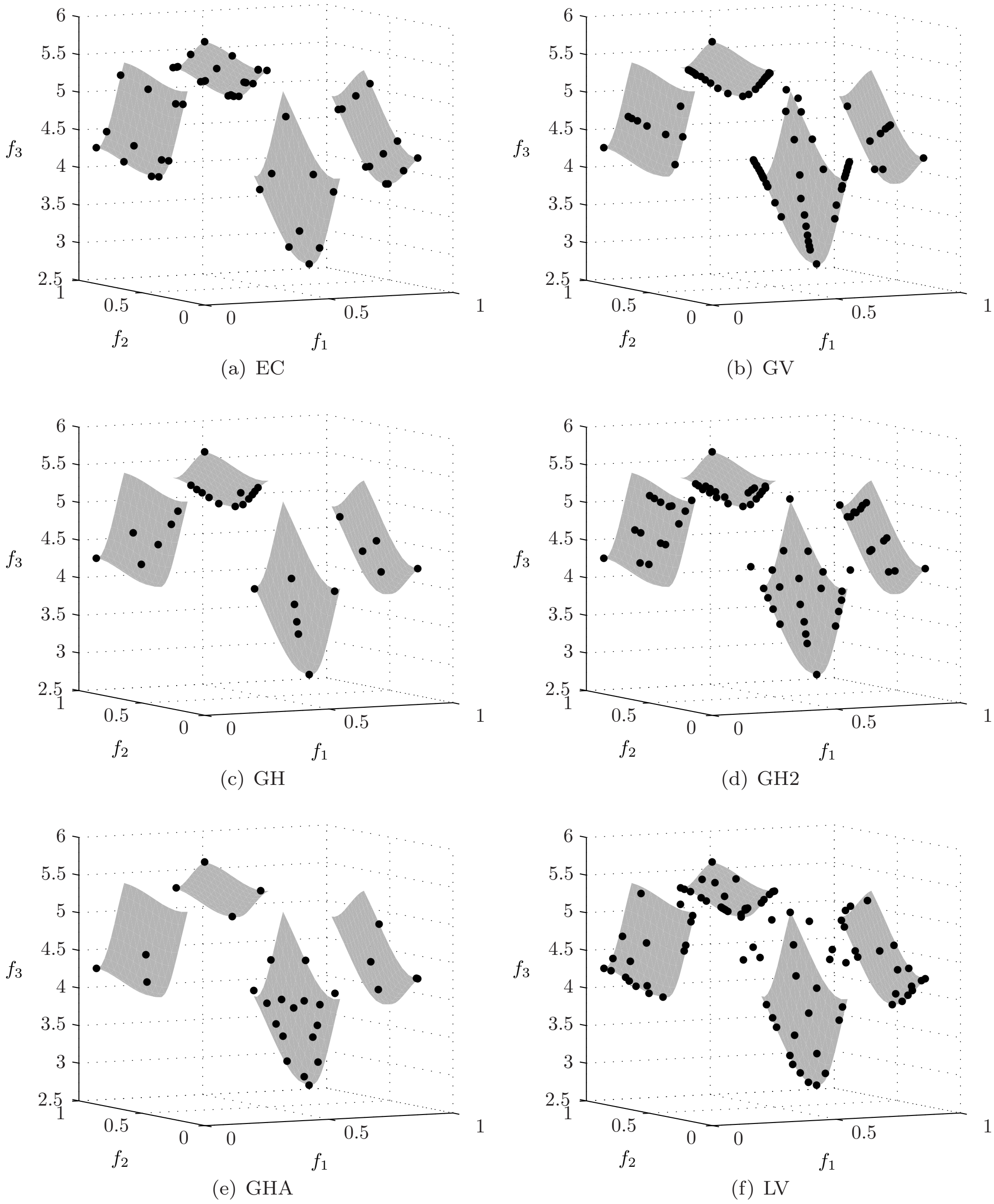


Figure 7.28: Representations of the nondominated set computed by different parametric algorithms for test case 5 and 100 subproblems

7.4 Conclusion and Further Ideas

In this chapter we applied different variants of a new parametric algorithm with an adaptive parameter scheme to continuous bicriteria and tricriteria optimization problems.

In the bicriteria case, we tested four different adaptive variants, in which the next box is selected either as the box with the largest area or the box for which the largest contribution to the dominated hypervolume is obtained. Moreover, we varied the reference point of the weighted Tchebycheff scalarization. The resulting four adaptive variants were compared to an a priori method and the method of Eichfelder (2006). We applied all variants to two test problems, one with a $\mathbb{R}^2_+$-convex and one with an $\mathbb{R}^2_+$-non-convex, disconnected nondominated set. All variants were evaluated with respect to cardinality, uniformity, coverage and relative dominated hypervolume. In general, we observed that all variants including the a priori method generate acceptable representations of the nondominated set. We obtained more uniform representations with our adaptive variants compared to the a priori approach. Comparing our methods among each other, it turned out that much better results were obtained when local ideal points instead of a fixed (global) ideal point are used. While the hypervolume-based variant contributes less points to the representation than the volume-based variant, results with the former variant were also very appealing.

In the tricriteria case, we evaluated adaptive and a priori approaches for test problems with $\mathbb{R}^3_+$-convex and $\mathbb{R}^3_+$-non-convex nondominated sets. We considered cardinality and the relative dominated hypervolume as quality criteria. The results show that adaptive methods are particularly superior to a priori methods when the nadir point is not available and the used upper bound z^M overestimates the true nadir point significantly. In cases where the nadir point is known, an a priori method performs quite well, in general, but might still be outperformed by an adaptive method with respect to cardinality and relative dominated hypervolume. In general, adaptive methods generate considerably less infeasible and/or redundant subproblems than a priori methods.

We conclude that adaptive algorithms are well suited to generate incomplete representations of the nondominated set. Particularly when only a moderate number of subproblems is solved, they provide a good coarse representation of the nondominated set.

With respect to efficiency, the proposed algorithms can still be improved. In the tricriteria case, we use the method of Beume et al. (2009) in order to compute the contribution of each point to the dominated hypervolume in each iteration and each box of the decomposition. An incremental computation would be beneficial such that the dominated hypervolumes are not recomputed from scratch in every iteration and for every box.

So far, only cardinality and dominated hypervolume have been evaluated as quality measures in the tricriteria case. Further research should address the incorporation of coverage, which requires the identification of neighboring points. As we also observed cases where it was not satisfactory to consider only points of the final representation for estimating coverage, the development of alternative approaches would be useful.

Furthermore, for more than two criteria it would be interesting to couple our algorithm with the sensitivity-based method of Eichfelder (2006) by using our approach in a (short) first stage and the method of Eichfelder (2006) in a second stage. Thereby, based on a (coarse) initial representation, the generation of a nearly equidistant representation of the nondominated set can be expected.

In this chapter we computed discrete sets of (nondominated) points. Based on these sets, piece-wise linear approximations could be constructed and comparisons with the gauge method of Klamroth et al. (2002) could be drawn. Lastly, the application to discrete problems, for which an incomplete representation is sought, could be considered.

8 Multiobjective Optimal Control of Sewer Networks

8.1 Introduction

In this chapter, we consider the optimal control of sewer networks which is a real-world application that involves multiple goals. Thereby, the task is to determine a real-time optimal control of the actuators such that a sewer network is operated best with respect to several objectives. The traditional approach consists in an offline control. Thereby, a database of control decisions in form of if-then-rules is created by simulating a huge number of different scenarios before the system is actually operated. The control decisions needed in the operational phase are then obtained from the database. In contrast, online control comprises the computation of the control settings during the operational phase. Therefore, an online monitoring of data as rainfall, runoff as well as information on water level, flow rate and water quality in the sewer network is needed. With the help of a process model that processes the available data, future state developments of the system are predicted. Based on the response of the process model, optimal control settings can be computed. Note that in the engineering context 'optimal' often means that the solution has been improved sufficiently with respect to some reference solution.

Online control of sewer networks is challenging due to several reasons: firstly, the shallow water flow in networks is described by a system of hyperbolic partial differential equations, i.e., one has to deal not only with nonlinearities but also with discontinuities of solutions, e.g., shock waves. Secondly, in online control the time to find control settings is typically limited to five minutes. Therefore, a compromise has to be found between a process model that describes the complex hydrodynamic process sufficiently well and, on the other hand, computes system states in the required short time. A third challenge is the size of the network, which, already for small cities, typically contains a large number of actuators, i.e., variables of the underlying control problem. A meaningful reduction is required to keep these networks manageable. We refer to Martin et al. (2012) for further details.

Despite of these challenges, optimal control of sewer networks is an interesting topic for a multicriteria analysis, as the problem formulation contains multiple objectives. Two main strategies are pursued in the literature. The first combines the objectives to a weighted sum, see, e.g., Marinaki and Papageorgiou (2005). The weights are determined by a trial-and-error procedure, which starts by assigning some initial values to the parameters. Then, similar to the creation of if-then-rules for the offline control, various representative inflow scenarios are tested in order to adjust the weights. Thereby, however, no systematic approach is reported. The second strategy is the application of evolutionary multicriteria optimization (EMO), see, e.g., Rauch and Harremoës (1999) or Muschalla (2008) for applications in the wastewater management context. However, since EMO methods are typically based on a multitude of function evaluations, which, in case of a hydrodynamic process model, require a large number of time consuming simulation runs, and since an interaction with a decision maker does not seem useful within an online optimization process, EMO appears to be less suited for the use in online optimal control.

