Representative Systems and Decision Support for Multicriteria Optimization Problems

Vom Fachbereich Mathematik
der Technischen Universität Kaiserslautern
zur Verleihung des akademischen Grades
Doktor der Naturwissenschaften
(Doctor rerum naturalium, Dr. rer. nat.)
genehmigte Dissertation

vorgelegt von

Tobias Kuhn

Gutachter: Prof. Dr. Stefan Ruzika
Prof. Dr. habil. Xavier Gandibleux

Datum der Disputation: 29.10.2015

D 386

Contents

1 Introduction

The International Federation of Red Cross and Red Crescent Societies defines a disaster or a catastrophe as "a sudden, calamitous event that seriously disrupts the functioning of a community or society and causes human, material, and economic or environmental losses that exceed the community's or society's ability to cope using its own resources" (IFRC, 2015). In most cases, such catastrophes occur unexpectedly to the residents and give reason to immediately evacuate all affected people from the danger zone. Recent examples for well-known catastrophes are the tsunami and nuclear plant emergency in Fukushima (see, e.g., The Guardian, 2011), the train derailment and explosion in Quebec (see, e.g., BBC News Online, 2013) or the volcanic eruption in Indonesia (see, e.g., The Guardian, 2014). However, a large-scale evacuation is not only necessary in response to a disaster but also for reasons of safety before some dangerous event, e.g., a bomb disposal. In Rhineland-Palatinate, several unexploded Second World War bombs are found every year such that thousands of people have to be evacuated from the danger zone and special forces have to defuse the bombs. Examples from last year include found bombs in Landau (see, e.g., Die Rheinpfalz, 2014e), Ludwigshafen (see, e.g., Die Rheinpfalz, 2014b,c), Zweibrücken (see, e.g., Die Rheinpfalz, 2014a), Germersheim (see, e.g., Die Rheinpfalz, 2014d), and Mainz (see, e.g., Die Rheinpfalz, 2014f). In order to be well-prepared in the case of a large-scale evacuation, one option consists of defining and analyzing possible scenarios and developing different solution strategies. Responsible decision makers usually use empirically created evacuation plans without any scientific justification and without knowledge about their quality. Therefore, it is highly recommended to deal with this topic scientifically and to develop mathematical approaches to understand the evacuation process. This is precisely the point where the research project DSS_Evac_Logistics comes into play.

DSS_Evac_Logistics

Many topics of this PhD thesis originated from the work in the project DSS_Evac_Logistics.[1] Hence, we shortly introduce this project to the reader. In general, the project's goal is a decision support system (DSS) which will help the decision makers to be prepared in the case of a large-scale evacuation. The resulting software will not

[1] For general information about this project see `http://projets.li.univ-tours.fr/dssvalog`.

be used during a real large-scale evacuation but for the preparation of an evacuation expertise. The decision makers can use it to keep in practice and gain experience. Therefore, the underlying optimization algorithms can be designed without focusing on speed but to meet some quality guarantees. The project is a French-German cooperation consisting of six different partners:

- German Academics or *D_Acad* (Research group on optimization of the Department of Mathematics at the University of Kaiserslautern and Prof. Dr. Stefan Ruzika from the University of Koblenz-Landau),
- German Industrial Partners or *D_Ind* (Software company INFORM GmbH)
- German End Users or *D_User* (Authorities and organizations responsible for safety in Kaiserslautern, called "Arbeitskreis Gefahrenabwehr Kaiserslautern")
- French Academics or *F_Acad* (The group "Scheduling and Control" of the Laboratory of Computer Science and the group "IPAPE" of the Laboratory CITERES, both at the University François Rabelais of Tours)
- French Industrial Partners or *F_Ind* (CERVVAL, company specialized in simulation software development)
- French End Users or *F_User* (BRGM, the French geological survey)

Each group has its own range of authority. In the following, we only explain each task in a brief way since a detailed description would be beyond the scope of this introduction. D_User and F_User provide the information needed for the design of the algorithms and for the definition of scenarios. They design possible evacuation scenarios due to a found bomb in Kaiserslautern, Germany, and a flood emergency triggered by an earthquake in Nice, France. The role of F_Acad and D_Acad consists in the broadest sense of fundamental mathematical research in this field. They analyze different subproblems arising from large-scale evacuation modeling and develop theoretical concepts including quality guarantees and appropriate algorithms with their implementations. D_Ind is responsible for a user-friendly graphical user interface (demonstrator) for the final DSS and F_Ind develops simulation software such that the evacuation scenarios can be evaluated appropriately. Moreover, each group is assigned to different working packages. The different groups with their assigned working packages are depicted in Figure 1.1. There, the arrows indicate the dependencies of the different working packages. Prof. Dr. Stefan Ruzika and I are responsible for the multicriteria working package (Evac-MCDM) within D_Acad.

The abstracted process which is necessary for the development of the DSS is sketched in Figure 1.2. Initially, the *real world problem* which is defined by F_User and D_User has to be transformed into a *mathematical model*. In this project, the focus lies on multicriteria models, i.e., models with several conflicting objective functions. These models are mainly developed in other working packages within F_Acad and D_Acad.

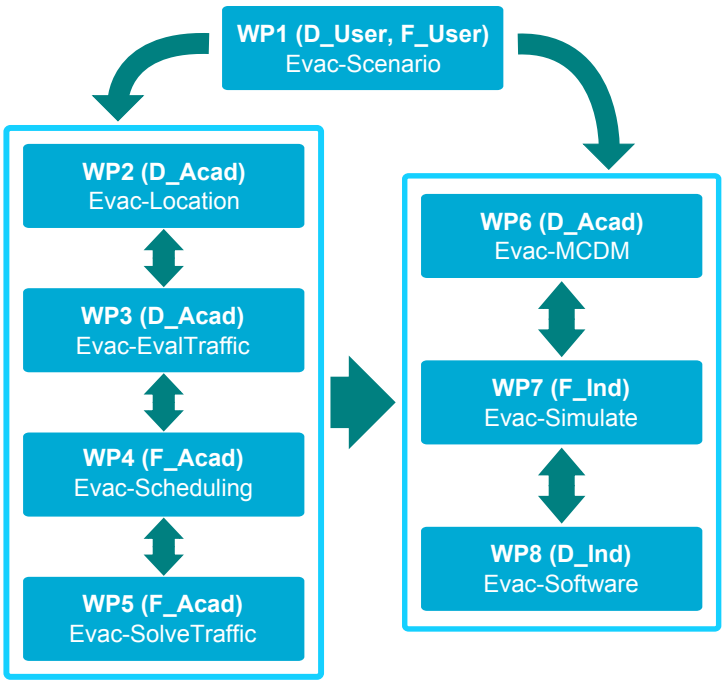


Figure 1.1: Working Packages (WP) in DSS_Evac_Logistics

The main task of Evac-MCDM is the development of a *generic algorithm* coping with such multicriteria optimization problems. Due to the conflicting nature of such problems, there does not exist a unique optimal solution in general. Instead, there are many so-called *nondominated solutions*, each of which cannot be improved in one objective function without deteriorating another objective function. The set of all nondominated solutions is called *nondominated set*. In many cases, this solution set is too big and computationally unattractive. One way to circumvent the computation of the whole nondominated set is to build a finite set, called *representative system*, such that the whole nondominated set is represented appropriately, i.e., fulfilling special quality guarantees. This representative system is then stored in a *database*. In the last step, we have to develop concepts to *support the decision makers* in exploring the whole database and in choosing one or more solutions from this database.

Furthermore, a concluding task of the project is to make the first move towards the theoretical analysis of the notion of an "integrative model" to understand how different optimization models can be combined with each other. This topic should also be further investigated in future research projects.

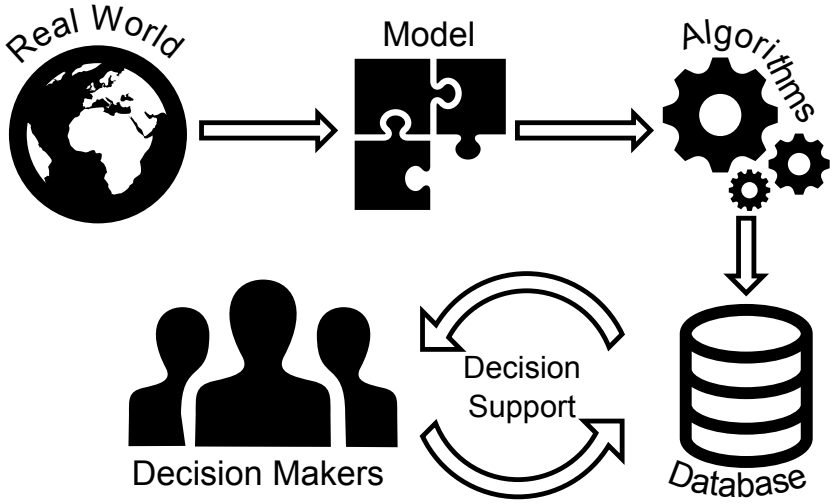


Figure 1.2: Sketch of the DSS

Contribution of the thesis

Motivated by evacuation logistics, the goal of this thesis is threefold:

- Innovative mathematical methods for computing representative systems are developed. All methods are theoretically supported so that their usage in applications leads to quantifiable and confident decisions.
- Practical tools for supporting the decision makers based on sound mathematical models and algorithms are explored, implemented and tested. Their benefits comprise fast computation, intuitive handling and effective application in action.
- Complex systems are often composed of partial systems. This composition is often not explicit and the mathematical treatment of this situation in the modeling process has received little attention. This thesis aims at providing a first step toward a rigorous analysis of the notion of combined complex systems.

Hence, this thesis investigates several theoretical and practical problems occurring in the context of conceiving and building a decision support system for real world problems modeled as multicriteria optimization problems. The main contribution of the thesis shall be outlined in the following.

Basic concepts, the notation used in this document as well as fundamental definitions regarding multicriteria optimization and, in particular, the approximation of the nondominated set are given in Chapter 2.

Chapters 3 to 5 are dedicated to the construction of representative systems for multicriteria optimization problems.

In Chapter 3, we extend the survey article of Ruzika and Wiecek (2005) and give an overview of existing methods for approximating the nondominated set published from 2003 to 2012. The reviewed articles are categorized in bicriteria and multicriteria approaches as well as discrete approximations and approximations of the 1^{st} order.

Chapter 4 first reviews the original Box-Algorithm for two objectives by Hamacher et al. (2007) and discusses several extensions regarding the coverage property, uniformity property, the enumeration of the whole nondominated set, and necessary modifications if the underlying scalarization problem cannot be solved to optimality.

In Chapter 5, the original Box-Algorithm is extended to the case of three objective functions to compute a representative system with desired coverage error. Besides the investigation of several theoretical properties, we prove the correctness of the algorithm, derive a bound on the number of iterations needed by the algorithm to meet the desired coverage error, propose some ideas for possible extensions, and apply the algorithm to the Bus Evacuation Problem.

Chapters 6 and 7 provide concepts for supporting the decision makers in their final choice after a representative system was computed and stored in a database.

The problem of selecting a subset with desired cardinality k from the computed representative system meeting a special quality criterion is subject of Chapter 6. There, we first investigate the bicriteria Hypervolume Subset Selection Problem (bicriteria k-HSSP) and derive two algorithms based on linear programming and a k-link shortest path problem, respectively, to efficiently solve the bicriteria k-HSSP. With the latter algorithm, we obtain the currently best known complexity bound for solving the bicriteria k-HSSP. Subsequently, we propose an integer programming formulation with a corresponding branch-and-bound scheme for the tricriteria k-HSSP and conclude with computational results.

The presentation of the computed representative system to the decision makers and the guidance through it is treated in Chapter 7. There, we first give a short overview of existing methods to visually display alternatives to the decision makers, propose a new illustration method based on bar charts, and develop two algorithms guiding the decision makers through the database of all computed representative points.

Finally, in Chapter 8, we step back and look from a meta-level on the issue of how to combine two given optimization problems and how the resulting combinations can be related to each other. We come up with several different combined formulations and give some ideas for the practical approach.

Acknowledgments

First and foremost, I would like to thank my supervisor Prof. Dr. Stefan Ruzika, who gave me this opportunity to work in the Optimization Research Group at the University of Kaiserslautern and to write this thesis. I am grateful for his encouragement and support throughout my studies. I also thank Prof. Dr. habil. Xavier Gandibleux for his effort in serving as a co-referee for this thesis.
Furthermore, I would like to thank Prof. Dr. Horst W. Hamacher for many fruitful discussions about several topics from this thesis and Karl Bringmann for the hint of the on-the-fly cost calculation which is utilized in Observation 6.14 in a similar manner.
My special thanks go to Marc Goerigk, Bob Grün and Abdullah Karatas for proofreading parts of this document.
For creating an excellent working atmosphere, I want to thank all my present and former colleagues in the Optimization Research Group. Especially, I would like to thank Sebastian Schmitt, Philipp Heßler and Corinna Heßler for the great office atmosphere.
I also want to take this opportunity to thank Luís Paquete and Carlos M. Fonseca for the beneficial collaboration and the hospitality during my visits to the University of Coimbra.
Furthermore, I gratefully acknowledge the financial support of the German Federal Ministry of Education and Research (BMBF) within the project DSS_Evac_Logistics (Decision Support System for Large-Scale Evacuation Logistics, FKZ 13N12229) . I am also thankful to the German Academic Exchange Service (DAAD, FKZ 57128839), allowing me to do parts of my research in Portugal.
Last but not least, I would like to express my deepest gratitude to my family, especially to Christine Mehlich and my parents Birgit and Johannes Kuhn, for their continuous support and encouragement.

Credits

The subject matter covered in Chapter 5 is based on joint work with Stefan Ruzika. This work is available as a technical report (Kuhn and Ruzika, 2014a) and is submitted for publication.

The results reported in Section 6.2 were obtained together with Carlos M. Fonseca, Luís Paquete and Stefan Ruzika. This work is available as a preprint (Kuhn et al., 2014) and is accepted for publication in *Evolutionary Computation* (Kuhn et al., 2015).

The results obtained in Chapter 8 is joint work with Stefan Ruzika. This work was presented at the *Joint GOR- and DASIG-Conference 2013* in Hamburg and is partially published in Kuhn and Ruzika (2014b).

This joint work represents many hours of fruitful discussions with my colleagues mentioned. I am grateful to all my coauthors for the good and beneficial collaboration and for allowing me to include our results in this thesis.

2 Preliminaries

In this chapter, basic concepts, definitions and the notation used throughout this thesis, in particular with respect to multicriteria optimization, are outlined. We assume that the reader is familiar with the basic concepts of combinatorial optimization, linear and integer programming, as well as complexity theory. For a detailed insight in these topics, we refer the reader to the text books of Hamacher and Klamroth (2006) and Nemhauser and Wolsey (1999).

2.1 Basics of Multicriteria Optimization

The focus in this thesis lies on multiple objective optimization problems. We mainly adopt the notation of Ehrgott (2005). By abuse of notation, in this thesis, the notions of "multicriteria optimization" and "multiple objective optimization" are both used to describe the field of optimization with more than one objective function.

Definition 2.1 (Multiple Objective Optimization Problem):
A general *multiple objective optimization problem* can be stated as

$$\begin{aligned} \text{(MOP)} \quad & \min \quad f(x) \\ & \text{s.t.} \quad x \in X \subseteq \mathbb{R}^n \end{aligned}$$

with a vector-valued objective function $f = (f_1, \ldots, f_p) : X \longrightarrow \mathbb{R}^p$ composed of $p \geq 2$ real-valued objective functions $f_i : X \longrightarrow \mathbb{R}$, $i = 1, \ldots, p$, also called objectives. In this context, $\mathbb{R}^p$ is referred to as the *outcome space* or also *objective space* and $\mathbb{R}^n$ as the *decision space*. The set X is called the *feasible set* and the image of it $Y := f(X)$ the *outcome set* containing the *outcomes*, i.e., images under f of all feasible solutions.

For the special cases $p = 2$ and $p = 3$, MOP is also called biobjective optimization problem and triobjective optimization problem, respectively. ◇

Since in general more than one real-valued objective functions are to be optimized simultaneously, the notion of optimality, i.e., the meaning of the term "min", has to be specified.

Definition 2.2:
The following notation allows comparisons between two vectors $u, v \in \mathbb{R}^p$ $(p > 1)$ based on the componentwise ordering.

$$\begin{aligned} u \leqq v \quad &:\Leftrightarrow \quad u_i \leq v_i \quad \forall i = 1, \ldots, p \\ u \leq v \quad &:\Leftrightarrow \quad u \neq v \text{ and } u_i \leq v_i \quad \forall i = 1, \ldots, p \\ u < v \quad &:\Leftrightarrow \quad u_i < v_i \quad \forall i = 1, \ldots, p \end{aligned}$$

Using these binary relations on $\mathbb{R}^p$, we define the cones $\mathbb{R}^p_{\leqq} := \{x \in \mathbb{R}^p : x \leqq 0\}$, $\mathbb{R}^p_{\leq} := \{x \in \mathbb{R}^p : x \leq 0\}$ and $\mathbb{R}^p_{<} := \{x \in \mathbb{R}^p : x < 0\}$. ◇

Definition 2.3 (Dominance):
For two vectors $u, v \in \mathbb{R}^p$, we say *u dominates v* if $u \leq v$. Moreover, for two feasible solutions $x^1, x^2 \in X$ of MOP, we also say x^1 *dominates* x^2 if $f(x^1)$ dominates $f(x^2)$. ◇

Definition 2.4 (Efficiency, Nondominance):
A feasible solution $x^* \in X$ of MOP is called *weakly efficient* and its outcome $f(x^*) \in Y$ is called *weakly nondominated* if there does not exist another solution $\bar{x} \in X$ with $f(\bar{x}) < f(x^*)$. x^* is called *efficient* and $f(x^*)$ *nondominated* if there does not exist another solution $\bar{x} \in X$ dominating x. The set of all efficient and nondominated solutions are denoted by X_E and Y_N, and they are called the *efficient set* and *nondominated set*, respectively. In other words, it is

$$\begin{aligned} X_E &:= \{x^* \in X | \nexists x \in X : f(x) \leq f(x^*)\} \text{ and} \\ Y_N &:= f(X_E) = \{y \in Y | \exists x \in X_E : y = f(x)\}. \end{aligned}$$ ◇

Solving a multiple objective optimization problem is generally understood as computing a *minimal complete set of efficient solutions*, i.e., the nondominated set has to be found and, for each nondominated point, exactly one corresponding efficient solution. Our methodology implies that a knowledgeable, intellectual instance named *decision makers* chooses one finally preferred efficient solution.

Definition 2.5:
For MOP, the *ideal, nadir* and *anti-ideal point* are defined as

$$y_i^I := \min_{x \in X} f_i(x), \qquad y_i^N := \max_{x \in X_E} f_i(x) \quad \text{and} \quad y_i^{AI} := \max_{x \in X} f_i(x) \quad \forall i = 1 \ldots, p\,,$$

respectively.

For $k \in \{1, \ldots, p\}$, an outcome $y \in Y$ with $y_k = y_k^I$ is called an *individual minimum* of the k-th objective function. ◇

Definition 2.6 (Lexicographic Minima):
For MOP, let $(i_1, \ldots, i_p) \in \{1, \ldots, p\}^p$ with $i_j \neq i_k$ for $j \neq k$. We call $x \in X$ the *lexicographic minimum w.r.t. the ordering* $(i_1, \ldots, i_p)$ if there does not exist another $\bar{x} \in X$ with $f_{i_{j^*}}(\bar{x}) < f_{i_{j^*}}(x)$, $j^* := \min\{j \in \{1, \ldots, p\} : f_{i_j}(x) \neq f_{i_j}(\bar{x})\}$. Additionally, if $(i_1, \ldots, i_p) = (1, \ldots, p)$, we call x the *lexicographic minimum in normal order*; if $(i_1, \ldots, i_p) = (p, \ldots, 1)$, we call x the *lexicographic minimum in reversed order*. ◇

Definition 2.7 (Externally Stable):
For MOP, let $\hat{Y} \subseteq Y$. The nondominated set Y_N is said to be *externally stable w.r.t.* $\hat{Y}$ if for each $y \in \hat{Y}$ there is a $y^* \in Y_N$ such that $y \in y^* + \mathbb{R}^p_{\geqq}$. If Y_N is externally stable w.r.t. Y, we simply call Y_N or also MOP to be *externally stable*. ◇

2.2 Approximation and Quality Measures

In practice, it is often not easy nor desirable to compute a minimal complete set of efficient solutions, i.e., an exact description of the nondominated set Y_N for a given multicriteria optimization problem MOP since this demands too much in many cases:

- For nonlinear problems, i.e., if X is defined by nonlinear inequalities and equalities and if f is a nonlinear function, the efficient set consists of infinitely many points and a closed form characterization of X_E is hardly ever possible. Computing a minimal complete set is then not a feasible task (Ehrgott and Wiecek, 2005).
- For linear, (mixed-)integer, and combinatorial problems, the set of efficient (extreme) points is countable, but in general exponential in the input size of the problem (Ehrgott and Gandibleux, 2000). Thus, any algorithm computing a minimal complete set of efficient solutions has exponential (worst-case) running time.
- In practical applications, the nondominated set is very large. Decision making based on many points is often a time-consuming, tedious task (Benson and Sayın, 1997).

As a consequence, many research articles focus on the computation of an approximation of Y_N in an appropriate way with simple structures and quality guarantees (see, e.g., the surveys of Ruzika and Wiecek (2005) and Ehrgott and Gandibleux (2004)). An approximation is in the broadest sense understood as any substitute for the nondominated set.

Definition 2.8 (Representative System):
For some MOP with outcome set Y, we call a finite approximation $Rep \subseteq Y$ *representative system* and its elements *representative points*.

◇

For the sake of fast and accurate decision making, such a representative system should come with some quality guarantees ensuring that the nondominated set Y_N is represented appropriately.

Definition 2.9 (Quality Measures):
Let Y_N denote the nondominated set of some MOP. Let $Rep \subseteq \mathbb{R}^p$ denote a representative system for MOP and $\|\cdot\|$ an arbitrary norm.

a) The *coverage error* of the representative system is defined as

$$\max_{y \in Y_N} \min_{z \in Rep} \|z - y\| .$$

b) The *uniformity* of the representative system is defined as

$$\min_{\substack{y,\hat{y} \in Rep \\ y \neq \hat{y}}} \|y - \hat{y}\| .$$

c) The *cardinality* of the representative system is defined as the cardinality of Rep, i.e., $|Rep|$.

d) The *representation error* of the representative system is defined as

$$\max_{z \in Rep} \min_{y \in Y_N} \|y - z\| . \qquad \diamond$$

The first three measures were proposed by Sayın (2000) and the representation error was introduced in Ruzika (2007). The *coverage error* measures the largest distance of any nondominated point to its closest representative point and, thus, quantifies how accurate the representative system represents the whole nondominated set. The *uniformity* measures the minimal distance between two different representative points, which quantifies the diversity of the representative system. The *cardinality* is related to the computational effort needed to obtain some representative system. The *representation error* meters the maximal distance of a representative point to its closest nondominated point. A "good" representative system has simultaneously a large uniformity as well as a small coverage error, cardinality and representation error. Note that these quality measures are conflicting by nature. For instance, the smaller the coverage error, the larger the cardinality of the representative system.

Some articles are also dealing with $(1+\varepsilon)$-*Pareto sets* in the decision space (see, e.g., Tsaggouris and Zaroliagis, 2009), i.e., a set of feasible solutions dominating every efficient solution up to the factor $(1+\varepsilon)$.

Definition 2.10 ($(1+\varepsilon)$-Pareto set):
Let $\varepsilon > 0$ and some MOP with feasible set X and objective function $f : X \to \mathbb{R}^p_{\leqq}$ be given. Then, a set $P_\varepsilon \subseteq X$ is called $(1+\varepsilon)$-*Pareto set* if it fulfills the following condition:

$$\forall \bar{x} \in X_E \; \exists x' \in P_\varepsilon : f(x') \leq (1+\varepsilon) f(\bar{x}) \qquad \diamond$$

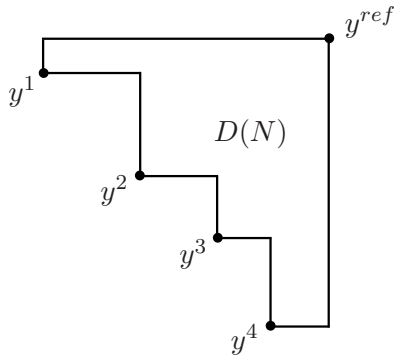


Figure 2.1: Dominated region of a given set $N = \{y^1, \ldots, y^4\}$ with reference point y^{ref}

Another quality measure which is widely used in heuristic approaches is the *hypervolume indicator*. Given a set of nondominated points in the objective space, the *hypervolume indicator* measures the size of the region in the corresponding space dominated by this set and bounded from above by some reference point (Zitzler and Thiele, 1998).

Definition 2.11 (Hypervolume Indicator):
Let $N = \{y^1, \ldots, y^m\} \subseteq Y_N$ be a set of nondominated points for some MOP and let y^{ref} be a *reference point* satisfying $y^{ref} > y^i$ for all $i = 1, \ldots, m$. The set

$$D(N) \coloneqq \bigcup_{i=1}^{m} \left\{ y \in \mathbb{R}^p : y^{ref} \geqq y \geqq y^i \right\}$$

is called the *dominated region* of N (w.r.t. y^{ref}) and the *hypervolume indicator* of N (w.r.t. y^{ref}) is defined as $S(N) \coloneqq \lambda^{\mathcal{L}}(D(N))$ where $\lambda^{\mathcal{L}}(\cdot)$ denotes the Lebesgue measure in $\mathbb{R}^p$. ◇

The dominated region of a 4-element set is depicted in Figure 2.1.

2.3 Scalarization Methods

For computing an appropriate approximation of a given MOP, almost all developed algorithms follow a general scheme (Ruzika, 2007). They solve a sequence of single-criterion optimization problems with appropriate choices of parameters until the desired quality is reached. Therefore, we shortly introduce some relevant single-objective scalarization methods which are utilized in such algorithms. For detailed properties of the considered scalarization methods, in particular necessary and sufficient conditions

to obtain (weakly) efficient solutions, and corresponding proofs, we refer the reader to Ehrgott (2005).

We assume some given MOP as described in Definition 2.1.

Definition 2.12 (Weighted Sum Method):
For some (normalized) weight vector $\lambda \in \mathbb{R}^p_{\geq}$, the *weighted sum method* or also *weighted sum scalarization* is given by:

$$\begin{aligned} \min \quad & \sum_{k=1}^{p} \lambda_k f_k(x) \\ \text{s.t.} \quad & x \in X \end{aligned} \qquad \diamond$$

In the *weighted sum method*, all p objective functions are combined by means of a convex combination into one objective function and the feasible set remains the same.

In the *ε-constraint method*, one objective function, e.g., the first objective function, is fixed and all others are incorporated as additional constraints besides the feasible set.

Definition 2.13 (ε-Constraint Method):
For some $\varepsilon \in \mathbb{R}^p$, the *$\varepsilon$-constraint method* or also *ε-constraint scalarization* is given by:

$$\begin{aligned} \min \quad & f_1(x) \\ \text{s.t.} \quad & x \in X \\ & f_k(x) \leq \varepsilon_k \qquad k = 2, \ldots, p \end{aligned} \qquad \diamond$$

An extension of this method is a lexicographic objective function, which is utilized in Chapters 4 and 5.

In the *weighted Tchebycheff method*, a feasible point is desired having minimal distance to a given reference point w.r.t. the corresponding weighted Tchebycheff metric.

Definition 2.14 (Weighted Tchebycheff Method):
For some (normalized) weight vector $\lambda \in \mathbb{R}^p_{\geq}$ and reference point $y^r \in \mathbb{R}^p$ with $Y \subseteq y^r + \mathbb{R}^p_{\geq}$, the *weighted Tchebycheff method* is given by:

$$\begin{aligned} \min \quad & \max_{k=1,\ldots,p} \lambda_k (f_k(x) - y^r_k) \\ \text{s.t.} \quad & x \in X \end{aligned} \qquad \diamond$$

Here, the reference point is normally chosen as the ideal point y^I. However, this method can also be used locally by restricting the feasible set appropriately and allowing also other reference points y^r not necessarily fulfilling $Y \subseteq y^r + \mathbb{R}^p_{\geq}$. To ensure efficiency of the obtained solutions, an *augmented weighted Tchebycheff method* can be used, in which a small augmentation term is added to the objective function (Steuer and Choo, 1983).

3 Overview of Existing Methods to Approximate the Nondominated Set

A detailed survey of methods for computing an approximation of the nondominated set up to the year 2003 can be found in Ruzika and Wiecek (2005). In this chapter, we extend this survey article and review methods from literature from 2003 to 2012. We focus on exact approaches without considering interactive and heuristic approaches (such as genetic algorithms, simulated annealing, tabu search, etc.). Moreover, also methods which are solely applicable for very restricted types of multicriteria optimization problems are excluded. We first consider approaches which are only applicable to bicriteria optimization problems and, afterwards, approaches which are also capable to handle problems with more than two objective functions. We categorize them by their utilized scalarization method and by their used approximation structure, i.e., 0^{th} or 1^{st} order (cf. Ruzika and Wiecek, 2005). Thereby, a first observation is that most reviewed articles concentrate on the outcome space, i.e., the nondominated set, instead on the decision space. In most cases, the nondominated set has a simpler structure and a smaller size than the efficient set (Benson, 1995). Note that this review is not complete since, for instance, papers which are not freely available (due to the license of my university) are not covered, but this review should reflect the continuous development in this field adequately.

3.1 Bicriteria Approaches

Most reviewed articles dealing with bicriteria optimization problems (BOP) consider an approximation of the 0^{th} order, i.e., representative systems with representative points. The approximation algorithms can be categorized due to the scalarization method which is used to compute an appropriate representative point in each iteration.

Three considered articles, Kouvelis and Sayın (2006), Ralphs et al. (2006), and Dutta and Kaya (2011), are based on the (weighted) Tchebycheff method as scalarization method. Kouvelis and Sayın (2006) (see also Sayın and Kouvelis (2005)) propose an algorithm for discrete BOPs. They use a two-stage scalarization method in which the first stage corresponds to the Tchebycheff method, and the second stage identifies the efficient solutions among all optimal solutions from the first stage. Initially, the authors determine the lexicographic minima and then iteratively choose two already found

neighboring points. The weights for the Tchebycheff method are defined appropriately to search for nondominated points between the two chosen points. The algorithm terminates if all neighboring points fulfill a coverage error check. If this check is ignored, their algorithm generates the whole nondominated set. Ralphs et al. (2006) use a similar approach and consider bicriteria integer optimization programs. Also their algorithm first determines the two lexicographic minima and processes neighboring points with appropriately chosen weights with the Tchebycheff method (but without a second stage). Their algorithm can identify all nondominated points or, by selecting unexplored intervals in a first-in-first-out manner and terminating the algorithm earlier, the algorithm can produce an approximation of Y_N. Dutta and Kaya (2011) consider BOPs with continuous objective functions and propose a Tchebycheff-type scalarization method along rays through a utopia point.[1] The desired cardinality of the representative system is an input of the algorithm. They use the two angles of the rays induced by the two lexicographic minima to generate a grid of equidistant angles. These angles are then exploited to define the parameters of their scalarization method to obtain a representative system with desired cardinality.

Another famous scalarization method is the weighted sum method, which is used by the authors of two reviewed articles, Kim and de Weck (2005) and Heermann et al. (2005). The approach of Kim and de Weck (2005) is based on an adaptive version of the weighted sum method, which focuses on yet unexplored regions. The algorithm is controlled by two parameters, an offset distance controlling the distribution of the final representative system and an overlapping distance which controls the uniformity. They first perform the classical weighted sum method with a uniform step size. Thereafter, they compute the distance of the segments between all neighboring outcomes and delete solutions according to the overlapping parameter. For a segment which needs further refinements, the weighted sum scalarization with additional upper bounds for the two objective functions is invoked. The algorithm terminates if the maximal length of all segments is at most the given offset distance. Heermann et al. (2005) provide an algorithm for BOPs with a polyhedral feasible set and convex quadratic objective functions. For solving the upcoming weighted sum scalarization problems, the authors use an interior point method. The intermediate points of the interior point method are recursively utilized to obtain warm-start points for other weighted sum scalarization problems which are successively constructed as perturbed problems. Hence, the algorithm solves several scalarization problems simultaneously and proceeds until the maximal distance of generated neighboring points has reached a given tolerance.

Five articles, Hamacher et al. (2007), Fernández and Tóth (2007), Eichfelder (2009b), Diakonikolas and Yannakakis (2009), and Faulkenberg and Wiecek (2012), are based on the ε-constraint method. Hamacher et al. (2007) consider discrete BOPs and construct rectangles to cover the whole nondominated set and to control the accuracy.

[1] For MOP, a *utopia point* is defined as any reference point in $y^I - \mathbb{R}^p_>$.

The rectangles are subdivided with an adaptively chosen lexicographic ε-constraint scalarization problem, bisecting the horizontal side of the currently considered rectangle. A detailed description of this algorithm is given in the introduction of Chapter 4. Fernández and Tóth (2007) do not aim at an approximation of the 0^{th} order but at an outer approximation containing the whole efficient set of a general BOP. Therefore, the authors solve the considered ε-constraint problems not to optimality but calculate all points which are optimal up to a predefined tolerance. This is done with the help of a branch-and-bound method from the literature. The authors incorporate the already obtained region by iteratively modifying the ε-constraint problem with a lower bound placed at the objective function which should be minimized. Eichfelder (2009b) considers the *Pascoletti-Serafini scalarization* (Pascoletti and Serafini, 1984), for which, besides other well-known problems, the ε-constraint scalarization can be considered as a special case of it. The author uses sensitivity results from Eichfelder (2009a) to propose an algorithm relying on adaptive parameter control for the general bicriteria Pascoletti-Serafini scalarization approach. After one problem is solved, the adaptive parameter control aims at generating almost equidistant points by choosing the next parameters such that a prespecified distance between neighboring points is fulfilled. It is outlined how this method explicitly works for some special cases of the considered scalarization approaches, including the ε-constraint scalarization. Diakonikolas and Yannakakis (2009) look at linear BOPs for which the corresponding ε-constraint problem can be solved efficiently. The authors give a simple greedy algorithm which solves several appropriately adjusted ε-constraint problems. For a given tolerance δ, the output is a $(1+\delta)$-Pareto set with minimal cardinality. Moreover, for (not necessarily linear) BOPs for which the ε-constraint problem cannot be solved exactly but it can be efficiently approximated in an appropriate way, the authors give another algorithm computing a $(1+\delta)$-Pareto set with no more points than two times the minimal possible cardinality. Faulkenberg and Wiecek (2012) propose two different approaches and concentrate for a predefined desired cardinality on the generation of equidistant (weakly) nondominated points for general BOPs with sufficient smoothness conditions. The first approach is based on an ε-constraint scalarization with an additional constraint setting a lower bound on the distance to a previously found point. The second approach utilizes a scalarization method based on bilevel programming to get equidistant weakly nondominated points. Here, the lower-level problem is a classical ε-constraint problem to obtain weakly efficient solutions, in which the distance to previous points is controlled in the upper-level problem.

Fernández and Tóth (2009) also consider another method to obtain an outer approximation by means of boxes of all weakly efficient solutions. Their method is based on an interval branch-and-bound algorithm and appropriate box divisions. The algorithm terminates when the maximal (relative) width of all boxes reaches some accuracy parameter.

3.2 Multicriteria Approaches

In contrast to bicriteria approaches, more authors dealing with multicriteria approaches focus on approximations of the 1^{st} order, i.e., outer or inner approximations by polyhedral sets or other sets. Therefore, we consider two categories, approximations of the 0^{th} and 1^{st} order maybe with additional resulting discrete approximation. In each subsection, we further collect approaches using similar scalarization methods.

3.2.1 Approximations of the 0^{th} Order

Four articles, Kim and de Weck (2006), Fliege (2006), Sylva and Crema (2007), and Masin and Bukchin (2008), use the weighted sum method in their scalarization problems. Kim and de Weck (2006) extend their bicriteria method from Kim and de Weck (2005) to work with more than two objectives. To determine the needed refinement, they identify patches spanned by already obtained outcomes, estimate the size of each patch and compare it to the mean value of the lengths of all patches. In each such patch, expected locations for nondominated solutions are determined by interpolation, which are then used to restrict the scalarization problem along the line connecting an expected solution and an estimate of the nadir point. In contrast to their bicriteria method, this algorithm can output some dominated solutions and it is not guaranteed to cover the whole nondominated set appropriately. However, they obtain a uniform distributed representative system and their method has the ability to investigate also non-convex regions. Fliege (2006) considers a convex feasible set with convex objective functions and assumes a special barrier function for the feasible set. As in Heermann et al. (2005), the author uses an interior point method to solve the weighted sum scalarizations as well as former obtained solutions as warm-start points. The algorithm computes a special covering of the weight-space and outputs the solutions of the corresponding scalarization problems. Sylva and Crema (2007) propose an algorithm for multicriteria mixed-integer programs. In each iteration, a special single-criterion mixed-integer programming problem is solved to obtain a point maximizing the distance to the whole dominated region of all already found nondominated points. Since this point needs not to be nondominated, a modified weighted sum method is invoked to find an appropriate nondominated point. The algorithm terminates if in some iteration the maximal distance to the current dominated region is less than a predefined tolerance or the desired cardinality is reached. Also Masin and Bukchin (2008) look at multicriteria problems for which an appropriate solver for the single-criterion case is available, e.g., multicriteria mixed-integer programs. In each iteration, they explicitly include a uniformity measure as constraint in the weighted sum method, which can be linearized using binary variables. This incorporated measure ensures that a nondominated point is chosen which is farthest away from the closest nondominated point already obtained.

The algorithm stops if the distance to the next point fulfills some given accuracy parameter. With different parameters for the incorporated uniformity measure, the authors can ensure different properties for the final representative system.

Messac and Mattson (2004) and Shao and Ehrgott (2007) aim to overcome the disadvantage to miss regions from the nondominated set (coverage) of two similar former published methods, the *normal constraint method* (NC) (Messac et al., 2003) and the *normal boundary intersection method* (NBI) (Das and Dennis, 1998). To generate the parameters for their approach, Messac and Mattson (2004) modify the NC method by using an extension of the reference plane which is defined through the convex hull of the individual minima. Therefore, they first create a hypercube enclosing the whole outcome set and enlarge the corresponding reference plane to enclose the whole projection of this hypercube. Moreover, they also remove regions that are dominated by the individual minima. The modified algorithm generates an even distribution of nondominated points while covering the whole nondominated set. Shao and Ehrgott (2007) investigate multicriteria linear programming problems and combine the *global shooting method* (Benson and Sayın, 1997) with the NBI method to guarantee coverage and evenly distributed points. The authors build a simplex enclosing the whole outcome set and use a subsimplex of it as reference plane. Thereafter, their method places equidistant reference points on the reference plain to find the closest outcomes along some given direction. In contrast to Messac and Mattson (2004), the authors give a proof for the selection of the parameters to obtain a desired uniformity and coverage error.

The developed method from Utyuzhnikov et al. (2009) relies on *physical programming* (Messac, 1996) and has conceptual similarities to the NBI and NC method. Their method also uses the reference plane defined through the individual minima to define parameters, including a box as search space, for a special single-objective optimization problem coming from physical programming, i.e., a special convex objective function is minimized within the current box guaranteeing (locally) efficient solutions. Their algorithm obtains an almost even approximation and is furthermore capable of generating the entire nondominated set.

The methods from Karasakal and Köksalan (2009) and Zhang and Gao (2006) are based on the (weighted) Tchebycheff method. Karasakal and Köksalan (2009) look at multicriteria linear programming problems. They first use a special augmented Tchebycheff method as scalarization problem with uniform spaced parameters to obtain an initial set of nondominated points. These points are then utilized to construct a special weighted L_q-surface (Karasakal and Köksalan, 2009). Thereafter, the authors select evenly distributed reference points on this surface and project these points onto Y_N. Zhang and Gao (2006) propose an adaptive scheme for selecting the weights and the reference point in the used Tchebycheff method such that uniform outcomes are obtained. Assuming sufficiently smoothness, the method iteratively utilizes the last

obtained solution in a two-stage approach. The method first calculates the normal direction for defining the weights and, afterwards, the corresponding tangent direction which is used to translate the reference point with a given step length.

For Lipschitz continuous objective functions, Galperin (2004) proposes a special set contraction algorithm working with boxes in the decision space. The algorithm recursively partitions the initial box into several subboxes, computes for each box a representative point and uses these points together with the Lipschitz condition to discard several subboxes not containing any efficient solution. Under special assumptions on the objective functions, it is shown that the union of subboxes converges to the efficient set. An iteration bound is proven for which a representative system is obtained fulfilling an accuracy depending on the diameter of the boxes.

Vassilvitskii and Yannakakis (2004) assume for their algorithm the existence of a special routine that either returns for a given reference point a solution dominating this point or reports that there does not exist any solution which is better than the reference point up to some tolerance (cf. Papadimitriou and Yannakakis, 2000). The authors present algorithms for different numbers of objective functions which are all based on the same idea of a geometric grid. For instance, for two objectives, their method divides the outcome space into rectangles with specific ratios and applies the special routine to the different corner points. Thereafter, they reduce the size of the obtained representative system with the help of a simple greedy algorithm. The output is a $(1+\varepsilon)$-Pareto set having reasonable size.

3.2.2 Approximations of the 1st Order

Four publications, Shao and Ehrgott (2008a), Shao and Ehrgott (2008b), Ehrgott et al. (2011), and Ehrgott et al. (2012), concentrate on cutting planes and vertex enumeration. All these methods are inspired by a former algorithm from Benson (1998) for calculating an outer approximation for multicriteria linear programming problems. Shao and Ehrgott (2008a) give some practical improvements for Benson's algorithm and propose an approximation version of it. They introduce two new sets for an outer and inner approximation. Instead aiming at the real vertices of the polyhedron in the outcome space, they also allow approximated vertices for which the distance to its projection onto the polyhedron is in some tolerance. After termination, the algorithm obtains feasible solutions which are weakly efficient up to an additive tolerance. The real nondominated set is "sandwiched" within the inner and outer approximation. Ehrgott et al. (2012) present a dual variant of Benson's outer approximation algorithm using results from geometric duality (cf. Heyde and Löhne, 2008). The dual variant works similar to Benson's algorithm but on the outcome set of the dual multicriteria linear programming problem. The algorithm outputs a non-degenerate inequality representation of the primal polyhedron from outcome space, all nondominated extreme

points of the primal problem as well as an appropriate weight set decomposition. Shao and Ehrgott (2008b) developed an approximation version of the proposed dual variant of Benson's algorithm. Instead of matching the real vertices of the dual polyhedron, the vertices are now allowed to fulfill some tolerance condition. The extreme points of the obtained outer approximation of the dual program define a set of solutions which are weakly efficient up to an additive tolerance for the primal problem. Thus, they obtain an inner approximation for the primal problem. In contrast to the aforementioned articles, Ehrgott et al. (2011) concentrate on the approximation of general convex multicriteria optimization problems. Their algorithm relies on the same ideas as the approximation version in Shao and Ehrgott (2008a), the approximation of the corresponding polyhedron in the outcome space by an outer and inner approximation. In each iteration, a special separating hyperplane is needed, which could be easily calculated for linear problems. For the case in which all functions are additionally continuously differentiable, the authors describe the calculation of a separating hyperplane; and, moreover, suggest a heuristic procedure for the non-differentiable case. Again, the algorithm outputs feasible solutions which are weakly efficient up to an additive tolerance and an inner as well as an outer approximation enclosing the nondominated set.

Rennen et al. (2011) and Lotov and Maiskaya (2012) use the weighted sum method to get approximations of the 1^{st} order. Rennen et al. (2011) describe ideas to overcome some problems which can occur in sandwich algorithms from literature calculating inner and outer approximations. For instance, the authors describe the incorporation of dummy points ensuring that all facets have a non-negative normal, which is crucial for the weighted sum method to guarantee weakly nondominated solutions. Moreover, also a criterion is given to identify dominated points in the inner approximation. A modified version of the sandwich algorithm from Solanki et al. (1993) is proposed and tested. Lotov and Maiskaya (2012) aim at a polyhedral inner approximation and propose two methods which work very similar. Both methods exploit a collection of unit directions obtained with a so-called *estimate refinement method* in a preprocessing step. These directions are used to define appropriate parameters for the weighted sum scalarization. The second algorithm chooses the directions in a more involved way leading to a better covering of the whole nondominated set.

Lehtonen and Pulkkinen (2010) utilize the augmented weighted Tchebycheff method to construct an outer approximation. In each iteration, their algorithm projects the vertices of the current outer approximation onto the nondominated set to refine the outer approximation. Under some special assumptions on the outcome set, the obtained nondominated points converge to the whole nondominated set.

A polyhedral inner and outer approximation for multicriteria integer optimization problems with monotone objective and constraint functions is the purpose of the method from Pospelov (2009). The author uses a branch-and-bound scheme to construct sequences of inner and outer approximations until the distance between these two

approximation structures has reached a desired accuracy.

Similar to Vassilvitskii and Yannakakis (2004), Legriel et al. (2010) assume the existence of a special routine that either returns for a given reference point a solution dominating this point or reports that there does not exist any solution which is better than the reference point up to some tolerance. This routine is used in their algorithm to update the inner and outer approximation iteratively by asking whether there is a solution for some special reference point. This reference point is selected due to the distance between the current inner and outer approximation such that the distance in the corresponding region is reduced.

The methods from Luc et al. (2005), Gourion and Luc (2008), and Gourion and Luc (2010) are all based on similar techniques to get outer approximations for their considered multicriteria optimization problems. The algorithm of Luc et al. (2005) successively constructs a sequence of polyhedra approximating a special super set of the outcome set, called the free disposal hull. To refine the current outer approximation, each vertex is orthogonally projected onto this super set. The algorithm stops when the maximal distance of all vertices to their corresponding projections has reached some tolerance. The obtained information can be used to construct a sequence of super sets converging to the set of all weakly nondominated points of the original problem. In contrast to Luc et al. (2005), Gourion and Luc (2008) successively construct a sequence of so-called free disposal polyhedra to approximate the free disposal hull of the outcome set. The outer approximation is refined by computing intersection points with a special hyperplane. This leads to a set of weakly nondominated points which converges to the whole set of all weakly nondominated points. Gourion and Luc (2010) extend this algorithm to obtain convergence to the closure of the nondominated set instead only to the weakly nondominated set.

4 Box-Algorithm for Bicriteria Optimization Problems

4.1 Introduction

In this chapter, we propose and discuss several extensions regarding the Box-Algorithm for bicriteria optimization problems (Hamacher et al., 2007); but first, we shortly review the original algorithm and prove some additional properties. Since the algorithm relies on an ε-constraint in the first coordinate, we refer to it as the *horizontal Box-Algorithm* or the *area-based Box-Algorithm*. In this chapter, we consider the following discrete bicriteria optimization problem

$$\begin{aligned} \text{(DBOP)} \quad & \min \quad f(x) = (f_1(x), f_2(x)) \\ & \text{s.t.} \quad x \in X \subseteq \mathbb{R}^n \end{aligned}$$

with $f : X \longrightarrow \mathbb{Z}^2$, where the existence of the ideal point y^I and nadir point y^N are assumed.

Definition 4.1 (Rectangle):
For two vectors $y^1, y^2 \in \mathbb{R}^2$, we call the set

$$\mathrm{R}(y^1, y^2) \coloneqq \left\{ y \in \mathbb{R}^2 : (y_1^1, y_2^2)^\top \leqq y \leqq (y_1^2, y_2^1)^\top \right\}$$

the *rectangle* spanned by y^1 and y^2. The two points y^1 and y^2 are called *left* and *right corner point* of the rectangle $\mathrm{R}(y^1, y^2)$, respectively.
We denote with $\mathrm{Ar}(\mathrm{R}(y^1, y^2)) \coloneqq (y_1^2 - y_1^1) \cdot (y_2^1 - y_2^2)$ the area of the rectangle.
For the maximum norm $\| \cdot \|_\infty$ on $\mathbb{R}^2$, the metric or distance $\|y^1 - y^2\|_\infty$ is referred to as the *corner point distance* of the rectangle $\mathrm{R}(y^1, y^2)$. ◇

For initialization, Hamacher et al. (2007) compute both lexicographic minima with outcome z^1 and z^2, add them to the representative system *Rep*, and use the rectangle $\mathrm{R}(z^1, z^2)$ as initial rectangle covering Y_N. In each iteration, an unexplored rectangle $\mathrm{R}(y^1, y^2)$ with greatest area value is investigated with the following lexicographic ε-constraint problem.

Definition 4.2 (Horizontal Subproblem):
For a rectangle $\mathrm{R}(y^1, y^2)$, let $\varepsilon := \lfloor (y_1^1 + y_1^2)/2 \rfloor$. Then, we define the *horizontal lexicographic ε-constraint scalarization* (P_ε^{hor}), also called *horizontal subproblem*, as

$$\begin{aligned} (P_\varepsilon^{hor}) \quad & \operatorname{lex\,min} \ (f_2(x),\ f_1(x)) \\ & \text{s. t.} \quad x \in X \\ & \qquad f_1(x) \leq \varepsilon \end{aligned}$$

where "lex min" calculates the lexicographic minimum in the given order under the given constraints. ◇

Lemma 4.3 (Hamacher et al. (2007)): *For an arbitrary rectangle* $\mathrm{R}(y^1, y^2)$*, let* $x^* \in X$ *be an optimal solution of* (P_ε^{hor})*. Then* $x^* \in X_E$*.*

If an optimal solution x^* of (P_ε^{hor}) is found and its image $z^* := f(x^*) \in Y_N$ is in $\mathrm{R}(y^1, y^2)$, then z^* is added to the representative system *Rep*. In this case, the current rectangle $\mathrm{R}(y^1, y^2)$ is substituted with the two rectangles $\mathrm{R}(y^1, z^*)$ and $\mathrm{R}(p^5, y^2)$ with $p^5 = (\varepsilon + 1, z_2^* - 1)^\top$ (see Figure 4.1) fulfilling the following result.

Lemma 4.4 (Hamacher et al. (2007)): *Let* z^* *be the image of an optimal solution of* (P_ε^{hor}) *corresponding to the rectangle* $\mathrm{R}(y^1, y^2)$*. Then, it holds*

$$Y_N \cap \mathrm{R}(y^1, y^2) \subseteq \mathrm{R}(y^1, z^*) \cup \mathrm{R}(p^5, y^2)\ .$$

Hence, in each iteration, the algorithm preserves a subdivision of rectangles containing the whole nondominated set. Moreover, for each rectangle, the right corner point is a known nondominated representative point.

The algorithm stops if the area of the largest rectangle in the current subdivision is less than or equal to some given accuracy value Δ.

Definition 4.5 (Area-Based Δ-Accuracy):
For DBOP, let a collection of rectangles $\mathcal{B}$ containing the whole nondominated set and a representative system *Rep* be given. Then for some $\Delta > 0$, *Rep* fulfills the *area-based Δ-accuracy* if for each rectangle $R \in \mathcal{B}$ there exists a representative point $z^* \in Rep$ with $z^* \in R$ and it holds $\mathrm{Ar}(R) \leq \Delta$. ◇

Moreover, the authors give an iteration bound for which the desired accuracy is guaranteed.

Theorem 4.6 (Hamacher et al. (2007)): *The horizontal Box-Algorithm terminates in finitely many steps. At termination, the algorithm yields a collection of rectangles and a representative system fulfilling the area-based Δ-accuracy in which all representative points are nondominated. More precisely, the algorithm performs at most* $\mathcal{O}\left(\frac{A}{\Delta}\right)$ *many iterations, where A denotes the area of the initial rectangle* $\mathrm{R}(z^1, z^2)$*.*

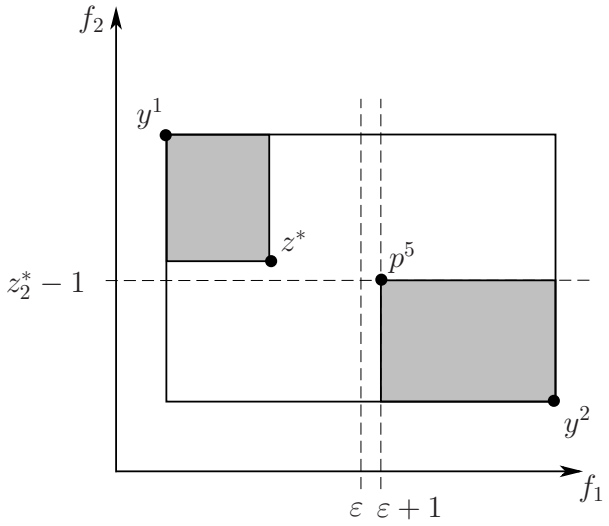


Figure 4.1: Horizontal updating of a rectangle $\mathrm{R}(y^1, y^2)$

From the construction of the rectangles during the algorithm, we get the *"distinct-rectangle-property"*.

Lemma 4.7 (Distinct-Rectangle-Property): *Let $\mathcal{B}^i$ denote the collection of unexplored and finished rectangles in iteration $i \geq 1$ of the horizontal Box-Algorithm, before the current rectangle is substituted. In iteration i, for each pair of distinct rectangles $\mathrm{R}(v^1, v^2),\ \mathrm{R}(w^1, w^2) \in \mathcal{B}^i$, exactly one of the following statements holds:*

(i) $v_1^2 < w_1^1$ *and* $v_2^2 > w_2^1$

(ii) $w_1^2 < v_1^1$ *and* $w_2^2 > v_2^1$

This property is called distinct-rectangle-property*.*

Proof:
The claim is shown by induction on the number of iterations i.

In the first iteration, the claim is trivially fulfilled.

Suppose we are after iteration i and all rectangles from $\mathcal{B}^i$ fulfill the distinct-rectangle-property. Let iteration i correspond to the refinement of the unexplored rectangle $\mathrm{R}(y^1, y^2) \in \mathcal{B}^i$ and let z^* denote the image of an optimal solution of the corresponding horizontal subproblem.

If $z^* \in \mathrm{R}(y^1, y^2)$, then from the construction of the two new rectangles substituting $\mathrm{R}(y^1, y^2)$, the distinct-rectangle-property is fulfilled for the new subdivision $\mathcal{B}^{i+1}$.

If $z^* \notin \mathrm{R}(y^1, y^2)$, the horizontal Box-Algorithm discards the left side $\mathrm{R}(y^1, y^3)$ with $y^3 = (\varepsilon, y_2^2)^\top$ of the current rectangle and substitutes the current rectangle with the right side $\mathrm{R}(y^4, y^2)$ with $y^4 = (\varepsilon + 1, y_2^1)$.[1] Hence, the distinct-rectangle-property is still valid for the new subdivision $\mathcal{B}^{i+1}$. □

The validity of the horizontal Box-Algorithm is related to the following result.

Theorem 4.8 (Correctness Property): *In an arbitrary iteration in the horizontal Box-Algorithm, let $z^* \in Y_N$ be the image of an optimal solution corresponding to the refinement of some rectangle $\mathrm{R}(y^1, y^2)$. Then, we have*

$$z^* \notin \mathrm{R}(y^1, y^3) \Longleftrightarrow \mathrm{R}(y^1, y^3) \cap Y_N = \emptyset$$

where $y^3 = (\varepsilon, y_2^2)^\top$. Note that the lexicographic minimum in normal order is always a feasible solution of any upcoming subproblem.

This property is called correctness property*.*

Proof:
Let $\mathcal{B}^i$ be defined as in Lemma 4.7. Consider an arbitrary iteration $i \geq 1$ corresponding to the refinement of rectangle $\mathrm{R}(y^1, y^2) \in \mathcal{B}^i$ with image $z^* \in Y_N$ of an optimal solution.

From Lemma 4.7, it holds the distinct-rectangle-property for iteration i.

If $z^* \notin \mathrm{R}(y^1, y^3)$, then it has to be in a rectangle $\mathrm{R}(s^1, s^2) \in \mathcal{B}^i$ with $s_1^2 < y_1^1$ due to the ε-constraint and the distinct-rectangle-property. Again, due to the distinct-rectangle-property, we get $z_2^* \geq s_2^2 > y_2^1$ implying $\mathrm{R}(y^1, y^3) \cap Y_N = \emptyset$ since otherwise, any element in $\mathrm{R}(y^1, y^3) \cap Y_N$ would contradict the optimality of z^* for the horizontal lexicographic ε-constraint scalarization.

If $z^* \in \mathrm{R}(y^1, y^3)$, we immediately get $\mathrm{R}(y^1, y^3) \cap Y_N \neq \emptyset$ since we know from Lemma 4.3 that z^* is nondominated. □

4.2 Extensions of the Box-Algorithm

4.2.1 Adding the Coverage Property

In this subsection, we do not aim at the area-based Δ-accuracy (cf. Definition 4.5). Here, we are interested in the incorporation of the coverage error into the Box-Algorithm, i.e., after termination, we aim at a representative system having a coverage error less than or equal to some predefined coverage-tolerance $\delta^C > 0$ (cf. Definition 2.9). If not

[1] Note that this special case is not treated in the pseudocode description in Hamacher et al. (2007), but it is obvious to consider this case (cf. Algorithm 4.1 on p. 33).

stated otherwise, the coverage error is measured with respect to the maximum norm $\|\cdot\|_\infty$.

Besides the horizontal subproblem (P_ε^{hor}), we also look at the *vertical subproblem.*

Definition 4.9 (Vertical Subproblem):
For a rectangle $\mathrm{R}(y^1, y^2)$, let $\varepsilon := \lfloor (y_2^1 + y_2^2)/2 \rfloor$. Then, we define the *vertical lexicographic ε-constraint scalarization* (P_ε^{vert}), also called *vertical subproblem*, as

$$\begin{aligned} (P_\varepsilon^{vert}) \quad \operatorname{lex\,min} \ & (f_1(x), f_2(x)) \\ \text{s.t.} \ & x \in X \\ & f_2(x) \leq \varepsilon \end{aligned} \qquad \diamond$$

Analogous to the analysis of the horizontal problem (Hamacher et al., 2007), we get the following result.

Observation 4.10:
For an arbitrary rectangle $\mathrm{R}(y^1, y^2)$ and $x^* \in X$, it holds:

- x^* optimal for $(P_\varepsilon^{vert}) \implies x^* \in X_E$
- $x^* \in X_E \implies x^*$ optimal for (P_ε^{vert}) with $\varepsilon := f_2(x^*)$ $\triangleleft$

Clearly, in the horizontal Box-Algorithm, we can guarantee the desired coverage property if we choose the parameter Δ sufficiently small (cf. Subsection 4.2.3). However, in many cases, this leads to a complete enumeration of Y_N or to a representative system with an unnecessarily high cardinality since cutting each rectangle with too much area horizontally into halves could affect the vertical length of the rectangle in some iterations only marginally. This issue is demonstrated in the following example.

Example 4.11:
Let $k > 0$ be given and arbitrary. Suppose we have given a DBOP inducing the following $(k+6)$-element nondominated set

$$\begin{aligned} Y_N = A \cup B \cup C \quad \text{with} \quad & A = \left\{ (0, M)^\top, (5 \cdot 2^k, 0)^\top \right\}, \\ & B = \left\{ \left(1, \frac{M}{2^1}\right)^\top, \left(2, \frac{M}{2^2}\right)^\top, \left(3, \frac{M}{2^3}\right)^\top, \left(4, \frac{M}{2^4}\right)^\top \right\} \text{ and} \\ & C = \left\{ \left(5 \cdot 2^i, k - i\right)^\top : 0 \leq i \leq k-1 \right\} \end{aligned}$$

where $M := 5 \cdot 2^{k+5}$. An example of this set for $k = 9$ is depicted in Figure 4.2.

In the following, we compute a representative system with coverage error less than or equal to $\delta^C := \frac{M}{2^4}$.

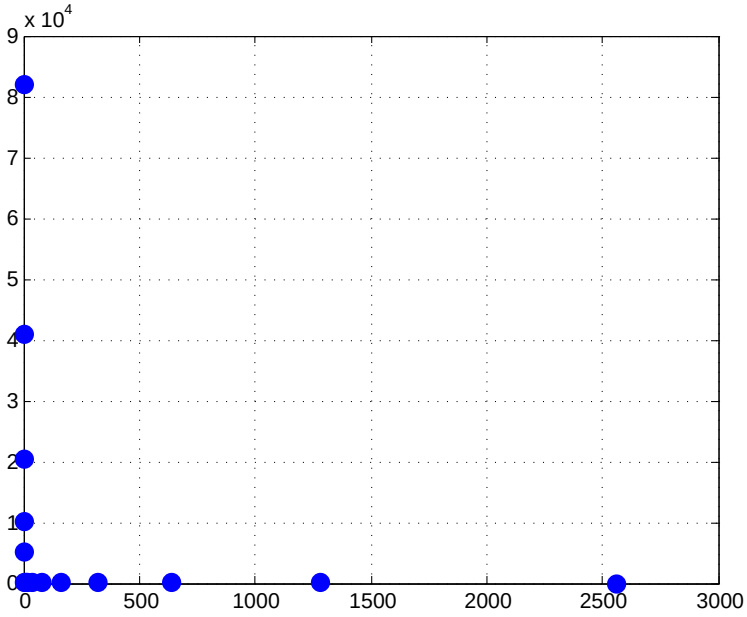


Figure 4.2: Y_N from Example 4.11. Note that the unit-distances of the two axes are not equal.

The two points from the set A are the outcomes of the two lexicographic minima of DBOP which span a rectangle with height $M = 5 \cdot 2^{k+5}$ and width $5 \cdot 2^k$. After the first iteration of the horizontal Box-Algorithm, we get a rectangle with right corner point $(5 \cdot 2^{k-1}, 1)^\top$ with height one less than the initial rectangle $M - 1$ and width $5 \cdot 2^{k-1}$ and, thus, with corner point distance $M - 1$.[2] Analogously, after k iterations, we get a rectangle with corner point distance $M - k > \frac{M}{2^4}$.

If we instead use an appropriately modified (vertical) Box-Algorithm which only uses the vertical subproblem (P_ε^{vert}), after four iterations, we get a relevant rectangle with height $\frac{M}{2^4}$ and width $5 \cdot 2^k - 4 < \frac{M}{2^4}$ and, thus, with corner point distance $\frac{M}{2^4}$. Hence, the desired accuracy is reached and the algorithm terminates and outputs the desired representative system.

To conclude, for an arbitrary parameter k, we can construct a problem instance for which the horizontal Box-Algorithm needs at least k iterations, and the vertical Box-Algorithm always terminates after four iterations. Moreover, in each iteration of the horizontal Box-Algorithm, the longest side of the currently considered rectangle is only reduced by 1.

[2] Here, w.l.o.g., we only look at the left rectangle since the right rectangle only defines a horizontal line and can be discarded since we always know the right corner point as a nondominated point.

Obviously, we can flip the instance such that the vertical Box-Algorithm needs at least k iterations, and the horizontal Box-Algorithm always terminates after 4 iterations. For this, consider the instance with the set

$$\begin{aligned} Y_N = A' \cup B' \cup C' \quad \text{with} \quad A' &= \left\{(0, 5 \cdot 2^k)^\top, (M, 0)^\top, \right\}, \\ B' &= \left\{ \left(\frac{M}{2^1}, 1\right)^\top, \left(\frac{M}{2^2}, 2\right)^\top, \left(\frac{M}{2^3}, 3\right)^\top, \left(\frac{M}{2^4}, 4\right)^\top \right\} \text{ and} \\ C' &= \left\{ \left(k - i, 5 \cdot 2^i\right)^\top : 0 \leq i \leq k-1 \right\}. \end{aligned}$$

Furthermore, we can combine both instances to one instance such that, during the algorithm, we indeed need to switch between the horizontal and the vertical subproblem to get the best performance. For this, we consider the alignment of the two rectangles induced by the corresponding outcomes of the lexicographic minima $y^1 \coloneqq (0, 5 \cdot 2^k)^\top$, $y^2 \coloneqq (M, 0)^\top$, $y^3 \coloneqq (0, M)^\top$ and $y^4 \coloneqq (5 \cdot 2^k, 0)^\top$ from A' and A, respectively. The rectangles $\mathrm{R}(y^1, y^2)$ and $\mathrm{R}(y^3, y^4)$ corresponding to the first and second instance, respectively, are translated in such a way that we get the new rectangles $\mathrm{R}(\hat{y}^1, \hat{y}^2)$ and $\mathrm{R}(\hat{y}^3, \hat{y}^4)$, respectively, with coordinates

$$\begin{aligned} \hat{y}^1 &\coloneqq (0, M + 2 + 5 \cdot 2^k)^\top, & \hat{y}^2 &\coloneqq (M, M+2)^\top, \\ \hat{y}^3 &\coloneqq (M+3, M+1)^\top \quad \text{and} & \hat{y}^4 &\coloneqq (M + 3 + 5 \cdot 2^k, 1)^\top. \end{aligned}$$

Additionally, to exactly obtain the translated rectangle $\mathrm{R}(\hat{y}^3, \hat{y}^4)$ in an iteration of the algorithm, we introduce the dummy point $\hat{y}^5 \coloneqq (2M + 5, 0)^\top$.

The constructed instance is sketched in Figure 4.3.

If the horizontal Box-Algorithm is applied to this constructed instance, in the first iteration, we choose $\varepsilon = \lfloor (2M+5)/2 \rfloor = M + 2$ and, thus, find point $\hat{y}^2$ which induces the two new rectangles $\mathrm{R}(\hat{y}^1, \hat{y}^2)$ and $\mathrm{R}(\hat{y}^3, \hat{y}^5)$.

Hence, after the first iteration, a translation of rectangle $\mathrm{R}(y^1, y^2)$ is added to the list, and it is only efficiently processed if we use the horizontal subproblem.

In a later iteration, we have to choose the rectangle $\mathrm{R}(\hat{y}^3, \hat{y}^5)$ with $\varepsilon = \lfloor (3M+8)/2 \rfloor = 3/2 \cdot M + 4$. The corresponding horizontal subproblem outputs the point $\hat{y}^4$, which induces the two new rectangles $\mathrm{R}(\hat{y}^3, \hat{y}^4)$ and $\mathrm{R}(\hat{y}^4 + (1, -1)^\top, \hat{y}^5)$.

Consequently, after considering rectangle $\mathrm{R}(\hat{y}^3, \hat{y}^5)$, a translation of rectangle $\mathrm{R}(y^3, y^4)$ is added to the list. This rectangle is only efficiently processed if we use the vertical subproblem.

To sum up, for the combined instance, appropriately switching the subproblem from horizontal to vertical and vice versa leads to a better performance. ◁

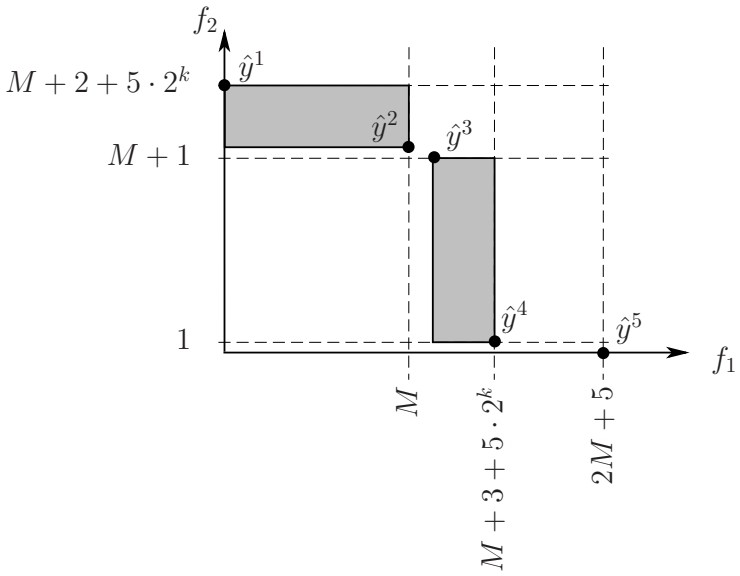


Figure 4.3: Constructed combined instance

The basic idea to overcome this issue in general is to divide each rectangle at the longest side, which leads to a bisection of the corner point distance every second time a rectangle is considered, i.e., after a "child" of this rectangle is considered (see proof of Theorem 4.17 below).

Remark 4.12:
This idea originates from the "Future Research"-section of Hamacher et al. (2007) and was also picked up, independently from this work, in a master's thesis (Ky, 2012). However, Ky (2012) considered another accuracy condition and does not consider a repair step such that the algorithm can output an incorrect representative system. Note that the algorithm from Ky (2012) will also be included in a forthcoming paper of Hamacher et al. (2015). Therein, a repair step (see below) will be added since we had pointed out the incorrect representative system to the authors which was also contained in a preliminary version of their forthcoming paper. Here, we introduce and analyze the combination of both subproblems with a more theoretically background as in Ky (2012) which is fundamental for the corresponding correctness property (see Theorem 4.16). Moreover, our analysis for an accuracy condition explicitly based on the coverage error leads to a tighter complexity bound as in Ky (2012). ◁

Following Hamacher et al. (2007), we consider a rectangle $\mathrm{R}(y^1, y^2)$ which should be refined by solving problem (P_ε^{vert}) with $\varepsilon := \lfloor (y_2^1 + y_2^2)/2 \rfloor$. Let $z^* \in Y_N$ denote the image of an optimal solution of (P_ε^{vert}).

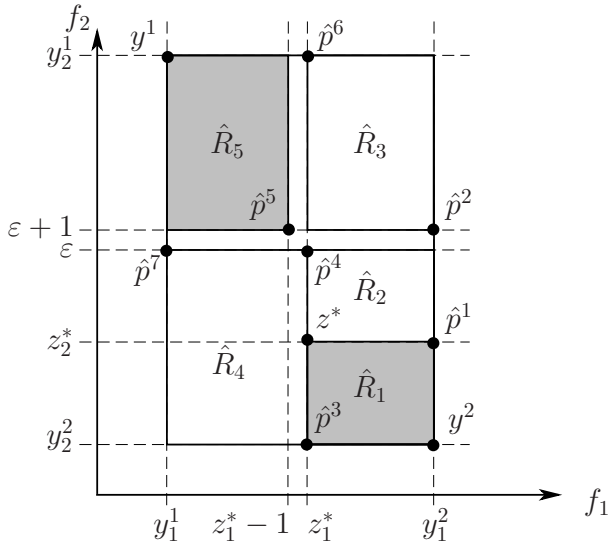


Figure 4.4: Vertical updating of a rectangle $\mathrm{R}(y^1, y^2)$

If $z^* \in \mathrm{R}(y^1, y^2)$, we divide the rectangle, similar to the horizontal case, into five new rectangles

$$\hat{R}_1 := \mathrm{R}(z^*, y^2), \quad \hat{R}_2 := \mathrm{R}(\hat{p}^4, \hat{p}^1), \quad \hat{R}_3 := \mathrm{R}(\hat{p}^6, \hat{p}^2),$$

$$\hat{R}_4 := \mathrm{R}(\hat{p}^7, \hat{p}^3), \quad \text{and} \quad \hat{R}_5 := \mathrm{R}(y^1, \hat{p}^5)$$

with

$$\hat{p}^1 := (y_1^2, z_2^*)^\top, \quad \hat{p}^2 := (y_1^2, \varepsilon+1)^\top, \quad \hat{p}^3 := (z_1^*, y_2^2)^\top, \quad \hat{p}^4 := (z_1^*, \varepsilon)^\top,$$

$$\hat{p}^5 := (z_1^*-1, \varepsilon+1)^\top, \quad \hat{p}^6 := (z_1^*, y_2^1)^\top, \quad \text{and} \quad \hat{p}^7 := (y_1^1, \varepsilon)^\top .$$

These five new rectangles are depicted in Figure 4.4.

Analogous to the horizontal case, due to the special structure of the vertical subproblem, we get the following result.

Corollary 4.13: *Let z^* be the image of an optimal solution of (P_ε^{vert}) corresponding to the rectangle $\mathrm{R}(y^1, y^2)$. Then, it holds $Y_N \cap \mathrm{R}(y^1, y^2) \subseteq \hat{R}_1 \cup \hat{R}_5$.*

In the horizontal Box-Algorithm, we got to each rectangle a nondominated point in the representative system *Rep* which corresponds to the right corner point of the

rectangle. If we appropriately substitute the horizontal subproblem with the newly introduced vertical subproblem, we get a corresponding property, i.e., for each rectangle, we get a nondominated point in *Rep* which corresponds to the left corner point of the rectangle.

However, for the combination of both subproblems, the horizontal and the vertical subproblem, we have to be careful since using different types of subproblems in different iterations can cause difficulties. More precisely, at termination, we could obtain rectangles without knowing a representative point lying inside these rectangles. This issue is sketched in Figure 4.5. There, the rectangle $\mathrm{R}(p^5, \hat{p}^5)$ does not include a point in *Rep* because both corner points, p^5 and $\hat{p}^5$, are induced by dummy points. In the worst case, this rectangle would not be considered anymore by the algorithm due to some termination condition and, thus, we would not get a representative system fulfilling the desired coverage error.