In Dächert and Klamroth (2012) we presented a preliminary, scenario-based offline investigation taking the multiple objectives arising in wastewater management into account. Therefore, we studied the applicability of the weighted sum, the ε-constraint and the augmented weighted Tchebycheff method with a priori and a posteriori parameter schemes for the determination of a 'good' representation of the nondominated set. Our work was part of a research project funded by the BMBF, the German Ministry of Education and Research. Its main goal was to develop innovative methods in the context of water supply and wastewater management based on state of the art mathematical and engineering knowledge. The contributions of all research groups are collected in Martin et al. (2012), to which we refer for all modeling and numerical aspects as well as for a detailed problem description.

This chapter is organized as follows. In Section 8.2, we present the objectives arising in wastewater management together with a short literature review. In Section 8.3, we describe the computational setup, including the single-objective optimizer and different parameter schemes. In Section 8.4, we show results of our scenario-based analysis and discuss them particularly with respect to the adaptive parameter selection that is promoted in this thesis. This chapter has already been published in Dächert and Klamroth (2012).

8.2 Objectives in Wastewater Management

In the literature several objectives for the optimal control of sewer networks are stated. Marinaki and Papageorgiou (2005) formulate five goals, namely avoiding overloads in storage elements without overflow, minimizing overflows, maximizing the utilization of the wastewater treatment plant, obtaining a desired distribution of the reserve storage volume and avoiding abrupt changes of outflows. In Pleau et al. (2005) the experiences of the Québec urban community are described, where since the year 2000 a real-time optimal control system has been installed and monitored. The system covers the minimization of overflows, the maximization of the use of the wastewater treatment plant capacity and the minimization of accumulated volumes in the tunnels as well as set point variations. Furthermore, the preferential treatment of some overflow sites and the dewatering of the upstream tunnel are included in the system. While in Marinaki and Papageorgiou (2005) and Pleau et al. (2005) pollution is not considered as an objective, it is listed as an important goal among the future trends in Schütze et al. (2004). Moreover, pollution in the receiving water is considered as one of the objectives in Rauch and Harremoës (1999) and Weinreich et al. (1997). The minimization of pollution and economical costs in an estuary is studied in Alvarez-Vazquez et al. (2010). Based on these references we consider the following objectives as particularly relevant in the context of optimal control of sewer networks:

1. Minimization of overflows, i.e., the amount of water which has to be released due to capacity limitations of the network

2. Minimization of pollution mass in the released water

3. Minimization of variations of inflow to the wastewater treatment plant (WWTP), as the WWTP works best when inflow is as constant as possible

4. Minimization of variations of all controllable weirs and pumps in order to get a smooth control profile, i.e., unnecessary and sudden opening and closing of the controllable elements should be avoided

5. Maximization of inflow to the WWTP such that unnecessary storage of water in the network is prevented

We assume that the data which is necessary to consider these objectives is available, i.e., especially the network contains sensors to measure the required data online. Concerning pollution load, this is an idealized situation since in practice online measurements of the chemical oxygen demand (COD) are usually not available.

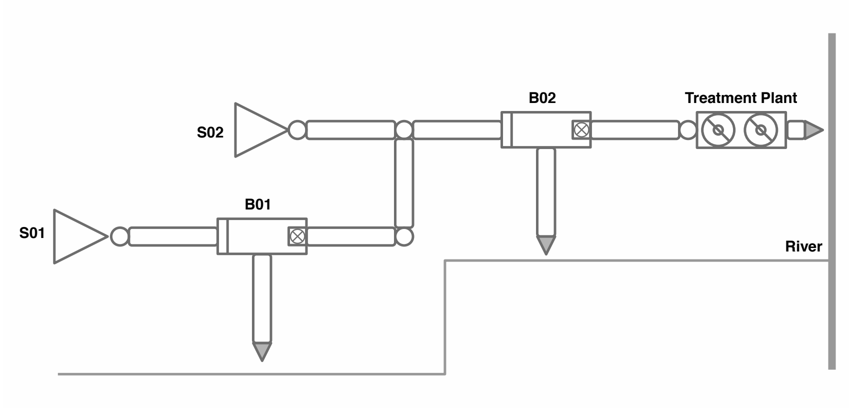


Figure 8.1: Test network from Heusch and Ostrowski (2012)

8.3 Computational Setup

Test Networks

In our computational study we consider a network from Heusch and Ostrowski (2012) with two inflows (S01, S02) and two storage units (B01, B02), depicted in Figure 8.1. Each of the storage units has a controllable pump and an overflow to a nearby river. We also study a subnetwork of it, which is depicted in Figure 8.2. It consists of one single inflow node, a channel connecting this inflow node to a storage unit, and a controllable pump at the end of the storage unit, through which water enters another channel, which leads to the wastewater treatment plant. Whenever the storage unit is overcharged, water leaves the storage unit through an overflow. The height of this overflow can be controlled by a weir. Note that this simple network can be seen as a detail of a bigger network. Thus, the inflow node does not necessarily represent a natural inflow, but a node of inflow of water coming from the upper part of a larger network. Analogously, the channel behind the pump does not necessarily lead directly to the wastewater treatment plant, but may connect to parts of a bigger network lying behind.

Formulation of Objective Functions

We use the following formulation of the objectives presented in Section 8.2. Let T denote the time steps considered and let S be the number of storage units in the network.

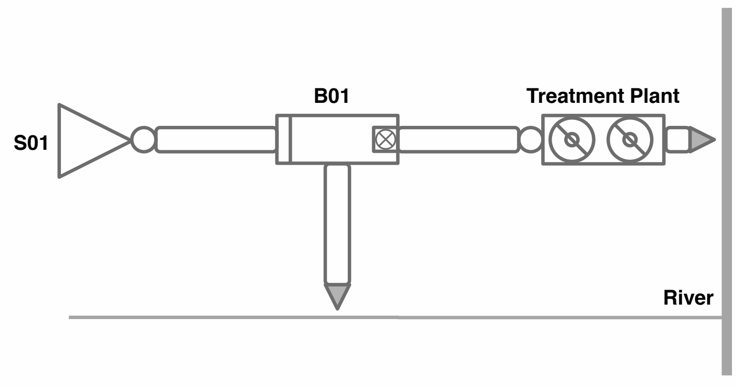


Figure 8.2: Considered subnetwork

1. The minimization of the total release of water is modelled by

$$f_1 = \sum_{i=1}^{S} \sum_{t=1}^{T} Q_{i,t}, \tag{8.1}$$

where $Q_{i,t}$ denotes the overflow rate at storage unit i in time step t averaged over the time interval Δt.

2. The minimization of the pollution mass of released water is obtained by

$$f_2 = \sum_{i=1}^{S} \sum_{t=1}^{T} \rho_{i,t} \cdot Q_{i,t}, \tag{8.2}$$

where $\rho_{i,t}$ denotes the pollution density given by the chemical oxygen demand (COD).

3. Variations of some specific controllable element $i \in S$ are modeled by

$$f_3^i = \sum_{t=1}^{T-1} \left(u_{i,t+1} - u_{i,t}\right)^2, \tag{8.3}$$

where $u_{i,t} \in [0,1]$ denotes the control. For describing variations in inflow to the WWTP we minimize the variation of flow through the last controllable element in the network before the WWTP is reached. Therefore, goals 3 and 4 described in Section 8.2 can both be modeled by (8.3).

4. The maximum utilization of the WWTP is described by

$$f_4 = \sum_{t=1}^{T-1} \left(u_{max} - u_{i,t}\right)^2, \tag{8.4}$$

where i denotes the last controllable element before the wastewater treatment plant is reached and u_{max} is the maximal capacity of this element. This goal prevents unnecessary storage in the network.

In the following, we only include objectives (8.1) to (8.3) into our numerical study. Thereby, the third goal is seen to be subordinate to the first two objectives. Consequently, we treat the problem as a bicriteria optimization problem and minimize lexicographically with respect to the third objective.

Hydrodynamic Process Model

In Hild and Leugering (2012), a hydrodynamic process model together with an implementation in C++ is proposed. We use this process model for our computational study. The code is supplemented by an implementation of the weighted sum, the ε-constraint and the augmented weighted Tchebycheff method, a list which maintains the set of nondominated points, functions updating the parameters of the subproblems and some interface functions. The user can choose which objectives are considered, which scalarization method and which parameter update scheme is used, see the description below.

Besides, in order to evaluate (8.2), the movement of the pollution particles has to be modeled. This is realized by a pure transport equation, which simplifies the underlying physical and chemical processes immensely and, thus, only gives a rough idea of the distribution of pollution mass in the network, but which was sufficient for our analysis.

Single-Objective Solver

As a flexible single-objective optimizer for nonlinear constrained optimization problems, IPOPT (Wächter and Biegler, 2006) is applied, which has already been tested for real-world multiobjective problems, see, e.g., Hakanen et al. (2007). IPOPT is a primal-dual interior-point algorithm with a filter line-search method for nonlinear programming. For details, we refer to Wächter and Biegler (2006). IPOPT converges, if a sufficient number of iterations is performed. If, however, due to time restrictions, a maximum number of iterations is specified in advance, this may not be sufficient for finding a stationary point. In this situation, only an intermediate solution is returned by IPOPT. Note that the returned solution may even not be feasible.