One possibility to overcome this problem is to invoke an additional "repair step" after each vertical division. For instance, the following problem can be solved after we have vertically divided a rectangle $\mathrm{R}(y^1, y^2)$ into the two rectangles $\mathrm{R}(y^1, \hat{p}^5)$ and $\mathrm{R}(z^*, y^2)$:

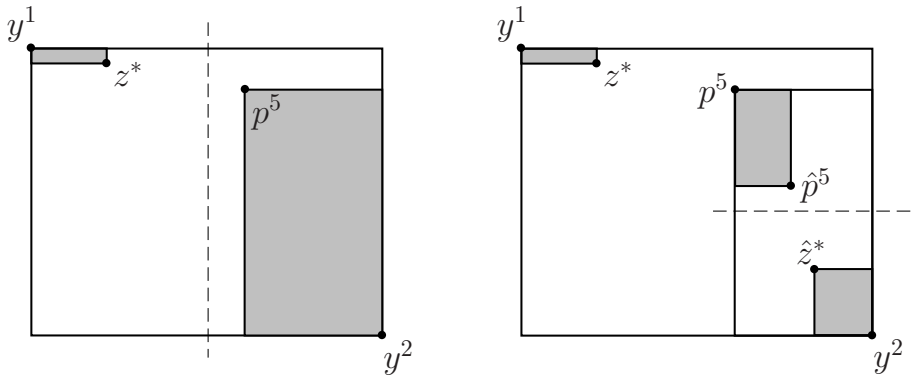
$$\begin{aligned} (P^{repair}_{\hat{p}^5}) \quad & \operatorname{lex\,min} \ (f_2(x), f_1(x)) \\ & \text{s.t.} \quad x \in X \\ & \qquad f_1(x) \leq \hat{p}^5_1 \end{aligned}$$

Similar considerations as above result in the following observation.

Observation 4.14:
Let $\tilde{z}$ be the outcome of an optimal solution of problem $(P^{repair}_{\hat{p}^5})$ for repairing some rectangle $\mathrm{R}(y^1, \hat{p}^5)$. Then, it holds:

- $\tilde{z} \in Y_N$
- $\tilde{z} \notin \mathrm{R}(y^1, \hat{p}^5) \implies \mathrm{R}(y^1, \hat{p}^5) \cap Y_N = \emptyset$
  (i.e., rectangle $\mathrm{R}(y^1, \hat{p}^5)$ can be discarded)
- $\tilde{z} \in \mathrm{R}(y^1, \hat{p}^5) \implies \mathrm{R}(y^1, \hat{p}^5) \cap Y_N = \mathrm{R}(y^1, \tilde{z}) \cap Y_N$
  (i.e., rectangle $\mathrm{R}(y^1, \hat{p}^5)$ can be substituted with rectangle $\mathrm{R}(y^1, \tilde{z})$) $\triangleleft$

A pseudocode description of the resulting *coverage-based Box-Algorithm* for two objectives using both subproblems to obtain a representative system with desired coverage error δ^C w.r.t. $\|\cdot\|_\infty$ is provided in Algorithm 4.1.



(a) Iteration 1 (b) Iteration 2

Figure 4.5: The algorithm divides the current rectangle at the longest side. In the first iteration, the division is due to the horizontal lexicographic ε-constraint problem and in the second iteration due to the vertical lexicographic ε-constraint problem.

Algorithm 4.1 Box-Algorithm (for two objectives, coverage)

Input: A discrete bicriteria optimization problem, $\delta^C > 0$.
Output: A representative system $Rep \subseteq Y_N$ with coverage error at most δ^C (cf. Theorem 4.17).

$S \leftarrow \emptyset$, $\mathcal{B} \leftarrow \emptyset$
Compute the lexicographic outcomes z^1 in normal and z^2 in reversed order.
$Rep \leftarrow \{z^1, z^2\}$
CheckAccuracy($\mathrm{R}(z^1, z^2)$) *// see Algorithm 4.2*
while $S \neq \emptyset$ **do** *// termination condition*
    Choose the rectangle $\mathrm{R}(y^1, y^2) \in S$ with maximal distance $\|y^1 - y^2\|_\infty$.
    $S \leftarrow S \setminus \{\mathrm{R}(y^1, y^2)\}$
    if $\|y^1 - y^2\|_\infty = y_1^2 - y_1^1$ **then** *// call horizontal subproblem*
        Solve (P_ε^{hor}) with $\varepsilon := \lfloor (y_1^1 + y_1^2)/2 \rfloor$ and obtain optimal outcome $z^* \in Y_N$.
        if $z^* \in \mathrm{R}(y^1, y^2)$ **then**
            $Rep \leftarrow Rep \cup \{z^*\}$
            CheckAccuracy($\mathrm{R}(y^1, z^*)$)
            CheckAccuracy($\mathrm{R}(p^5, y^2)$)
        else *// $z^* \notin \mathrm{R}(y^1, y^2)$*
            CheckAccuracy($\mathrm{R}((\varepsilon + 1, y_2^1)^\top, y^2)$)

Algorithm 4.1 Box-Algorithm (for two objectives, coverage) – continued

16: **else** // $\|y^1 - y^2\|_\infty = y_2^1 - y_2^2 \Longrightarrow$ *call vertical subproblem*
17: Solve (P_ε^{vert}) with $\varepsilon := \lfloor (y_2^1 + y_2^2)/2 \rfloor$ and obtain optimal outcome $z^* \in Y_N$.
18: **if** $z^* \neq y^2$ **then**
19: $Rep \leftarrow Rep \cup \{z^*\}$
20: CheckAccuracy($\mathrm{R}(z^*, y^2)$)
21: Solve $(P_{\hat{p}^5}^{repair})$ and obtain optimal outcome $\tilde{z} \in Y_N$.
22: **if** $\tilde{z} \in \mathrm{R}(y^1, \hat{p}^5)$ **then**
23: $Rep \leftarrow Rep \cup \{\tilde{z}\}$
24: CheckAccuracy($\mathrm{R}(y^1, \tilde{z})$)
25: **return** $(Rep, \mathcal{B})$

Algorithm 4.2 CheckAccuracy

Input: A rectangle $\mathrm{R}(a, b)$.
1: **if** $\|a - b\|_\infty \leq \delta^C$ **then** // *accuracy condition*
2: $\mathcal{B} \leftarrow \mathcal{B} \cup \{\mathrm{R}(a, b)\}$
3: **else**
4: $S \leftarrow S \cup \{\mathrm{R}(a, b)\}$

As in Theorem 4.8, we prove a corresponding *correctness property* in the modified algorithm.

Lemma 4.15 (Distinct-Rectangle-Property): *Let $\mathcal{B}^i$ denote the collection of unexplored and finished rectangles in iteration $i \geq 1$ of the coverage-based Box-Algorithm (Algorithm 4.1), before the current rectangle is substituted. In iteration i, for each pair of distinct rectangles $\mathrm{R}(v^1, v^2)$, $\mathrm{R}(w^1, w^2) \in \mathcal{B}^i$, exactly one of the following statements holds:*

(i) $v_1^2 < w_1^1$ *and* $v_2^2 > w_2^1$

(ii) $w_1^2 < v_1^1$ *and* $w_2^2 > v_2^1$

This property is called distinct-rectangle-property.

Proof:
Essentially, we can use the same proof as in Lemma 4.7, but we have to distinguish between the two cases in which the horizontal and vertical subproblem is called, respectively. For both cases, the way of argumentation is analogous to the proof of Lemma 4.7 with a slight modification if the vertical subproblem is called. Suppose we are in this case and have solved the vertical subproblem corresponding to rectangle $\mathrm{R}(y^1, y^2)$ with image z^* of an optimal solution. Instead of distinguishing between the

two cases $z^* \in \mathrm{R}(y^1, y^2)$ and $z^* \notin \mathrm{R}(y^1, y^2)$, we have to distinguish between the two cases $z^* \neq y^2$ and $z^* = y^2$. With these modifications the proof works analogously to the proof of Lemma 4.7. □

Theorem 4.16 (Correctness Property): *In an arbitrary iteration in the coverage-based Box-Algorithm (Algorithm 4.1), let $z^* \in Y_N$ be the image of an optimal solution corresponding to the refinement of some rectangle $\mathrm{R}(y^1, y^2)$. Then, we have:*

(i) If the horizontal subproblem was called:

$$z^* \notin \mathrm{R}(y^1, y^3) \Longleftrightarrow \mathrm{R}(y^1, y^3) \cap Y_N = \emptyset$$

where $y^3 = (\varepsilon, y_2^2)^\top$.

(ii) If the vertical subproblem was called:

$$z^* = y^2 \Longleftrightarrow \mathrm{R}(y^4, y^2) \cap Y_N = \{y^2\}$$

where $y^4 = (y_1^1, \varepsilon)^\top$.
In particular, if $z^* \neq y^2$, we get $z^* \in \mathrm{R}(y^4, y^2) \cap Y_N$.

Note that the lexicographic minimum in normal order and the lexicographic minimum in reversed order are always feasible for any upcoming horizontal and vertical subproblem, respectively.

Proof:
Let $\mathcal{B}^i$ be defined as in Lemma 4.15. Consider an arbitrary iteration $i \geq 1$ corresponding to the refinement of rectangle $\mathrm{R}(y^1, y^2) \in \mathcal{B}^i$. Obviously, due to the construction of the rectangles, we get $y^2 \in Y_N$.

From Lemma 4.15, it holds the distinct-rectangle-property for iteration i.

If the horizontal subproblem was called, we can use the same proof as in Theorem 4.8. Suppose the vertical subproblem was called with z^* being the image of a corresponding optimal solution.

If $z^* = y^2$, we immediately get $\mathrm{R}(y^4, y^2) \cap Y_N = \{y^2\}$, due to the lexicographic structure of the subproblem.

If $z^* \neq y^2$, then z^* has to fulfill $z_1^* < y_1^2$ since otherwise, y^2 would be better than z^* w.r.t. the corresponding vertical subproblem. Suppose z^* is not in $\mathrm{R}(y^4, y^2)$, then z^* has to be in another rectangle $\mathrm{R}(s^1, s^2) \in \mathcal{B}^i$ with $s_1^1 \leq z_1^* < y_1^2$. Due to the distinct-rectangle-property, we get $z_2^* \geq s_2^2 > y_2^1 \geq \varepsilon$ which contradicts feasibility of z^* for the vertical subproblem. Hence, we get $y^2 \neq z^* \in \mathrm{R}(y^4, y^2) \cap Y_N$. □

Theorem 4.17: *Algorithm 4.1 terminates in finitely many steps. It outputs a collection of rectangles $\mathcal{B}$ containing all nondominated points. At termination, the representative system Rep has a coverage error of at most δ^C (w.r.t. $\|\cdot\|_\infty$). More precisely,*

the algorithm performs at most $\mathcal{O}\left(\left(\frac{L}{\delta^C}\right)^2\right)$ *many iterations, where* L *equals the corner point distance* $\|z^1 - z^2\|_\infty$ *of the initial rectangle* $\mathrm{R}(z^1, z^2)$.

Proof:
Due to the construction of the rectangles, a subdivision $\mathcal{B} \cup S$ is obtained which contains Y_N at each state of the algorithm (cf. Lemma 4.4, Corollary 4.13 and Theorem 4.16). Moreover, in each iteration, for each rectangle $\mathrm{R}(y^1, y^2) \in \mathcal{B} \cup S$, we know $y^2 \in Y_N \cap Rep$. The termination condition (line 5) together with the accuracy condition (line 1 in Algorithm 4.2) of the algorithm guarantee that the coverage error of the representative system is at most δ^C if the algorithm terminates. Thus, it suffices to show the termination of the algorithm.

The functioning of the algorithm can be represented by a binary tree for which each node corresponds to a rectangle in some iteration. More precisely, the root node of the tree corresponds to the initial rectangle and after the first iteration, we substitute this rectangle by at most two new rectangles which are then represented by at most two child nodes of the father node. Finiteness of the algorithm corresponds to a bounded height of the corresponding binary tree which is shown in the following.

Claim: The height of the corresponding binary tree is bounded by $2 \cdot \lceil \log_2(L/\delta^C) \rceil + 1$.

Proof:
Let $\mathrm{R}(y^1, y^2)$ be a rectangle corresponding to some node of the binary tree in an arbitrary but odd level greater than or equal to 3.[3] Let $\mathrm{R}(w^1, w^2)$ and $\mathrm{R}(v^1, v^2)$ be the rectangles corresponding to its father and grandfather nodes, respectively. Consider now the iteration in which the rectangle corresponding to the grandfather node is processed. Without loss of generality, let the horizontal subproblem be called (i.e., $\|v^1 - v^2\|_\infty = v_1^2 - v_1^1$) resulting in rectangles including the rectangle $\mathrm{R}(w^1, w^2)$. Due to the special ε-constraint in the horizontal subproblem and the definition of the rectangles, it holds

$$w_1^2 - w_1^1 \leq \frac{\|v^1 - v^2\|_\infty}{2} < \|v^1 - v^2\|_\infty \tag{4.1}$$

and also $w_2^1 - w_2^2 \leq \|v^1 - v^2\|_\infty$.

Claim: $\|y^1 - y^2\|_\infty \leq \frac{\|v^1 - v^2\|_\infty}{2}$

Proof:
Case 1: $w_2^1 - w_2^2 \leq \frac{\|v^1 - v^2\|_\infty}{2} \implies \|y^1 - y^2\|_\infty \leq \|w^1 - w^2\|_\infty \leq \frac{\|v^1 - v^2\|_\infty}{2}$

Case 2: $w_2^1 - w_2^2 > \frac{\|v^1 - v^2\|_\infty}{2} \implies \frac{\|v^1 - v^2\|_\infty}{2} < w_2^1 - w_2^2 \leq \|v^1 - v^2\|_\infty \overset{(4.1)}{\implies}$ The vertical subproblem was called when the rectangle $\mathrm{R}(w^1, w^2)$ was considered. Thus, due to the special ε-constraint in the vertical subproblem, we get $\|y^1 - y^2\|_\infty \leq \frac{\|v^1 - v^2\|_\infty}{2}$. $\diamondsuit$

[3] The level of the root node is 1, its children are on level 2 and so on.

By induction, we get that an arbitrary rectangle $\mathrm{R}(y^1, y^2)$ corresponding to a node in level $2k-1$ ($k \in \mathbb{N}$) fulfills

$$\|y^1 - y^2\|_\infty \leq \frac{L}{2^{k-1}} \,.$$

Hence, if $k \geq \log_2(L/\delta^C) + 1$, we get $\|y^1 - y^2\|_\infty \leq \delta^C$ and, therefore, each rectangle at level $2 \cdot \lceil \log_2(L/\delta^C) \rceil + 1$ fulfills the desired coverage error and is not subdivided anymore, which shows boundedness of the height of the tree. ♢

Moreover, the tree can have at most

$$\sum_{i=0}^{2\cdot\lceil \log_2(L/\delta^C)\rceil} 2^i = 2^{2\cdot\lceil \log_2(L/\delta^C)\rceil+1} - 1 \leq 2^{2\cdot\log_2(L/\delta^C)+3} - 1 = 8 \cdot \left(\frac{L}{\delta^C}\right)^2 - 1$$

nodes and, correspondingly, the number of iterations of the algorithm is bounded by the same term. □

Remark 4.18:
Note that also Ky (2012) proved in his master's thesis, independently from this work, the termination of the Box-Algorithm combining the horizontal and vertical subproblem. Since he used the perimeter as accuracy condition for termination, he got a complexity bound of

$$\mathcal{O}\left(\left(\frac{P}{4 \cdot \delta^C}\right)^{\log_{4/3}(2)}\right) = \mathcal{O}\left(\left(\frac{4 \cdot L}{4 \cdot \delta^C}\right)^{\log_{4/3}(2)}\right)$$

where P is the initial perimeter. Hence, for the coverage error, the complexity bound from Theorem 4.17 is tighter than the one from Ky (2012) since $\log_{4/3}(2) > 2.4 > 2$. ◁

If we desire a representative system with coverage error w.r.t. an arbitrary p-norm $\|\cdot\|_p$, $1 \leq p \leq \infty$, we can still use Algorithm 4.1 to obtain such a representative system, we only have to choose δ^C sufficiently small. A rule how δ^C can be chosen is induced by the following (well-known) lemma.

Lemma 4.19: *Let $1 \leq p < \infty$ and $n \in \mathbb{N}$. Then, it holds:*

$$\|x\|_p \leq n^{1/p} \cdot \|x\|_\infty \qquad \textit{for all } x \in \mathbb{R}^n$$

In particular, if $\|x\|_\infty \leq \delta^C$, then $\|x\|_p \leq n^{1/p} \cdot \delta^C$.

Proof:
Let $i_0 \coloneqq \underset{i=1,\ldots,n}{\arg\max} |x_i|$, then we have

$$\|x\|_p = \left(\sum_{i=1}^{n} \underbrace{|x_i|^p}_{\leq |x_{i_0}|^p}\right)^{1/p} \overset{\text{mon. incr.}}{\leq} (n \cdot |x_{i_0}|^p)^{1/p} = n^{1/p} \cdot |x_{i_0}| = n^{1/p} \cdot \|x\|_\infty \,. \qquad \square$$

4.2.2 Adding the Uniformity Property

In this subsection, the uniformity measure is incorporated into the horizontal Box-Algorithm, i.e., the points in *Rep* should not build clusters and fulfill a given minimal distance between each other (cf. Definition 2.9).

Hamacher et al. (2007) noted that after the termination of the horizontal Box-Algorithm, a filtering technique can be applied to reduce the cluster density. However, a filtering can destroy the guarantees from the algorithm (see Figure 4.6). Hence, we incorporate the uniformity during the running time of the horizontal Box-Algorithm. The aim is a representative system *Rep* having, besides the area-based Δ-accuracy, a uniformity greater than or equal to some predefined uniformity-tolerance $\delta^U > 0$ (w.r.t. $\|\cdot\|_\infty$). We assume that $\delta^U \cdot \delta^U \leq \Delta$ since we do not want to discard rectangles with area greater than Δ.

In this section, we assume the additional constraint $f_2(x) \geq y_2^2$ in each upcoming horizontal subproblem (P_ε^{hor}) for a current rectangle $R(y^1, y^2)$.

During the algorithm, after we have solved (P_ε^{hor}) for the current rectangle $\mathrm{R}(y^1, y^2)$ with image $z^* \in Y$ of an optimal solution, we look at the following questions:

- When do we add z^* to the representative system *Rep* (be careful: not adding z^* to the set *Rep* can destroy the accuracy of the algorithm)?
- When do we add the two newly constructed rectangles $\mathrm{R}(y^1, z^*)$ and $\mathrm{R}(p^5, y^2)$ with $p^5 = (\varepsilon + 1, z_2^* - 1)^\top$ to the list S of unexplored rectangles?

For each corner point y^1 and y^2 of every upcoming rectangle $\mathrm{R}(y^1, y^2)$, we have to save additional parameters $\delta^{y^1}, \delta^{y^2} \in \mathbb{R}^2_{\geqq}$, respectively, telling us how the "discarding area" looks like.

Definition 4.20:
For some $y \in \mathbb{R}^2$ and $\delta^y \in \mathbb{R}^2_{\geqq}$, we denote the neighborhood

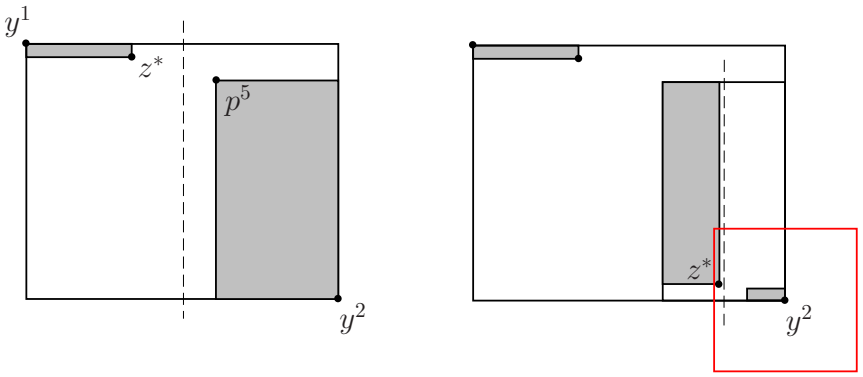
$$B_{\delta^y}(y) := \left\{ \hat{y} \in \mathbb{R}^2 : |\hat{y}_1 - y_1| < \delta_1^y,\ |\hat{y}_2 - y_2| < \delta_2^y \right\}$$

as the *δ-rectangle of y*. Then, for a rectangle $\mathrm{R}(y^1, y^2)$ with given δ^{y^1} and δ^{y^2}, the *discarding area* of the rectangle is defined as

$$\left(B_{\delta^{y^1}}(y^1) \cup B_{\delta^{y^2}}(y^2)\right) \cap \mathrm{R}(y^1, y^2)\ . \qquad \diamond$$

If no discarding area is known for some corner point, the corresponding parameter is set to 0. For the initial rectangle $\mathrm{R}(z^1, z^2)$, we set $\delta^{z^1} := \delta^{z^2} := (\delta^U, \delta^U)^\top$.

The pseudocode description of the new algorithm is provided in Algorithm 4.3.



(a) Iteration 1 (b) Iteration 2 (and Termination)

Figure 4.6: Suppose the horizontal Box-Algorithm terminates after the second iteration. If we now apply to *Rep* some filtering procedure forcing a uniformity greater than or equal to some δ^U, which is indicated for y^2 with the red rectangle, we have to delete either y^2 or z^* (from the second iteration). But after this deletion, we lose the area-based Δ-accuracy since we do not have a representative point of at least one rectangle from the second iteration.

Algorithm 4.3 Box-Algorithm (for two objectives, area and uniformity)

Input: A discrete bicriteria optimization problem, $\Delta, \delta^U > 0$ with $\delta^U \cdot \delta^U \leq \Delta$.
Output: A representative system $Rep \subseteq Y_N$ with area-based Δ-accuracy and uniformity at least δ^U (cf. Corollary 4.25).

$S \leftarrow \emptyset$, $\mathcal{B} \leftarrow \emptyset$
Compute the lexicographic outcomes z^1 in normal and z^2 in reversed order.
$\delta^{z^1} \leftarrow \delta^{z^2} \leftarrow (\delta^U, \delta^U)^\top$
$Rep \leftarrow \{z^1, z^2\}$
CheckAccuracy($\mathrm{R}(z^1, z^2)$,δ^{z^1},δ^{z^2}) // *see Algorithm 4.4*
while $S \neq \emptyset$ **do**
    Choose $\left(\mathrm{R}(y^1, y^2), \delta^{y^1}, \delta^{y^2}\right) \in S$ with maximal area $\mathrm{Ar}(\mathrm{R}(y^1, y^2))$.
    $S \leftarrow S \setminus \left\{\left(\mathrm{R}(y^1, y^2), \delta^{y^1}, \delta^{y^2}\right)\right\}$
    Solve (P_ε^{hor}) with $\varepsilon := \lfloor (y_1^1 + y_1^2)/2 \rfloor$ and obtain optimal outcome $z^* \in Y$.
    if $z^* \in \mathrm{R}(y^1, y^2)$ **then**
        if $z^* \in B_{\delta^{y^1}}(y^1)$ **then** // *Case 1:* $z^* \in B_{\delta^{y^1}}(y^1)$
            if $p^5 \in B_{\delta^{y^1}}(y^1)$ **then**
                $\delta^{p^5} \leftarrow \left(\delta_1^{y^1} - (p_1^5 - y_1^1),\ \delta_2^{y^1} - (y_2^1 - p_2^5)\right)^\top$
            else
                $\delta^{p^5} \leftarrow 0$
            CheckAccuracy($\mathrm{R}(p^5, y^2)$,δ^{p^5},δ^{y^2})

Algorithm 4.3 Box-Algorithm (for two objectives, area and uniformity) – continued

else if $z^* \in B_{\delta^{y^2}}(y^2)$ **then** // *Case 2:* $z^* \in B_{\delta^{y^2}}(y^2) \setminus B_{\delta^{y^1}}(y^1)$
  $\delta^{z^*} \leftarrow \left(\delta_1^{y^2} - (y_1^2 - z_1^*),\ \delta_2^{y^2} - (z_2^* - y_2^2)\right)^\top$
  Solve the subproblem with disjunctive constraints (4.2) and obtain $\tilde{z} \in Y$.
  if $\tilde{z} \notin B_{\delta^{y^1}}(y^1)$ **and** $\tilde{z} \in \mathrm{R}(y^1, z^*)$ **then**
    $Rep \leftarrow Rep \cup \{\tilde{z}\}$
    $\delta^{\tilde{z}} \leftarrow (\delta^U, \delta^U)^\top$
    CheckAccuracy($\mathrm{R}(y^1, \tilde{z})$,$\delta^{y^1}$,$\delta^{\tilde{z}}$)
else // *Case 3:* $z^* \notin B_{\delta^{y^1}}(y^1) \cup B_{\delta^{y^2}}(y^2)$
  $Rep \leftarrow Rep \cup \{z^*\}$
  $\delta^{z^*} \leftarrow (\delta^U, \delta^U)^\top$
  CheckAccuracy($\mathrm{R}(y^1, z^*)$,δ^{y^1},δ^{z^*})
  if $p^5 \in B_{\delta^{z^*}}(z^*)$ **then**
    $\delta^{p^5} \leftarrow \left(\delta_1^{z^*} - (p_1^5 - z_1^*),\ \delta_2^{z^*} - (z_2^* - p_2^5)\right)^\top$
  else
    $\delta^{p^5} \leftarrow 0$
  CheckAccuracy($\mathrm{R}(p^5, y^2)$,δ^{p^5},δ^{y^2})
else // $z^* \notin \mathrm{R}(y^1, y^2)$
  if $p^2 \coloneqq (\varepsilon + 1, y_2^1)^\top \in B_{\delta^{y^1}}(y^1)$ **then**
    $\delta^{p^2} \leftarrow \left(\delta_1^{y^1} - (p_1^2 - y_1^1),\ \delta_2^{y^1}\right)^\top$
  else
    $\delta^{p^2} \leftarrow 0$
  CheckAccuracy($\mathrm{R}(p^2, y^2)$,δ^{p^2},δ^{y^2})
return $(Rep, \mathcal{B})$

Algorithm 4.4 CheckAccuracy

Input: A rectangle $\mathrm{R}(a, b)$ with δ^a, δ^b.
if $\mathrm{Ar}(\mathrm{R}(a, b)) \leq \Delta$ **then**
  $\mathcal{B} \leftarrow \mathcal{B} \cup \left\{\left(\mathrm{R}(a, b), \delta^a, \delta^b\right)\right\}$
else
  $S \leftarrow S \cup \left\{\left(\mathrm{R}(a, b), \delta^a, \delta^b\right)\right\}$

The algorithm works similar to the original horizontal Box-Algorithm, in particular, the upcoming lexicographic subproblems are solved in the same way as in the original algorithm. Due to the initial assumption $\delta^U \cdot \delta^U \leq \Delta$, we immediately get the following result.

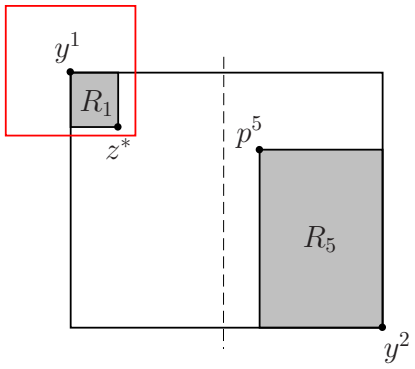


Figure 4.7: This figure shows what can happen in the first case.

Lemma 4.21: *For Algorithm 4.3, let $\delta^U, \Delta > 0$ be given with $\delta^U \cdot \delta^U \leq \Delta$. In an arbitrary iteration corresponding to the refinement of some $\left(\mathrm{R}(y^1, y^2), \delta^{y^1}, \delta^{y^2}\right)$ with image z^* of an optimal solution for the corresponding horizontal subproblem, it holds*

$$\mathrm{Ar}(\mathrm{R}(p^5, y^2)) > \Delta \Longrightarrow p^5 \notin B_{\delta^{y^2}}(y^2)$$

where $p^5 = (\varepsilon + 1, z_2^* - 1)^\top$.

Proof:
Obviously, during the algorithm, there does not occur any corner point y possessing a δ-rectangle $B_{\delta^y}(y)$ with δ_i^y greater than δ^U for some $i \in \{1, 2\}$. □

In the following, we present and analyze the different cases which can occur during the algorithm. We consider an arbitrary iteration corresponding to the refinement of some rectangle $\left(\mathrm{R}(y^1, y^2), \delta^{y^1}, \delta^{y^2}\right)$. Let S be the list of yet unexplored rectangles and z^* be the image of an optimal solution for the corresponding horizontal subproblem. As in Hamacher et al. (2007), we denote with R_1 and R_5 the two new rectangles $\mathrm{R}(y^1, z^*)$ and $\mathrm{R}(p^5, y^2)$ with $p^5 = (\varepsilon + 1, z_2^* - 1)^\top$, respectively. In the first and third case, we assume $\mathrm{Ar}(R_5) > \Delta$ since otherwise, we do not need to consider this rectangle anymore.

Case 1: z^* is in the δ-rectangle of y^1 (see Figure 4.7)

If z^* is contained in the δ-rectangle of y^1, we discard z^* and R_1 since the latter is then completely contained in the δ-rectangle of y^1.

If p^5 is not contained in the δ-rectangle of y^1, we add R_5 to the list S of unexplored rectangles with $\delta^{p_5} := 0$.

If p^5 is inside the δ-rectangle of y^1, we add R_5 to the list S and store

$$\delta^{p^5} := \left(\delta_1^{y^1} - (p_1^5 - y_1^1), \delta_2^{y^1} - (y_2^1 - p_2^5)\right)^\top.$$

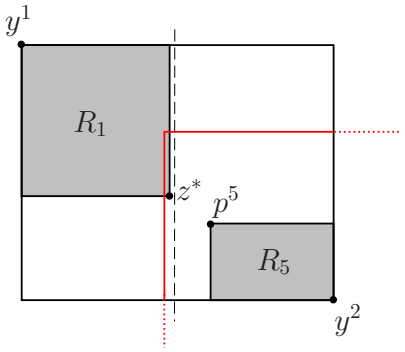


Figure 4.8: This figure shows what can happen in the second case.

Case 2: z^* is only in the δ-rectangle of y^2 (see Figure 4.8)

If z^* is only contained in the δ-rectangle of y^2, we can immediately imply that R_5 is completely contained in the δ-rectangle of y^2. Therefore, R_5 can be discarded.

We first store (temporarily)

$$\delta^{z^*} := \left(\delta_1^{y^2} - (y_1^2 - z_1^*),\ \delta_2^{y^2} - (z_2^* - y_2^2)\right)^\top,$$

but we do not add z^* to *Rep*. Here, we have to add an additional "repair step" since we want to delete z^* but it is connected to a relevant unexplored rectangle. We add disjunctive constraints to our model (cf. Nemhauser and Wolsey, 1999), i.e., we have to introduce two new binary variables, d_1 and d_2, and add the following constraints to problem (P_ε^{hor}) with $\varepsilon := z_1^*$:

$$f_1(x) \leq z_1^* - \delta_1^{z^*} + \delta_1^{z^*} \cdot (1 - d_1) \tag{4.2a}$$
$$f_2(x) \geq z_2^* + \delta_2^{z^*} - \delta_2^{z^*} \cdot (1 - d_2) \tag{4.2b}$$
$$d_1 + d_2 \geq 1 \tag{4.2c}$$
$$d_1,\ d_2 \in \{0, 1\} \tag{4.2d}$$

These constraints allow us to exclude the δ-rectangle of z^* intersected with R_1. Note that if $d_1 = 1$ ($d_2 = 1$), then x has to fulfill $f_1(x) \leq z_1^* - \delta_1^{z^*}$ ($f_2(x) \geq z_2^* + \delta_2^{z^*}$) and if $d_1 = 0$ ($d_2 = 0$), we get the redundant constraint $f_1(x) \leq z_1^*$ ($f_2(x) \geq z_2^*$).

The image $\tilde{z}$ of an optimal solution of this extended subproblem replaces z^* and serves as right corner point for the new unexplored rectangle $\mathrm{R}(y^1, \tilde{z})$. If no point is found in the current rectangle or $\tilde{z}$ is in the δ-rectangle of y^1, we discard the rectangle since, in this case, it does not contain any nondominated point of interest; otherwise, we add the new rectangle to the list S (or $\mathcal{B}$ if the desired accuracy is reached) and $\tilde{z}$ to *Rep* with $\delta^{\tilde{z}} := (\delta^U, \delta^U)^\top$.

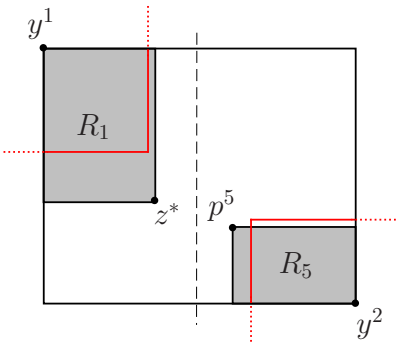


Figure 4.9: This figure shows what can happen in the third case.

Observation 4.22:
Note that in this case, due to the introduced disjunctive constraints, $\tilde{z}$ does not have to be nondominated since it could be dominated by a point in $B_{\delta^{z^*}}(z^*) \cap \mathrm{R}(y^1, z^*)$. However, it serves as a representative point for this current rectangle and it obviously holds $Y_N \cap \mathrm{R}(y^1, \tilde{z}) = Y_N \cap \left(\mathrm{R}(y^1, z^*) \setminus B_{\delta^{z^*}}(z^*)\right)$. ◁

Case 3: z^* is in no δ-rectangle (see Figure 4.9)

In this case, we add z^* to *Rep* and, if the desired accuracy is not yet reached, we add R_1 to the list S with $\delta^{z^*} := (\delta^U, \delta^U)^\top$. For R_5, we only have to check the δ-rectangle of z^* since the δ-rectangle of y^1 is to the left or above z^* and, hence, does not affect rectangle R_5.

If p^5 is not contained in the δ-rectangle of z^*, we add R_5 to the list S with $\delta^{p^5} := 0$; otherwise, we add R_5 to the list S and store

$$\delta^{p^5} := \left(\delta_1^{z^*} - (p_1^5 - z_1^*), \delta_2^{z^*} - (z_2^* - p_2^5)\right)^\top.$$

For proving finiteness and correctness of Algorithm 4.3, we need the following two results.

Lemma 4.23 (Distinct-rectangle-property): *Let $\mathcal{B}^i$ denote the collection of unexplored and finished rectangles in iteration $i \geq 1$ of Algorithm 4.3, before the current rectangle is substituted. In iteration i, for each pair of distinct rectangles $\mathrm{R}(v^1, v^2)$ and $\mathrm{R}(w^1, w^2)$ from $\mathcal{B}^i$, exactly one of the following statements holds:*

(i) $v_1^2 < w_1^1$ *and* $v_2^2 > w_2^1$

(ii) $w_1^2 < v_1^1$ *and* $w_2^2 > v_2^1$

This property is called distinct-rectangle-property.

Proof:
Since the construction of the rectangles is analogous to the horizontal Box-Algorithm, the same way of argumentation as in the proof for Lemma 4.7 can be applied leading to the desired result. □

Lemma 4.24: *For Algorithm 4.3, let* $\delta^U, \Delta > 0$ *be given with* $\delta^U \cdot \delta^U \leq \Delta$. *In an arbitrary iteration* $i \geq 1$, *let* Rep^i *and* $\mathcal{B}^i$ *denote the collection of representative points and the collection of unexplored and finished rectangles in this iteration, respectively, before the current subproblem is processed. Then, for all* $(\mathrm{R}(v,w), \delta^v, \delta^w) \in \mathcal{B}^i$ *with* $\mathrm{Ar}(\mathrm{R}(v,w)) > \Delta$, *it holds*

$$(B_{\delta^v}(v) \cup B_{\delta^w}(w)) \cap \mathrm{R}(v,w) = \bigcup_{z \in Rep^i} B_\delta(z) \cap \mathrm{R}(v,w) \quad \text{with } \delta \coloneqq (\delta^U, \delta^U)^\top .$$

Proof:
Firstly, due to the distinct-rectangle-property, we know that in each iteration i for all $\mathrm{R}(v,w) \in \mathcal{B}^i$, there does not exist a $z \in Rep^i$ with $B_\delta(z) \cap \mathrm{R}(v,w) \neq \emptyset$ and $B_\delta(z) \cap \{v,w\} = \emptyset$.

The claim is shown by induction on the number of iterations i.

In the first iteration, the claim is trivially fulfilled.

Suppose we are after iteration i and the claim holds for all rectangles in $\mathcal{B}^i$. Let iteration i correspond to the refinement of rectangle $\mathrm{R}(y^1, y^2) \in \mathcal{B}^i$ and let z^* denote the image of an optimal solution of the corresponding horizontal subproblem, if necessary, after the corresponding disjunctive constraints were added.

If z^* and the corresponding rectangle to the left of the ε-constraint are discarded, $Rep^{i+1} = Rep^i$ and the δ-rectangles of the corner points of the new rectangle are obviously constructed in such a way that they fulfill the claim.