In the case of wastewater management problems, and particularly in the context of a real-time optimal control, we have to interrupt the minimization process after some predetermined number of iterations due to the limited amount of time that is available for the individual optimization runs. We, thus, have to expect that the outcomes of our computations are intermediate solutions and are, in general, no stationary points. For practical problems this is usually not critical since in the wastewater management context with its dynamics and uncertainties the goal is generally not to find the absolute optimum. A reasonable improvement as compared to the uncontrolled case is usually satisfactory. From a theoretical point of view, however, it is important to note that we often deal with dominated or even infeasible outcomes, despite the theoretical properties of the applied scalarization methods.

Approximation of the Nondominated Set

Due to the interruption of the solution process, we can not expect to find a discrete representation of the nondominated set but are satisfied with a discrete approximation, which should give a rough idea of the shape of the nondominated set. Note that due to possible dominated outcomes it is mandatory to test each point entering the approximation for nondominance with respect to the points contained in the current approximation. In particular, every point of the approximation may be dominated by some outcome that is found in a later subproblem and, thus, may leave the approximation at a later stage.

Parameter Variation Schemes

In order to find suitable parameters of the subproblems, we employ different strategies. Thereby, the parameters to be selected are the weights λ in the weighted sum approach, the scalar ε related to the second objective in the ε-constraint method and the directions d in the augmented weighted Tchebycheff method which correspond to the weights. Note that in the latter, the reference point was set to the origin and the augmentation parameter to $\rho = 10^{-3}$. We test three different rules for the parameter selection.

R1: Simple a priori approach: For the first and simplest rule a total number $N \geq 1$ of subproblems to be solved is given, and the parameters are chosen with equidistant spread. For the weighted sum approach, λ_1 varies between 0 and 1 with an even increment of $\frac{1}{N}$ in each iteration. The second parameter is computed by $\lambda_2 = 1 - \lambda_1$. The ε-constraint approach is not evaluated with this simple method as it does not take the magnitude of the objective function values into account. Therefore, it is very likely to construct infeasible problems when fixing ε to any value not related to the left-hand side of the ε-constraint. The directions in the augmented weighted Tchebycheff problem are set to $d_1 = \lambda_2$ and $d_2 = \lambda_1$, where the same values as for the weighted sum parameters are chosen.

R2: Improved a priori approach: The second rule follows the parameter update of the a priori box algorithm described in Hamacher et al. (2007). While the box algorithm is developed for the ε-constraint method, the parameter scheme can be transferred to the weighted sum and to the augmented weighted Tchebycheff approach. Let z^1 and z^2 be the lexicographically minimal solutions with respect to the first and second objective, respectively, and let again $N \geq 1$ denote the number of subproblems to be solved. Let $\triangle x := z_1^2 - z_1^1$ and $\triangle y := z_2^1 - z_2^2$. The parameters of the weighted sum method are set to

$$\lambda_1 = \frac{k\triangle y}{\triangle x + \triangle y},\ \lambda_2 = 1 - \lambda_1,$$

the parameter of the ε-constraint method to

$$\varepsilon = z_2^2 + \frac{k}{N}\triangle y$$

and the directions in the augmented weighted Tchebycheff problem to

$$d_1 = k \cdot \triangle x,\ d_2 = (N-k) \cdot \triangle y$$

for $k = 1, \dots, N-1$, respectively. As the parameters are computed based on the lexicographically minimal solutions, the magnitude of the individual objective function values is taken into account.

R3: Adaptive approach: All parameters are computed with respect to the local ideal and local nadir point of two adjacent points in the approximation. Thereby, a volume-based selection rule is applied, i.e., we choose the next box as the one with the largest volume. Let z^i and z^{i+1} be two points of the current approximation with $z_1^i < z_1^{i+1}$, without loss of generality. Then, the directions of the augmented weighted Tchebycheff problem are set to

$$d_1 = z_1^{i+1} - z_1^i,\ d_2 = z_2^i - z_2^{i+1},$$

the weights of the weighted sum method to

$$\lambda_1 = \frac{d_2}{d_1 + d_2},\ \lambda_2 = 1 - \lambda_1$$

and the parameter of the ε-constraint method is set to

$$\varepsilon = z_2^{i+1} + \frac{1}{2}(z_2^i - z_2^{i+1}).$$

If the solutions of the single-criterion solver are globally optimal, see the examples in Section 7.2, either a new point in the considered box or one of its defining points is computed. In the latter case, the considered box can be excluded from the search region. In the former case, the new point is inserted into the (discrete) approximation, the box volumes are updated and the search continues until a predefined maximum box volume is attained. If, due to numerical issues described above, the subproblem solved in a certain box yields a point that lies outside this box, we have to exclude the considered box from the search region as well, because otherwise, it would be selected in each subsequent iteration, since its volume does not change.

8.4 Computational Results

First, we analyze the effects between pairs of objectives in more detail. For this purpose, we present results for two objectives and for the subnetwork depicted in Figure 8.2.

Total Release versus Total Pollution Mass

The two goals (8.1) and (8.2) concerning quantity and quality of water release, respectively, can be conflicting, but this is not necessarily the case for all data sets and/or network structures. If there exists, for example, an optimal control strategy such that no water has to be released at all, we clearly attain the goal of minimizing pollutants at the same time. Besides this trivial case, there are also non-trivial cases in which, by minimizing with respect to (8.1), we also achieve the minimal total pollution. These examples are less interesting for our analysis as the nondominated set then shrinks to one single point, i.e., the ideal point is feasible. In the following, we discuss two examples for which it is not possible to minimize total release and total pollution at the same time.

Scenario 1: This scenario represents a heavy and sudden rainfall, which causes a high inflow. The concentration of COD (chemical oxygen demand) is low at the first time steps and then rises significantly. This may reflect removal of deposits from the channel walls in upper parts of the network. We set the total time to $T = 3000$, i.e., we consider 3000 time steps. The flow and pollution input data is updated each 60 time steps, so we set $\triangle t = 60$. The inflow Q_t, $t = 1, 2, \dots, T$, (in m^3 per second) and the pollution density ρ_t, $t = 1, 2, \dots, T$, (in kg per m^3) are given explicitly for $t = \triangle t, 2\triangle t, \dots$ and are interpolated linearly for all other time steps. The values used for the example are depicted in Figure 8.3. The weir is controlled each 100 time

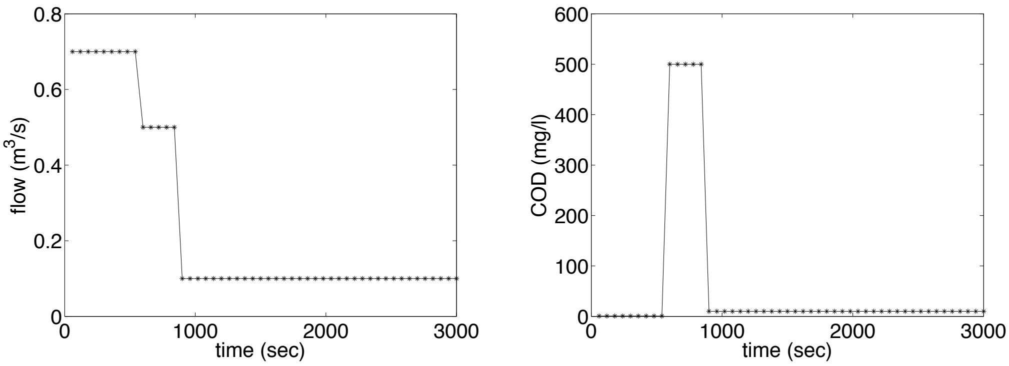


Figure 8.3: Inflow Q_t and pollution density ρ_t over time

steps, so there are 30 variables. The weir height is scaled to $[0, 1]$, where $u_t = 0$ means that the weir is completely opened at time t and $u_t = 1$ denotes that the weir is closed.

Minimizing only with respect to total release yields the solution $z^1 = (93.8, 11.85)^\top$, so a total release of 93.8 m^3 and a total pollution mass of 11.85 kg. Fixing the optimal objective function value $f_1 = 93.8$ and optimizing with respect to the pollution mass, i.e., solving $\min\{f_2(x) : f_1(x) \leq 93.8, x \in X\}$, results in the solution $z^2 = (95.2, 5.1)^\top$. Although this solution is not feasible for the constrained problem (recall that this effect is due to the limit on the maximum number of iterations of IPOPT which may not be sufficient to guarantee convergence to a stationary point), it is a feasible solution for the optimal control problem. Compared to $z^1 = (93.8, 11.85)^\top$, it only has a small impairment with respect to f_1 (1.5%), but a large improvement with respect to f_2 (57.0%). When computing an approximation of the set of nondominated points we obtain for the weighted sum method with the a posteriori parameter update rule (R3) the solution $z^3 = (94.2, 0.1)^\top$, which we expect to be the (local) ideal point. Thus, the second objective is nearly reduced to its minimum, zero pollution, and the objective function value of the first objective is only slightly worse compared to z^1 (0.4%). Note that z^3 dominates z^2, i.e., z^2 leaves the approximation when z^3 is inserted.

A closer look at the two control strategies for z^1 and z^3 reveals that the water quantities to be released are simply shifted, see Figure 8.4. By releasing more clean water in earlier time steps, capacity is made available for the time steps when polluted water streams in. Thereby, nearly all the polluted water can be kept in the network. But whether such an ideal strategy is possible depends on the parameters of the scenario and can not be guaranteed in general, as the following example shows.

Scenario 2: We consider the same inflow data as in Scenario 1 but limit the capacity

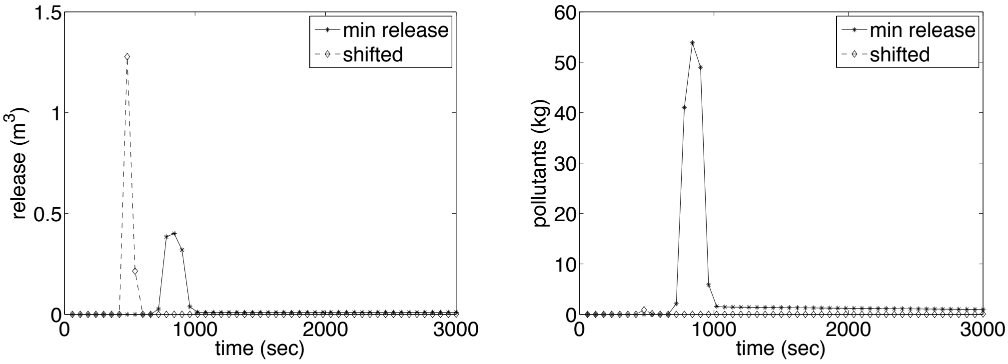


Figure 8.4: Total release (left) and pollution mass (right) for solutions z^1 (min release) and z^3 (shifted)

of the pump in the first 700 time steps. Instead of 90 l/s, only 18 l/s (20%) can be pumped out of the storage unit. This represents the case that less water can leave the storage unit in the first time steps because the capacity of the network behind the pump is exhausted.

In this example we can not shift water volumes in order to reduce the pollution of the released water to zero. We thus have a significant conflict between the two criteria. Improving the solution with respect to one criterion causes an impairment with respect to the other criterion. While the optimization with respect to the first objective yields $z^1 = (140.5, 14.0)^\top$, the best value for the second objective is obtained for solution $z^2 = (217.0, 0.2)^\top$. Note that, similar to the previous example, we can find a good compromise solution $z^3 = (147.4, 3.1)^\top$ that improves the water quality immensely (by 78.9%), whereas the released quantity only rises by 4.9%. An approximation generated with the weighted sum method is depicted in Figure 8.5. Note that even though many more subproblems are solved, the approximation only contains four points since in most subproblems dominated points are computed.

Total Release versus Constant Inflow to the WWT

The third objective, the minimization of variations in inflow to the wastewater treatment plant, was investigated separately. Different from the goal to minimize pollution mass, we see this goal clearly subordinate to minimizing the total release of water. Therefore, we search only among the optimal solutions of the minimal total release problem for an optimal solution of minimal variance of inflow to the wastewater treatment plant. Consequently, the lexicographic optimization approach is applied in this case. As in the previous case, there exist scenarios for which this secondary objective is automatically optimized in the primary optimization process. If, for example, in

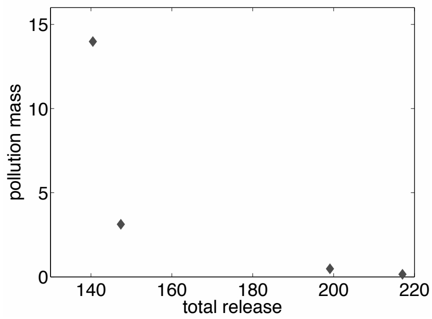


Figure 8.5: Approximation for Scenario 2 obtained by a weighted sum method; due to numerical issues dominated points are obtained which are not depicted

each time step water has to be released, then there are no variations in the inflow to the wastewater treatment plant since the channel connecting to it is always fully charged. To avoid trivial cases in the following, we consider only scenarios in which release of water does not occur in every time step. Figure 8.6 shows the considered inflow data. In this example, the pump is controlled.

Minimizing the total release yields a solution for which no water is released at all ($f_1 = 0$). There are infinitely many controls leading to this result, and the optimal control returned by the solver depends on the starting solution. Figure 8.6 shows two solutions minimizing the total release obtained from different starting solutions. Fixing the total release to zero and optimizing with respect to the second objective then yields in both cases a much smoother control than before. Note that these smoothed solutions would probably not have been found by only considering the total release of water and, thus, the lexicographic approach significantly improves the solution.

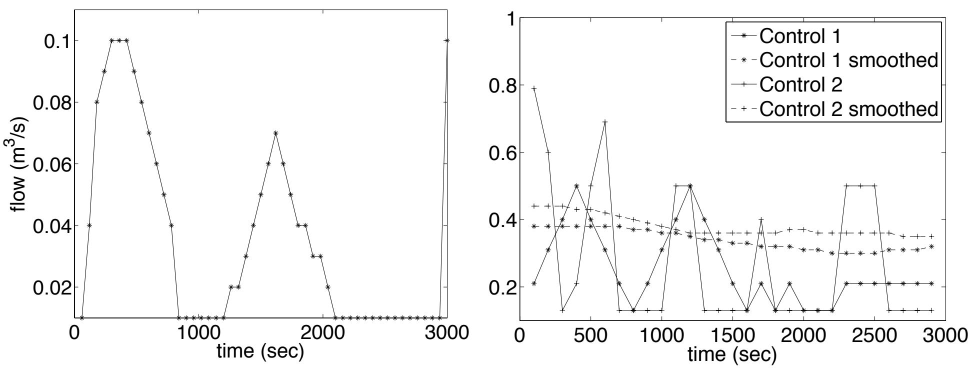


Figure 8.6: Inflow (left) and two different controls minimizing total release and their smoothed counterpart

Comparison of the Parameter Selection Rules

As expected, the simple a priori parameter scheme (R1) with the evenly distributed parameters not related to the magnitude of the objective function values performs quite badly for the weighted sum as well as for the augmented weighted Tchebycheff method. Recall that the ε-constraint method is not evaluated with this simple rule. For $N = 10$, no new solution different from the solution obtained for weights $(1, 0)^\top$ is computed. So the variation of the weights is useless because no new solution is found due to the relative 'overweighting' of the first objective. This is also observed for the augmented weighted Tchebycheff method with evenly distributed directions.

The improved a priori rule (R2) performs well. After having computed the two lexicographic minima, different new points that are mutually nondominated are found with all three scalarization methods.

With the adaptive rule (R3) the termination criterion is typically reached very quickly. Hence, only few iterations are performed and, thus, only few new points are generated. However, the generated points have a good quality, in general. For example, the quasi-ideal solution of the first scenario was computed with this rule.

For generating an approximation of the Pareto set in the wastewater management problem it seems to be the best option to use an a priori parameter scheme (R2), because, on the one hand, it automatically includes information about the magnitude of the objective function values and, on the other hand, it screens the interesting region without premature termination. However, the adaptive method (R3) also contributes points not found by the a priori method.

Comparison of the Scalarizations

In general, we can say that all three methods, the weighted sum, the ε-constraint and the augmented weighted Tchebycheff method, solve the problems in a satisfying way. All three methods contribute points to the approximation. However, we notice that for the considered example problems the weighted sum method seems to generate better solutions, i.e., contributes more points to the final approximation. This can be explained by the structure of the scalarized optimization problems. In the weighted sum method no new constraints are introduced. This in turn also means that every control associated to an intermediate solution is feasible both for the original optimal control problem as well as for the scalarization. For methods in which constraints are added, the initial and intermediate solutions may be infeasible and, thus, the solver has to find an optimal and feasible solution simultaneously. Even if the infeasibility with respect to the scalarization is not a problem because the solution is feasible for the optimal control problem, this may, nevertheless, influence the quality of the final objective function value obtained after a prescribed number of iterations.

Numerical Results for the Academic Test Network and Three Criteria

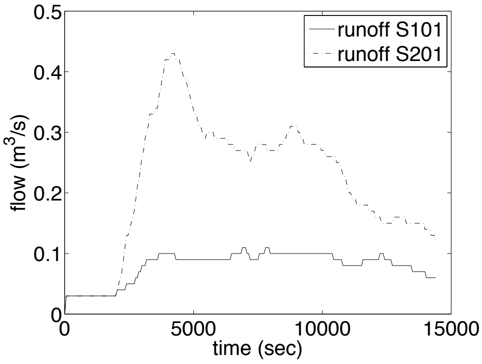
Finally, we present the results obtained for the academic network from Heusch and Ostrowski (2012), see Figure 8.1. Four different data sets for inflow values are given, see Figure 8.7. We consider the data recorded during the first four hours of the time series given in Heusch and Ostrowski (2012), Figure 9. One control step takes 10 minutes. The problem consists of 24 variables which equal the number of control steps during the considered four hours. Our reference problem is the uncontrolled case, i.e., $90\,l/s$ leave B01 constantly. For every instance, a tricriteria optimization problem is solved where the third objective is lexicographically optimized after the approximation of the nondominated set with respect to the first two objectives has been computed.

In the first example, the uncontrolled solution is $(1192.34, 349.15)^\top$. With the lexicographic approach we get $(304.93, 64.87)^\top$, which corresponds to an improvement of 74.4% with respect to the first and 81.4% with respect to the second objective. With the weighted sum method we find the even better solution $(280.30, 58.98)^\top$ for weights $(0.6, 0.4)^\top$. In the second example, the same inflow data is used as in Example 1 but the lower storage unit B01 is set one meter deeper and the power of the pump behind the upper storage unit B02 is increased from $90l/s$ to $120l/s$. The uncontrolled solution is $(850.49, 236.67)^\top$. With the lexicographic approach we get $(0, 0)^\top$ which is the ideal solution. In the third example, the uncontrolled solution is $(850.70, 48.27)^\top$. With the lexicographic approach we get $(287.52, 14.38)^\top$. No better solution was found so we expect this solution to be the ideal solution. In the fourth example, the uncontrolled solution is $(771.76, 88.50)^\top$. With the lexicographic approach we get $(312.09, 24.12)^\top$. With the weighted sum method we find the improved solution $(303.76, 23.32)^\top$ for weights $(0.1, 0.9)^\top$.