If z^* is not discarded, we know $z^* \notin B_{\delta^{y^1}}(y^1) \cup B_{\delta^{y^2}}(y^2)$ and $Rep^{i+1} = Rep^i \cup \{z^*\}$. Then, due to the distinct-rectangle-property, $B_\delta(z^*)$ cannot affect a corner point of any rectangle $R \in \mathcal{B}^i \setminus \{\mathrm{R}(y^1, y^2)\}$. Moreover, since $y^2 \in Rep^i$ and $z^* \notin B_{\delta^{y^2}}(y^2)$, $B_\delta(z^*)$ cannot affect y^2. If $y^1 \in B_\delta(z^*)$, then we can immediately imply $\mathrm{Ar}(\mathrm{R}(y^1, z^*)) \leq \Delta$. If the rectangle $\mathrm{R}(p^5, y^2)$ to the right of the ε-constraint was not discarded and $\mathrm{Ar}(\mathrm{R}(p^5, y^2) > \Delta$ with $p^5 = (\varepsilon + 1, z_2^* - 1)^\top$, then the δ-rectangles of the corner points p^5 and y^2 are constructed in such a way that they fulfill the claim. □

During the algorithm, we do not destroy the accuracy Δ since we have made the assumption $\delta^U \cdot \delta^U \leq \Delta$ and, hence, every discarded point/area lies in a rectangle with area less than or equal to Δ containing some element of *Rep*.

Corollary 4.25: *Algorithm 4.3 terminates after finitely many steps. At termination, the representative system Rep has a uniformity of at least* δ^U *and, together with the collection of rectangles* $\mathcal{B} \cup \bigcup_{z \in Rep} \overline{B_{(\delta^U, \delta^U)^\top}(z)}$, *fulfills the area-based* Δ*-accuracy.*

Proof:
Termination of the algorithm and the fulfillment of the area-based Δ-accuracy as well as the uniformity property follow directly due to termination of the horizontal Box-Algorithm (cf. Theorem 4.6) and the result from Lemma 4.24 together with the applied modifications compared to the horizontal Box-Algorithm. □

We conclude this subsection with two remarks about the incorporation of the uniformity.

Remark 4.26:
In the following, the idea how both properties, the coverage error as well as the uniformity, can be incorporated into the Box-Algorithm is shortly outlined. We aim to combine the two algorithms/ideas to get a representative system with coverage error at most δ^C and uniformity at least δ^U with $0 < \delta^U \leq \delta^C$. The latter condition guarantees that we do not discard complete rectangles with corner point distance greater than δ^C.

Since the above considerations and results are in an analogous way valid after the vertical subproblem was called, we immediately get the desired algorithm by combining the coverage-based Box-Algorithm (Algorithm 4.1) with analogous ideas as discussed above to fulfill the desired uniformity. However, after the vertical subproblem is called, we get an additional point from the repair step. In the horizontal subproblem, p^5 only has a passive impact to the definition of the new δ-rectangles since we know it only as a dummy point. In the vertical subproblem, we know that the analogue of p^5 (after the repair step is applied) is a nondominated point which we potentially want to add to *Rep*. This point is easily and straightforward incorporated in the distinction of cases from above.

Remark 4.27:
In each upcoming subproblem, we could always incorporate the δ-rectangles with the help of disjunctive constraints as in (4.2). However, since we do not want to restrict to certain classes of problems, we assume that, in general, adding the corresponding ε-constraint and disjunctive constraints, including the creation of two new binary variables, makes the problem even harder than adding only the ε-constraint. Therefore, we recommend to only add the disjunctive constraints where they are really needed. ◁

4.2.3 Enumeration of the Whole Nondominated Set

As already mentioned in the introduction (see Chapter 1) and in Section 2.2, enumerating the whole nondominated set is often no efficient way to tackle a multicriteria optimization problem but, nevertheless, the area-based as well as the coverage-based Box-Algorithm can be turned into an exact algorithm enumerating the whole nondominated set Y_N. To achieve this, the predefined accuracy parameter has to be

chosen sufficiently small, i.e., $\Delta < 1$ and $\delta^C < 1$ for the area- and the coverage-based Box-Algorithm, respectively.

Remark 4.28:
Hamacher et al. (2007) noted that in the area-based Box-Algorithm, it suffices to choose $\Delta = 1$ to obtain the whole nondominated set; but this is incorrect, which can be observed in the following example.

Suppose we have given an instance inducing the following nondominated set

$$Y_N = \left\{ y^1 := (0,3)^\top, y^2 := (3,2)^\top, y^3 := (4,1)^\top, y^4 := (5,0)^\top \right\}.$$

The area-based Box-Algorithm with $\Delta = 1$ and subproblem (P_ε^{hor}) is then applied to the given instance. First, the outcomes y^1 and y^4 of the two lexicographic minima are found and added to *Rep*. In the first iteration, ε is chosen as $\lfloor 2.5 \rfloor = 2$ and the invoked subproblem returns no new point. Therefore, only rectangle $\mathrm{R}(y^2, y^4)$ remains and is investigated with $\varepsilon = 4$ identifying the new representative point y^3. Thereafter, the algorithm terminates since no rectangle remains with area greater than 1. Although point y^2 was used to define a rectangle during the algorithm, it was only considered as a dummy point and not identified to be a nondominated point; hence, it was missed adding y^2 to *Rep*. ◁

The coverage-based Box-Algorithm divides each rectangle along the longest side. If we aim at minimizing the coverage error, Example 4.11 showed that dividing at the shortest side can lead to a lot of unnecessary iterations. In the current setting in which we are interested in the whole nondominated set, the division of the longest side can lead to unnecessary many iterations, which is illustrated in the following example.

Example 4.29:
Let $n > 1$ be given and arbitrary. Suppose an instance is given which has only three nondominated points $y^1 := (0,2)^\top$, $y^2 := (2^n - 1, 1)^\top$ and $y^3 := (2^n, 0)^\top$. The coverage-based Box-Algorithm with $\delta^C < 1$ is applied to find the whole set Y_N. In the initialization step, the Box-Algorithm finds the outcomes y^1 and y^3 of the two lexicographic minima defining the initial rectangle with width 2^n and height 2. Since $2^n > 2$, the horizontal subproblem with $\varepsilon = 2^{n-1}$ is called and only one new rectangle is inserted into the list of unexplored rectangles, namely $\mathrm{R}((2^{n-1}+1, 1)^\top, y^3)$. Similarly, in iteration $k \leq n$, always the horizontal subproblem is called with $\varepsilon_k = \sum_{i=1}^{k} 2^{n-i}$ and the new rectangle $\mathrm{R}((\varepsilon_k + 1, 1)^\top, y^3)$ is added to the list of unexplored rectangles. After n iterations (with $\varepsilon_n = 2^n - 1$), the third point y^2 is found and no rectangle remains unexplored.

If, in each iteration, the vertical subproblem would have been called, the Box-Algorithm would be finished after two iterations. ◁

As the above example indicates, the division of the shortest side can lead to a smaller number of iterations. For an arbitrary given rectangle, the number of integral nondominated points inside this rectangle is bounded by the length of the shortest side + 1. Hence, if we always divide the shortest side, the number of iterations can be bounded by the length of the shortest side of the initial rectangle.

Theorem 4.30: *Let $\delta^C < 1$. A modified version of Algorithm 4.1 which chooses the subproblem placing the ε-constraint always on the shortest side of any rectangle finds the whole nondominated set Y_N in at most $\ell := \min\{z_1^2 - z_1^1, z_2^1 - z_2^2\}$ iterations, where z^1 and z^2 are the corner points of the initial rectangle $\mathrm{R}(z^1, z^2)$.*

Proof:
We only have to prove termination since the accuracy parameter $\delta^C < 1$ ensures the enumeration of all nondominated points. Clearly, during the algorithm, the length of each side of an arbitrary upcoming rectangle is integral. In an arbitrary iteration, if the shortest side of a rectangle is equal to 1, the next call of a subproblem on this rectangle completely explores this rectangle. If the shortest side of a rectangle is equal to 0, there is no need to explore this rectangle further since we always know the right corner point of a given rectangle as a nondominated point.

In each iteration, the rectangle with the maximal shortest side greater than or equal to 1 is selected and is substituted with at most two new rectangles. Due to the integrality gap in the update step, the summed up length of the shortest sides of both new rectangles is one less than the shortest side of the selected rectangle. Hence, the summed up length of the shortest sides of all unexplored rectangles in the list S after iteration k is bounded by $\ell - k$. Therefore, after ℓ iterations, we have no unexplored rectangle in our list S with maximal shortest side greater than or equal to 1. □

4.2.4 Solving Subproblems with a Positive Gap

In each iteration of each Box-Algorithm presented, we have to solve a corresponding lexicographic ε-constraint scalarization. For this task, adequate solvers should be used, e.g., CPLEX for linear or (mixed-)integer programming problems. In this subsection, we consider the situation in which a subproblem is hard to solve to optimality but it can be efficiently solved for a given positive gap. We provide ideas how the Box-Algorithm has to be modified to treat this situation appropriately. We consider DBOP and assume w.l.o.g. that $y^I > 0$. We investigate the horizontal Box-Algorithm with subproblem

(P_ε^{hor}) with additional upper bound $f_2(x) \leq y_2^1$ for a considered rectangle $\mathrm{R}(y^1, y^2)$.[4]

$$\begin{aligned} (P_\varepsilon^{hor}) \quad \operatorname{lex\,min} \ & (f_2(x),\ f_1(x)) \\ \text{s.t.} \ & x \in X \\ & f_1(x) \leq \varepsilon \\ & f_2(x) \leq y_2^1 \end{aligned}$$

In practice, this subproblem is solved in two steps (S1) and (S2) with appropriate chosen parameters $\varphi_i^\ell, \varphi_i^u \in \mathbb{R} \cup \{\pm\infty\}$, $i = 1, \ldots, 4$.

$$\begin{aligned} \text{(S1)} \quad f_2^* \coloneqq \min \ & f_2(x) \\ \text{s.t.} \ & x \in X \\ & \varphi_1^\ell \leq f_1(x) \leq \varphi_1^u \\ & \varphi_2^\ell \leq f_2(x) \leq \varphi_2^u \end{aligned} \qquad \begin{aligned} \text{(S2)} \quad f_1^* \coloneqq \min \ & f_1(x) \\ \text{s.t.} \ & x \in X \\ & \varphi_3^\ell \leq f_1(x) \leq \varphi_3^u \\ & \varphi_4^\ell \leq f_2(x) \leq \varphi_4^u \end{aligned}$$

For the investigation of a rectangle $\mathrm{R}(y^1, y^2)$, first, (S1) is solved with $\varphi_1^\ell = \varphi_2^\ell = -\infty$, $\varphi_1^u = \varepsilon$ and $\varphi_2^u = y_2^1$. Thereafter, (S2) is solved with $\varphi_3^\ell = \varphi_4^\ell = -\infty$, $\varphi_3^u = \varepsilon$ and $\varphi_4^u = f_2^*$ where f_2^* is induced from the first step.

We look at three different cases:

Case 1: The first step (S1) cannot be solved to optimality.

Case 2: The second step (S2) cannot be solved to optimality.

Case 3: Both steps cannot be solved to optimality.

For each case, we examine the problems which can occur and present ideas how the horizontal Box-Algorithm can be modified such that a representative system of desired accuracy is obtained. These modifications also include possible additional assumptions. Here, we only demand representative points in Y and not necessarily in Y_N.

Case 1: Suppose that in each iteration, for arbitrary parameters, the first step (S1) of the horizontal subproblem can only be solved up to a gap of $\theta > 0$, i.e., if (S1) is feasible, the solver outputs a feasible solution with objective value f_2^θ fulfilling

$$\frac{f_2^\theta - f_2^*}{f_2^\theta} \leq \theta$$

where f_2^* denotes the optimal objective value for (S1). For the investigation of a rectangle, we get two information from this. First, a feasible solution which can be

[4] Note that this upper bound can also be used in the horizontal Box-Algorithm, which does not change anything there.

used to cut off a dominated region and, second, a lower bound $\left\lceil (1-\theta)\cdot f_2^\theta \right\rceil$ for the optimal objective value f_2^*.

Under this assumption, a first issue is the computation of the initial rectangle such that Y_N is completely covered since we cannot compute the lexicographic minima exactly. Let z^1 and z^2 be the images of the lexicographic minima in normal and reversed order, respectively. In a normal setting, both are computed with the help of the two subproblems (S1) and (S2). For z^1, subproblem (S2) is solved with parameters $-\varphi_3^\ell = -\varphi_4^\ell = \varphi_3^u = \varphi_4^u = \infty$ and, thereafter, (S1) is solved with parameters $\varphi_1^\ell = \varphi_2^\ell = -\infty$, $\varphi_1^u = f_1^*$ and $\varphi_2^u = \infty$, where f_1^* is the obtained optimal objective value of (S2). Similarly, for z^2, subproblem (S1) is solved with parameters $-\varphi_1^\ell = -\varphi_2^\ell = \varphi_1^u = \varphi_2^u = \infty$ and, thereafter, (S2) is solved with parameters $\varphi_3^\ell = \varphi_4^\ell = -\infty$, $\varphi_3^u = \infty$ and $\varphi_4^u = f_2^*$, where f_2^* is the obtained optimal objective value of (S1).

Under the above assumption, let $\hat{z}^1$ and $\hat{z}^2$ be the images of the obtained feasible solutions by solving the optimization problems corresponding to the lexicographic minima in normal and reversed order, respectively. Then, z^1 can be substituted appropriately with the resulting point $\hat{z}^1$ with $\hat{z}_1^1 = z_1^1$ and $\hat{z}_2^1 \geq z_2^1$. Hence, we can use $\hat{z}^1$ as left corner point for the initial rectangle. For $\hat{z}^2$, we can only guarantee $(1-\theta)\cdot\hat{z}_2^2 \leq z_2^2 \leq \hat{z}_2^2$ and a lower bound $\hat{z}_1^2 \leq z_1^2$ for the first coordinate of z^2. Therefore, $\hat{z}^2$ cannot be used to define the right corner point appropriately. For an appropriate right corner point, we still need an upper bound for z_1^2. Here, we have to additionally assume that the corresponding outcome set Y is bounded from above in the first coordinate. Then, we can maximize the first objective and use this information, in addition to the information we gain from $\hat{z}^2$, to get the right corner point for the new initial rectangle

$$\left(\max_{x\in X} f_1(x), \left\lceil (1-\theta)\cdot\hat{z}_2^2 \right\rceil\right)^\top. \tag{4.3}$$

Note that this point is in general only a dummy point.

In an arbitrary iteration, for a rectangle $\mathrm{R}(y^1, y^2)$, let $z' \in Y$ be the outcome of the obtained feasible solution for the corresponding first step (S1). Suppose $z' \in \mathrm{R}(y^1, y^2)$, then we define $f_2^\theta := z_2'$, w.l.o.g. $\left\lceil (1-\theta)\cdot f_2^\theta \right\rceil - 1 \geq y_2^2$. Hence, due to dominance and the structure of the problem, we can already imply the exclusion of the following regions

$$\mathrm{R}\Big((z_1', y_2^1)^\top, (y_1^2, f_2^\theta)^\top\Big) \quad \text{and} \quad \mathrm{R}\Big((y_1^1, \left\lceil (1-\theta)\cdot f_2^\theta \right\rceil - 1)^\top, (\varepsilon, y_2^2)^\top\Big) \tag{4.4}$$

which are illustrated in Figure 4.10.

If in each iteration, (S2) can be efficiently solved to optimality, we can exclude another region from further investigation. Let $z^* \in Y$ denote the outcome of the optimal

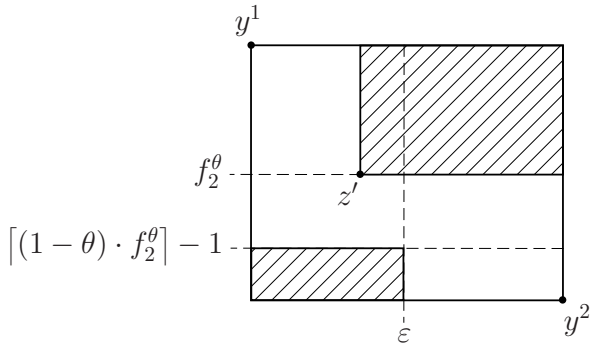


Figure 4.10: Two regions which can be excluded after the first step.

solution of the second step with $\varphi_4^u = f_2^\theta$, w.l.o.g. z^* lying in the interior of the rectangle $\mathrm{R}\Big((y_1^1, f_2^\theta)^\top, (\varepsilon, \Big\lceil (1-\theta)\cdot f_2^\theta \Big\rceil)^\top\Big)$.

Then, besides two analogous regions as above, we can additionally cut off

$$\mathrm{R}\Big((y_1^1, f_2^\theta)^\top, (z_1^* - 1, \Big\lceil (1-\theta)\cdot f_2^\theta \Big\rceil)^\top\Big)$$

since another solution inside this region would contradict optimality of z^* for (S2). Hence, three new rectangles remain to be explored further (see also Figure 4.11)

$$B_1 \coloneqq \mathrm{R}\Big(y^1, (z_1^* - 1, f_2^\theta + 1)^\top\Big), \quad B_2 \coloneqq \mathrm{R}\Big(z^* + (0,-1)^\top, \Big(\varepsilon, \Big\lceil (1-\theta)\cdot f_2^\theta \Big\rceil\Big)^\top\Big)$$
$$\text{and} \quad B_3 \coloneqq \mathrm{R}\Big((\varepsilon + 1, z_2^* - 1)^\top, y^2\Big).$$

After each iteration, we get at most three new rectangles, each of which has at most half the size of the original rectangle. However, for the upcoming rectangles, the correctness property can be violated, i.e., even though the real optimal solution would lie in the currently considered rectangle, the point found by the solver must not lie in this rectangle, which is illustrated in the following example.

Example 4.31:
Suppose we have given an instance with 5 points

$$Y = \left\{ y^1 \coloneqq \begin{pmatrix} 0 \\ 300 \end{pmatrix}, y^2 \coloneqq \begin{pmatrix} 200 \\ 100 \end{pmatrix}, y^3 \coloneqq \begin{pmatrix} 300 \\ 95 \end{pmatrix}, y^4 \coloneqq \begin{pmatrix} 400 \\ 90 \end{pmatrix}, y^5 \coloneqq \begin{pmatrix} 600 \\ 0 \end{pmatrix} \right\}$$

and we can solve (S1) with a gap of $\theta = 0.1$. Then, w.l.o.g., after the algorithm has found the outcomes y^1 and y^5 of the two lexicographic minima, the horizontal subproblem corresponding to $\varepsilon = 300$ is invoked. Suppose that, due to θ, both steps

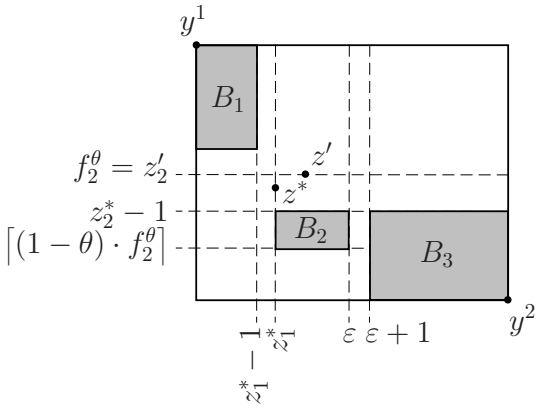


Figure 4.11: Three remaining rectangles after the second step.

(S1) and (S2) output point y^2 instead of the desired point y^3. Then, the three new rectangles $\mathrm{R}\Big(y^1, y^2 + (-1,1)^\top\Big)$, $\mathrm{R}\Big(y^2 + (0,-1)^\top, (300,90)^\top\Big)$ and $\mathrm{R}\Big((301,99)^\top, y^5\Big)$ are induced. In a next step, the latter rectangle is investigated with desired optimal outcome y^4. However, due to θ, the solver can return point y^3 not contained in the latter rectangle. $\triangleleft$

Hence, in our setting, it is not easy to guarantee the correctness property. However, if we introduce also lower bounds on both objective functions corresponding to the lower bounds of the rectangle, we can guarantee to find a feasible point, if existent, in the half of such a rectangle. Hence, for a considered rectangle $\mathrm{R}(y^1, y^2)$, for (S1) and (S2), we use the parameters $\varphi_1^\ell = \varphi_3^\ell = y_1^1$, $\varphi_2^\ell = \varphi_4^\ell = y_2^2$, $\varphi_1^u = \varphi_3^u = \varepsilon$, $\varphi_2^u = y_2^1$ and $\varphi_4^u = f_2^\theta$ where f_2^θ is induced from the first step.

The introduction of lower bounds can lead to representative points which dominate other representative points. To overcome this problem, we do not select the unexplored rectangle with largest area but the left most rectangle not yet fulfilling the desired accuracy. Then, if we find a new representative point, we have to cut off the dominated part not only in its corresponding rectangle but also in all other rectangles containing a portion of the dominated region of this new found point. This means that in a rectangle which contains a portion of this dominated region, we have to reduce the second coordinate of the left corner point appropriately.

During the algorithm, in contrast to the original Box-Algorithm, the right corner points of each rectangle may only be dummy points and no known representative points. Hence, in the end when the termination condition is fulfilled, we additionally have to consider all rectangles without a representative point and search for a feasible point in each of them.

With all these modifications, we can process all upcoming rectangles appropriately, also such kind of rectangles $\mathrm{R}(y^1, y^2)$ with $(1-\theta) \cdot y_2^1 \leq y_2^2$, which is always fulfilled for rectangles of type B_2. This is possible since in step (S1), the solver always returns a feasible solution if (S1) is feasible.

Case 2: We proceed to consider the second case mentioned at the beginning of this subsection. Suppose that in each iteration, for arbitrary parameters, the second step (S2) of the horizontal subproblem can only be solved up to a gap of $\theta > 0$, i.e., if (S2) is feasible, the solver outputs a feasible solution with objective value f_1^θ fulfilling

$$\frac{f_1^\theta - f_1^*}{f_1^\theta} \leq \theta$$

where f_1^* denotes the optimal objective value for (S2). As in the first case, for the investigation of a rectangle, we get two information from this. First, a feasible solution which can be used to cut off a dominated region and a lower bound $\left\lceil (1-\theta) \cdot f_1^\theta \right\rceil$ for the optimal objective value f_1^*.

For the initial rectangle, we get an analogous issue as for the first case. Under the above assumption, let $\hat{z}^1$ and $\hat{z}^2$ be the images of the obtained feasible solutions by solving the optimization problems corresponding to the lexicographic minima in normal and reversed order, respectively. With the same way of argumentation as in the first case, we can use $\hat{z}^2$ and the dummy point

$$\left(\left\lceil (1-\theta) \cdot \hat{z}_1^1 \right\rceil, \max_{x \in X} f_2(x) \right)^\top \tag{4.5}$$

to define an initial rectangle covering Y_N while assuming that the corresponding outcome set Y is bounded from above in the second coordinate.

Since we can exactly solve (S1), we look at the outcome z^* of the obtained feasible solution for a rectangle $\mathrm{R}(y^1, y^2)$ after (S2) was executed. Here, we can exclude the following regions

$$\mathrm{R}\left((z_1^*, y_2^1)^\top, (y_1^2, z_2^*)^\top \right) \quad \text{and} \quad \mathrm{R}\left((y_1^1, z_2^* - 1)^\top, (\varepsilon, y_2^2)^\top \right).$$

Hence, we get the same partition as in the original Box-Algorithm (cf. Lemma 4.4) and we can process as there. With this knowledge, also the first case can be processed as in the horizontal Box-Algorithm if we are allowed to switch the two objective functions at the beginning.

Case 3: Suppose now that in each iteration, for arbitrary parameters, both (S1) and (S2) can only be solved up to a gap of $\theta > 0$. The initial rectangle is obviously obtained by the two dummy points (4.3) and (4.5) from above considerations with the assumption that the anti-ideal point y^{AI} exists and can be computed.

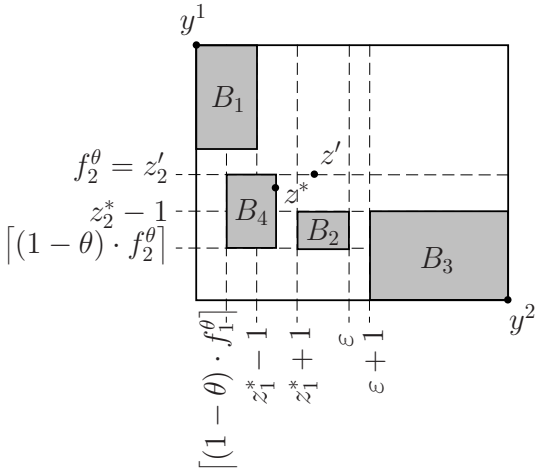


Figure 4.12: Four remaining rectangles after the second step.

For a rectangle $\mathrm{R}(y^1, y^2)$, after the first step (S1) is solved, we are in the same situation as in the first case and we can already imply the exclusion of the two regions in (4.4) illustrated in Figure 4.10. After the second step (S2) is solved, let z^* be the outcome of the obtained feasible solution, w.l.o.g. lying in the interior of the rectangle

$$\mathrm{R}\Big((y_1^1, f_2^\theta)^\top, \Big(\varepsilon, \Big\lceil (1-\theta)\cdot f_2^\theta \Big\rceil\Big)^\top\Big)\,,$$

with objective value $f_1^\theta \coloneqq z_1^*$, w.l.o.g. $\Big\lceil (1-\theta)\cdot f_1^\theta \Big\rceil - 1 \geq y_1^1$. Besides analogous regions as after the first step (S1), we can additionally exclude the region

$$\mathrm{R}\Big((y_1^1, f_2^\theta)^\top, \Big(\Big\lceil (1-\theta)\cdot f_1^\theta \Big\rceil - 1, \Big\lceil (1-\theta)\cdot f_2^\theta \Big\rceil\Big)^\top\Big)\,.$$

Hence, four new rectangles remain to be explored further (see Figure 4.12)

$$B_1 \coloneqq \mathrm{R}\Big(y^1, (f_1^\theta - 1, f_2^\theta + 1)^\top\Big), \quad B_2 \coloneqq \mathrm{R}\Big(z^* + (1,-1)^\top, \Big(\varepsilon, \Big\lceil (1-\theta)\cdot f_2^\theta \Big\rceil\Big)^\top\Big)$$
$$B_3 \coloneqq \mathrm{R}\Big((\varepsilon + 1, z_2^* - 1)^\top, y^2\Big) \quad \text{and}$$
$$B_4 \coloneqq \mathrm{R}\Big(\Big(\Big\lceil (1-\theta)\cdot f_1^\theta \Big\rceil, f_2^\theta\Big)^\top, \Big(f_1^\theta, \Big\lceil (1-\theta)\cdot f_2^\theta \Big\rceil\Big)^\top\Big)\,.$$

With the same modifications as for the first case, we can process all rectangles to reach the desired accuracy.

5 Box-Algorithm for Tricriteria Optimization Problems

5.1 Introduction

In this chapter, the Box-Algorithm from Hamacher et al. (2007) (see Chapter 4) is extended to the case of three objectives to obtain a representative system with desired coverage error (cf. Definition 2.9). Therefore, we consider the MOP from Definition 2.1 with $p = 3$, i.e., the following *3-criteria optimization problem*

$$\begin{aligned} \text{(3OP)} \quad & \min \quad f(x) = (f_1(x),\ f_2(x),\ f_3(x)) \\ & \text{s.t.} \quad x \in X \subseteq \mathbb{R}^n \end{aligned}$$

where X denotes the feasible set and $f : X \longrightarrow \mathbb{R}^3$ the objective function.

We note here that the consideration is not only restricted to discrete optimization problems as in the previous chapter and in the paper Hamacher et al. (2007) but to all optimization problems for which the scalarization problems can be solved using an appropriate black-box solver (for linear and (mixed-)integer programming problems, e.g., CPLEX).

The algorithm computes a set of boxes which contains the nondominated set Y_N. Thus, we first define the concept of a box in $\mathbb{R}^3$.

Definition 5.1 (Box):
Let $\ell, u \in \mathbb{R}^3$ with $\ell \leqq u$. We refer to the cuboid

$$\mathrm{B}(\ell, u) := \ell + \mathbb{R}^3_{\geqq} \cap u - \mathbb{R}^3_{\geqq} = \{y \in \mathbb{R}^3 | \ell \leqq y \leqq u\}$$

as the *box* defined by ℓ and u. The two points ℓ and u are called *lower left* and *upper right corner point* of the box $\mathrm{B}(\ell, u)$, respectively.
We denote with $\mathrm{Vol}(\mathrm{B}(\ell, u)) := (u_1 - \ell_1) \cdot (u_2 - \ell_2) \cdot (u_3 - \ell_3)$ the volume of the box.
For the maximum norm $\|\cdot\|_\infty$ on $\mathbb{R}^3$, the metric or distance $\|\ell - u\|_\infty$ is referred to as the *corner point distance* of the box $\mathrm{B}(\ell, u)$. ◇

To this end, recently published approaches also using this concept of boxes in the context of multiple objective programming are briefly reviewed in the following.

Laumanns et al. (2006) as well as the follow-up work of Dhaenens et al. (2010), Kirlik and Sayın (2014) and Boland et al. (2014b) do not compute a representative system of the nondominated set but aim at finding the exact nondominated set. Their method fixes one coordinate, projects the p-dimensional outcome space onto the remaining $(p-1)$-dimensional space and partitions the projection by means of boxes into a grid. Then, this grid is used to search for new nondominated points by solving ε-constraint subproblems. The grid is updated adaptively until the nondominated set is found completely. Laumanns et al. (2006) notice that also a heuristic can be derived to get a representative system of the nondominated set. In contrast to Laumanns et al. (2006), Dhaenens et al. (2010) apply a partitioning directly in the p-dimensional outcome space and suggest calculating the true nadir point as described in Ehrgott and Tenfelde-Podehl (2003) in a first phase which also yields some intermediate solution points. These solution points are used to determine an initial grid in a lower-dimensional projected subspace. Later, Dächert and Klamroth (2013) improve the splitting of the boxes used in Dhaenens et al. (2010) and propose a generic algorithm (not necessarily using ε-constraint problems but also, e.g., augmented weighted Tchebycheff problems) which can also find the whole nondominated set for problems with three objectives while the outcome space is decomposed into relevant boxes. Recently, Boland et al. (2014a) proposed another method for finding all nondominated points for integer programs with three objectives. They use so-called L-shapes and rectangles to partition a two-dimensional projected search space. Moreover, they evaluate experimentally the quality of the representative system with the help of the hypervolume indicator if the algorithm is stopped before all nondominated points are found. All these algorithms use only dominance information and information about the calculated points to split boxes and to guide the search. The considered subproblems either have no lower bound in the fixed coordinate or no lower bound at all (besides the ideal point). These methods do not aim at and are not capable for bounding and controlling the coverage error accurately during the running time; a bound on the number of iterations needed for a predefined desired coverage error is not provided.

Besides these methods, the algorithm of Barichard and Hao (2003) which uses evolutionary concepts considers union of boxes containing the nondominated set. Without solving any problem, they first bisect the initial box and all resulting boxes according to a selected objective until they have reached the maximal population size. Thereafter, they try to instantiate each box (with an unspecified instantiation procedure) and then remove dominated boxes, which reduces the current population. The procedure is repeated until the number of (not deleted) instantiated boxes has reached the given maximal population size. The execution of this algorithm can be very expensive since the division of the boxes is made independently, a priori, of the found solutions. Moreover, they do not use the coverage error but the cardinality of the representative system as stopping criterion.

In this chapter, the Box-Algorithm for two objectives of Hamacher et al. (2007) is extended to the case of three objectives to obtain a representative system with desired coverage error. To the best of our knowledge, this is the first algorithm for tricriteria optimization problems relying on boxes subdividing the whole nondominated set for which an iteration bound to achieve a desired coverage error is proven. Besides dominance information stemming from the ordering relation $\leqq$ in $\mathbb{R}^3$ as well as information about optimality and feasibility of the lexicographic ε-constraint problem, our algorithm chooses iteratively appropriately placed ε-constraints to subdivide the outcome space and to reduce the coverage error. Moreover, also the lower bounds of the considered boxes are modified such that a partition of the remaining unexplored space into non-overlapping boxes is obtained. The aim of the algorithm is not to enumerate all nondominated solutions but to compute a representative system of Y_N while reducing the coverage error. Therefore, the upcoming ε-constraint problems in our algorithm cannot be limited to one fixed first coordinate which is minimized, the algorithm has to choose the appropriate first coordinate and, thereby, the appropriate subdivision of boxes, i.e., the two ε-constraints, adaptively in each iteration. Since lower bounds are used for the boxes, optimal solutions of the ε-constraint subproblems might be dominated. However, in the end, the unexplored space is reduced, a subdivision of boxes comprising the whole nondominated set and a representative system with desired accuracy is received. A straightforward extension of Hamacher et al. (2007) could use the volume of the boxes as accuracy condition. This accuracy condition can also be implemented in our new algorithm, and the needed analogue results for three objectives are also proven (see Section 5.6). However, since we experienced that the volume of boxes is not a robust and reliable quality measure, as lower-dimensional boxes would get an empty volume and boxes with two huge sides and one very small side would not be penalized (cf. Example 4.11), the focus in this chapter lies on the coverage error. Furthermore, it is explained how the other quality measures, i.e., cardinality and uniformity, can be treated in the theoretical analysis of the algorithm. Since the new algorithm can produce dominated representative points, the representation error can be greater than 0.

5.2 Initialization

The first step of the algorithm consists of computing the initial box. Naturally, we want to start with a box $\mathrm{B}(\ell^0, u^0)$ containing Y_N.

Lemma 5.2: *For* 3OP*, let* $\ell^0 \in y^I - \mathbb{R}^3_{\leqq}$ *and let* $u^0 \in y^N + \mathbb{R}^3_{\leqq}$*. Then* $Y_N \subseteq \mathrm{B}(\ell^0, u^0)$.

For the case of two objectives, we have initialized ℓ^0 and u^0 with the ideal point y^I and nadir point y^N, respectively, which was corresponding to the smallest bounding box

containing Y_N. For arbitrary dimensions, the ideal point can be easily computed but computing the real nadir point for more than two objectives is very difficult and no efficient method is known from literature tackling this task for general problems (only for special problems like linear or discrete multiple objective optimization problems, see, e.g., Alves and Costa (2009); Köksalan and Lokman (2015); Kirlik and Sayın (2015)). There exists a general method from Ehrgott and Tenfelde-Podehl (2003), but for computing the nadir point for three objectives, first, the nondominated sets of three appropriate bicriteria problems have to be computed, which is in most settings no efficient way. In the literature, usually an estimate of the nadir point is used calculated with the help of a payoff table (see, e.g., Ehrgott, 2005). However, this estimate comes with no guarantee (Korhonen et al., 1997) and cannot serve as an upper bound of the real nadir point and, therefore, we could not guarantee $Y_N \subseteq \mathrm{B}(\ell^0, u^0)$ since only a subset of Y_N could be contained in the initial box.

In a practical situation, if the problem is bounded from above, one may choose the initial upper right corner point u^0 as the anti-ideal point y^{AI} which gives the guarantee that the whole set Y, and in particular Y_N, is then contained in the initial box $\mathrm{B}(y^I, y^{AI})$ (see Lemma 5.2). Otherwise, the decision makers can be asked for an acceptable upper bound or some heuristics (see, e.g., Korhonen et al., 1997; Deb et al., 2010) can be used to estimate the nadir point, but then with no guarantee to cover the whole nondominated set. For the remainder of this chapter, we assume a given subroutine InitialBox() which determines the initial box $\mathrm{B}(\ell^0, u^0)$ sufficiently large such that it subsumes the nondominated set entirely. In this chapter, we further assume that $Y \cap (u^0 - \mathbb{R}^3_{\leqq})$ is compact such that the considered scalarization problems either obtain an optimal solution or report infeasibility. This assumption leads to the following result.

Lemma 5.3: *For* 3OP*, let* $u^0 \in \mathbb{R}^3$ *with* $Y \cap (u^0 - \mathbb{R}^3_{\leqq})$ *compact. Then, the nondominated set* Y_N *is externally stable w.r.t.* $Y \cap (u^0 - \mathbb{R}^3_{\leqq})$*.*

Proof:
This result follows immediately from Theorem 2.21 in Ehrgott (2005). □

After initialization and during the execution of the algorithm, every box generated by the algorithm consists of edges and faces which are parallel to one of the three coordinate axes.

5.3 Update Step

As long as some termination condition is not satisfied, the algorithm chooses some box $\mathrm{B}(\ell, u)$—referred to as the *current box*—due to some *selection rule* SelectBox(S)

(e.g., the box with the largest corner point distance) in order to refine the representative system locally, i.e., in the area contained within this current box.

Definition 5.4:
Let $\ell, u \in \mathbb{R}^3, \ell \leqq u$ and let $\varepsilon_i = \frac{\ell_i + u_i}{2}$ for $i = 1, 2, 3$. Then, we define the following three subproblems $(P^1_{\varepsilon_1,\varepsilon_2}), (P^2_{\varepsilon_1,\varepsilon_3})$, and $(P^3_{\varepsilon_2,\varepsilon_3})$ called *lexicographic ε-constraint scalarizations with lower bounds* as

$$
\begin{aligned}
(P^1_{\varepsilon_1,\varepsilon_2}) \quad \operatorname{lex\,min} \quad & (f_3(x),\ f_2(x),\ f_1(x)) \\
\text{s.t.} \quad & x \in X \\
& \left.\begin{array}{l} \ell_1 \leq f_1(x) \leq \varepsilon_1 \\ \ell_2 \leq f_2(x) \leq \varepsilon_2 \\ \ell_3 \leq f_3(x) (\leq u_3) \end{array}\right\} =: f(x) \in \mathrm{B}(\ell, u)^{(\varepsilon_1, \varepsilon_2, u_3)}
\end{aligned}
$$

and, analogously,

$$
\begin{aligned}
(P^2_{\varepsilon_1,\varepsilon_3}) \quad \operatorname{lex\,min} \quad & (f_2(x),\ f_1(x),\ f_3(x)) \\
\text{s.t.} \quad & x \in X \\
& f(x) \in \mathrm{B}(\ell, u)^{(\varepsilon_1, u_2, \varepsilon_3)}
\end{aligned}
\qquad
\begin{aligned}
(P^3_{\varepsilon_2,\varepsilon_3}) \quad \operatorname{lex\,min} \quad & (f_1(x),\ f_3(x),\ f_2(x)) \\
\text{s.t.} \quad & x \in X \\
& f(x) \in \mathrm{B}(\ell, u)^{(u_1, \varepsilon_2, \varepsilon_3)}
\end{aligned}
$$

◇

For the local refinement of a box, the two longest edges of the current box are determined and, depending thereon, one of the three lexicographic ε-constraint scalarizations with lower bounds is solved, namely the one with ε-constraints bisecting these two longest edges.

Without loss of generality, we may assume for the subsequent exposition of our ideas that the first subproblem $(P^1_{\varepsilon_1,\varepsilon_2})$ applies.

Proposition 5.5: *Let z^* be the outcome of an optimal solution of $(P^1_{\varepsilon_1,\varepsilon_2})$. Then, there does not exist a $y \in Y_N \setminus \{z^*\}$ such that*

$$
y \in \mathrm{B}(z^*, u) \cup \mathrm{B}\Big(\ell, (\varepsilon_1, \varepsilon_2, z^*_3)^\top\Big) \setminus \mathrm{B}\Big((\ell_1, z^*_2, z^*_3)^\top, (z^*_1, \varepsilon_2, z^*_3)^\top\Big) .
$$

Proof:
The set $\mathrm{B}(z^*, u)$ is dominated by z^* and the set

$$
\mathrm{B}\Big(\ell, (\varepsilon_1, \varepsilon_2, z^*_3)^\top\Big) \setminus \mathrm{B}\Big((\ell_1, z^*_2, z^*_3)^\top, (z^*_1, \varepsilon_2, z^*_3)^\top\Big)
$$

cannot contain a nondominated point since this contradicts optimality of z^* for the lexicographic ε-constraint scalarization with lower bounds. □

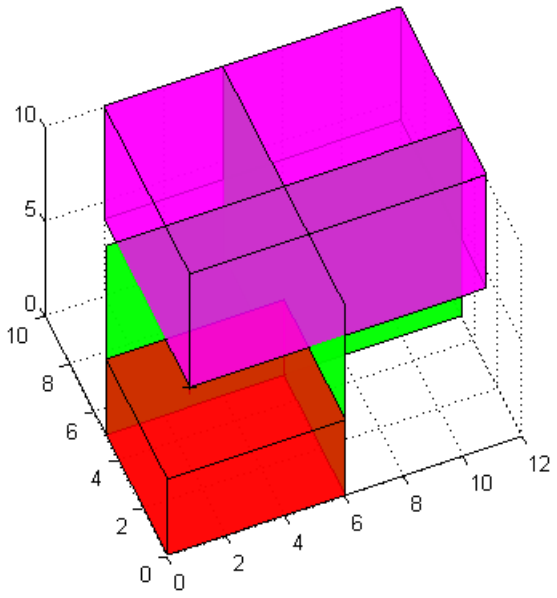


Figure 5.1: Excluded regions for Example 5.6 with the two ε-constraints indicated in green color.

Example 5.6:
We consider an initial box with $\ell = 0$ and $u = (12, 10, 10)^\top$. We solve subproblem $(P^1_{\varepsilon_1,\varepsilon_2})$ with $\varepsilon_1 = 6$ and $\varepsilon_2 = 5$. Suppose the lexicographic ε-constraint scalarization with lower bounds has an optimal solution with outcome $z^* = (2, 3, 4)^\top$. Then, due to Proposition 5.5, the two regions

$$\mathrm{B}\Big((2,3,4)^\top, (12,10,10)^\top\Big) \quad \text{and} \quad \mathrm{B}\Big((0,0,0)^\top, (6,5,4)^\top\Big) \setminus \mathrm{B}\Big((0,3,4)^\top, (2,5,4)^\top\Big)$$

do not contain unknown nondominated points and can thus be excluded from further consideration. These two sets are depicted in Figure 5.1 in magenta and red color, respectively. ◁

Excluding the two regions mentioned in Proposition 5.5 from the current box $\mathrm{B}(\ell, u)$, the remaining subset of the current box can be subdivided into new boxes the collection of which contains $Y_N \cap \mathrm{B}(\ell, u)$. The search space is thus reduced. The boxes of this new subdivision of the current box are located in each quarter of the current box (induced by the two ε-constraints).

Definition 5.7:
Let $\ell, u \in \mathbb{R}^3, \ell \leqq u$ and let $\varepsilon_i = \frac{\ell_i + u_i}{2}$ for $i = 1, 2, 3$. Suppose $(P^1_{\varepsilon_1,\varepsilon_2})$ is solved. Then,

the four *quarters of the current box* $\mathrm{B}(\ell, u)$ are defined as

$$Q^{1,1} \coloneqq \mathrm{B}\Big(\ell, (\varepsilon_1, \varepsilon_2, u_3)^\top\Big), \quad Q^{1,2} \coloneqq \mathrm{B}\Big((\varepsilon_1, \ell_2, \ell_3)^\top, (u_1, \varepsilon_2, u_3)^\top\Big),$$

$$Q^{1,3} \coloneqq \mathrm{B}\Big((\varepsilon_1, \varepsilon_2, \ell_3)^\top, u\Big), \quad Q^{1,4} \coloneqq \mathrm{B}\Big((\ell_1, \varepsilon_2, \ell_3)^\top, (\varepsilon_1, u_2, u_3)^\top\Big). \qquad \diamond$$

Having solved the lexicographic ε-constraint scalarization with lower bounds, we specify a subdivision of the current box $\mathrm{B}(\ell, u)$ into smaller boxes. This subdivision is not unique and other subdivisions than the one proposed next are eligible. It should be pointed out that the subsequent analysis applies to alternative subdivisions as long as each box is completely contained in a quarter and does not overlap with another box under consideration.

Definition 5.8:
Let $\ell, u \in \mathbb{R}^3, \ell \leqq u$ and let $\varepsilon_i = \frac{\ell_i + u_i}{2}$ for $i = 1, 2, 3$. Suppose $(P^1_{\varepsilon_1,\varepsilon_2})$ is solved and let $z^* \in \mathbb{R}^3$ denote the outcome of an optimal solution for $(P^1_{\varepsilon_1,\varepsilon_2})$. Then, the *subdivision of the current box* $\mathrm{B}(\ell, u)$ consists of the boxes

$$B^{1,1} \coloneqq \mathrm{B}\Big((\ell_1, z_2^*, z_3^*)^\top, (z_1^*, \varepsilon_2, u_3)^\top\Big),\ B^{1,2} \coloneqq \mathrm{B}\Big((\ell_1, \ell_2, z_3^*)^\top, (\varepsilon_1, z_2^*, u_3)^\top\Big) \text{ for } Q^{1,1},$$
$$B^{1,3} \coloneqq \mathrm{B}\Big((\varepsilon_1, \ell_2, \ell_3)^\top, (u_1, \varepsilon_2, z_3^*)^\top\Big),\ B^{1,4} \coloneqq \mathrm{B}\Big(\varepsilon_1, \ell_2, z_3^*)^\top, (u_1, z_2^*, u_3)^\top\Big) \text{ for } Q^{1,2},$$
$$B^{1,5} \coloneqq \mathrm{B}\Big((\varepsilon_1, \varepsilon_2, \ell_3)^\top, (u_1, u_2, z_3^*)^\top\Big) \text{ for } Q^{1,3}, \text{ and}$$
$$B^{1,6} \coloneqq \mathrm{B}\Big((\ell_1, \varepsilon_2, \ell_3)^\top, (\varepsilon_1, u_2, z_3^*)^\top\Big),\ B^{1,7} \coloneqq \mathrm{B}\Big((\ell_1, \varepsilon_2, z_3^*)^\top, (z_1^*, u_2, u_3)^\top\Big) \text{ for } Q^{1,4}.$$

$\diamond$

Remark 5.9:
In Definitions 5.7 and 5.8, we assumed that $(P^1_{\varepsilon_1,\varepsilon_2})$ was solved (due to clarity and brevity). The corresponding quarters $Q^{2,i}$ and $Q^{3,i}$, $i = 1, \ldots, 4$, and subdivisions $B^{2,j}$ and $B^{3,j}$, $j = 1, \ldots, 7$, for the subproblems $(P^2_{\varepsilon_1,\varepsilon_3})$ and $(P^3_{\varepsilon_2,\varepsilon_3})$, respectively, are defined analogously (for completeness see Appendix A.1). In general, the first superscript index i of a quarter $Q^{i,j}$ and a box $B^{i,k}$ refers to the number of the lexicographic ε-constraint scalarization with lower bounds $(P^i_{\cdot,\cdot})$, $i \in \{1, 2, 3\}$, which was solved. The second superscript indices $j \in \{1, 2, 3, 4\}$ and $k \in \{1, \ldots, 7\}$, simply enumerate the quarters and the boxes, respectively. $\triangleleft$

In Figure 5.2, the unexplored region and the subdivision for Example 5.6 is depicted.

We conclude this section with some remarks about the used scalarization problem and the corresponding subdivision.

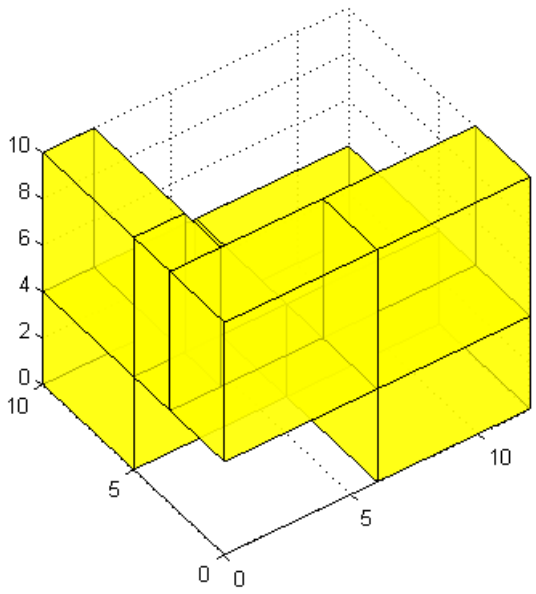


Figure 5.2: Unexplored Region

Remark 5.10:
If the point z^* is located on the boundary of $Q^{1,1}$, the subdivision should be further adjusted to avoid redundant boxes. These modifications are described in Section 5.7 such that the focus here is on the case that z^* is in the interior of $Q^{1,1}$. ◁

Remark 5.11:
If 3OP has three integer-valued objective functions, i.e., $f : X \longrightarrow \mathbb{Z}^3$, instead of solving three single-objective subproblems to obtain the lexicographic solution, we can also use an idea from Özlen and Azizoğlu (2009) and solve only one single-objective subproblem of the following form to obtain the lexicographic solution:

$$\begin{aligned} (P^1_{\varepsilon_1,\varepsilon_2}) \quad \min \quad & f_3(x) + w_2 \cdot f_2(x) + w_1 \cdot f_1(x) \\ \text{s.t.} \quad & x \in X \\ & f(x) \in \mathrm{B}(\ell, u)^{(\varepsilon_1,\varepsilon_2,u_3)} \end{aligned}$$

with $w_2 = \frac{1}{u_2-\ell_2+1}$ and $w_1 = \frac{1}{(u_2-\ell_2+1)(u_1-\ell_1+1)}$.
Clearly, this idea can also be utilized for the case of two objective functions. ◁

Remark 5.12:
Obviously, also a part of the border of some boxes can be excluded from further investigations. However, we assume here that the used black-box solver is not efficient or even not capable in handling strict inequality constraints and propose therefore

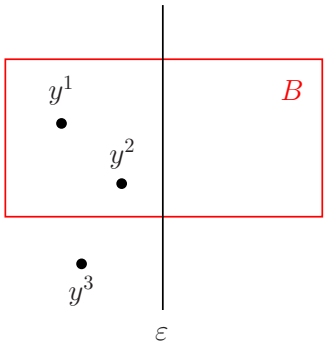


Figure 5.3: Lower Bound Problem

always the closure of these boxes. Nevertheless, the further results are also valid if we use strict inequalities to exclude the issued part of the border of the corresponding boxes. Strict inequalities of the kind $f_i(x) < a$ for some $a \in \mathbb{R}$ and $i \in \{1, \ldots, p\}$ can be numerically handled by introducing some small predefined tolerance β and using the inequality $f_i(x) \leq a - \beta$ instead (cf. Appendix A.2). $\triangleleft$

Remark 5.13:
The lower bound constraints in the considered subproblems are necessary to guarantee to find a solution in the desired box or to conclude that no feasible solution in this box exists.
However, with lower bound constraints on the objectives, we can obtain dominated points after solving the scalarization problem. This issue is demonstrated in the following example for simplicity for the case with two objectives: We consider the scenario given in Figure 5.3, where $y^1, y^3 \in Y_N$ and $y^2 \notin Y_N$. The corresponding lexicographic ε-constraint scalarization with lower bounds w.r.t. box B would output y^2 as optimal outcome which is dominated by y^3.

Despite this issue, we use the subproblem with the lower bound constraints to find new representative points in the currently considered box. Even if the found points do not have to be nondominated, we use them to exclude areas and to obtain a new subdivision of the current box. We assume that the decision makers do not care if some points of the representative system are not nondominated since their desired accuracy is reached. Later, we give more details how this problem can be overcome theoretically where we also derive a correctness property similar to Theorem 4.8 (see Section 5.9). However, we have to keep in mind that these theoretical considerations do not lead to an efficient algorithm and, because of this, we look further in the analysis of the Box-Algorithm using lexicographic ε-constraint scalarizations with lower bounds. $\triangleleft$

5.4 Correctness

In this subsection, we prove some properties and the correctness of the resulting Box-Algorithm. We make the same assumptions as in the preceding subsections. After problem $(P^1_{\varepsilon_1,\varepsilon_2})$ was solved for the current box $\mathrm{B}(\ell, u)$, the box is subdivided into the seven new boxes $B^{1,1}$ to $B^{1,7}$ defined as above for the quarters $Q^{1,1}$ to $Q^{1,4}$. Doing this, no nondominated point is missed.

Lemma 5.14: *Let z^* be the outcome of an optimal solution of $(P^1_{\varepsilon_1,\varepsilon_2})$ corresponding to the box $\mathrm{B}(\ell, u)$. Then, it holds $Y_N \cap \mathrm{B}(\ell, u) \subseteq \bigcup_{i=1}^{7} B^{1,i}$.*

Proof:
Follows directly from Proposition 5.5. □

As long as the corner point distance of all boxes exceed a value of $\delta^C > 0$, the algorithm chooses some box $\mathrm{B}(\ell, u)$ and solves one of the three lexicographic ε-constraint scalarizations with lower bounds. Using the outcome z^* of an optimal solution, a subdivision of the current box $\mathrm{B}(\ell, u)$ is obtained. The *coverage-based Box-Algorithm* for three objectives to obtain a representative system with desired coverage error δ^C w.r.t. $\|\cdot\|_\infty$ is stated as a pseudocode description in Algorithm 5.1.

Algorithm 5.1 Box-Algorithm (for three objectives, coverage)

Input: A multiple objective optimization problem with three objectives, $\delta^C > 0$.
Output: A representative system *Rep* with coverage error at most δ^C (cf. Theorem 5.15).

```
1: Rep ← ∅, B ← ∅
2: B(ℓ^0, u^0) ← InitialBox()
3: S ← {B(ℓ^0, u^0)}
4: while S ≠ ∅ do      // termination condition
5:     B(ℓ, u) ← SelectBox(S)      // selection rule
6:     S ← S \ {B(ℓ, u)}
7:     if ‖ℓ − u‖_∞ ≤ δ^C then      // accuracy condition
8:         Solve (P^1_{u1,u2}) and obtain optimal outcome ẑ.      // completion step
9:         if ẑ ≠ NULL then      // ẑ ∈ Y
10:            Rep ← Rep ∪ {ẑ}
11:            B ← B ∪ {B(ℓ, u)}
```

Algorithm 5.1 Box-Algorithm (for three objectives, coverage) – continued

12: **else** // *update step*
13: **if** $\min\{u_1 - \ell_1, u_2 - \ell_2\} \geq u_3 - \ell_3$ **then**
14: $r \leftarrow 1$, $i \leftarrow 1$, $j \leftarrow 2$
15: **else if** $\min\{u_1 - \ell_1, u_3 - \ell_3\} \geq u_2 - \ell_2$ **then**
16: $r \leftarrow 2$, $i \leftarrow 1$, $j \leftarrow 3$
17: **else** // $\min\{u_2 - \ell_2, u_3 - \ell_3\} \geq u_1 - \ell_1$
18: $r \leftarrow 3$, $i \leftarrow 2$, $j \leftarrow 3$
19: Solve $(P^r_{\varepsilon_i,\varepsilon_j})$ with $\varepsilon_k = \dfrac{\ell_k + u_k}{2}$, $k = i, j$, and obtain optimal outcome z^*.
20: **if** $z^* \neq NULL$ **then** // $z^* \in Y$
21: $Rep \leftarrow Rep \cup \{z^*\}$
22: **for** $i = 1, \ldots, 7$ **do**
23: $S \leftarrow S \cup \{B^{r,i}\}$
24: **else** // $z^* = NULL$, *i.e.,* $(P^r_{\varepsilon_i,\varepsilon_j})$ *infeasible*
25: **for** $i = 2, 3, 4$ **do**
26: $S \leftarrow S \cup \{Q^{r,i}\}$
27: **return** $(Rep, \mathcal{B})$

Finiteness, a bound on the complexity as well as correctness of the algorithm are proven in the following.

Theorem 5.15: *Algorithm 5.1 terminates in finitely many steps. It outputs a collection of boxes $\mathcal{B}$ containing all nondominated points. At termination, the representative system Rep has a coverage error of at most δ^C (w.r.t. $\|\cdot\|_\infty$). More precisely, the algorithm performs at most $\mathcal{O}\left(\left(\frac{L}{\delta^C}\right)^{2\cdot\log_2(7)}\right)$ many iterations, where L equals the corner point distance $\|\ell^0 - u^0\|_\infty$ of the initial box $\mathrm{B}(\ell^0, u^0)$.*

Proof:
Due to the construction of the boxes, a subdivision is obtained which contains Y_N at each state of the algorithm (see Lemma 5.14). The termination condition (line 4) together with the accuracy condition (line 7) and completion step (lines 8 to 11) of the algorithm guarantee that the coverage error of the representative system is at most δ^C if the algorithm terminates. Thus, it suffices to show the termination of the algorithm.

Analogous to the proof of Theorem 4.17, we consider the tree for which each node corresponds to a box in some iteration. More precisely, the root node of the tree corresponds to the initial box and after the first iteration (independently of the subproblem considered), this box is substituted with at most seven new boxes which are then represented by at most seven child nodes of the father node and so on. The

question about finiteness of the algorithm corresponds then to the question about a bounded height of the corresponding tree. This is shown in the following.

Let $\mathrm{B}(y^1, y^2)$ be a box corresponding to some node of the tree in an arbitrary but odd level greater than or equal to 3.[1] Let $\mathrm{B}(w^1, w^2)$ and $\mathrm{B}(v^1, v^2)$ be the boxes corresponding to its father and grandfather nodes, respectively. Consider now the iteration in which the box corresponding to the grandfather node is processed. Without loss of generality, let problem $(P^1_{\varepsilon_1,\varepsilon_2})$ be called, i.e.,

$$v_3^2 - v_3^1 \leq \min_{i=1,2}\{v_i^2 - v_i^1\} \leq \max_{i=1,2,3}\{v_i^2 - v_i^1\} = \|v^1 - v^2\|_\infty \,,$$

resulting in boxes including the box $\mathrm{B}(w^1, w^2)$. Due to the constraints in the lexicographic ε-constraint scalarization with lower bounds and the definition of the boxes, it holds

$$\max_{i=1,2}\{w_i^2 - w_i^1\} \leq \frac{\|v^1 - v^2\|_\infty}{2} \tag{5.1}$$

and also $w_3^2 - w_3^1 \leq \|v^1 - v^2\|_\infty$.

Claim: $\|y^1 - y^2\|_\infty \leq \frac{\|v^1 - v^2\|_\infty}{2}$

Proof:
Case 1: $w_3^2 - w_3^1 \leq \frac{\|v^1 - v^2\|_\infty}{2} \implies \|y^1 - y^2\|_\infty \leq \|w^1 - w^2\|_\infty \leq \frac{\|v^1 - v^2\|_\infty}{2}$.

Case 2: $w_3^2 - w_3^1 > \frac{\|v^1 - v^2\|_\infty}{2} \implies \frac{\|v^1 - v^2\|_\infty}{2} < w_3^2 - w_3^1 \leq \|v^1 - v^2\|_\infty \overset{(5.1)}{\implies}$ Subproblem $(P^2_{\varepsilon_1,\varepsilon_3})$ or $(P^3_{\varepsilon_2,\varepsilon_3})$ was called when the box $\mathrm{B}(w^1, w^2)$ was considered. Thus, due to the constraints in the lexicographic ε-constraint scalarization with lower bounds, it follows $\|y^1 - y^2\|_\infty \leq \frac{\|v^1 - v^2\|_\infty}{2}$ and the claim is proven. ♢

By induction, we get that an arbitrary box $\mathrm{B}(y^1, y^2)$ corresponding to a node in level $2k-1$ ($k \in \mathbb{N}$) fulfills

$$\|y^1 - y^2\|_\infty \leq \frac{L}{2^{k-1}} \,.$$

Hence, if $k \geq \log_2(L/\delta^C) + 1$, then $\|y^1 - y^2\|_\infty \leq \delta^C$ is induced and, therefore, each box at level $2 \cdot \lceil \log_2(L/\delta^C) \rceil + 1$ fulfills the accuracy condition which shows boundedness of the height of the tree.

Moreover, the tree can have at most

$$\sum_{i=0}^{2\cdot\lceil \log_2\left(L/\delta^C\right)\rceil} 7^i = \frac{7^{2\cdot\lceil \log_2\left(L/\delta^C\right)\rceil+1} - 1}{6} \in \mathcal{O}\left(\left(\frac{L}{\delta^C}\right)^{2\cdot\log_2(7)}\right)$$

nodes and, correspondingly, the number of iterations of the algorithm is bounded by the same term. □

[1] The level of the root node is 1, its children are on level 2 and so on.

Remark 5.16:
In Theorem 5.15, the norm $\|\cdot\|_\infty$ is considered. If a representative system with coverage error with respect to another p-norm $\|\cdot\|_p$, $1 \leq p < \infty$, is desired, the value δ^C has to be adapted in Algorithm 5.1, e.g., according to Lemma 4.19 from previous chapter. $\lhd$

Let *Rep* be the representative system obtained from Algorithm 5.1 and let Rep_N denote all points of *Rep* which are not dominated by any other point in this set. Obviously, deleting points can change the coverage error of the representative system. Yet, the following quality guarantee for the set Rep_N can be derived.

Corollary 5.17: *Let Rep be an arbitrary representative system with coverage error less than or equal to δ^C (w. r. t. $\|\cdot\|_\infty$) for the considered problem. Then, it is*

$$Y_N \subseteq \left(Rep_N - (\delta^C, \delta^C, \delta^C)^\top\right) + \mathbb{R}^3_{\geqq}$$

where Rep_N denotes all points of Rep which are not dominated by any other representative point.

Proof:
Let $y \in Y_N$. Due to the coverage property, there exists $z \in Rep$ with $\|y - z\|_\infty \leq \delta^C$, which implies $z - (\delta^C, \delta^C, \delta^C)^\top \leqq y$. If $z \notin Rep_N$, then there exists $\hat{z} \in Rep_N$ with $\hat{z} \leq z$ fulfilling $\hat{z} - (\delta^C, \delta^C, \delta^C)^\top \leqq y$. $\square$

5.5 Representation Error

In Remark 5.13, it is noted that the resulting representative system can contain dominated points. Therefore, the representation error

$$\max_{z \in Rep} \min_{y \in Y_N} \|z - y\|$$

(for $\|\cdot\| \coloneqq \|\cdot\|_p$ for some $1 \leq p \leq \infty$) can be positive. In this section, we are interested in how the representation error can be approximated or bounded from above for a given representative system *Rep* with coverage error less than or equal to δ^C (w.r.t. $\|\cdot\|$).

Definition 5.18:
For 3OP, let $(Rep, \mathcal{B})$ be the output of Algorithm 5.1. Let $z \in Rep$ and $\mathrm{B}(z) \subseteq \mathcal{B}$ be the set containing either the box for which z was computed or the corresponding child boxes contained in the corresponding quarter of the box for which z was computed. Then, we define $\mathcal{B}^z \coloneqq \{B \in \mathcal{B} : B \cap (z - \mathbb{R}^3_{\geq}) \neq \emptyset \text{ and } B \notin \mathrm{B}(z)\}$. If $\mathcal{B}^z \neq \emptyset$, we call z a *critical representative point*. $\diamond$

Remark 5.19:
For a box $\mathrm{B}(\ell, u)$ and a point $z \in \mathit{Rep}$, property $B \cap (z - \mathbb{R}^3_{\geq}) \neq \emptyset$ can be easily checked by checking $\ell \leq z$. Moreover, the sets $\mathrm{B}(z)$, $z \in \mathit{Rep}$, can be constructed during the algorithm, e.g., by saving to each upcoming box a reference to each representative point $z \in \mathit{Rep}$ for which this box is contained in the corresponding set $\mathrm{B}(z)$. These references have to be passed on from a box to its "child boxes". ◁

Lemma 5.20: *For* 3OP*, let* $(\mathit{Rep}, \mathcal{B})$ *be the output of Algorithm 5.1 and let* $z \in \mathit{Rep}$ *be some representative point. Then, we have:*

$$\mathcal{B}^z = \emptyset \Longrightarrow z \in Y_N$$

Proof:
Let $z \in \mathit{Rep}$ be a representative point with $\mathcal{B}^z = \emptyset$. Suppose $z \notin Y_N$. Then, due to Lemma 5.3, there must exist a $z^* \in Y_N$ with $z^* \leq z$. From Theorem 5.15, we know that $Y_N \subseteq \bigcup_{B \in \mathcal{B}} B$ such that there exists some $B \in \mathcal{B}$ with $z^* \in B \cap (z - \mathbb{R}^3_{\geq})$. Due to the structure of the lexicographic ε-constraint scalarizations with lower bounds, B cannot be contained in $\mathrm{B}(z)$. Thus, $B \in \mathcal{B}^z$ which is a contradiction. □

Proposition 5.21: *For* 3OP*, let* $(\mathit{Rep}, \mathcal{B})$ *be the output of Algorithm 5.1 and let there exist some critical representative point. Then, we have*

$$\max_{z \in \mathit{Rep}} \min_{y \in Y_N} \|z - y\| \leq \min \left\{ \max_{\substack{z \in \mathit{Rep} \\ \mathcal{B}^z \neq \emptyset}} \max_{\mathrm{B}(\ell,u) \in \mathcal{B}^z} \|\ell - z\|, \max_{\substack{z \in \mathit{Rep} \\ \mathcal{B}^z \neq \emptyset}} \max_{\hat{z} \in B_{\delta^C}^{\mathit{Rep}}(z - \mathbb{R}^3_{\geq})} \|z - \hat{z}\| + \delta^C \right\}$$

where $B_{\delta^C}^{\mathit{Rep}}(z - \mathbb{R}^3_{\geq}) := \left\{ y \in \mathit{Rep} : \exists \tilde{y} \in z - \mathbb{R}^3_{\geq} \text{ with } \|y - \tilde{y}\| \leq \delta^C \right\}$.

Proof:
Let $z \in \mathit{Rep}$ be an arbitrary critical representative point. Then, we immediately get

$$\min_{y \in Y_N} \|z - y\| \leq \max_{\mathrm{B}(\ell,u) \in \mathcal{B}^z} \|\ell - z\|$$

since for a box $\mathrm{B}(\ell, u)$ the distance $\|\ell - z\|$ is an upper bound for the distance of any point contained in $\mathrm{B}(\ell, u)$ and dominating z. Hence, the first upper bound is proven.
We define $y^* := \operatorname{arg\,min}_{y \in Y_N \cap (z - \mathbb{R}^3_{\geq})} \|z - y\|$ (assuming existence) and $z^{y^*} := \operatorname{arg\,min}_{\tilde{z} \in \mathit{Rep}} \|\tilde{z} - y^*\|$, then it follows

$$\begin{aligned} \min_{y \in Y_N} \|z - y\| &\leq \|z - y^*\| = \|z - y^* + z^{y^*} - z^{y^*}\| \\ &\leq \|z - z^{y^*}\| + \|y^* - z^{y^*}\| \leq \|z - z^{y^*}\| + \delta^C \\ &\leq \max_{\hat{z} \in B_{\delta^C}^{\mathit{Rep}}(z - \mathbb{R}^3_{\geq})} \|z - \hat{z}\| + \delta^C \,. \end{aligned}$$

□

We conclude this section with some remarks about Proposition 5.21 and the representation error in general.

Remark 5.22:
For $\|\cdot\| = \|\cdot\|_\infty$, the second upper bound in Proposition 5.21 can be substituted with

$$\max_{\substack{z \in Rep \\ \mathcal{B}^z \neq \emptyset}} \max_{\hat{z} \in Rep \cap \left(z + (\delta^C, \delta^C, \delta^C)^\top - \mathbb{R}^3_{\geq}\right)} \|z - \hat{z}\|_\infty + \delta^C \ . \qquad \lhd$$

Remark 5.23:
Another possibility to approximate the representation error is to use, for example, Benson's method (see, e.g., Ehrgott, 2005) for all critical representative points, which returns some nondominated point dominating the current critical representative point (if existent). In this case, the maximum over all distances between the critical representative points and the corresponding found nondominated points also serves as an upper bound of the representation error. $\lhd$

Remark 5.24:
Moreover, the representation error can be forced to be zero if some filtering procedure is applied after the calculation of the representative system, which identifies all dominated points and deletes them afterwards. This can also be done for example with the help of Benson's method which additionally gives a replacement point dominating the deleted point. However, this filtering technique can destroy the accuracy since points representing a box having a "nondominating part" can be deleted, but, taking replacement points into account, a result similar to Corollary 5.17 can still be proven. $\lhd$

Remark 5.25:
Another idea to keep the representation error low is to check in each iteration step, after we have obtained an outcome z^* in the current box, if there are other boxes $\mathrm{B}(\ell, u)$ with $z^* \leqq \ell$. In this case, we can delete the whole box. If $z^* \leqq u$, then we can cut off a special portion of $\mathrm{B}(\ell, u)$ and subdivide it into new boxes (more details in Section 5.8). $\lhd$

5.6 Volume-Based Box-Algorithm

As already mentioned before, also the volume can be incorporated as accuracy condition in the Box-Algorithm for three objectives. A straightforward pseudocode description of the *volume-based Box-Algorithm* for three objectives using only subproblem $(P^1_{\varepsilon_1,\varepsilon_2})$ can be found in the appendix in Algorithm A.1. Note that this subproblem can also be substituted with either $(P^2_{\varepsilon_1,\varepsilon_2})$ or $(P^3_{\varepsilon_1,\varepsilon_2})$ or, as in the coverage-based Box-Algorithm,

with adaptively choosing the subproblem. However, for the following results it does not matter which subproblem is used.

Note that for the volume-based Box-Algorithm, Proposition 5.5 and Lemma 5.14 still holds.

The volume reduction obtained by the subdivision corresponding to some lexicographic ε-constraint scalarization with lower bounds can be analyzed.

Lemma 5.26: *Let* $\mathrm{B}(\ell, u)$ *denote the current box in some iteration in Algorithm 5.1 or A.1 and let* $B^{j,i}$, $i = 1, \dots, 7$, *denote the subsequent subdivision after an optimal solution for the corresponding lexicographic* ε*-constraint scalarization with lower bounds* $(P^j_{\cdot,\cdot})$ *was obtained. Then, we get:*

(i) $\mathrm{Vol}(B^{j,i}) \leq \frac{1}{4} \cdot \mathrm{Vol}(\mathrm{B}(\ell, u)) \quad$ *for all* $\quad i = 1, \dots, 7$

(ii) $\sum_{i=1}^{7} \mathrm{Vol}(B^{j,i}) \leq \frac{3}{4} \cdot \mathrm{Vol}(\mathrm{B}(\ell, u))$

Moreover, for each of the quarters $Q^{j,1}$, $Q^{j,2}$, *and* $Q^{j,3}$, *we can find two pairs of boxes for which the combined volume fulfills formula (i).*

Proof:
We only prove the second statement (ii) since the first statement (i) follows directly from the construction of the boxes.

Without loss of generality, we assume that $(P^1_{\varepsilon_1,\varepsilon_2})$ was called with image z^* of an optimal solution.
The volume of the boxes $B^{\mathrm{lex}} := \mathrm{B}\Big(\ell, (\varepsilon_1, \varepsilon_2, z_3^*)^\top\Big)$ and $B^{\mathrm{dom}} := \mathrm{B}(z^*, u)$ are cut off fulfilling

$$\begin{aligned}
&\mathrm{Vol}(B^{\mathrm{lex}}) + \mathrm{Vol}(B^{\mathrm{dom}}) \\
&\qquad \geq \quad \mathrm{Vol}(B^{\mathrm{lex}}) + \mathrm{Vol}\Big(\mathrm{B}\Big((\varepsilon_1, \varepsilon_2, z_3^*)^\top, u\Big)\Big) \\
&\qquad = \quad \mathrm{Vol}\Big(B^{\mathrm{lex}} + (\varepsilon_1 - \ell_1, \varepsilon_2 - \ell_2, 0)^\top\Big) + \mathrm{Vol}\Big(\mathrm{B}\Big((\varepsilon_1, \varepsilon_2, z_3^*)^\top, u\Big)\Big) \\
&\qquad = \quad \mathrm{Vol}\Big(\mathrm{B}\Big((\varepsilon_1, \varepsilon_2, \ell_3)^\top, (u_1, u_2, z_3^*)^\top\Big)\Big) + \mathrm{Vol}\Big(\mathrm{B}\Big((\varepsilon_1, \varepsilon_2, z_3^*)^\top, u\Big)\Big) \\
&\qquad = \quad \mathrm{Vol}(Q^{1,3})
\end{aligned}$$

where $B^{\mathrm{lex}} + (\varepsilon_1 - \ell_1, \varepsilon_2 - \ell_2, 0)^\top$ denotes the box after shifting B^{lex} in direction $(\varepsilon_1 - \ell_1, \varepsilon_2 - \ell_2, 0)^\top$.

Hence, we get

$$\begin{aligned}
\sum_{i=1}^{7} \mathrm{Vol}(B^{1,i}) &= \mathrm{Vol}(\mathrm{B}(\ell, u)) - \Big(\mathrm{Vol}(B^{\mathrm{lex}}) + \mathrm{Vol}(B^{\mathrm{dom}})\Big) \\
&\leq \mathrm{Vol}(\mathrm{B}(\ell, u)) - \mathrm{Vol}(Q^{1,3}) = \frac{3}{4} \cdot \mathrm{Vol}(\mathrm{B}(\ell, u)) \,. \qquad \square
\end{aligned}$$

Definition 5.27 (Volume-Based Δ-Accuracy):
For 3OP, let a collection of boxes $\mathcal{B}$ containing the whole nondominated set and a representative system *Rep* be given. Then for some $\Delta > 0$, *Rep* fulfills the *volume-based* Δ-*accuracy* if for each box $B \in \mathcal{B}$ there exists a representative point $z^* \in Rep$ with $z^* \in B$ and it holds $\mathrm{Vol}(B) \leq \Delta$. ◇

Theorem 5.28: *Algorithm A.1 terminates in finitely many steps. It outputs a collection of boxes $\mathcal{B}$ containing all nondominated points. At termination, the representative system Rep fulfills the volume-based Δ-accuracy. More precisely, the algorithm performs at most $\mathcal{O}\left(\left(\frac{V_0}{\Delta}\right)^{\log_4(7)}\right)$ many iterations, where V_0 equals the volume $\mathrm{Vol}(\mathrm{B}(\ell^0, u^0))$ of the initial box $\mathrm{B}(\ell^0, u^0)$.*

Proof:
Similar to the proof of Theorem 5.15, we look at the tree rooted at the initial box $\mathrm{B}(\ell^0, u^0)$, for which each node corresponds to a box in some iteration.