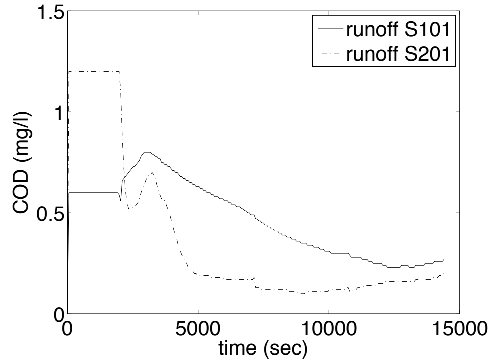
Note that for all test problems, the lexicographic minimization with respect to the third objective, the minimization of variations of control, did not lead to an improvement of the third objective while maintaining the objective values in the first two components. Therefore the resulting controls are not stated here.

8.5 Conclusion and Further Ideas

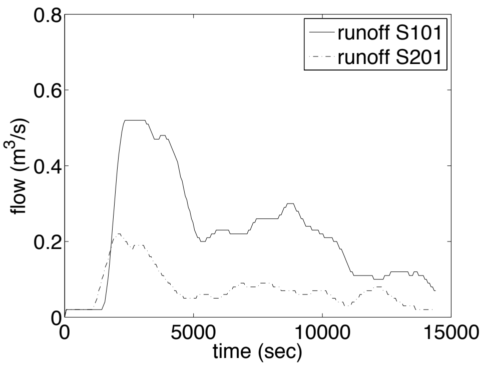
The numerical study presented in this chapter demonstrates that the successful application of adaptive methods highly depends on the quality of the computed points. When, for some reason, the single-criterion solver cannot compute global minima



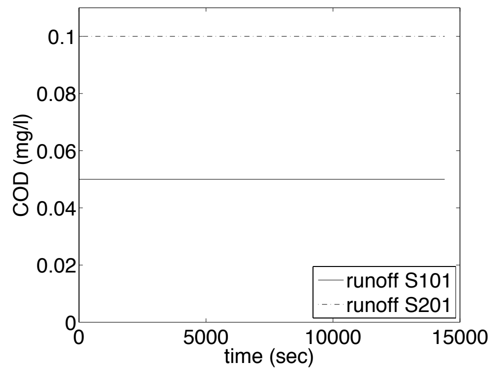
(a) Inflow Example 1 and 2



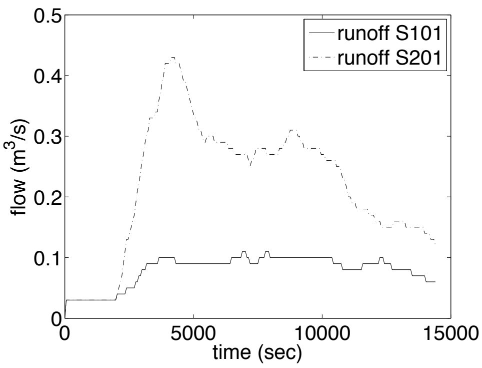
(b) COD Example 1 and 2



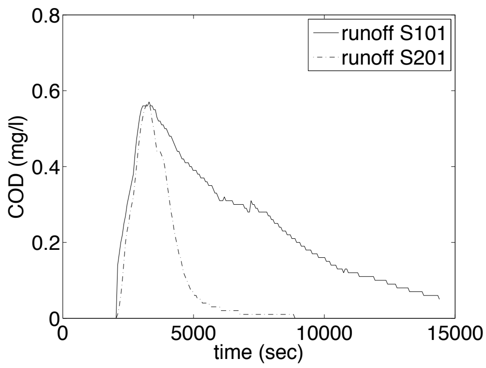
(c) Inflow Example 3



(d) COD Example 3



(e) Inflow Example 4



(f) COD Example 4

Figure 8.7: Inflow and pollution density at the two runoffs S01 and S02

that correspond to nondominated points of the multicriteria problem, an adaptive parametric algorithm might not work properly. In this situation, an a priori parameter scheme might yield better results. Moreover, the final approximation might only contain few points. From a theoretical perspective and in comparison to the results of Chapter 7, the results obtained in this chapter are disappointing. Nevertheless, since we were able to improve the uncontrolled solutions considerably, the results might be satisfactory for the practical application. In all considered examples, significant reductions in the pollution of released water could be achieved at the price of only a small increase in the total overflow, i.e., the total amount of released water.

Future research should address the observed numerical problems from the point of view of multiobjective approximation algorithms. By combining appropriate scalarizing functions with an adaptive approach, an approximation of the nondominated set that is in a certain sense robust with respect to non-optimality in the subproblems is aimed at.

9 Conclusion

In this thesis, we elaborated a new adaptive parametric algorithm based on a systematic decomposition of the search region. The algorithm consists in the iterative solution of a scalarization for different parameter choices. Thereby, common scalarizations like the classic ε-constraint or the (augmented) weighted Tchebycheff method can be used. The algorithm is universally applicable to generate (discrete) representations of discrete and continuous, convex and non-convex multicriteria optimization problems.

An important ingredient for achieving efficient parametric algorithms are adaptive parameter schemes. The notion 'adaptive' means that the parameters of the chosen scalarization are constructed dependent on the nondominated points that are known so far. Given a subset of the search region that is defined by components of known nondominated points, the parameters of the scalarization are chosen such that either a new nondominated point in the considered subset is generated or that this subset can be discarded from the search region. While adaptive parameter schemes have already been derived for the ε-constraint and the weighted Tchebycheff method, no formulas for selecting all parameters of the augmented weighted Tchebycheff method in an adaptive way have been available so far. In most applications, only the weights are selected adaptively but the augmentation parameter is fixed to a small positive constant. However, this frequently causes that nondominated points are missed and/or dominated points are obtained. Other approaches use the augmentation parameter in order to incorporate trade-off information, but chose the weights arbitrarily. However, an adaptive choice of the weights is needed to direct the search to a certain region within a parametric algorithm. In this thesis, we derived explicit formulas such that *all* parameters of an augmented weighted Tchebycheff scalarization, i.e., the weights and the augmentation parameter, can be chosen dependent on the nondominated points generated so far. Thereby, for discrete problems the parameters are computed such that it can be guaranteed that a possibly existing nondominated point in a certain region is generated. For the application in the continuous case we use a generalized formulation of the augmented weighted Tchebycheff method and demonstrate under which conditions given trade-off information expressed with respect to certain nondominated points can be translated into suitable parameters.

If the nondominated set of the considered multicriteria optimization problem is finite, a complete representation can be computed with the help of the proposed algorithm. In this case, the number of subproblems solved is of particular interest. In the tricriteria case we were able to improve the best known upper bound on the number of subproblems. We showed that at most three subproblems per nondominated point need to be solved if an initial search region based on the ideal point and an arbitrary upper bound on the set of feasible outcomes is available. We emphasize that the nadir point is not required in the algorithm. This is important as, in general, the nadir point is difficult to compute as soon as more than two criteria are considered. The linear bound on the number of subproblems could be established with the help of a new split criterion, which allows to avoid the generation of redundant boxes. In order to find new nondominated points any scalarization can be used by which a certain box of the search region can be investigated. This includes, e.g., Tchebycheff approaches with and without augmentation term. If, in particular, the ε-constraint method is used as scalarization, we could further improve the upper bound on the number of subproblems. In this case we demonstrated that at most two subproblems per nondominated point are required to generate a complete representation of the nondominated set.

Our numerical results confirmed the theoretical findings. We demonstrated with the help of discrete bi- and tricriteria problems that the entire nondominated set is computed reliably when the parameters are chosen adaptively according to the formulas developed in this thesis. In contrast, a fixed choice of the augmentation parameter caused that a certain percentage of nondominated points could not be generated. In the tricriteria case we successfully validated the new linear upper bound with the help of multidimensional knapsack problems. A comparison with very recent approaches from this field showed that our proposed algorithm can compete with state of the art algorithms.

We also applied our algorithm to generate incomplete representations for continuous problems. In particular, we compared an adaptive parameter scheme to an a priori parameter scheme, which is frequently used in practice. We showed that the parametric algorithm using an adaptive parameter scheme performs particularly well in the tricriteria case when the nadir point is not available and a substitute overestimating it is used. However, our study also revealed that the main feature of adaptive parameter schemes, which relies in their adaptivity to the nondominated set, is at the same time their weakness when the solutions of the subproblems are suboptimal, hence, the outcomes are not non-dominated. We experienced this difficulty for a real-world application, in which the single-criterion solver had to be interrupted and, thus, frequently returned non-optimal solutions. In this situation a very simple parameter selection might be a better choice than a sophisticated adaptive selection. However, when the outcomes of the subproblems are nondominated or near to nondominated points, adaptive parameter schemes are superior to a priori schemes, in general. By adapting to the shape of the nondominated set, the generation of infeasible and/or redundant subproblems can be avoided and, typically, a good coverage of the nondominated set can be obtained within rather few iterations.