Claim: At level k in the above defined tree, we only have nodes corresponding to boxes with volume less than or equal to $\frac{1}{4^{k-1}} \cdot V_0$.

Proof:
We show the claim by induction on k.
For $k = 1$, the claim is trivially fulfilled. Suppose, the induction hypothesis holds for level k, i.e., each box corresponding to a node at level k has a volume less than or equal to $\frac{1}{4^{k-1}} \cdot V_0$. Consider an arbitrary box B at level $k+1$. The corresponding node of this box has to be a child node of some node at level k corresponding to a box B'. From induction hypothesis, we get $\mathrm{Vol}(B') \leq \frac{1}{4^{k-1}} \cdot V_0$. Using the result from Lemma 5.26 (i), we immediately get

$$\mathrm{Vol}(B) \leq \frac{1}{4} \cdot \mathrm{Vol}(B') \leq \frac{1}{4} \cdot \frac{1}{4^{k-1}} \cdot V_0 = \frac{1}{4^k} \cdot V_0 \,. \qquad \diamondsuit$$

Hence, if $k \geq \log_{1/4}(\Delta/V_0) + 1$, all boxes at level k have a volume less than or equal to Δ, i.e., the height of the tree is bounded by $k^* := \left\lceil \log_{1/4}(\Delta/V_0) \right\rceil + 1$.

Moreover, the tree can have at most

$$\sum_{i=0}^{k^*-1} 7^i = \frac{7^{k^*} - 1}{6} \in \mathcal{O}\left(7^{k^*}\right) = \mathcal{O}\left(7^{\log_{1/4}(\Delta/V_0)}\right) = \mathcal{O}\left(\left(\frac{V_0}{\Delta}\right)^{\log_4(7)}\right)$$

nodes and, correspondingly, the number of iterations of the algorithm is bounded by the same term. □

5.7 Lower-Dimensional Update Step

Up to now, as mentioned in Remark 5.10, in the description of the coverage-based Box-Algorithm, we only treated the situation z^* lying in the interior of the corresponding quarter $Q^{1,1}$ and not on the boundary. Clearly, all considerations remain true if z^* is located on the boundary of the corresponding quarter. However, in this case, during the update step in Algorithm 5.1, a lot of redundant boxes are added to the list $\mathcal{S}$ and explored further. We consider an arbitrary iteration and assume here, w.l.o.g., that the first subproblem $(P^1_{\varepsilon_1,\varepsilon_2})$ applies.

Definition 5.29:
In an arbitrary iteration in Algorithm 5.1, after a new solution of $(P^1_{\varepsilon_1,\varepsilon_2})$ was found, a new box in $\mathrm{B}(\ell,u) \in \{B^{1,1},\dots,B^{1,7}\}$ is called a *real box* if $\ell < u$. In particular, a real box has always a positive volume.
If $\mathrm{B}(\ell,u)$ is no real box but there exist $i,j \in \{1,2,3\}$ with $i \neq j$, $\ell_i < u_i$ and $\ell_j < u_j$, then $\mathrm{B}(\ell,u)$ is called a *plane box*. If there exists a unique index $i \in \{1,2,3\}$ with $\ell_i < u_i$, then $\mathrm{B}(\ell,u)$ is called a *line box*. We call $\mathrm{B}(\ell,u)$ a *point box* if $\ell = u$. ◇

In this section, we investigate the situation in which problem $(P^1_{\varepsilon_1,\varepsilon_2})$ on a real box $\mathrm{B}(\ell,u)$ yields an outcome located on the boundary of $Q^{1,1}$. In this case, some boxes are created as no real boxes and we have to check which of them are redundant and which of them are necessary for the further exploration.

Observation 5.30:
In Algorithm 5.1, after a new solution of $(P^1_{\varepsilon_1,\varepsilon_2})$ on a real box $\mathrm{B}(\ell,u)$ with outcome z^* was found, the following cases for the seven different boxes $B^{1,1}$ to $B^{1,7}$ can occur:

- $B^{1,1}$ is no real box if ...
  - ... $\ell_1 = z_1^* \Longrightarrow B^{1,1}$ can be deleted due to dominance.
  - ... $z_2^* = \varepsilon_2 \Longrightarrow B^{1,1}$ can be deleted since it is contained in the box $B^{1,7}$.
  - ... $z_3^* = u_3 \Longrightarrow B^{1,1}$ can contain unknown nondominated points. Hence, if $\ell_1 < z_1^*$ and $z_2^* < \varepsilon_2$, then $B^{1,1}$ has to be added to the list of unexplored boxes (cf. Proposition 5.5).
- $B^{1,2}$ is no real box if ...
  - ... $\ell_2 = z_2^* \Longrightarrow B^{1,2}$ can be deleted since one part belongs to $B^{1,1}$ and the other part is dominated.
  - ... $z_3^* = u_3 \Longrightarrow B^{1,2}$ can be deleted due to the structure of the lexicographic problem $(P^1_{\varepsilon_1,\varepsilon_2})$.
- $B^{1,3}$ is no real box if ...

  - ... $\ell_3 = z_3^* \Longrightarrow B^{1,3}$ can be deleted since one part belongs to $B^{1,4}$ and the other part is dominated.

- $B^{1,4}$ is no real box if ...
  - ... $\ell_2 = z_2^* \Longrightarrow B^{1,4}$ can be deleted due to dominance.
  - ... $z_3^* = u_3 \Longrightarrow B^{1,4}$ can be deleted since it is contained in the box $B^{1,3}$.
- $B^{1,5}$ is no real box if ...
  - ... $\ell_3 = z_3^* \Longrightarrow B^{1,5}$ can be deleted due to dominance.
- $B^{1,6}$ is no real box if ...
  - ... $\ell_3 = z_3^* \Longrightarrow B^{1,6}$ can be deleted since one part belongs to $B^{1,7}$ and the other part is dominated.
- $B^{1,7}$ is no real box if ...
  - ... $\ell_1 = z_1^* \Longrightarrow B^{1,7}$ can be deleted due to dominance.
  - ... $z_3^* = u_3 \Longrightarrow B^{1,7}$ can be deleted since it is contained in the box $B^{1,6}$.

If for some $B^{1,i}$, $i \in \{1, \dots, 7\}$, no mentioned case applies, then $B^{1,i}$ describes a real box and has to be added to the list of unexplored boxes. ◁

The suggested modifications from this observation are straightforward incorporated in the pseudocode description of Algorithm 5.1. However, induced from this observation, we can come in the situation in which the selected current box $\mathrm{B}(\ell, u)$ is no real box but a plane box, a line box or just a point box. Note that this cannot happen in the volume-based Box-Algorithm since such boxes are not considered anymore due to the zero volume. Suppose, in Algorithm 5.1 in line 5, we have selected a plane box $\mathrm{B}(\ell, u)$, i.e., there exists some $i_0 \in \{1, 2, 3\}$ with $\ell_{i_0} = u_{i_0}$ and $\ell_i < u_i$ for all $i \in \{1, 2, 3\} \setminus \{i_0\}$. Due to the selection of the invoked lexicographic ε-constraint scalarization with lower bounds, we do not divide this box in the constant coordinate i_0. Without loss of generality, we assume i_0 equals 3 such that problem $(P^1_{\varepsilon_1,\varepsilon_2})$ is selected by the algorithm. Here, the first step of problem $(P^1_{\varepsilon_1,\varepsilon_2})$, i.e., minimization of f_3, can be omitted since all points in the current box share the same third coordinate $\ell_3 = u_3$. Then, after an outcome z^* was found, we get the following result.

Lemma 5.31: *Let* $\mathrm{B}(\ell, u)$ *denote the current plane box in some iteration in Algorithm 5.1. Suppose the lexicographic* ε*-constraint scalarization with lower bounds* $(P^1_{\varepsilon_1,\varepsilon_2})$ *applies and yields an optimal solution with outcome* $z^* \in Y$*. Then, it holds*

$$Y_N \cap \mathrm{B}(\ell, u) \subseteq B^{1,1} \cup B^{1,4} \cup B^{1,7} .$$

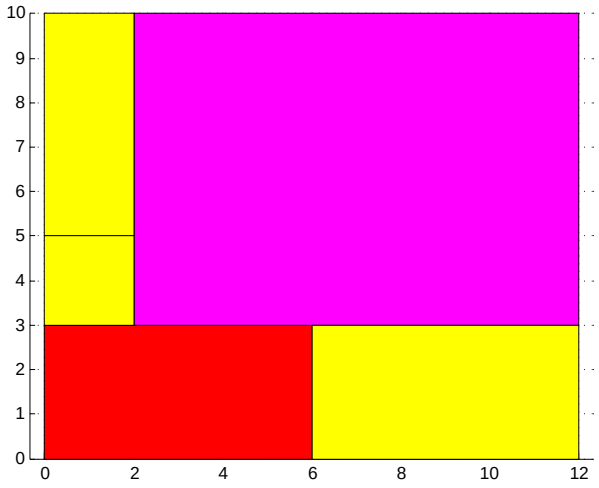


Figure 5.4: 2D-Separation in the Box-Algorithm with three objectives: Only the three yellow boxes remain. The magenta and red boxes can be excluded due to dominance information and the special structure of the lexicographic ε-constraint scalarization with lower bounds, respectively.

Proof:
Since $(P^1_{\varepsilon_1,\varepsilon_2})$ was called, we have $\ell_3 = u_3$ and $\ell_i < u_i$ for $i = 1, 2$. From Proposition 5.5 and the special structure of the lexicographic ε-constraint scalarization with lower bounds, there does not exist any nondominated point $y \in Y_N \setminus \{z^*\}$ contained in $\mathrm{B}(z^*, u) \cup \mathrm{B}\Big(\ell, (\varepsilon_1, z_2^*, \ell_3)^\top\Big)$. Hence, we get

$$Y_N \cap \mathrm{B}(\ell, u) \subseteq \{z^*\} \cup \mathrm{B}(\ell, u) \setminus \Big(\mathrm{B}(z^*, u) \cup \mathrm{B}\Big(\ell, (\varepsilon_1, z_2^*, \ell_3)^\top\Big)\Big) \subseteq B^{1,1} \cup B^{1,4} \cup B^{1,7} \,. \quad \square$$

Note that $Q^{1,3}$ is then completely dominated by $z^* \in Q^{1,1}$ and that in the definition of each box, the third coordinate of the lower left and upper right corner point is the same.

Example 5.32:
In an arbitrary iteration, let the current box with $\ell = 0$ and $u = (12, 10, 0)^\top$ be given. Suppose problem $(P^1_{\varepsilon_1,\varepsilon_2})$ with $\varepsilon_1 = 6$ and $\varepsilon_2 = 5$ returns an optimal solution with outcome $z^* = (2, 3, 0)^\top$. Then, after the subdivision, only three non-redundant boxes remain

$$\begin{aligned} B^{1,1} &= \mathrm{B}\Big((0,3,0)^\top, (2,5,0)^\top\Big) && \text{for } Q^{1,1}, \\ B^{1,4} &= \mathrm{B}\Big((6,0,0)^\top, (12,3,0)^\top\Big) && \text{for } Q^{1,2} \text{ and} \\ B^{1,7} &= \mathrm{B}\Big((0,5,0)^\top, (2,10,0)^\top\Big) && \text{for } Q^{1,4}. \end{aligned}$$

The remaining boxes and the excluded region are depicted in Figure 5.4. $\triangleleft$

If the current box is a plane box and, w.l.o.g., problem $(P^1_{\varepsilon_1,\varepsilon_2})$ applies, the found outcome z^* can also be located on the boundary of $Q^{1,1}$. Similar to Observation 5.30, this can lead to redundant boxes since some boxes are created as no plane boxes.

Observation 5.33:
In Algorithm 5.1, after a new solution of $(P^1_{\varepsilon_1,\varepsilon_2})$ on a plane box $\mathrm{B}(\ell, u)$ with outcome z^* was found, the following cases for the three different boxes $B^{1,1}$, $B^{1,4}$ and $B^{1,7}$ can occur:

- $B^{1,1}$ is no plane box if ...
  - ... $\ell_1 = z_1^* \Longrightarrow B^{1,1}$ can be deleted due to dominance.
  - ... $z_2^* = \varepsilon_2 \Longrightarrow B^{1,1}$ can be deleted due to the structure of the lexicographic problem $(P^1_{\varepsilon_1,\varepsilon_2})$.
- $B^{1,4}$ is no plane box if ...
  - ... $\ell_2 = z_2^* \Longrightarrow B^{1,4}$ can be deleted due to dominance.
- $B^{1,7}$ is no plane box if ...
  - ... $\ell_1 = z_1^* \Longrightarrow B^{1,7}$ can be deleted due to dominance.

If for some $B^{1,i}$, $i \in \{1, 4, 7\}$, no mentioned case applies, then $B^{1,i}$ describes a plane box and has to be added to the list of unexplored boxes. ◁

Remark 5.34:
By the above considerations, if the initial box defines a real or a plane box, a line box as well as a point box cannot occur in the selection step since they would always be redundant. However, for completeness, we shorty discuss the case of a line box since a line box can occur if another subdivision is used or if the initial box defines a line box. The investigation of a point box is trivial and is not considered here.

If our current box is a line box, then we can immediately call the completion step (lines 8 to 11 in Algorithm 5.1) and it is easy to see that, after an optimal solution was found, there is no need to construct a new subdivision since, in this case, we can exclude the whole line box from the further investigation. ◁

5.8 Extensions of the Box-Algorithm

The basic concept of the coverage-based Box-Algorithm can be equipped with several improvements and extensions which may be reasonable in appropriate settings and circumstances. The following proposals are thus facultative.

An obvious extension, which is also used in other methods to handle each objective unbiased, is the measurement of the distances in a normalized fashion, i.e., incorporating the range of each objective function.

Algorithm 5.2 Deletion of dominated boxes - Part 1

Input: An outcome z^*.
1: **for** $\mathrm{B}(\ell, u) \in \mathcal{B} \cup S$ **do**
2: **if** $z^* \leqq \ell$ **then**
3: Delete $\mathrm{B}(\ell, u)$

If our optimization problem 3OP has three integer-valued objective functions, in each iteration, the unexplored region can be shrunk even further using the integrality property of the outcomes.

5.8.1 Incorporating Dominance during the Algorithm

The Box-Algorithm uses the concept of dominance only locally: after an optimal solution of the lexicographic ε-constraint scalarization with lower bounds was found, the part of the current box being dominated by this solution is neglected from further consideration. However, this solution also dominates points outside the current box. This observation can help to reduce the number of boxes significantly without losing performance.

Hence, the step in Algorithm 5.1 is considered in which a new feasible outcome z^* in the currently considered box is found (cf. line 20). It is natural to use this solution to define the notion of dominated boxes (see also Barichard and Hao, 2003).

Definition 5.35 (Dominated Box):
In an arbitrary iteration in Algorithm 5.1, let S be the set of stored unexplored boxes which are not yet considered and $\mathcal{B}$ be the set of finished boxes fulfilling the accuracy condition. Then, after an optimal solution with image $z^* \in Y$ in line 19 or 8 was found, a box $\mathrm{B}(\ell, u) \in \mathcal{B} \cup S$ is called *dominated* by z^* if $z^* \leqq \ell$. ◇

Hence, a dominated box cannot contain points of interest since all possible points in the box are dominated by the found solution. Then, right after the outcome z^* was found, before lines 21 and 10, we invoke Algorithm 5.2 as a subroutine which considers the lists S of unexplored boxes and $\mathcal{B}$ of finished boxes and deletes all boxes dominated by z^*. Additionally, before adding a new box B to $\mathcal{B}$ or S in lines 11, 23 and 26, Algorithm 5.3 is invoked as a subroutine which first investigates the set *Rep* and checks if the new box is already dominated.

On the other hand, during the algorithm, also partially dominated boxes can be obtained.

Definition 5.36 (Partially Dominated Box):
In an arbitrary iteration in Algorithm 5.1, let S be the set of stored unexplored boxes which are not yet considered and $\mathcal{B}$ be the set of finished boxes fulfilling the accuracy

Algorithm 5.3 Deletion of dominated boxes - Part 2

Input: A box $\mathrm{B}(\ell, u)$.
1: **for** $z \in \mathit{Rep}$ **do**
2: **if** $z^* \leqq \ell$ **then**
3: Delete $\mathrm{B}(\ell, u)$

condition. Then, after an optimal solution with image $z^* \in Y$ in line 19 or 8 was found, a box $\mathrm{B}(\ell, u) \in \mathcal{B} \cup S$ is called *partially dominated* by z^* if $z^* \not\leqq \ell$ but $z^* \leqq u$. $\diamond$

In this case, a specific subset of a partially dominated box $\mathrm{B}(\ell, u)$ can be deleted, namely the box

$$\mathrm{B}\Big(z^{*,proj} \coloneqq (\max\{\ell_1, z_1^*\},\ \max\{\ell_2, z_2^*\},\ \max\{\ell_3, z_3^*\})^\top, u\Big)$$

where the lower left corner point $z^{*,proj}$ is called the $\mathrm{B}(\ell, u)$-*projection* of z^*.

Partial dominance is incorporated in Algorithm 5.1 analogously to dominance with appropriately modified subroutines. However, for practical reasons, to avoid cutting off only very small portions, some threshold for partial dominance (e.g., 20 %) can be defined for which it is worth to delete this portion from further considerations. Furthermore, before substituting, we should also perform a feasible-test of the partially dominated box to reduce future computation time.

Obviously, there may exist boxes which are completely infeasible and in the coverage-based Box-Algorithm, we have to put in a lot of effort to prove this (in most cases, we cannot even prove this but we shrink this infeasible box until the given tolerance is reached). Therefore, it is recommended to perform some feasible-test after we have identified a partially dominated box and after we have encountered an infeasible subproblem, i.e., we have proven infeasibility for a quarter of the current box. For this feasible-test, e.g., a lexicographic ε-constraint scalarization with lower bounds can be used for which we do not restrict the feasible domain to the corresponding quarter but rather to the whole box (see also the completion step in lines 8 to 11 in Algorithm 5.1).

If we do not cut off all partially dominated boxes, we can come in the following situation. We look at a box $\mathrm{B}(\ell, u)$ which is partially dominated by some point $p \in \mathit{Rep}$. We solve the corresponding subproblem and obtain a new outcome z^* lying in the partially dominated region, i.e., which is dominated by p. In this case, we can use point p instead of point z^* to define the boxes for the new subdivision. Hence, we have to look at the $\mathrm{B}(\ell, u)$-projection p^{proj} of p which also fulfills $p^{proj} \leqq z^*$. Similar to Proposition 5.5, we immediately get the following result.

Proposition 5.37: *Let z^* be the outcome of an optimal solution of $(P^1_{\varepsilon_1,\varepsilon_2})$ which is dominated by some point $p \in Y$. Let p^{proj} be the $\mathrm{B}(\ell,u)$-projection of p. Then, there does not exist a $y \in Y_N \setminus \{p\}$ such that*

$$y \in \mathrm{B}(p^{proj},u) \cup \mathrm{B}\Big(\ell,(\varepsilon_1,\varepsilon_2,z_3^*)^\top\Big) \setminus \mathrm{B}\Big((\ell_1,z_2^*,z_3^*)^\top,(z_1^*,\varepsilon_2,z_3^*)^\top\Big) .$$

Similar to Definition 5.8, for $(P^1_{\varepsilon_1,\varepsilon_2})$, this results in the following subdivision of the current box $\mathrm{B}(\ell,u)$:

$$B^{1,1} \coloneqq \mathrm{B}\Big((\ell_1,p_2^{proj},z_3^*)^\top,(p_1^{proj},\varepsilon_2,u_3)^\top\Big), \quad B^{1,2} \coloneqq \mathrm{B}\Big((\ell_1,\ell_2,z_3^*)^\top,(\varepsilon_1,p_2^{proj},u_3)^\top\Big)$$

for $Q^{1,1}$,

$$B^{1,3} \coloneqq \mathrm{B}\Big((\varepsilon_1,\ell_2,\ell_3)^\top,(u_1,\varepsilon_2,p_3^{proj})^\top\Big), \quad B^{1,4} \coloneqq \mathrm{B}\Big((\varepsilon_1,\ell_2,p_3^{proj})^\top,(u_1,p_2^{proj},u_3)^\top\Big)$$

for $Q^{1,2}$,

$$B^{1,5} \coloneqq \mathrm{B}\Big((\varepsilon_1,\varepsilon_2,\ell_3)^\top,(u_1,u_2,p_3^{proj})^\top\Big)$$

for $Q^{1,3}$ and

$$B^{1,6} \coloneqq \mathrm{B}\Big((\ell_1,\varepsilon_2,\ell_3)^\top,(\varepsilon_1,u_2,p_3^{proj})^\top\Big), \quad B^{1,7} \coloneqq \mathrm{B}\Big((\ell_1,\varepsilon_2,p_3^{proj})^\top,(p_1^{proj},u_2,u_3)^\top\Big)$$

for $Q^{1,4}$.

Note that the subdivision from Definition 5.8 defines a superset of this subdivision.

5.8.2 Other Quality Measures

Another quality measure is the concept of uniformity (cf. Definition 2.9). The Box-Algorithm can be modified to address this measure as well. To this end, let $0 < \delta^U \leq \delta^C$ be given. The aim is a representative system *Rep* fulfilling the coverage property and

$$\min_{\substack{y,\hat{y}\in Rep\\ y\neq\hat{y}}} \|y-\hat{y}\|_\infty \geq \delta^U . \tag{5.2}$$

It is assumed that $\delta^U \leq \delta^C$ since otherwise, the coverage property could be violated. Again, the step in the algorithm is considered in which a new feasible outcome z^* is obtained (cf. lines 20 and 9 in Algorithm 5.1). Due to (5.2), the region $\{z \in \mathbb{R}^3 : \|z-z^*\|_\infty \leq \delta^U\}$ defining the box

$$\mathrm{B}\Big((z_1^*-\delta^U,z_2^*-\delta^U,z_3^*-\delta^U)^\top,(z_1^*+\delta^U,z_2^*+\delta^U,z_3^*+\delta^U)^\top\Big) \tag{5.3}$$

should be deleted from further investigation.

This box has now to be incorporated in the current update step in a straightforward way which results in another subdivision. Furthermore, the lists of all unexplored boxes S and finished boxes $\mathcal{B}$ have to be investigated and the boxes overlapping with box (5.3) have to be updated appropriately.

If the quality measure "cardinality" should be incorporated, the algorithm can be aborted if the desired cardinality is reached, i.e., an additional termination condition corresponding to the desired cardinality has to be added to line 4 in Algorithm 5.1. Obviously, this can violate the desired coverage property since the different quality measures tend to be contradicting. In this case, however, the coverage error of the currently obtained representative system can be computed by looking at the last obtained subdivision (see also Corollary 5.38 below).

5.8.3 Selection Rules

In the coverage-based Box-Algorithm, the next box for investigation is selected due to some selection rule (see line 5 in Algorithm 5.1). As mentioned at the beginning of Section 5.3, for instance, the box with the largest corner point distance can be selected, i.e., in a practical implementation, the set S can be implemented as a list for which the elements are ordered according to their corner point distances. We call this selection rule the *max-dist selection rule*. The max-dist selection rule is helpful if the algorithm should be aborted prematurely (e.g., by aiming at the cardinality of the representative system) since, in this case, a guaranteed bound for the obtained coverage error can be stated.

Corollary 5.38: *For Algorithm 5.1, let the subroutine* SelectBox() *always select the box with largest corner point distance (max-dist selection rule). Suppose the algorithm is aborted prematurely after* $\Gamma \geq 1$ *iterations and for all remaining boxes* $B \in S$*, we additionally invoke the completion step, i.e., line 8 to 11 from Algorithm 5.1. Then, the algorithm outputs a collection of boxes* $\mathcal{B}$ *containing all nondominated points. At termination, the representative system Rep has a coverage error of at most* $L \cdot 2^{-\lfloor(\log_7(6\Gamma+1))/2\rfloor}$ *(w.r.t.* $\|\cdot\|_\infty$*), where* L *equals the corner point distance* $\|\ell^0 - u^0\|_\infty$ *of the initial box* $\mathrm{B}(\ell^0, u^0)$*.*

Proof:
Due to the construction of the boxes, a subdivision is obtained which contains Y_N at each state of the algorithm (see Lemma 5.14). Analogous to the proof of Theorem 5.15, we consider the tree for which each node corresponds to a box in some iteration. Due to the utilized selection rule and the proof of Theorem 5.15, we have: If the Box-Algorithm selects a box from level $2k+1$ or higher, $k \in \mathbb{N}_0$, then all remaining

boxes (including the selected box) have a corner point distance of at most $L/2^k$. From level 1 to level $2k$, the number of nodes in the tree is bounded by

$$\sum_{i=0}^{2k-1} 7^i = \frac{7^{2k}-1}{6} .$$

Let $k^* \in \mathbb{N}_0$ be the largest number fulfilling

$$\frac{7^{2k^*}-1}{6} \leq \Gamma .$$

This means, in some iteration less than or equal to Γ, the algorithm has already selected a box corresponding to a node at a level $2k^*+1$ or higher, or, without the premature abortion, in iteration $\Gamma+1$, the algorithm would select a box corresponding to a node at a level $2k^*+1$ or higher. Thus, at termination, all boxes have a corner point distance of at most

$$\frac{L}{2^{k^*}} = L \cdot 2^{-\lfloor (\log_7(6\Gamma+1))/2 \rfloor} . \qquad \square$$

However, selecting the box with maximal corner point distance can lead to the generation of a lot of dominated solutions since a found solution can be dominated by a new solution which is found in a later iteration. In the following, we outline the idea of a new selection rule, called *nondominated selection rule*, for which a clever sorting of the list S of unexplored boxes is chosen in such a way that the output *Rep* contains only few or even no points which are dominated by other points in *Rep*. We assume here the incorporation of the deletion of dominated as well as partially dominated boxes as described in Section 5.8.1. Furthermore, we assume that SelectBox(S) is implemented in such a way that it always selects the first element of the given list S.

Consider an arbitrary iteration of Algorithm 5.1, w.l.o.g. after an outcome z^* was found by solving problem ($P^1_{\varepsilon_1,\varepsilon_2}$) (see line 19). Let $B^{1,1}$ to $B^{1,7}$ denote the new boxes. In the modified algorithm, a new box is not added to the list S by looking at its corner point distance but with respect to some dominance relation. No point in the interior of the boxes $B^{1,2}$, $B^{1,3}$ and $B^{1,6}$ can be dominated by a point in any other of the remaining six boxes. Hence, these three boxes (w.l.o.g. in the given ordering) are added at the beginning of the list S, which defines the ordering of the next selected boxes. Thereafter, box $B^{1,1}$ is added since all points in its interior can only be dominated by points from $B^{1,2}$, which is then completely explored (up to the desired precision) when box $B^{1,1}$ is considered. With the same argumentation the boxes $B^{1,4}$, $B^{1,7}$ and $B^{1,5}$ are added after the position of $B^{1,1}$. Then, the algorithm first explores box $B^{1,2}$ completely, since also the boxes from the new partition are added at the beginning of the list S; and, thereafter, the algorithm goes over to explore box $B^{1,3}$ and so on.

Assuming that the insertion in lines 23 and 26 in Algorithm 5.1 is implemented in such a way that the new element is always added to the beginning of the corresponding list S, we change the for-loop in line 22 to

$$\textbf{for } i = 5, 7, 4, 1, 6, 3, 2 \textbf{ do}$$

such that the above derived ordering is obtained. Moreover, with the same argumentation, we substitute the for-loop in line 25 with

$$\textbf{for } i = 3, 2, 4 \textbf{ do} .$$

Analogous arguments show that both substitutions are also valid if problem $(P^2_{\varepsilon_1,\varepsilon_3})$ or $(P^3_{\varepsilon_2,\varepsilon_2})$ was called.

A remaining issue of the nondominated selection rule are the outcomes lying on the boundary of some box since such a point can belong to more than one box and, thus, such found outcomes can be dominated in a later iteration step. However, we assume here that such situations do not occur during the run of the algorithm. In practice, this issue can be handled numerically if some small predefined box-gap between two neighbored boxes is used (cf. Appendix A.2).

To conclude, using the nondominated selection rule, each found point cannot be dominated by a point from a later iteration step. However, a new point can still be dominated by a former found point. If the threshold for partial dominance is chosen sufficiently small, the modified algorithm returns a representative system for which only few or even no points in the set *Rep* are dominated by any other point in *Rep*. Hence, with appropriately chosen threshold, the nondominated selection rule is able to combine the coverage property with the quality guarantee from Corollary 5.17.

5.9 Theoretical Scalarization Problem

In this section, we derive a theoretical scalarization problem which gives us a similar *correctness property* as for the case with two objectives in Theorem 4.8.

In the coverage-based Box-Algorithm in Section 5.3, for the used scalarization problems, we have to use lower bounds in the constraints to ensure that the new point lies in the current box. However, we have seen in Remark 5.13 that the new point does not have to be nondominated since it can be dominated from a point in another box.

Hence, for a given box $\mathrm{B}(\ell, u)$ with $Y \cap (u - \mathbb{R}^3_{\geqq})$ compact (cf. Lemma 5.3) and $\varepsilon_i = \frac{\ell_i + u_i}{2}$ for $i = 1, 2$, we desire a more suitable scalarization method which can be used either

- to find a point $x^* \in X_E$ with $f(x^*) \in \mathrm{B}(\ell, u)^{(\varepsilon_1, \varepsilon_2, u_3)}$ or
- to prove that there does not exist such a point.

In other words, it is desirable to solve the following problem efficiently:

$$\begin{aligned}(\hat{P}^1_{\varepsilon_1,\varepsilon_2}) \quad & \operatorname{lex\,min}\ (f_3(x),\ f_2(x),\ f_1(x)) \\ & \text{s.t.}\ \ x \in X_E \\ & \qquad f(x) \in \mathrm{B}(\ell,u)^{(\varepsilon_1,\varepsilon_2,u_3)}\end{aligned}$$

This formulation $(\hat{P}^1_{\varepsilon_1,\varepsilon_2})$ has the disadvantage that the efficient set is assumed to be known in advance, which contradicts the fact that a representative system of Y_N should be computed.

However, ignoring the lexicographic objective function, we can express the condition $x \in X_E$ implicitly with the help of a bilevel optimization problem[2], in which the lower-level problem checks whether the current point is nondominated. This check is based on Benson's method (cf. Ehrgott, 2005). A similar idea for checking efficiency of a proper face for linear problems was used in Sayın (1996). Here, this idea is incorporated for general problems in the lower-level of the following scalarization problem.

$$\begin{aligned}(\hat{P}^{1,bilevel}_{\varepsilon_1,\varepsilon_2}) \quad & \min_{x,z^x}\ z^x && \text{(5.4a)}\\ & \text{s.t.}\ \ x \in X && \text{(5.4b)}\\ & \qquad f(x) \in \mathrm{B}(\ell,u)^{(\varepsilon_1,\varepsilon_2,u_3)} && \text{(5.4c)}\\ & \qquad z^x \coloneqq \max_{\tilde{x}} \sum_{i=1}^{3} f_i(x) - f_i(\tilde{x}) && \text{(5.4d)}\\ & \qquad\quad \text{s.t.}\ \ \tilde{x} \in X && \text{(5.4e)}\\ & \qquad\qquad f(\tilde{x}) \leqq f(x) && \text{(5.4f)}\end{aligned}$$

$(\hat{P}^{1,bilevel}_{\varepsilon_1,\varepsilon_2})$ defines a so-called *min-max problem*, a special case of a bilevel optimization problem for which the lower-level and upper-level objective functions coincide.

In this context, we say 3OP is *discrete*, if it has three integer-valued objective functions, i.e., $f : X \longrightarrow \mathbb{Z}^3$. Hence, if we additionally assume that 3OP is discrete, we can also

[2] See, e.g., Dempe (2003) and Colson et al. (2007) for an overview of bilevel optimization.

add implicitly the minimization of the third objective function:

$$
\begin{aligned}
(\hat{P}^{1,bilevel}_{IP,\varepsilon_1,\varepsilon_2}) \quad \min_{x,z^x} \quad & z^x + \delta \cdot f_3(x) && \text{(5.5a)}\\
\text{s.t.} \quad & x \in X && \text{(5.5b)}\\
& f(x) \in \mathrm{B}(\ell,u)^{(\varepsilon_1,\varepsilon_2,u_3)} && \text{(5.5c)}\\
& z^x \coloneqq \max_{\tilde{x}} \sum_{i=1}^{3} f_i(x) - f_i(\tilde{x}) && \text{(5.5d)}\\
& \text{s.t.} \quad \tilde{x} \in X && \text{(5.5e)}\\
& \qquad f(\tilde{x}) \leqq f(x) && \text{(5.5f)}
\end{aligned}
$$

where $0 < \delta < \frac{1}{2 \cdot \max\{|\ell_3|,|u_3|\}}$ and $\ell_3 < u_3$.
We first note some obvious but nice properties of these scalarization problems.

Observation 5.39:

- If X is induced by linear constraints and f is a linear function, then all constraints and the objective function remain linear but now with bilevel structure.
- All terms of the sum in (5.4d) and (5.5d) are non-negative.
- The absolute value of the second summand $\delta \cdot f_3(x)$ in the objective function (5.5a) fulfills:

$$|\delta \cdot f_3(x)| = \delta \cdot |f_3(x)| \leq \delta \cdot \max\{|\ell_3|, |u_3|\} < \frac{1}{2}$$

- For a feasible solution $\hat{x}$ of $(\hat{P}^{1,bilevel}_{IP,\varepsilon_1,\varepsilon_2})$, it holds $z^{\hat{x}} \in \mathbb{N}_0$ since for all $x \in X$ it holds $f(x) \in \mathbb{Z}^3$ and $z^x \geq 0$. ◁

For these two scalarization methods, we are now able to give a similar result as in Theorem 4.8.

Theorem 5.40 (Correctness Property): *$(\hat{P}^{1,bilevel}_{\varepsilon_1,\varepsilon_2})$ is infeasible or it exists an optimal solution $x^* \in X$ with $z^{x^*} > 0$ if and only if $\mathrm{B}(\ell,u)^{(\varepsilon_1,\varepsilon_2,u_3)} \cap Y_N = \emptyset$.*
If 3OP *is discrete, the statement also holds for $(\hat{P}^{1,bilevel}_{IP,\varepsilon_1,\varepsilon_2})$.*

Proof:
We only consider the case for $(\hat{P}^{1,bilevel}_{IP,\varepsilon_1,\varepsilon_2})$ since the proof for $(\hat{P}^{1,bilevel}_{\varepsilon_1,\varepsilon_2})$ works analogously.
"$\Rightarrow$":
Suppose there exists some $x^* \in X_E$ with $f(x^*) \in \mathrm{B}(\ell,u)^{(\varepsilon_1,\varepsilon_2,u_3)} \cap Y_N$. Then,

$(x, \tilde{x}, z^x) := (x^*, x^*, 0)$ is a feasible solution for $(\hat{P}^{1,bilevel}_{IP,\varepsilon_1,\varepsilon_2})$ with $z^{x^*} = 0$. Furthermore, no other feasible solution $\hat{x} \in X$ with $z^{\hat{x}} > 0$ (i.e., $z^{\hat{x}} \geq 1$) can obtain a better objective value since the second term of the objective function always fulfills $\delta \cdot f_3(\hat{x}) \in \left(-\frac{1}{2}, \frac{1}{2}\right)$.

"$\Leftarrow$":
Let $\mathrm{B}(\ell, u)^{(\varepsilon_1,\varepsilon_2,u_3)} \cap Y_N = \emptyset$.
Case 1: $\mathrm{B}(\ell, u)^{(\varepsilon_1,\varepsilon_2,u_3)} \cap Y = \emptyset$. Then, $(\hat{P}^{1,bilevel}_{IP,\varepsilon_1,\varepsilon_2})$ is infeasible.
Case 2: Let $x^* \in X$ with $f(x^*) \in \mathrm{B}(\ell, u)^{(\varepsilon_1,\varepsilon_2,u_3)} \cap Y \setminus Y_N$ be arbitrary. Then, there exists some $x' \in X$ with $f(x') \leq f(x^*)$. Thus, the associated z^{x^*} fulfills $z^{x^*} \geq 1 > 0$. □

Corollary 5.41: $\mathrm{B}(\ell, u)^{(\varepsilon_1,\varepsilon_2,u_3)} \cap Y_N \neq \emptyset$ *if and only if* $(\hat{P}^{1,bilevel}_{\varepsilon_1,\varepsilon_2})$ *is feasible and all optimal solutions* $x^* \in X$ *have* $z^{x^*} = 0$.
If 3OP *is discrete, the statement also holds for* $(\hat{P}^{1,bilevel}_{IP,\varepsilon_1,\varepsilon_2})$.