Notation

$\mathbb{N}$	set of natural numbers $\{1, 2, \dots\}$
$\mathbb{N}_0$	$\mathbb{N} \cup \{0\}$
$\mathbb{Z}$	set of integer numbers $\{\dots, -2, -1, 0, 1, 2, \dots\}$
$\mathbb{R}$	set of real numbers
$\mathbb{R}^n$	n-dimensional Euclidean space
$\mathbb{R}^n_>$	$\{x \in \mathbb{R}^n : x_i > 0, i = 1, \dots, n\}$
$\mathbb{R}^n_+$, $\mathbb{R}^n_{\geqq}$	$\{x \in \mathbb{R}^n : x_i \geq 0, i = 1, \dots, n\}$
X	feasible set (subset of $\mathbb{R}^n$)
Z, $f(X)$	set of feasible outcomes (subset of $\mathbb{R}^m$)
X_E	efficient set, see (2.4)
Z_N	nondominated set, see (2.5)
$\mathcal{R}$	(discrete) representation of the nondominated set, see Definition 2.1
$\mathcal{A}$	approximation of the nondominated set, see Definition 2.2
C	cone, typically restricted to the Pareto cone $C = \mathbb{R}^m_+$
$Y^1 + Y^2$	algebraic sum of two sets $Y^1, Y^2 \subseteq \mathbb{R}^m$
$\mathbb{R}^m_+$-convex	a set $Y \subseteq \mathbb{R}^m$ for which $Y + \mathbb{R}^m_+$ is convex
$<$	$z < \bar{z} \;:\Leftrightarrow\; z_i < \bar{z}_i \quad \forall\, i = 1, \dots, m$
$\leq$	$z \leq \bar{z} \;:\Leftrightarrow\; z_i \leq \bar{z}_i \quad \forall\, i = 1, \dots, m$ and $\exists\, j \in \{1, \dots, m\} : z_j < \bar{z}_j$
$\leqq$	$z \leqq \bar{z} \;:\Leftrightarrow\; z_i \leq \bar{z}_i \quad \forall\, i = 1, \dots, m$
$m \in \mathbb{N}$	number of objectives, typically $m \geq 2$
$f : X \to \mathbb{R}^m$	(vector-valued) objective function
z	point in $\mathbb{R}^m$
z^I	ideal point, see (2.11)
z^U	utopian point, point that strictly dominates z^I
z^N	nadir point, see (2.12)

z^M	point of individual maxima, see (2.14)
$T_{ij}^G(\mathrm{z})$	trade-off in z with respect to objectives f_i, f_j, see (2.8)
$T_{ij}(x,x')$	ratio of change with respect to x, x' and objectives f_i, f_j, see (2.7)
$d_C(\mathcal{R}, Z_N)$	coverage of a representation, see (2.15)
$d_U(\mathcal{R})$	uniformity of a representation, see (2.17)
r_H	relative dominated hypervolume of a representation
$\lVert\cdot\rVert_{w,\infty}$	weighted Tchebycheff norm, see (4.1)
$\lVert\cdot\rVert_{w,\rho}$	augmented weighted Tchebycheff norm, see (4.2)
$\lVert\cdot\rVert_{w,\rho}^G$	generalized augmented weighted Tchebycheff norm, see (4.25)
$\mathcal{L}_\alpha$	contour of the (augmented) weighted Tchebycheff norm
$\mathcal{L}_\alpha^+$	contour of the (augmented) weighted Tchebycheff norm in $\mathbb{R}_+^m$
α	level of the contour of a norm
z^q	inflection point of the contour of a Tchebycheff norm
w	weight (vector in $\mathbb{R}^m$)
ρ	augmentation parameter (real number)
r	reference point (vector in $\mathbb{R}^m$)
d	direction (vector in $\mathbb{R}^m$)
ε	right-hand side vector of the ε-constraint method (vector in $\mathbb{R}^m$)
η	perturbation parameter (vector in $\mathbb{R}^m$)
$\bar{\eta}$	perturbation parameter (scalar)
B	box, i.e., rectangular subset of a decomposition of the search region
B_0	initial search region
u,$u(B)$	upper bound (of box B), vector in $\mathbb{R}^m$
$V(B)$	individual subset of box B
$v(B)$	lower bound of the individual subset of box B (vector in $\mathbb{R}^m$)
$\mathcal{B}_s$	decomposition of the search region in iteration s
$\overline{\mathcal{B}}_s$	set of boxes which are split in iteration s
$B_i^s(B)$	neighbor of box B with respect to component i in iteration s

Acknowledgement

My special thanks go to my supervisor Prof. Dr. Kathrin Klamroth for giving me this unique opportunity and supporting my work with advice and discussion whenever needed. I also thank Prof. Dr. Margaret M. Wiecek for being second reviewer of this thesis.

Moreover, I thank all current and former members of the group of Optimization and Approximation of the University of Wuppertal as well as my former fellows from the University of Erlangen-Nuremberg Dr. Martin Bischoff, Dr. Simone Gast, Dr. Jochen Gorski, Dipl. Math. Markus Kaiser, Prof. Dr. Manfred Mendel, Dr. Michael Stiglmayr, Britta Schulze M.Sc., Teresa Schnepper M.Sc., Prof. Dr. Peter Beisel, Prof. Dr. Margareta Heilmann, Dr. Alexander Thekale, Dr. Martin Wagner and Kirsten Wilshaus for the friendly atmosphere in the working group and helpful advice whenever needed.

During my PhD studies I had the opportunity to spend several weeks as a guest with the group of industrial optimization at the University of Jyväskylä, Finland. I thank Prof. Dr. Kaisa Miettinen for hosting me in her group as well as all current and former members for welcoming me as a temporary fellow.

In retrospect I noticed that my thesis was significantly influenced by two papers that were communicated personally to me. The first is the paper of Ralphs et al. (2006), which I received from Prof. Dr. Margaret M. Wiecek. The second is the paper of Laumanns et al. (2006), which was given to me by Dr. Pradyumn K. Shukla. Interestingly, this paper remained somehow ‘unregarded’ on my desk before being such a rich source for my thesis over one year later. I am very grateful that my attention was drawn to both publications. Furthermore, I would like to thank Prof. Dr. Gabriele Eichfelder and Prof. Dr. Stefan Ruzika for giving me copies of their thesis. Both were a valuable source for my work. In addition, I pay tribute to Prof. Dr. Carlos M. Fonseca who made me familiar with relevant results from the field of computational geometry.

I am also grateful to the co-authors of three joint publications, Dr. Jochen Gorski and Prof. Dr. Kathrin Klamroth, for the very productive work.

For my numerical experiments I acknowledge the source code for generating bicriteria knapsack instances from Prof. Dr. Serpil Sayın as well as instances of a tricriteria multidimensional knapsack problem provided by PD Dr. Marco Laumanns. Moreover, I thank the former undergraduate student Lars Balzer M.Sc. for support in programming within the BMBF grant.

During the first three years of my PhD studies I was funded by the BMBF, grant 03KLPAK3, which is kindly acknowledged.

Last, but not least, I thank my family, in particular my parents, my companion and my friends for their support.

When thanking people there is always the risk of forgetting somebody. So my final thanks go to all the people who are not mentioned here explicitly, but supported me in some way.

Bibliography

Alvarez-Vazquez, L. J., Garcia-Chan, N., Martinez, A., and Vazquez-Mendez, M. E. Multi-objective Pareto-optimal control: an application to wastewater management. *Computational Optimization and Applications*, 46:135–157, 2010.

Alves, M. J. and Climaco, J. An interactive reference point approach for multiobjective mixed-integer programming using branch-and-bound. *European Journal of Operational Research*, 124:478–494, 2000.

Alves, M. J. and Climaco, J. A review of interactive methods for multiobjective integer and mixed-integer programming. *European Journal of Operational Research*, 180:99–115, 2007.

Aneja, Y. P. and Nair, K. P. K. Bicriteria Transportation Problem. *Management Science*, 25:73–78, 1979.

Armann, R. Solving Multiobjective Programming Problems by Discrete Representation. *Optimization*, 20:483–492, 1989.

Beume, N., Fonseca, C. M., López-Ibáñez, M., Paquete, L., and Vahrenhold, J. On the Complexity of Computing the Hypervolume Indicator. *IEEE Transactions on Evolutionary Computation*, 13(5):1075–1082, 2009.

Boissonnat, J.-D., Sharir, M., Tagansky, B., and Yvinec, M. Voronoi Diagrams in Higher Dimensions under Certain Polyhedral Distance Functions. *Discrete & Computational Geometry*, 19:485–519, 1998.

Bowman, V. J. On the relationship of the Tchebycheff norm and the efficient frontier of multiple-criteria objectives. In Thieriez, H. and Zionts, S., editors, *Multiple Criteria Decision Making*, pages 76–85, Berlin, Heidelberg, 1976. Springer Verlag.

Bozkurt, B., Fowler, J. W., Gel, E. S., Kim, G., Köksalan, M., and Wallenius, J. Quantitative Comparison of Approximate Solution Sets for Multicriteria Optimization Problems with Weighted Tchebycheff Preference Function. *Operations Research*, 58:650–659, 2010.

Bringmann, K. Bringing Order to Special Cases of Klee's Measure Problem. Technical report, Max Planck Institute for Informatics Saarbrücken, Germany, 2013. URL `http://arxiv.org/abs/1301.7154`.