Theorem 5.42: *Let* 3OP *be discrete. If there exists an optimal solution* $x^* \in X$ *for* $(\hat{P}^{1,bilevel}_{IP,\varepsilon_1,\varepsilon_2})$ *with* $z^{x^*} = 0$*, then this solution fulfills:*

(i) $x^* \in X_E$

(ii) $x^* = \arg\min\left\{ f_3(x) : x \in X_E \text{ and } f(x) \in \mathrm{B}(\ell, u)^{(\varepsilon_1,\varepsilon_2,u_3)} \right\}$

In particular, if 3OP *is not discrete, (i) also holds for an optimal solution* x^* *for* $(\hat{P}^{1,bilevel}_{\varepsilon_1,\varepsilon_2})$.

Proof:
Let x^* be an optimal solution for $(\hat{P}^{1,bilevel}_{IP,\varepsilon_1,\varepsilon_2})$ (or $(\hat{P}^{1,bilevel}_{\varepsilon_1,\varepsilon_2})$, respectively) with $z^{x^*} = 0$.
(i):
Suppose $x^* \notin X_E$. Then, there exists some $x' \in X$ with $f(x') \leq f(x^*)$. Thus, the associated z^{x^*} fulfills $z^{x^*} > 0$ which contradicts the assumption.

(ii):
Suppose there exists a $x' \in X_E$ with $f(x') \in \mathrm{B}(\ell, u)^{(\varepsilon_1,\varepsilon_2,u_3)}$ and $f_3(x') < f_3(x^*)$. Thus, the associated $z^{x'}$ fulfills $z^{x'} = 0$ and it holds

$$\delta \cdot f_3(x') < \delta \cdot f_3(x^*)$$

which contradicts optimality of x^* for $(\hat{P}^{1,bilevel}_{IP,\varepsilon_1,\varepsilon_2})$. □

We conclude this section with a remark about the practicability of the scalarization problems $(\hat{P}^{1,bilevel}_{\varepsilon_1,\varepsilon_2})$ and $(\hat{P}^{1,bilevel}_{IP,\varepsilon_1,\varepsilon_2})$.

Remark 5.43:
Due to its bilevel structure, problem $(\hat{P}^{1,bilevel}_{\varepsilon_1,\varepsilon_2})$ is hard to solve. Since such problems should be solved repeatedly in each iteration in the Box-Algorithm for three objectives,

we decided to investigate the simpler subproblem ($P^1_{\varepsilon_1,\varepsilon_2}$) further, which can be solved using standard black-box solver (like CPLEX). However, in the literature, there exist some methods to tackle this min-max problem for special assumptions on the constraints and objectives (see, e.g., Falk, 1973; Shimizu and Aiyoshi, 1981; Belenky, 1997), but, through the bilevel structure and the doubled constraints due to the introduction of the variable $\tilde{x}$, no method with such an efficiency we need. For instance, Falk (1973) considers the case with linear constraints and continuous variables

$$\max_x \min_y \{c^\top x + d^\top x : Ax + By \leqq b,\ x, y \geqq 0\} . \tag{5.6}$$

He first showed that an optimal solution must be located at a vertex of the corresponding polyhedron and developed a branch-and-bound algorithm solving this problem. Nevertheless, the algorithm is very time-consuming since the branching is done directly on the variables forcing them to be not in the basis. Hence, in the branch-and-bound scheme, for each father node, m many child nodes are obtained, where m denotes the rank of the constraint matrix. Since in the special subproblem ($\hat{P}^{1,bilevel}_{\varepsilon_1,\varepsilon_2}$), the constraints are doubled, this method and also other methods are nice from a theoretical point of view but are expected to be not efficient.

Despite the nice theoretical results for problem ($\hat{P}^{1,bilevel}_{IP,\varepsilon_1,\varepsilon_2}$), we are not aware of a method solving such integral bilevel optimization problems in reasonable time. Theoretically, one can ask if methods as, e.g., the approach of Falk (1973) can be used to develop a branch-and-bound algorithm for bilevel integer linear problems. However, the linear relaxation (5.6) of the following integer linear max-min problem

$$\max_x \min_y \{c^\top x + d^\top x : Ax + By \leq b,\ x, y \in \mathbb{N}_0\} \tag{5.7}$$

cannot serve neither as lower nor as upper bound for the integer case problem, which can be observed in the following example:

We consider two different max-min problems. The first problem is given by

$$\max_x \min_y \{y : 2y - x \geq 0,\ 2y + x \leq 6,\ x, y \in \mathbb{N}_0\} . \tag{5.8}$$

The optimal solution is attained at $(x, y) = (2, 1)$ with objective value 1. The optimal solution of the corresponding linear problem, where the constraints $x, y \in \mathbb{N}_0$ are substituted with $x, y \geq 0$, is attained at $(x, y) = (3, 1.5)$ with objective value 1.5.

The second problem is given by

$$\max_x \min_y \{y : 2y - x \geq 0,\ 2y + x \leq 7,\ x \leq 3,\ x, y \in \mathbb{N}_0\} . \tag{5.9}$$

Here, the optimal solution is attained at $(x, y) = (3, 2)$ with objective value 2. The optimal solution of the corresponding linear problem, where the constraints $x, y \in \mathbb{N}_0$ are substituted with $x, y \geq 0$, is again attained at $(x, y) = (3, 1.5)$ with objective value 1.5. ◁

5.10 Application to the Bus Evacuation Problem

In this section, in the context of the project DSS_Evac_Logistics, we apply the coverage-based Box-Algorithm for three objectives to the (mixed-)integer programming (IP) formulation for the *Bus Evacuation Problem (BEP)*, which was also developed for the project DSS_Evac_Logistics (details can be found in Goerigk et al. (2013)).

Goerigk et al. (2013) formulated the following IP:

$$
\begin{aligned}
&\min\ t_{\max} \\
&\text{s. t.} \\
&t_{\max} \geq \sum_{r\in[R]} \left(t_{to}^{br} + t_{back}^{br}\right) + \sum_{i\in[S],j\in[T]} d_i^{start} x_{ij}^{b1} && \forall b \in [B] \\
&t_{to}^{br} = \sum_{i\in[S],j\in[T]} d_{ij}^{ST} x_{ij}^{br} && \forall b \in [B], r \in [R] \\
&t_{back}^{br} \geq d_{ji}^{TS} \left(\sum_{k\in[S]} x_{kj}^{br} + \sum_{\ell\in[T]} x_{i\ell}^{b,r+1} - 1 \right) && \forall b \in [B], i \in [S], j \in [T], r \in [R-1] \\
&\sum_{i\in[S],j\in[T]} x_{ij}^{br} \leq 1 && \forall b \in [B], r \in [R] \\
&\sum_{i\in[S],j\in[T]} x_{ij}^{br} \geq \sum_{i\in[S],j\in[T]} x_{ij}^{b,r+1} && \forall b \in [B], r \in [R-1] \\
&\sum_{b\in[B],j\in[T],r\in[R]} x_{ij}^{br} \geq l_i && \forall i \in [S] \\
&\sum_{b\in[B],i\in[S],r\in[R]} x_{ij}^{br} \leq u_j && \forall j \in [T] \\
&x_{ij}^{br} \in \{0,1\} && \forall b \in [B], i \in [S], j \in [T], r \in [R] \\
&t_{\max},\ t_{to}^{br},\ t_{back}^{br} \in \mathbb{R} && \forall b \in [B], r \in [R]
\end{aligned}
$$

The meanings of the input parameters are described shortly:

- B, R, S and T are the scenario-specific numbers of the available buses, rounds, sources and sinks or also called shelters, respectively. The operator $[\cdot]$ maps a number to the set from 1 to the input, e.g., $[S] = \{1, \ldots, S\}$.
- For a fixed source $i \in [S]$ and sink $j \in [T]$, d_{ij}^{ST} and d_{ji}^{TS} denote the distances from i to j and j to i, respectively.
- For a source $i \in [S]$, l_i describes the number of evacuees in i in terms of integer multiples of bus loads.

- For a sink $j \in [T]$, u_j describes the capacity of shelter j in terms of integer multiples of bus loads.

An optimal solution of this IP describes a bus schedule which minimizes the maximum travel time over all buses while transporting all evacuees to the sinks and respecting the sink capacities.

Since we want to use CPLEX to solve each upcoming subproblem, different testings showed that the model is too hard to solve (which is also described in Goerigk et al. (2013)). Since in most cases, this hardness is caused due to the use of rounds, we suggest using a simpler reduced model without rounds:

$$\begin{aligned}
\min \quad & t_{\max} && && (5.10)\\
\text{s.t.} \quad & t_{\max} \geq \sum_{i\in[S],j\in[T]} d_{ij}^{red} x_{ij}^b && \forall b \in [B] && (5.11)\\
& \hat{x}_{ij}^b \leq C^b x_{ij}^b && \forall b \in [B], i \in [S], j \in [T] && (5.12)\\
& \sum_{b\in[B],j\in[T]} \hat{x}_{ij}^b = l_i && \forall i \in [S] && (5.13)\\
& \sum_{b\in[B],i\in[S]} \hat{x}_{ij}^b \leq u_j && \forall j \in [T] && (5.14)\\
& x, \hat{x} \in \mathbb{N}_0^{B+S+T} && && (5.15)\\
& t_{\max} \in \mathbb{R} && && (5.16)
\end{aligned}$$

where C^b denotes the capacity of bus b and d_{ij}^{red} describes the *reduced distance*, i.e., the distance from source i to sink j plus the average distance from j to all other sources computed as

$$d_{ij}^{red} := d_{ij}^{ST} + \sum_{k\in S} \frac{d_{jk}^{TS}}{S} .$$

In this context, for some source $i \in [S]$ and sink $j \in [T]$, l_i and u_j describe the number of evacuees in i and the capacity of shelter j in terms of evacuees, respectively.

Clearly, due to the missing rounds, this model is not able to model the distances exactly as in the previous IP, but it has some nice additional features which are important for some newly introduced objective functions: Firstly, each bus $b \in [B]$ can have an individual capacity C^b and secondly, we do not restrict to busloads since each variable $\hat{x}_{ij}^b$ models the number of evacuees going from source i to sink j by bus b. x_{ij}^b counts how often bus b takes the route from i to j.

Furthermore, besides the minimization of the maximal evacuation time, we introduce new objectives which can be incorporated in the above model with the help of the following objective functions and, if necessary, the following additional constraints:

- minimize average evacuation time:

$$\min \sum_{b\in[B],i\in[S],j\in[T]} \frac{d_{ij}^{red} x_{ij}^{b}}{B}$$

- minimize/maximize average shelter utilization:

$$\min / \max \sum_{b\in[B],i\in[S],j\in[T]} \frac{\hat{x}_{ij}^{b}}{u_j \cdot T}$$

- minimize maximal shelter utilization:

$$\begin{aligned} \min \quad & \delta^1 \\ \text{s.t.} \quad & \sum_{b\in[B],i\in[S]} \frac{\hat{x}_{ij}^{b}}{u_j} \leq \delta^1 && \forall j \in [T] \\ & \delta^1 \in \mathbb{R} \end{aligned}$$

- maximize minimal shelter utilization:

$$\begin{aligned} \max \quad & \delta^2 \\ \text{s.t.} \quad & \sum_{b\in[B],i\in[S]} \hat{x}_{ij}^{b} \leq v_j \cdot u_j && \forall j \in [T] \\ & \sum_{b\in[B],i\in[S]} \frac{\hat{x}_{ij}^{b}}{u_j} + (1 - v_j) \geq \delta^2 && \forall j \in [T] \\ & v \in \{0,1\}^T \\ & \delta^2 \in \mathbb{R} \end{aligned}$$

- minimize number of shelters:

$$\begin{aligned} \min \quad & \sum_{j\in[T]} v_j \\ \text{s.t.} \quad & \sum_{b\in[B],i\in[S]} \hat{x}_{ij}^{b} \leq v_j \cdot u_j && \forall j \in [T] \\ & v \in \{0,1\}^T \end{aligned}$$

sub-scenario	s_0	s_1	s_2	s_3
1	400	480	470	400
2	790	400	320	240
3	320	720	390	320
4	320	400	710	320
5	240	550	320	640

Table 5.1: Distribution of evacuees to sources

t_0	t_1	t_2	t_3	t_4
160	880	480	160	320

Table 5.2: Capacities of the shelters

- minimize number of buses:

$$\begin{aligned} \min \quad & \sum_{b\in[B]} y_b \\ \text{s. t.} \quad & \sum_{i\in[S],j\in[T]} x_{ij}^b \leq y_b \cdot \sum_{i\in[S],j\in[T]} \left(\frac{\min\{l_i,u_j\}}{C^b}+1\right) \qquad \forall b\in[B] \\ & y \in \{0,1\}^B \end{aligned}$$

  Note that for each solution there always exists an equivalent solution fulfilling $C^b \cdot (x_{ij}^b - 1) \leq \min\{l_i, u_j\}$ for all $i \in [S], j \in [T]$.

Hence, before applying the Box-Algorithm, the decision makers can choose three objective functions from these eight suggested functions. Moreover, all listed objective functions can be obviously grouped into three groups: time, shelter and bus. Therefore, we suggest choosing one objective function from each group. Note that if the maximization of the total evacuation time is not considered, constraint (5.11) is not needed.

In the following, we give computational results and comparisons for some extensions for a real-world scenario for Kaiserslautern. The scenario is from Goerigk and Grün (2012). In this scenario, 1750 people have to be evacuated and use four (in the danger zone) almost evenly distributed different sources (bus stops) s_0 to s_3. We look at five different distributions from the people to the sources as shown in Table 5.1, i.e., we get five different sub-scenarios. The evacuees are brought to five shelters (gymnasiums) t_0 to t_4 outside the danger zone with capacities as shown in Table 5.2. Moreover, the scenario assumes three buses, each with capacity 80. The distances between the sources and sinks are given in Table 5.3. A picture of the used sources and shelters is given in Figure 5.5.

We ran Algorithm 5.1 for all sub-scenarios with all relevant combinations of three objective functions, i.e., one objective function from each group time, shelter and bus,

	t_0	t_1	t_2	t_3	t_4
s_0	524	725	335	184	1080
s_1	309	734	370	124	1115
s_2	349	504	402	279	1090
s_3	606	609	165	344	910

Table 5.3: Distances from sources to sinks

resulting in 50 different instances. Moreover, for each such created instance, we used four different parameter-settings for the Box-Algorithm (cf. Section 5.8):

P1 *with:* partial dominance, nondominated selection rule; *without:* feasible-tests

P2 *with:* partial dominance, nondominated selection rule, feasible-tests

P3 *with:* nondominated selection rule, feasible-tests; *without:* partial dominance

P4 *with:* partial dominance, max-dist selection rule, feasible-tests

Hence, we got 200 different computational results.

The threshold for partial dominance was set to 0.1 and the corner point distances were always measured in a normalized fashion with desired maximal distance $\delta^C = 0.1$. Moreover, we limit the maximal number of iterations to 300 and in each completion step (see lines 8 to 11 in Algorithm 5.1), we only search for one feasible point without special objective function.

All experiments were conducted on a compute server with a 16-core Intel Xeon E5-2670 processor, running at 2.60 GHz with 96 GB RAM and Ubuntu 12.04. The Box-Algorithm itself is written in C++ using the programming interface ROPI[3] such that different solvers (CPLEX, SCIP, Xpress, Gurobi) can be used, compiled with g++ v. 4.6.3. We used CPLEX v. 12.6 with 4 threads and a time-limit of 300 seconds for solving each upcoming subproblem.

From the 200 different computational results, we evaluated the following properties:

- **P1** causes 60 % of the different instances to reach the iteration limit of 300 steps. With **P2**, all instances terminate in (much) less than 300 steps. Looking only at the other 40 % for which **P1** does not reach the iteration limit, the use of parameter-setting **P2** saves in the average more than 27 % computation time and more than 60 % iteration steps compared to **P1** (see also Figure 5.6a).
- Unfortunately, parameter-setting **P2** has in our tested 50 instances no advantages to parameter-setting **P3** since only very few subdivisions due to partial dominance

[3] `http://optimierung.mathematik.uni-kl.de/~goerigk/ropi/`

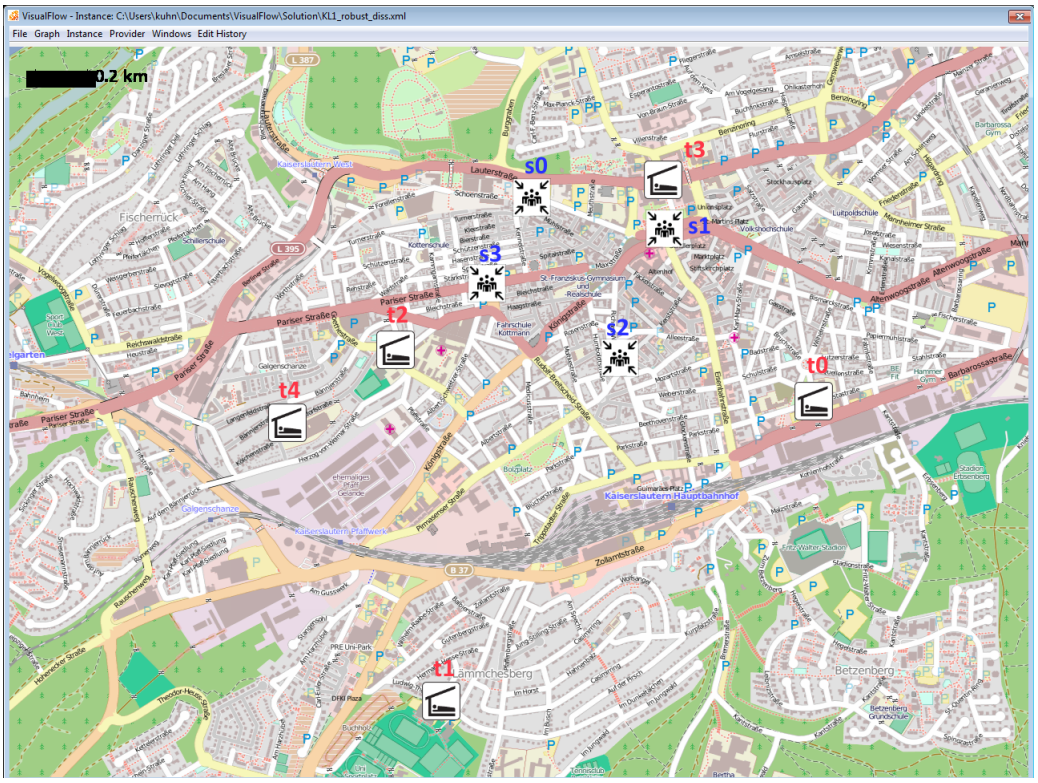
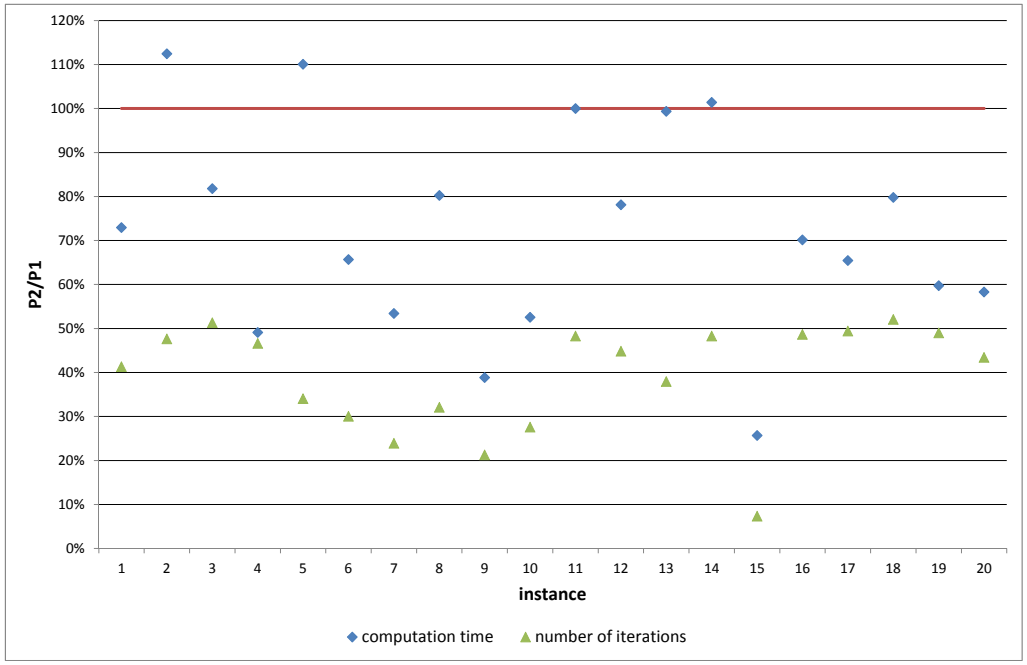


Figure 5.5: Sources (marked blue) and sinks (marked red) in the considered scenario

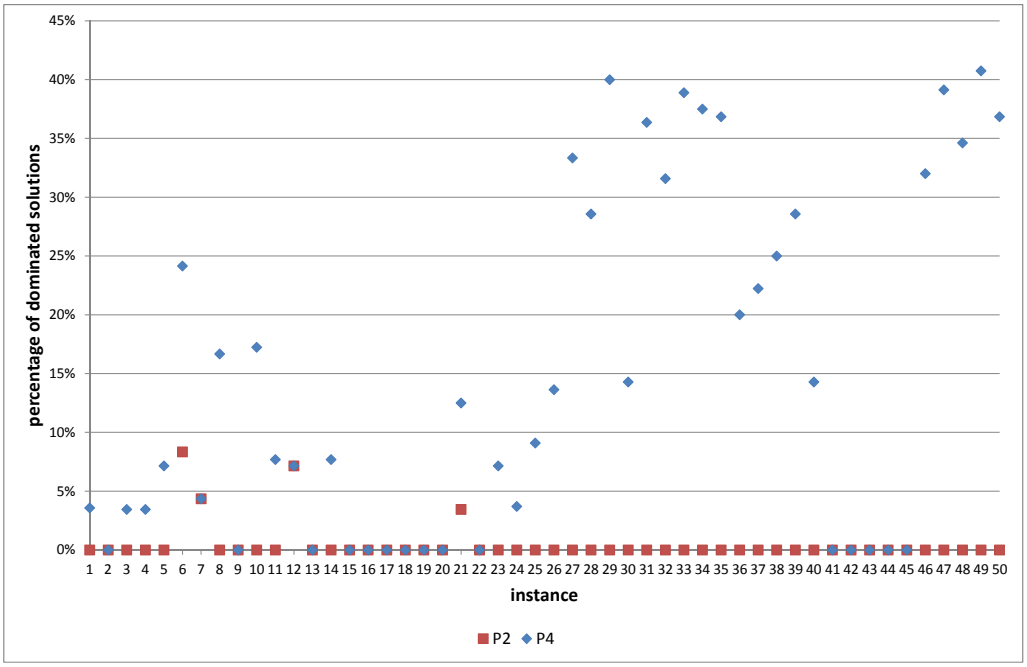
are invoked (in average 2.72 in each instance). However, for other problem scenarios partial dominance can lead to a better performance.

- Parameter-setting **P2** needed in the average 80 % more time compared to **P4**. However, **P4** created more dominated solutions; in the average 15 % of the solutions generated with **P4** and only 0.4 % of the solutions generated with **P2** are dominated by other solutions. Moreover, **P4** results in 36 % of the instances in greater than or equal to 20 % dominated solutions (see also Figure 5.6b).

To sum up, in our scenarios, the feasible-tests saved a lot of time, partial dominance did not lead to a better result since only few partial divisions are invoked and the max-dist selection rule is helpful to save time but can produce many dominated solutions compared to the nondominated selection rule.



(a) The computation time (number of iterations) of **P2** is divided by the computation time (number of iterations) of **P1**.



(b) Comparison w.r.t. the percentage of representative points dominated by other representative points.

Figure 5.6: Comparison between **P1** and **P2** (a), and **P2** and **P4** (b).

6 The Hypervolume Subset Selection Problem

6.1 Introduction

In this chapter, we investigate the problem of selecting a subset with desired cardinality k from a given set of m points, the Hypervolume Subset Selection Problem (HSSP). This subset is chosen due to the *hypervolume indicator* measuring the dominated region in the objective space bounded by some reference point (see Definition 2.11). Hence, we want to find a subset which obeys the predefined cardinality and maximizes the hypervolume indicator. This problem originates from population-based heuristic approaches and is used in the selection step of several multiobjective evolutionary algorithms (see, e.g., Huband et al., 2003; Beume et al., 2007; Igel et al., 2007; Bader and Zitzler, 2011). However, in this thesis, we do not concentrate on the application to population-based heuristic approaches but on the problem itself. Algorithms for the problem can be used to support the decision makers in finding a subset with desired cardinality from a given representative system while fulfilling the quality guarantee that this subset maximizes the hypervolume indicator.

Since in the corresponding literature, it is common to consider a multiple objective *maximization* problem, we also assume here, in contrast to all other chapters, a given multiple objective maximization problem MOP. All definitions from Chapter 2 are easily transferred to this case.

As introduced in Definition 2.11, we denote with $D(N)$ and $S(N)$ the dominated region and the hypervolume indicator of a finite set $N \subseteq \mathbb{R}^p$ w.r.t. the Lebesgue measure $\lambda^{\mathcal{L}}(\cdot)$ in $\mathbb{R}^p$, respectively.

Definition 6.1 (The *p*-criteria *k*-HSSP):
Let $N = \{y^1, \ldots, y^m\} \subseteq \mathbb{R}^p$ be a set of nondominated points for some MOP. Let $k \in \{1, \ldots, m\}$ and let $y^{ref} \in \mathbb{R}^p$ be an appropriate reference point. The *p-criteria hypervolume subset selection problem with desired cardinality k* (p-criteria k-HSSP) consists of selecting a subset $N' \subseteq N$ with $|N'| = k$ such that the value of the hypervolume indicator $S(N')$ of this subset is maximal, i.e.,

$$S(N') = \max_{\substack{N'' \subseteq N \\ |N''|=k}} S(N'') \ .$$

For the special cases $p = 2$ and $p = 3$, we call the problem the bicriteria and tricriteria k-HSSP, respectively. ◇

We first investigate the bicriteria k-HSSP and provide two new formulations for it, a linear programming formulation and a k-link shortest path formulation. For the latter formulation, we propose an algorithm for which we obtain the currently best known complexity bound for solving the bicriteria k-HSSP. In the subsequent section, we consider the tricriteria k-HSSP for which we propose an integer programming formulation with a corresponding branch-and-bound scheme.

6.2 The Bicriteria Hypervolume Subset Selection Problem

In this section, we consider the bicriteria k-HSSP for a set $N = \{y^1, \dots, y^m\} \subseteq \mathbb{R}^2$.

To the best of our knowledge, only two approaches have been proposed in the literature tackling the bicriteria k-HSSP. Bader (2009) introduced a dynamic programming algorithm with $\mathcal{O}(m^2k)$ time complexity. This algorithm is based on the fact that the contribution of the hypervolume indicator of the left-most point of a given nondominated subset only depends on this point and its immediate neighbor. Very recently and independently from our approach, Bringmann et al. (2014) proposed another approach to the same problem with a time complexity of $\mathcal{O}(m(k + \log m))$. This algorithm computes in the ℓth iteration all maximal hypervolume indicator values using at most ℓ points with respect to m appropriately chosen reference points. This can be done for each reference point by computing the maximum of $\mathcal{O}(m)$ different linear function evaluations. The running time is achieved by using a linear time algorithm to compute the upper envelope of lines.

In this section, we provide two new formulations for the bicriteria k-HSSP, an integer programming formulation which can be solved by solving its linear programming relaxation and a k-link shortest path formulation on a special digraph. In the k-link shortest path formulation, the arc costs have a special property, called the Monge property. This property allows us to solve the bicriteria k-HSSP with a simple dynamic programming approach in $\mathcal{O}((m-k+1)k+m \log m)$ time, which slightly improves upon the result of Bringmann et al. (2014). Thus, with the k-link shortest path approach, we obtain the currently best known complexity bound for solving the bicriteria case of the k-HSSP.

Remark 6.2:
In the succeeding subsections, we assume a given set of nondominated points $N = \{y^1, \dots, y^m\} \subseteq \mathbb{R}^2$ of a bicriteria maximization problem and a reference point $y^{ref} < y^i$ for all $i = 1, \dots, m$.
We further assume that the points in N are sorted in increasing order of the first component, i.e., $y_1^i < y_1^j$ for $i < j$, which can be achieved in $\mathcal{O}(m \log m)$ time (see, e.g., Cormen et al., 2009). ◁

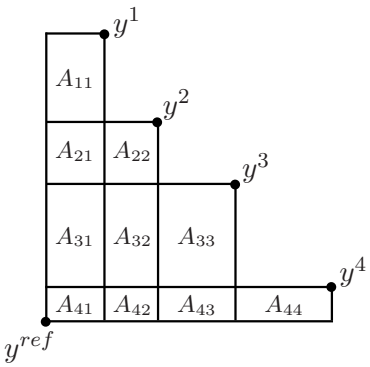


Figure 6.1: Partition of the dominated region for a given set $N = \{y^1, y^2, y^3, y^4\}$

6.2.1 Preprocessing: Decomposition of the Dominated Region

In the following, a preprocessing step which is crucial for the new formulations is introduced. We partition the dominated region $D(N)$ into certain rectangles.

Definition 6.3 (Partition of the Dominated Region):
For the bicriteria k-HSSP, let A_{ij}, $i, j \in \{1, \dots, m\}$, $i \geq j$, denote the rectangle defined by the subregion of $D(N)$ which is exclusively dominated by all points in $\{y^j, \dots, y^i\}$ and no other point in N, i.e., $A_{ij} = \left\{ y \in \mathbb{R}^2 : (y_1^{j-1}, y_2^{i+1})^\top \leqq y \leqq (y_1^j, y_2^i)^\top \right\}$, with $y_1^0 \coloneqq y_1^{ref}$ and $y_2^{m+1} \coloneqq y_2^{ref}$.
For each such rectangle A_{ij}, we define the weight w_{ij} as the area of rectangle A_{ij}, i.e., $w_{ij} \coloneqq \lambda^{\mathcal{L}}(A_{ij}) = (y_1^j - y_1^{j-1}) \cdot (y_2^i - y_2^{i+1})$. ◇

Proposition 6.4: *For the rectangles A_{ij}, $i, j \in \{1, \dots, m\}$, $i \geq j$, from Definition 6.3, it holds*

$$D(N) = \bigcup_{\substack{i,j \in \{1,\dots,m\} \\ i \geq j}} A_{ij}$$

and $\lambda^{\mathcal{L}}(A_{i_1 j_1} \cap A_{i_2 j_2}) = 0$ for $(i_1, j_1)^\top \neq (i_2, j_2)^\top$.

Proof:
Follows directly from Definition 6.3 and the sorting of the set N (cf. Remark 6.2). □

An example of this partition is given in Figure 6.1. Note that we can calculate all the weights w_{ij}, $i \geq j$, in $\mathcal{O}(m^2)$ time.

6.2.2 An Integer Programming Formulation

In this subsection, we present an integer programming (IP) formulation for the bicriteria k-HSSP and show that we can efficiently solve this formulation by solving its linear programming (LP) relaxation. Following the notation from Definition 6.3, we denote with w_{ij} the weight of the corresponding rectangle A_{ij}, $i \geq j$. The following IP formulation models the corresponding bicriteria k-HSSP:

$$
\begin{aligned}
(IP_k) \quad \max \quad & \sum_{i=1}^{m} \sum_{j=1}^{i} w_{ij} x_{ij} && && (6.1)\\
\text{s. t.} \quad & \sum_{\ell=1}^{m} x_{\ell\ell} = k && && (6.2)\\
& \sum_{\ell=j}^{i} x_{\ell\ell} \geq x_{ij} && i = 2, \ldots, m;\ j = 1, \ldots, i-1 && (6.3)\\
& x_{ij} \in \{0, 1\} && i = 1, \ldots, m;\ j = 1, \ldots, i &
\end{aligned}
$$

Thereby, variable $x_{\ell\ell}$ is equal to 1 if and only if $y^\ell \in N$ is selected and variable x_{ij}, $i \geq j$, determines whether the subregion A_{ij} is covered by some point in $\{y^j, \ldots, y^i\}$, which is guaranteed by the constraints (6.3). Constraint (6.2) ensures the compliance of the selection of exactly k points and the objective function (6.1) calculates the hypervolume indicator of the current feasible selection, which has to be maximized.

For proving that this IP can be efficiently solved with the help of the corresponding LP relaxation, we need the notion of a totally unimodular matrix.

Definition 6.5 (Totally Unimodular Matrix):
A matrix $A \in \mathbb{R}^{p \times q}$ is called *totally unimodular* if the determinant of each square submatrix of A belongs to the set $\{0, 1, -1\}$. $\diamond$

Theorem 6.6 (Integrality (Nemhauser and Wolsey, 1999)):
Let $b, b' \in \mathbb{Z}^p$ and $d, d' \in \mathbb{Z}^q$. If the matrix $A \in \mathbb{R}^{p \times q}$ is totally unimodular and $P := \{x \in \mathbb{R}^q : b' \leqq Ax \leqq b, d' \leqq x \leqq d\}$ is non-empty, then P is an integral polyhedron*, i.e., each of its non-empty faces contains an integral point.*

In the following, we consider the corresponding LP relaxation of (IP_k). We show that the constraint matrix of this LP in some standard form is totally unimodular. The LP

relaxation is given by the following formulation:

$$
\begin{aligned}
(LP_k) \quad \max \quad & \sum_{i=1}^{m} \sum_{j=1}^{i} w_{ij} x_{ij} && && (6.4)\\
\text{s.t.} \quad & \sum_{\ell=1}^{m} x_{\ell\ell} = k && && (6.5)\\
& \sum_{\ell=j}^{i} x_{\ell\ell} - x_{ij} - s_{ij} = 0 && i = 2,\ldots,m;\ j = 1,\ldots,i-1 && (6.6)\\
& 0 \le x_{ij} \le 1 && i = 1,\ldots,m;\ j = 1,\ldots,i &\\
& s_{ij} \ge 0 && i = 2,\ldots,m;\ j = 1,\ldots,i-1 &
\end{aligned}
$$

where the new variables s_{ij} are *surplus variables.*

Theorem 6.7: *The constraint matrix of* (LP_k) *is totally unimodular.*

Proof:
If we rearrange the columns of the constraint matrix in a certain way, first the variables $x_{\ell\ell}$, $\ell = 1,\ldots,m$, and then the variables x_{ij} and s_{ij}, $i = 2,\ldots,m$, $j = 1,\ldots,i-1$, according to the ordering of the constraints (6.6), the structure of the constraint matrix corresponding to (LP_k) is given by

$$
\left(\begin{array}{ccc|ccc|ccc}
1 & \ldots & 1 & 0 & \ldots & 0 & 0 & \ldots & 0 \\
\hline
\multicolumn{3}{c|}{C} & \multicolumn{3}{c|}{\text{-}I} & \multicolumn{3}{c}{\text{-}I}
\end{array}\right)
$$

where C is a $\frac{m(m-1)}{2} \times m$-matrix and $-I$ is the negative of the $\frac{m(m-1)}{2} \times \frac{m(m-1)}{2}$-identity-matrix. Let us denote by $\tilde{C}$ the submatrix $\begin{pmatrix} e \\ C \end{pmatrix}$, where $e \in \mathbb{R}^m$ is the vector of all ones, and by D the submatrix $\left(\begin{array}{ccc|ccc} 0 & \ldots & 0 & 0 & \ldots & 0 \\ \hline \multicolumn{3}{c|}{\text{-}I} & \multicolumn{3}{c}{\text{-}I} \end{array}\right)$.

Since the matrix D has no more than one nonzero entry in each column, namely -1, this matrix is totally unimodular (cf. Nemhauser and Wolsey, 1999, Proposition 2.6). Observe that the matrix $\tilde{C}$ defines an *interval matrix*, i.e., all entries are in {0,1} and in each row the 1's appear consecutively. Thus, $\tilde{C}$ is also totally unimodular (cf. Nemhauser and Wolsey, 1999, Corollary 2.10).

Let an arbitrary squared submatrix B of the constraint matrix of (LP_k) be given.

Case 1: B is completely contained in $\tilde{C}$ or completely contained in D and, therefore, $\det(B) \in \{0, \pm 1\}$ since both matrices are totally unimodular.

Case 2: B possesses $s > 0$ columns from matrix $\tilde{C}$ and $t > 0$ columns from matrix D, w.l.o.g. no duplicate column from D.