Chalmet, L., Lemonidis, L., and Elzinga, D. An algorithm for the bi-criterion integer programming problem. *European Journal of Operational Research*, 25:292–300, 1986.

Chankong, V. and Haimes, Y. Y. *Multiobjective Decision Making: Theory and Methodology.* Elsevier Science Publishing, New York, 1983.

Cohon, J. L. *Multiobjective Programming and Planning.* Academic Press, New York, 1978.

Cohon, J. L., Church, R. L., and Sheer, D. P. Generating multiobjective tradeoffs: An algorithm for bicriterion problems. *Water Resources Research*, 15:1001–1010, 1979.

Corley, H. W. A new scalar equivalence for Pareto optimization. *IEEE Transactions on Automatic Control*, 25:829–830, 1980.

Dächert, K. and Klamroth, K. Multicriteria Optimization in Wastewater Management. In Martin, A., Klamroth, K., Lang, J., Leugering, G., Morsi, A., Oberlack, M., Ostrowski, M., and Rosen, R., editors, *Mathematical Optimization of Water Networks*, pages 167–196. Birkhäuser, 2012.

Dächert, K. and Klamroth, K. A linear bound on the number of scalarizations needed to solve discrete tricriteria optimization problems. Technical report, Bergische Universität Wuppertal, Germany, 2013. URL `http://arxiv.org/abs/1305.5266`.

Dächert, K., Gorski, J., and Klamroth, K. An augmented weighted Tchebycheff method with adaptively chosen parameters for discrete bicriteria optimization problems. *Computers and Operations Research*, 39:2929–2943, 2012.

Das, I. and Dennis, J. E. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural Optimization*, 14:63–69, 1997.

Das, I. and Dennis, J. E. Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998.

Deb, K. *Multi-objective Optimization using Evolutionary Algorithms.* Wiley, Chichester, UK, 2001.

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. Scalable Test Problems for Evolutionary Multi-Objective Optimization. Technical Report 112, ETH Zürich 2001.

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. Scalable Test Problems for Evolutionary Multi-Objective Optimization. In Abraham, A., Jain, L., and Goldberg, R., editors, *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, pages 105–146. Springer, 2005.

Dhaenens, C., Lemesre, J., and Talbi, E.-G. K-PPM: A new exact method to solve multi-objective combinatorial optimization problems. *European Journal of Operational Research*, 200:45–53, 2010.

Edgeworth, F. Y. *Mathematical Psychics: An Essay on the Application of Mathematics to the Moral Sciences.* C. Kegan Paul & Co., London, 1881.

Ehrgott, M. *Multicriteria Optimization.* Springer, Berlin, 2005.

Ehrgott, M. and Gandibleux, X. A survey and annotated bibliography of multiobjective combinatorial optimization. *OR Spektrum*, 22:425–460, 2000.

Ehrgott, M. and Ruzika, S. Improved ε-Constraint Method for Multiobjective Programming. *Journal of Optimization Theory and Applications*, 138:375–396, 2008.

Ehrgott, M. and Tenfelde-Podehl, D. Computation of ideal and Nadir values and implications for their use in MCDM methods. *European Journal of Operational Research*, 151:119–139, 2003.

Eichfelder, G. *Parametergesteuerte Lösung nichtlinearer multikriterieller Optimierungsprobleme (in German).* PhD thesis, University of Erlangen-Nürnberg, Germany, 2006.

Eichfelder, G. An Adaptive Scalarization Method in Multiobjective Optimization. *SIAM Journal on Optimization*, 19(4):1694–1718, 2009a.

Eichfelder, G. Scalarizations for adaptively solving multi-objective optimization problems. *Computational Optimization and Applications*, 44:249–273, 2009b.

Eswaran, P., Ravindran, A., and Moskowitz, H. Algorithms for Nonlinear Integer Bicriterion Problems. *Journal of Optimization Theory and Applications*, 63(2): 261–279, 1989.

Faulkenberg, S. L. and Wiecek, M. M. On the quality of discrete representations in multiple objective programming. *Optimization and Engineering*, 11(3):423–440, 2010.

Faulkenberg, S. L. and Wiecek, M. M. Generating equidistant representations in biobjective programming. *Computational Optimization and Applications*, 51(3): 1173–1210, 2012.

Fliege, J. and Svaiter, B. F. Steepest descent methods for multicriteria optimization. *Mathematical Methods of Operations Research*, 51:479–494, 2000.

Fliege, J., Drummond, M. G., and Svaiter, B. F. Newton's method for multicriteria optimization. *SIAM Journal on Optimization*, 20(2):602–626, 2009.

Fonseca, C. M. Personal communication, 2013.

Fonseca, C. M., Guerreiro, A. P., López-Ibáñez, M., and Paquete, L. On the Computation of the Empirical Attainment Function. In Takahashi, R. H. C., Deb, K., Wanner, E. F., and Greco, S., editors, *Evolutionary Multi-Criterion Optimization*, volume 6576 of *Lecture Notes in Computer Science*, pages 106–120. Springer, 2011.

Gass, S. and Saaty, T. The Computational Algorithm for the Parametric Objective Function. *Naval Research Logistics Quarterly*, 2:39–45, 1955.

Geoffrion, A. M. Solving Bicriterion Mathematical Programs. *Operations Research*, 15:39–54, 1967.

Geoffrion, A. M. Proper Efficiency and the Theory of Vector Maximization. *Journal of Mathematical Analysis and its Applications*, 22:618–630, 1968.

Gorski, J. *Multiple Objective Optimization and Implications for Single Objective Optimization*. PhD thesis, Bergische Universität Wuppertal, 2010.

Gourion, D. and Luc, D. T. Finding Efficient Solutions by Free Disposal Outer Approximation. *SIAM Journal on Optimization*, 6:2939–2958, 2010.

Guerreiro, A. P., Fonseca, C. M., and Emmerich, M. T. M. A Fast Dimension-Sweep Algorithm for the Hypervolume Indicator in Four Dimensions. In *Proceedings of the 24th Canadian Conference on Computational Geometry (CCCG 2012)*, Charlottetown, Prince Edward Island, Canada, 2012.

Haimes, Y. Y., Lasdon, L. S., and Wismer, D. On a bicriteria formulation of the problems of integrated systems identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, 1:296–297, 1971.

Hakanen, J., Kawajiri, Y., Miettinen, K., and Biegler, L. T. Interactive multiobjective optimization for simulated moving bed processes. *Control & Cybernetics*, 36(2):283–302, 2007.

Hamacher, H. W., Pedersen, C. R., and Ruzika, S. Finding representative systems for discrete bicriteria optimization problems by box algorithms. *Operations Research Letters 35*, 3:336–344, 2007.

Hansen, M. P. and Jaszkiewicz, A. Evaluating the quality of approximations to the non-dominated set. Technical Report IMM-REP-1998-7, IMM Technical University of Denmark, 1998.

Heusch, S. and Ostrowski, M. Optimal Control of Sewer Networks Engineers View. In Martin, A., Klamroth, K., Lang, J., Leugering, G., Morsi, A., Oberlack, M., Ostrowski, M., and Rosen, R., editors, *Mathematical Optimization of Water Networks*, pages 111–127. Birkhäuser, 2012.

Hild, J. and Leugering, G. Real-Time Control of Urban Drainage Systems. In Martin, A., Klamroth, K., Lang, J., Leugering, G., Morsi, A., Oberlack, M., Ostrowski, M., and Rosen, R., editors, *Mathematical Optimization of Water Networks*, pages 129–150. Birkhäuser, 2012.

Isermann, H. and Steuer, R. E. Computational experience concerning payoff tables and minimum criterion values over the efficient set. *European Journal of Operational Research*, 33:91–97, 1987.

Jahn, J. *Vector Optimization: Theory, applications and extensions*. Springer, Berlin, 2004.

Kaliszewski, I. A Modified Weighted Tchebycheff Metric for Multiple Objective Programming. *Computers and Operations Research*, 14(4):315–323, 1987.

Kaliszewski, I. Using trade-off information in decision-making algorithms. *Computers and Operations Research*, 27:161–182, 2000.

Kaliszewski, I. and Michalowski, W. Efficient Solutions and Bounds on Tradeoffs. *Journal of Optimization Theory and Applications*, 94(2):381–394, 1997.

Kim, I. Y. and de Weck, O. L. Adaptive weighted sum method for multiobjective optimization: a new method for Pareto front generation. *Structural and Multidisciplinary Optimization*, 31:105–116, 2006.

Kirlik, G. and Sayın, S. A new algorithm for generating all nondominated solutions of multiobjective discrete optimization problems. *European Journal of Operational Research*, 232:479–488, 2014.

Klamroth, K. and Wiecek, M. M. Dynamic Programming Approaches to the Multiple Criteria Knapsack Problem. *Naval Research Logistics*, 47:57–76, 2000.

Klamroth, K., Tind, J., and Wiecek, M. M. Unbiased approximation in multicriteria optimization. *Mathematical Methods of Operations Research*, 56:413–437, 2002.

Klein, D. and Hannan, E. An algorithm for the multiple objective integer linear programming problem. *European Journal of Operational Research*, 9:378–385, 1982.

Kuhn, H. W. The Hungarian method for the assignment problem. *Naval Research Logistics*, 2:83–97, 1955.

Laumanns, M., Thiele, L., and Zitzler, E. An Adaptive Scheme to Generate the Pareto Front Based on the Epsilon-Constraint Method. In Branke, J., Deb, K., Miettinen, K., and Steuer, R. E., editors, *Practical Approaches to Multi-Objective Optimization*, number 04461 in Dagstuhl Seminar Proceedings, Dagstuhl, Germany, 2005. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany. URL `http://drops.dagstuhl.de/opus/volltexte/2005/246`.

Laumanns, M., Thiele, L., and Zitzler, E. An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *European Journal of Operational Research*, 169:932–942, 2006.

Lokman, B. and Köksalan, M. Finding all nondominated points of multi-objective integer programs. *Journal of Global Optimization*, 57:347–365, 2013.

Luque, M., Ruiz, F., and Steuer, R. E. Modified interactive Chebyshev algorithm (MICA) for convex multiobjective programming. *European Journal of Operational Research*, 204:557–564, 2010.

Luque, M., Miettinen, K., Ruiz, A. B., and Ruiz, F. A two-slope achievement scalarizing function for interactive multiobjective optimization. *Computers and Operations Research*, 39(7):1673–1681, 2012.

Marinaki, M. and Papageorgiou, M. *Optimal Real-time Control of Sewer Networks*. Advances in Industrial Control. Springer Verlag, Berlin, Heidelberg, 2005.

Martin, A., Klamroth, K., Lang, J., Leugering, G., Morsi, A., Oberlack, M., Ostrowski, M., and Rosen, R. *Mathematical Optimization of Water Networks*, volume 162 of *International Series of Numerical Mathematics*. Birkhäuser, 2012.

Mavrotas, G. Effective implementation of the ε-constraint method in Multi-Objective Mathematical Programming problems. *Applied Mathematics and Computation*, 213:455–465, 2009.

Miettinen, K. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, 1999.

Miettinen, K., Mäkelä, M., and Kaario, K. Experiments with classification-based scalarizing functions in interactive multiobjective optimization. *European Journal of Operational Research*, 175:931–974, 2006.

Murthy, I. and Her, S. Solving min-max shortest-path problems on a network. *Naval Research Logistics*, 39:669–683, 1992.

Muschalla, D. Optimization of integrated urban wastewater systems using multiobjective evolution strategies. *Urban Water Journal*, 5(1):57–65, 2008.

Nemhauser, G. L. and Wolsey, L. A. *Integer and Combinatorial Optimization*. Wiley, 1999.

Özlen, M. and Azizoğlu, M. Multi-objective integer programming: A general approach for generating all non-dominated solutions. *European Journal of Operations Research*, 199:25–35, 2009.

Ozlen, M., Burton, B. A., and MacRae, C. A. G. Multi-Objective Integer Programming: An Improved Recursive Algorithm. *Journal of Optimization Theory and Applications*, 160(2):470–482, 2014.

Özpeynirci, Ö. and Köksalan, M. Pyramidal tours and multiple objectives. *Journal of Global Optimization*, 48:569–582, 2010a.

Özpeynirci, Ö. and Köksalan, M. An Exact Algorithm for Finding Extreme Supported Nondominated Points of Multiobjective Mixed Integer Programs. *Management Science*, 56:2302–2315, 2010b.

Pareto, V. *Cours d'Economie Politique*. F. Rouge, Lausanne, 1896.

Pascoletti, A. and Serafini, P. Scalarizing Vector Optimization Problems. *Journal of Optimization Theory and Applications*, 42:499–524, 1984.

Pasternak, H. and Passy, U. Bicriterion mathematical programs with Boolean variables. In Cochrane, J. L. and Zeleny, M., editors, *Multiple Criteria Decision Making*, pages 327–348, Columbia, SC, 1973. University of South Carolina.

Pleau, M., Colas, H., Lavallée, P., Pelletier, G., and Bonin, R. Global optimal real-time control of the Quebec urban drainage system. *Environmental Modelling & Software*, 20:401–413, 2005.

Podkopaev, D. An approach to finding trade-off solutions by a linear transformation of objective functions. *Control and Cybernetics*, 36(2):347–355, 2007.

Przybylski, A., Gandibleux, X., and Ehrgott, M. Computational Results for Four Exact Methods to Solve the Three-Objective Assignment Problem. In Barichard, V., Ehrgott, M., Gandibleux, X., and T'Kindt, V., editors, *Multiobjective Programming and Goal Programming: Theoretical Results and Practical Applications*, Lecture Notes in Economics and Mathematical Systems 618, pages 79–88. Springer, 2009.

Przybylski, A., Gandibleux, X., and Ehrgott, M. A two phase method for multi-objective integer programming and its application to the assignment problem with three objectives. *Discrete Optimization*, 7:149–165, 2010a.

Przybylski, A., Gandibleux, X., and Ehrgott, M. A Recursive Algorithm for Finding All Nondominated Extreme Points in the Outcome Set of a Multiobjective Integer Programme. *INFORMS Journal on Computing*, 22(3):371–386, 2010b.

Ralphs, T., Saltzman, M., and Wiecek, M. M. An improved algorithm for solving biobjective integer programs. *Annals of Operations Research*, 147:43–70, 2006.

Rauch, W. and Harremoës, P. Genetic algorithms in real time control applied to minimize transient pollution from urban wastewater systems. *Water Research*, 33 (5):1265–1277, 1999.

Rennen, G., van Dam, E. R., and den Hertog, D. Enhancement of Sandwich Algorithms for Approximating Higher Dimensional Convex Pareto Sets. Technical Report 2009-52, Tilburg University 2009.

Ruzika, S. *On Multiple Objective Combinatorial Optimization.* PhD thesis, Technische Universität Kaiserslautern, 2007.

Ruzika, S. and Wiecek, M. M. A Survey of Approximation Methods in Multiobjective Programming. *Journal of Optimization Theory and Applications*, 126(3):473–501, 2005.

Sayın, S. Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. *Mathematical Programming*, 87:543–560, 2000.

Sayın, S. and Kouvelis, P. The Multiobjective Discrete Optimization Problem: A Weighted Min-Max Two-Stage Optimization Approach and a Bicriteria Algorithm. *Management Science*, 51(10):1572–1581, 2005.

Schandl, B., Klamroth, K., and Wiecek, M. M. Norm-based Approximation in Bicriteria Programming. *Computational Optimization and Applications*, 20:23–42, 2001.

Schandl, B., Klamroth, K., and Wiecek, M. M. Norm-based Approximation in Multicriteria Programming. *Computers and Mathematics with Applications*, 44:925–942, 2002.

Schütze, M., Campisano, A., Colas, H., Schilling, W., and Vanrolleghem, P. A. Real time control of urban wastewater systems – where do we stand today? *Journal of Hydrology*, 299:335–348, 2004.

Solanki, R. S. Generating the Noninferior Set in Mixed Integer Biobjective Linear Programs: An Application to a Location Problem. *Computers and Operations Research*, 18(1):1–15, 1991.

Solanki, R. S., Appino, P. A., and Cohon, J. L. Approximating the noninferior set in multiobjective linear programming problems. *European Journal of Operational Research*, 68(3):356–373, 1993.

Steuer, R. E. *Multiple Criteria Optimization: Theory, Computation and Application.* John Wiley and Sons, New York, 1986.

Steuer, R. E. and Choo, E. An Interactive Weighted Tchebycheff Procedure for Multiple Objective Programming. *Mathematical Programming*, 26:326–344, 1983.

Sylva, J. and Crema, A. A method for finding the set of non-dominated vectors for multiple objective integer linear programs. *European Journal of Operational Research*, 158:46–55, 2004.

Sylva, J. and Crema, A. A method for finding well-dispersed subsets of non-dominated vectors for multiple objective mixed integer linear programs. *European Journal of Operational Research*, 180:1011–1027, 2007.

Szczepanski, M. and Wierzbicki, A. P. Application of multiple criteria evolutionary algorithms to vector optimisation, decision support and reference point approaches. *Journal of Telecommunications and Information Technology*, 3:16–33, 2003.

Tanaka, M., Watanabe, H., Furukawa, Y., and Tanino, T. GA-based decision support system for multicriteria optimization. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pages 1556–1561, 1995.

Tenfelde-Podehl, D. A Recursive Algorithm for Multiobjective Combinatorial Optimization Problems with Q Criteria. Technical report, Institut für Mathematik, Technische Universität Graz, 2003.

Ulungu, E. L. and Teghem, J. The two phases method: An efficient procedure to solve bi-objective combinatorial optimization problems. *Foundations of Computing and Decision Sciences*, 20(2):149–165, 1995.

Wächter, A. and Biegler, L. T. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

Weinreich, G., Schilling, W., Birkely, A., and Moland, T. Pollution based real time control strategies for combined sewer systems. *Water Science and Technology*, 36 (8-9):331–336, 1997.

Wendell, R. E. and Lee, D. N. Efficiency in multiple objective optimization problems. *Mathematical Programming*, 12:406–414, 1977.

Wierzbicki, A. P. The use of reference objectives in multiobjective optimization. In *Multiple Criteria Decision Making*, pages 468–486, Berlin, 1980. Springer.

Zeleny, M. Compromise programming. In Cochrane, J. L. and Zeleny, M., editors, *Multiple Criteria Decision Making*, pages 262–301, Columbia, SC, 1973. University of South Carolina.

Zitzler, E. and Thiele, L. Multiobjective optimization using evolutionary algorithms - a comparative case study. In *PPSN-V*, pages 292–301, Amsterdam, 1998.