We choose some column $j > s$ from B belonging to D and expand the determinant of B with respect to the jth column (*Laplace expansion*). Since this column has only one nonzero entry, say b_{ij}, we get $\det(B) = (-1)^{i+j+1} \cdot \det(M_{ij})$, where M_{ij} is the minor of matrix B formed by eliminating row i and column j from B. The minor M_{ij} also corresponds to a squared submatrix of the constraint matrix and if we follow the above Laplace expansion, after t steps, we end with a submatrix $\tilde{B}$ of B matching "Case 1", i.e., $\det(\tilde{B}) \in \{0, \pm 1\}$. Then, by construction, we get $\det(B) = \pm \det(\tilde{B}) \in \{0, \pm 1\}$.

Since B was an arbitrarily chosen squared submatrix, we have shown the totally unimodular property of the constraint matrix. □

Corollary 6.8 (Integrality): *The polyhedron corresponding to (LP_k) is integral. In particular, (IP_k) can be solved by solving its linear programming relaxation (LP_k), which always possesses an integral optimal solution.*

Proof:
This follows from Theorem 6.6 and Theorem 6.7 and the following upper bound on the surplus variables:

$$s_{ij} = \sum_{\ell=j}^{i} x_{\ell\ell} - x_{ij} \leq k - x_{ij} \leq k \quad i = 2, \ldots, m;\ j = 1, \ldots, i-1 \qquad \square$$

6.2.3 A k-link Shortest Path Formulation with the Monge Property

In the following, we show how the bicriteria k-HSSP can be modeled using a k-link shortest path formulation in an appropriate directed graph (digraph). In a *k-link shortest path problem*, the aim is to find the shortest path from a given source to a given sink with exactly k arcs. This problem on our special digraph can then be solved using a dynamic programming (DP) approach. Since the digraph has a special structure, the Monge property, in each DP-step, we can use the Matrix-Searching Algorithm from Aggarwal et al. (1987) to find the right entry in linear time.

We first explain the construction of the corresponding digraph $G^N = (V, E)$ related to the set N. The graph construction is based on the observation that for each choice of a subset $\{y^{s_1}, \ldots, y^{s_k}\} \subseteq N$, $s_i < s_j$ for $i < j$, the contribution to the hypervolume indicator of the consecutive points $\{y^{s_i+1}, \ldots, y^{s_{i+1}-1}\}$, for two indices with $s_i + 1 < s_{i+1}$, only depends on the coordinates of these consecutive points and the

points y^{s_i} and $y^{s_{i+1}}$. For each element $y^c \in N$, we create a node $c \in V$. In addition, we also add two other nodes 0 and $m+1$ to V, which serve as source and target node, respectively. We add the arcs $e_{uv} := (u, v)$ for all $u, v \in \{0, \ldots, m+1\}$ with $u < v$ to E.

Definition 6.9 (Exclusive Volume):
According to Definition 6.3, for a subset $\{y^u, y^{u+1}, \ldots, y^{v-1}, y^v\} \subseteq N$, $u, v \in \{1, \ldots, m\}$, $u \leq v$, we define the *exclusive volume* $EV(y^u, y^v)$ of this set by

$$EV(y^u, y^v) := \sum_{i=u}^{v} \sum_{j=u}^{i} w_{ij} \ .$$

For notational reasons, for $u, v \in \{0, \ldots, m+1\}$, $u > v$, we define $EV(y^u, y^v) := 0$.
For the graph $G^N = (V, E)$, we define the cost c_{uv} of an arc $e_{uv} \in E$ as follows

$$c_{uv} := EV(y^{u+1}, y^{v-1}) = \sum_{i=u+1}^{v-1} \sum_{j=u+1}^{i} w_{ij} \ . \qquad \diamond$$

The cost c_{uv} describes the exclusive contribution of the whole set $\{y^{u+1}, \ldots, y^{v-1}\} \subseteq N$ to the hypervolume indicator $S(N)$. Moreover, the cost $c_{0,m+1}$ equals the hypervolume indicator $S(N)$ of the whole set N (cf. Proposition 6.4). An example for the graph construction is depicted in Figure 6.2.

Proposition 6.10: *Let $N' = \{y^{s_1}, \ldots, y^{s_k}\} \subseteq N$ be a selection of k points with $s_1 < s_2 < \ldots < s_k$. Then, it holds*

$$S(N') = S(N) - EV(y^1, y^{s_1 - 1}) - \sum_{i=1}^{k-1} EV(y^{s_i+1}, y^{s_{i+1}-1}) - EV(y^{s_k+1}, y^m) \ .$$

Proof:
The claim is shown by induction on k.

For $k = m$, i.e., $N' = N$, the claim is trivially fulfilled.

Suppose the claim holds for all selections of k points. Let N' be a selection of $1 \leq k-1 < m$ points and, according to the ordering of the elements in the set N, let $y^\ell \in N \setminus N'$ be the first point not contained in N'. Let $N' \cup \{y^\ell\} = \{y^{s_1}, \ldots, y^{s_k}\}$ with $s_1 < s_2 < \ldots < s_k$ and let $r \in \{1, \ldots, k\}$ with $y^\ell = y^{s_r}$. Then, we immediately get $s_i = i$ for $1 \leq i \leq r$ and it holds $S(N') = S(N' \cup \{y^\ell\}) - (y_1^\ell - y_1^{s_{r-1}}) \cdot (y_2^\ell - y_2^{s_{r+1}})$ with $y_1^{s_0} := y_1^{ref}$ and $y_2^{s_{k+1}} := y_2^{ref}$. From the induction hypothesis, we get

$$\begin{aligned} S(N') = S(N) - EV(y^1, y^{s_1 - 1}) - \sum_{i=1}^{k-1} EV(y^{s_i+1}, y^{s_{i+1}-1}) \\ - EV(y^{s_k+1}, y^m) - (y_1^{s_r} - y_1^{s_{r-1}}) \cdot (y_2^{s_r} - y_2^{s_{r+1}}) \ . \end{aligned}$$

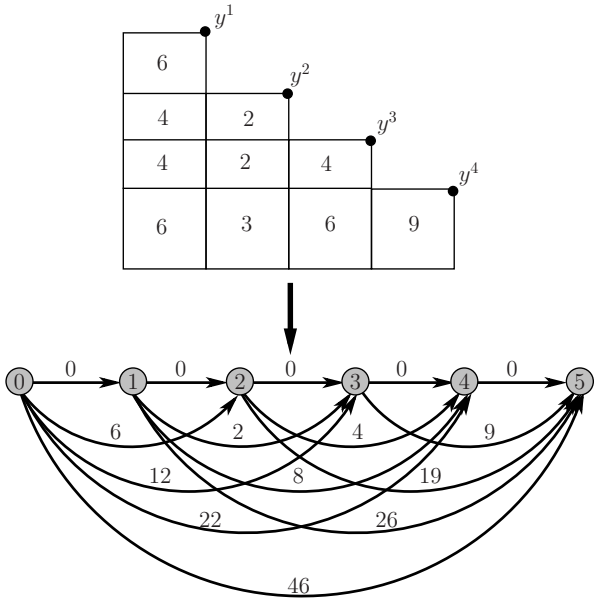


Figure 6.2: Example for the graph construction

Case 1 ($1 < r < k$):
Obviously, $EV(y^{s_{r-1}+1}, y^{s_r-1}) = 0$ since $s_r = s_{r-1} + 1$. Moreover, it holds

$$EV(y^{s_r+1}, y^{s_{r+1}-1}) + (y_1^{s_r} - y_1^{s_{r-1}}) \cdot (y_2^{s_r} - y_2^{s_{r+1}}) = EV(y^{s_r}, y^{s_{r+1}-1}) \; .$$

Case 2 ($r = 1$):
Then, we get $s_r = s_1 = 1$ and $EV(y^1, y^{s_1-1}) = 0$. Moreover, it holds

$$EV(y^{s_1+1}, y^{s_2-1}) + (y_1^{s_r} - y_1^{s_{r-1}}) \cdot (y_2^{s_r} - y_2^{s_{r+1}}) = EV(y^1, y^{s_2-1}) \; .$$

Case 3 ($r = k$):
Then, we get $s_r = s_k = k$ and $EV(y^{s_{k-1}+1}, y^{s_k-1}) = 0$. Moreover, it holds

$$EV(y^{s_k+1}, y^m) + (y_1^{s_r} - y_1^{s_{r-1}}) \cdot (y_2^{s_r} - y_2^{s_{r+1}}) = EV(y^k, y^m) = EV(y^{s_{k-1}+1}, y^m) \; .$$

Hence, for each case, we get the desired formula proving the claim. □

Observation 6.11:
From construction of the digraph G^N and Proposition 6.10, we get that each choice in the bicriteria k-HSSP of a subset $N' \coloneqq \{y^{s_1}, \ldots, y^{s_k}\} \subseteq N$ with cardinality k corresponds to a path in G^N with exactly $k + 1$ arcs that starts in node 0, visits the

Algorithm 6.1 DP for the special k-link shortest path problem

Input: $G^N = (V, E)$, $k \in \{1, \ldots, m\}$.
Output: Length $D(k, m+1)$ of the shortest path from 0 to $m+1$ with exactly k arcs.
1: $D(1, v) \leftarrow c_{0v}$ for all $v \in \{1, \ldots, m-k+2\}$
2: **for** $\ell = 2, \ldots, k$ **do**
3: **for** $v = \ell, \ldots, m+1-k+\ell$ **do**
4: $D(\ell, v) \leftarrow \min_{u=\ell-1,\ldots,v-1} \{D(\ell-1, u) + c_{uv}\}$

nodes s_1 to s_k, and ends in the node $m+1$. Since the cost of a used arc e_{uv} is defined as the exclusive volume of the jumped over nodes $u+1$ to $v-1$, the hypervolume indicator $S(N)$ of the whole set N minus the total cost of the path corresponds then to the hypervolume indicator $S(N')$ of the subset N'. Hence, the $(k+1)$-link shortest path problem on G^N models the bicriteria k-HSSP. ◁

In the following, for simplicity, we look at the k-link shortest path problem on G^N which corresponds to the bicriteria $(k-1)$-HSSP, i.e., with desired cardinality $k-1$.

Since in our special k-link shortest path problem the *Bellman principle of optimality* is valid (subpaths of length $\ell < k$ of optimal paths are again optimal for the corresponding ℓ-link shortest path problem), we can use a straightforward DP approach to solve this problem (see Algorithm 6.1). In an iteration corresponding to some $\ell \in \{2, \ldots, k\}$ and $v \in \{\ell, \ldots, m+1-k+\ell\}$, the length $D(\ell, v)$ of the shortest path from 0 to v with exactly ℓ arcs is calculated. However, this would not lead directly to a much better running time than Bader's DP algorithm since finding the minimum in line 4 in a naive way is done in $\mathcal{O}(m-k+2)$ time, resulting in an overall running time of $\mathcal{O}((m-k+1)^2 k)$ for the bicriteria k-HSSP. Therefore, to get a better running time, we have to improve this DP scheme and have to investigate the digraph G^N further.

In the following, we prove some special structure for G^N, the so-called *(concave) Monge property* (Aggarwal et al., 1994).

Theorem 6.12 (Monge Property): *For the digraph G^N, let four arcs with the following shape be given*

$$e_{ij},\ e_{i,j-1},\ e_{i+1,j},\ e_{i+1,j-1}$$

for some i, j with $j > i+2$ (see also Figure 6.3). Then, we have

$$c_{ij} > c_{i,j-1} + c_{i+1,j} - c_{i+1,j-1}\ .$$

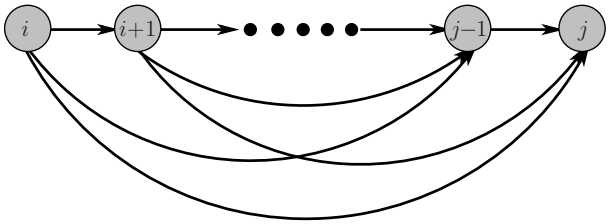


Figure 6.3: Selected arcs in Theorem 6.12

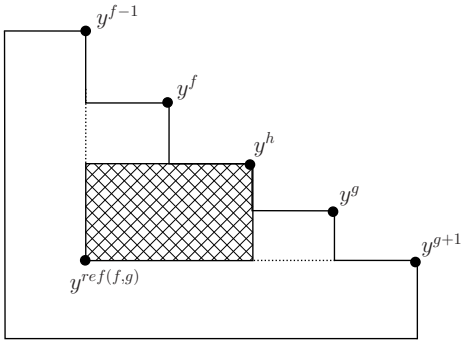


Figure 6.4: Example for $B^{(f,g)}(y^h)$ (shaded area)

Proof:
For $1 \leq f \leq h \leq g \leq m$, we define $B^{(f,g)}(y^h)$ as the area of the rectangle induced by the two corner points y^h and the special reference point $y^{ref(f,g)} := (y_1^{f-1}, y_2^{g+1})^\top$ with $y_1^0 = y_1^{ref}$ and $y_2^{m+1} = y_2^{ref}$ (cf. Figure 6.4).

We immediately get the following three formulas:

$$\begin{aligned}
EV(y^{i+1}, y^{j-1}) &= EV(y^{i+2}, y^{j-2}) + B^{(i+1,j-2)}(y^{i+1}) + B^{(i+1,j-1)}(y^{j-1}) \\
EV(y^{i+1}, y^{j-2}) &= EV(y^{i+2}, y^{j-2}) + B^{(i+1,j-2)}(y^{i+1}) \\
EV(y^{i+2}, y^{j-1}) &= EV(y^{i+2}, y^{j-2}) + B^{(i+2,j-1)}(y^{j-1})
\end{aligned}$$

Moreover, we know:

$$\begin{aligned}
B^{(i+2,j-1)}(y^{j-1}) &= B^{(i+1,j-1)}(y^{j-1}) - B^{(i+1,j-1)}(y^{i+1}) \cap B^{(i+1,j-1)}(y^{j-1}) \\
&= B^{(i+1,j-1)}(y^{j-1}) - w_{j-1,i+1}
\end{aligned}$$

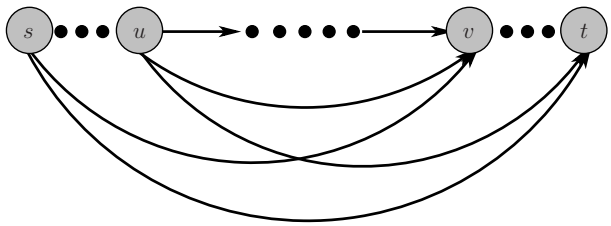


Figure 6.5: Selected arcs in Corollary 6.13

With these, we can state the following chain:

$$\begin{aligned} c_{ij} &= EV(y^{i+1}, y^{j-1}) \\ &= EV(y^{i+2}, y^{j-2}) + B^{(i+1,j-2)}(y^{i+1}) + B^{(i+1,j-1)}(y^{j-1}) \\ &= EV(y^{i+2}, y^{j-2}) + B^{(i+1,j-2)}(y^{i+1}) + B^{(i+2,j-1)}(y^{j-1}) + w_{j-1,i+1} \\ &= EV(y^{i+1}, y^{j-2}) + EV(y^{i+2}, y^{j-1}) - EV(y^{i+2}, y^{j-2}) + w_{j-1,i+1} \\ &= c_{i,j-1} + c_{i+1,j} - c_{i+1,j-1} + w_{j-1,i+1} \qquad (6.7) \\ &> c_{i,j-1} + c_{i+1,j} - c_{i+1,j-1} \qquad \square \end{aligned}$$

Adapting a proof of Aggarwal and Park (1989), we can state the following equivalent property.

Corollary 6.13: *For the digraph G^N, let four arcs with the following shape be given*

$$e_{st},\ e_{sv},\ e_{ut},\ e_{uv}$$

with $s < u < v < t$ (see also Figure 6.5). Then, we have

$$c_{st} > c_{sv} + c_{ut} - c_{uv}\ .$$

In particular, we get the following formula

$$c_{st} = c_{sv} + c_{ut} - c_{uv} + (y_1^u - y_1^s) \cdot (y_2^v - y_2^t) \qquad (6.8)$$

with $y_1^0 = y_1^{ref}$ and $y_2^{m+1} = y_2^{ref}$.

Proof:
From formula (6.7), we get

$$c_{ij} + c_{i+1,j-1} = c_{i,j-1} + c_{i+1,j} + w_{j-1,i+1}$$

for all $i, j \in \{0, \ldots, m+1\}$ with $j > i+2$. Thus, for $j > u+1$, it holds

$$\sum_{i=s}^{u-1} (c_{ij} + c_{i+1,j-1}) = \sum_{i=s}^{u-1} (c_{i,j-1} + c_{i+1,j} + w_{j-1,i+1})$$

Algorithm 6.2 Calculation of the costs c_{0v} and $c_{v,m+1}$

Input: $N = \{y^1, \ldots, y^m\}$.
Output: The costs c_{0v} and $c_{v-1,m+1}$ for all nodes $v \in \{1, \ldots, m+1\}$.

```
1: c_01 ← 0, c_{m,m+1} ← 0, y_2^{m+1} ← y_2^ref
2: for v = 1, ..., m do
3:     c_{0,v+1} ← c_0v + (y_1^v − y_1^ref) · (y_2^v − y_2^{v+1})
4: for v = m, ..., 2 do
5:     c_{v−1,m+1} ← c_{v,m+1} + (y_1^v − y_1^{v−1}) · (y_2^v − y_2^ref)
```

which gives us, by canceling identical terms,

$$c_{sj} + c_{u,j-1} = c_{s,j-1} + c_{uj} + \sum_{i=s}^{u-1} w_{j-1,i+1} \ .$$

Summation over j yields

$$\sum_{j=v+1}^{t} (c_{sj} + c_{u,j-1}) = \sum_{j=v+1}^{t} \left(c_{s,j-1} + c_{uj} + \sum_{i=s}^{u-1} w_{j-1,i+1} \right)$$

implying

$$c_{st} + c_{uv} = c_{sv} + c_{ut} + \sum_{j=v+1}^{t} \sum_{i=s}^{u-1} w_{j-1,i+1} \ .$$

Note that the summation index j fulfills $j \geq v+1 > u+1$. The term

$$\sum_{j=v+1}^{t} \sum_{i=s}^{u-1} w_{j-1,i+1}$$

is identical to the area of the rectangle

$$\left\{ y \in \mathbb{R}^2 : (y_1^s, y_2^t)^\top \leqq y \leqq (y_1^u, y_2^v)^\top \right\}$$

proving formula (6.8). □

Observation 6.14:
For Algorithm 6.1, we can use formula (6.8) to calculate each cost c_{uv} on-the-fly. By setting $s = 0$ and $t = m+1$, we get for $0 < u < v < m+1$

$$c_{uv} = c_{0v} + c_{u,m+1} - c_{0,m+1} + (y_1^u - y_1^{ref}) \cdot (y_2^v - y_2^{ref}) \ .$$

Hence, we only need to precompute all costs c_{0v} and $c_{u,m+1}$ for all $0 \leq u < v \leq m+1$, which can be done in $\mathcal{O}(m)$ time (see Algorithm 6.2). ◁

Algorithm 6.3 MINCOMPUTE(A) (Aggarwal et al., 1987)

Input: A totally monotone matrix $A \in \mathbb{R}^{p \times q}$.
Output: The minimal entries of all columns (cf. Theorem 6.16).
1: $S \leftarrow$ REDUCE(A) // *see Algorithm 6.4*
2: **if** $q = 1$ **then**
3: **return** S
4: $T \leftarrow S[1, 2, \ldots, q; 2, 4, \ldots, 2\lfloor q/2 \rfloor]$
5: $MinEven \leftarrow$MINCOMPUTE(T)
6: From the known positions of the minima in the even columns of S, find the minima in its odd columns and **return** .

Going back to Algorithm 6.1, to find for a fixed $\ell \leq k$ the new entries $D(\ell, v)$, $v = \ell, \ldots, m + 1 - k + \ell$, we have to find in a matrix M^ℓ, in which only the entries $M^\ell(u, v) \coloneqq M^\ell_{uv} \coloneqq D(\ell - 1, u) + c_{uv}$, $v \in \{\ell, \ldots, m + 1 - k + \ell\}$, $u \in \{\ell - 1, \ldots, v - 1\}$ are relevant, for each column v the minimal value, which is then assigned to $D(\ell, v)$. Thus, matrix M^ℓ can be represented by a square matrix in $\mathbb{R}^{(m-k+2)\times(m-k+2)}$, where each undefined entry $M^\ell(u, v)$, $v \in \{\ell, \ldots, m - k + \ell\}$, $u \in \{v, \ldots, m - k + \ell\}$, is set to $m^\infty + (u - \ell)$, where m^∞ denotes a sufficiently large number with $m^\infty > M^\ell_{uv}$ for all $v \in \{\ell, \ldots, m + 1 - k + \ell\}$, $u \in \{\ell - 1, \ldots, v - 1\}$.

For an efficient searching algorithm which finds these minimal values for each column, we first need to define the notion of a totally monotone matrix.

Definition 6.15 (Totally Monotone Matrix):
Let $A \in \mathbb{R}^{p \times q}$ be given. For a column $1 \leq j \leq q$, let $\min(j)$ denote the index of the greatest row containing the minimal value of column j, i.e.,

$$\min(j) \coloneqq \max\{i \in \{1, \ldots, p\} : A_{ij} = \min_{s=1,\ldots,p} A_{sj}\}\,.$$

Matrix A is called *monotone* if $1 \leq j_1 < j_2 \leq q$ implies $\min(j_1) \leq \min(j_2)$. Moreover, matrix A is called *totally monotone* if each submatrix is monotone. $\diamond$

Theorem 6.16 (Matrix-Searching Algorithm (Aggarwal et al., 1987)):
Let $A \in \mathbb{R}^{p \times q}$, $p \geq q$, denote a totally monotone matrix. Then, the Matrix-Searching Algorithm (Algorithm 6.3) finds the minimal entries in all columns in $\mathcal{O}(p)$ time.

Remark 6.17:
In line 4 in Algorithm 6.3, the notation $S[1, 2, \ldots, q; 2, 4, \ldots, 2\lfloor q/2 \rfloor]$ denotes the matrix which consists of all q rows and all even columns $2, 4, \ldots, 2\lfloor q/2 \rfloor$ of $S \in \mathbb{R}^{q \times q}$. Furthermore, in line 6 in Algorithm 6.3, once the positions of the minima in the even columns are known, the minima in the odd columns are restricted to be in at most $q + \lfloor (q-1)/2 \rfloor$ entries of S and, thus, can be found in $\mathcal{O}(q)$ time (Aggarwal et al., 1987). $\triangleleft$

Algorithm 6.4 REDUCE(A) (Aggarwal et al., 1987)

Input: A totally monotone matrix $A \in \mathbb{R}^{p \times q}$.
Output: A square matrix $R \in \mathbb{R}^{q \times q}$ where irrelevant rows are deleted.

```
1: R ← A; ℓ ← q
2: while R has more than q rows do
3:     switch R_ℓℓ do
4:         case R_ℓℓ ≥ R_{ℓ+1,ℓ} and ℓ > 1
5:             ℓ ← ℓ − 1
6:         case R_ℓℓ ≥ R_{ℓ+1,ℓ} and ℓ = 1
7:             Delete row ℓ from R
8:         case R_ℓℓ < R_{ℓ+1,ℓ}
9:             Delete row ℓ + 1 from R
10:            if ℓ < q then
11:                ℓ ← ℓ + 1
```

Theorem 6.18: *M^ℓ is totally monotone for a fixed $\ell \in \{2, \ldots, k\}$.*

Proof:
Choose some arbitrary submatrix with rows $i_1, i_2, \ldots, i_s$, w.l.o.g. without columns containing only entries greater than or equal to m^∞.

Choose two columns from the submatrix $j_1 < j_2$. Suppose that $\min(j_1) > \min(j_2)$ and let $i_g \coloneqq \min(j_2)$ and $i_h \coloneqq \min(j_1)$. From definition of matrix M^ℓ, we get $\ell - 1 \leq i_g < i_h < j_1 < j_2$.

Then, we get the four entries

$$M^\ell(i_g, j_1) = D(\ell - 1, i_g) + c_{i_g, j_1}, \qquad M^\ell(i_g, j_2) = D(\ell - 1, i_g) + c_{i_g, j_2},$$
$$M^\ell(i_h, j_1) = D(\ell - 1, i_h) + c_{i_h, j_1}, \qquad M^\ell(i_h, j_2) = D(\ell - 1, i_h) + c_{i_h, j_2}.$$

Due to the assumption $\min(j_1) > \min(j_2)$, we know $M^\ell(i_h, j_2) > M^\ell(i_g, j_2)$ and, moreover, $M^\ell(i_g, j_1) \geq M^\ell(i_h, j_1)$, i.e., we have

$$D(\ell - 1, i_h) + c_{i_h, j_2} > D(\ell - 1, i_g) + c_{i_g, j_2}$$
$$-D(\ell - 1, i_h) - c_{i_h, j_1} \geq -D(\ell - 1, i_g) - c_{i_g, j_1}$$

which gives us summed up the result

$$c_{i_g, j_2} - c_{i_g, j_1} < c_{i_h, j_2} - c_{i_h, j_1} \,. \tag{6.9}$$

Furthermore, we are now in the situation $i_g < i_h < j_1 < j_2$ and from Corollary 6.13, we immediately get

$$c_{i_g, j_2} - c_{i_g, j_1} > c_{i_h, j_2} - c_{i_h, j_1} \,.$$

This leads together with (6.9) to a contradiction and we get $\min(j_1) \leq \min(j_2)$. □

Corollary 6.19: *Using the Matrix-Searching Algorithm from Theorem 6.16 in the DP approach (see Algorithm 6.1), the k-link shortest path problem on the digraph G^N can be solved in $\mathcal{O}((m-k+2)k)$ time.*

Remark 6.20:
Note that also Aggarwal et al. (1994) mention parenthetically the existence of a $\mathcal{O}(mk)$ time algorithm for the special k-link shortest path problem on G^N. This algorithm can be obtained using results from Aggarwal et al. (1987) and Aggarwal and Park (1988), but we could not find a result in there which explicitly shows that the Monge property induces the totally monotone property for the special matrix M^ℓ such that we decided to explicitly prove Theorem 6.18. ◁

Hence, incorporating the idea from Observation 6.14 and the complexity for the sorting mentioned in Remark 6.2, we can state the following result.

Theorem 6.21: *The bicriteria k-HSSP can be solved in $\mathcal{O}((m-k+1)k + m\log m)$ time.*

Observation 6.22:
Note that a simple modification of the algorithm induced from Theorem 6.21 can obtain all solutions for the bicriteria k-HSSP for all $k = 1, \ldots, m$ in $\mathcal{O}(m^2)$ time. ◁

Remark 6.23:
The whole graph construction cannot be applied to the case with three objectives for $N \subseteq \mathbb{R}^3$ since Proposition 6.10 does not hold for this case. Thus, the problem cannot be easily transformed to a k-link shortest path problem which can be observed in the following example. Consider the following three points

$$y^1 = (1,2,2)^\top, \qquad y^2 = (2,1,3)^\top \quad \text{and} \quad y^3 = (3,3,1)^\top .$$

Looking at the corresponding dominated regions/boxes (w.r.t. $y^{ref} = 0$), we can imply that each pair from the three induced boxes possesses a non-empty intersection only belonging to both considered boxes. Hence, there is no unique sorting of the points as in the bicriteria case. Nevertheless, suppose that we assign an arbitrary sorting to the three nodes in the corresponding digraph with five nodes (including source and target nodes). Clearly, to model the subset selection corresponding to all points except one, the arc jumping over one node must have cost equal to the exclusive volume of the corresponding point. Then, the path corresponding to the subset selection by choosing only the mid-point (w.r.t. the digraph nodes) has the wrong value since we only need two arcs for jumping only over the second and second last node, respectively. We would miss subtracting the volume of the exclusive intersection of the two not selected points. ◁

6.3 The Tricriteria Hypervolume Subset Selection Problem

In this section, we extend the integer programming (IP) formulation introduced in the previous section to the tricriteria k-HSSP for a set $N = \{y^1, \dots, y^m\} \subseteq \mathbb{R}^3$ and a reference point $y^{ref} \in \mathbb{R}^3$ (cf. Definition 6.1). To the best of our knowledge, for $p > 2$ objective functions, there does not exist an algorithm for the p-criteria k-HSSP that is faster than enumerating all possible subsets of desired size (Bringmann and Friedrich, 2010). Hence, it is an open question whether this problem is NP-hard for more than two objectives. Utilizing the linear programming (LP) relaxation of the corresponding IP formulation, we give a simple branch-and-bound algorithm solving the tricriteria k-HSSP.

6.3.1 An Integer Programming Formulation

We first state the IP formulation for the tricriteria k-HSSP. The idea is the same as for the bicriteria k-HSSP (cf. Section 6.2.2), we want to subdivide the space with respect to all given values from the set of nondominated points $N = \{y^1, \dots, y^m\} \subseteq \mathbb{R}^3$. This subdivision is easy to understand geometrically but to get correctly indexed variables, we have to introduce some notation and definitions. We first define the matrix $\Lambda \in \mathbb{R}^{m \times 3}$ with entries

$$\lambda_{ij} \coloneqq \min\Big\{ y_j^\ell : \ell = 1, \dots, m \text{ with } |\{q \neq \ell : y_j^q \leq y_j^\ell\}| \geq i - 1 \Big\}$$

for $i = 1, \dots, m$ and $j = 1, 2, 3$. The entry λ_{ij} describes the ith smallest value from the set $\{y_j^\ell : \ell = 1, \dots, m\}$, for $i = 1, \dots, m$ and $j = 1, 2, 3$.

For two given corner points $\ell, u \in \mathbb{R}^3$, we use the same notation for a box $\mathrm{B}(\ell, u)$ as in Chapter 5 (cf. Definition 5.1).

Then, we define a splitting of the box $\mathrm{B}\Big(y^{ref}, (\lambda_{m1}, \lambda_{m2}, \lambda_{m3})^\top\Big)$ into m^3 subboxes of the shape

$$A_{ij\ell} \coloneqq \mathrm{B}\Big((\lambda_{i-1,1}, \lambda_{j-1,2}, \lambda_{\ell-1,3})^\top, (\lambda_{i1}, \lambda_{j2}, \lambda_{\ell 3})^\top\Big)$$

with $i, j, \ell = 1, \dots, m$ and $\lambda_{0r} \coloneqq y_r^{ref}$ for $r = 1, 2, 3$.

Lemma 6.24: *Let $(i', j', \ell')^\top \in \{1, \dots, m\}^3$. Then, we have*

$$\mathrm{B}\Big(y^{ref}, (\lambda_{i'1}, \lambda_{j'2}, \lambda_{\ell' 3})^\top\Big) = \bigcup_{\substack{(i,j,\ell)^\top \in \{1,\dots,m\}^3: \\ (i,j,\ell)^\top \leqq (i',j',\ell')^\top}} A_{ij\ell} \,.$$

Proof:
From the definition of the matrix Λ, we immediately get

$$\bigcup_{\substack{(i,j,\ell)^\top \in \{1,\dots,m\}^3: \\ (i,j,\ell)^\top \leqq (i',j',\ell')^\top}} A_{ij\ell} \subseteq \mathrm{B}\Big(y^{ref}, (\lambda_{i'1}, \lambda_{j'2}, \lambda_{\ell'3})^\top\Big) .$$

Let $y \in \mathrm{B}\Big(y^{ref}, (\lambda_{i'1}, \lambda_{j'2}, \lambda_{\ell'3})^\top\Big)$ be given. Then, from the definition of Λ, for each criterion $q \in \{1,2,3\}$, there exists an index $s_q \in \{1,\dots,m\}$ with $\lambda_{s_q-1,q} \leq y_q \leq \lambda_{s_q,q}$. Since $y \leqq (\lambda_{i'1}, \lambda_{j'2}, \lambda_{\ell'3})^\top$, we can assume $(s_1,s_2,s_3)^\top \leqq (i',j',\ell')^\top$. Thus, we get $y \in \mathrm{B}\Big((\lambda_{s_1-1,1}, \lambda_{s_2-1,2}, \lambda_{s_3-1,3})^\top, (\lambda_{s_1,1}, \lambda_{s_2,2}, \lambda_{s_3,3})^\top\Big) = A_{s_1 s_2 s_3}$. □

Since we are only interested in the dominated region $D(N)$ of N, some of these subboxes are redundant. To avoid redundant subboxes, we first define a hierarchy for each criterion. Suppose we have given three surjective functions h^j, $j = 1,2,3$, assigning for each criterion j an ordering to the set N:

$$h^j : N \longrightarrow \{1,\dots,m\}, \quad y^\ell \longmapsto h^j(y^\ell)$$

where $h^j(y^\ell)$ maps to an element of $\{i : \lambda_{ij} = y^\ell_j\}$ which is not assigned to another y^r with $r \neq \ell$. For abbreviation, we define the vector-valued function $h \coloneqq (h^1, h^2, h^3)^\top$.

Definition 6.25 (Partition of the Dominated Region):
The *set of relevant subboxes* is defined as $\mathcal{A} \coloneqq \{A_{ij\ell} : \exists y^r \in N \text{ with } (i,j,\ell)^\top \leqq h(y^r)\}$. For each subbox $A_{ij\ell} \in \mathcal{A}$, we define the weight $w_{ij\ell}$ as the volume of subbox $A_{ij\ell}$, i.e., $w_{ij\ell} \coloneqq \lambda^{\mathcal{L}}(A_{ij\ell}) = (\lambda_{i1} - \lambda_{i-1,1}) \cdot (\lambda_{j2} - \lambda_{j-1,2}) \cdot (\lambda_{\ell 3} - \lambda_{\ell-1,3})$. ◇

Proposition 6.26: *For the relevant subboxes $\mathcal{A}$ from Definition 6.25, it holds*

$$D(N) = \bigcup_{A_{ij\ell} \in \mathcal{A}} A_{ij\ell}$$

and $\lambda^{\mathcal{L}}(A_{i_1 j_1 \ell_1} \cap A_{i_2 j_2 \ell_2}) = 0$ for $(i_1,j_1,\ell_1)^\top \neq (i_2,j_2,\ell_2)^\top$.

Proof:
$\lambda^{\mathcal{L}}(A_{i_1 j_1 \ell_1} \cap A_{i_2 j_2 \ell_2}) = 0$ for $(i_1,j_1,\ell_1)^\top \neq (i_2,j_2,\ell_2)^\top$ follows immediately from the definition of the subboxes $A_{ij\ell}$ and the matrix Λ.

$\subseteq$:
Let $y \in D(N)$ be given. Then, there exists a $y^r \in N$ with $y^{ref} \leqq y \leqq y^r$, i.e., $y \in \mathrm{B}(y^{ref}, y^r)$. Let $(i',j',\ell')^\top \coloneqq h(y^r)$, then we get $y^r = (\lambda_{i'1}, \lambda_{j'2}, \lambda_{\ell'3})^\top$ and, thus, $y \in \mathrm{B}\Big(y^{ref}, (\lambda_{i'1}, \lambda_{j'2}, \lambda_{\ell'3})^\top\Big)$. Then, from Lemma 6.24, we get that there exists some $(i,j,\ell)^\top \in \{1,\dots,m\}^3$ with $(i,j,\ell)^\top \leqq (i',j',\ell')^\top = h(y^r)$ and $y \in A_{ij\ell}$.

$\supseteq$:
Let $A_{ij\ell} \in \mathcal{A}$ be given. Then, there exists a $y^r \in N$ with $(i,j,\ell)^\top \leqq h(y^r)$. Let $(i',j',\ell')^\top := h(y^r)$, then we get $y^r = (\lambda_{i'1}, \lambda_{j'2}, \lambda_{\ell'3})^\top$. From Lemma 6.24, we get $A_{ij\ell} \subseteq \mathrm{B}\Big(y^{ref}, (\lambda_{i'1}, \lambda_{j'2}, \lambda_{\ell'3})^\top\Big) = \mathrm{B}(y^{ref}, y^r) \subseteq D(N)$. □

Now, we are able to state the IP formulation for the tricriteria k-HSSP:

$$(IP_k^3) \quad \max \quad \sum_{A_{ij\ell} \in \mathcal{A}} w_{ij\ell} \cdot x_{ij\ell} \tag{6.10}$$

$$\text{s.t.} \quad \sum_{r=1}^{m} x_{h(y^r)} = k \tag{6.11}$$

$$x_{ij\ell} \leq \sum_{\substack{y^r \in N: \\ (i,j,\ell)^\top \leqq h(y^r)}} x_{h(y^r)} \qquad \forall A_{ij\ell} \in \mathcal{A} \setminus \{A_{h(y^s)} : y^s \in N\} \tag{6.12}$$

$$x_{ij\ell} \in \{0,1\} \qquad \forall A_{ij\ell} \in \mathcal{A} \tag{6.13}$$

where the variables and constraints are analogous to the IP formulation for the bicriteria k-HSSP on p. 96. In particular, $x_{h(y^r)}$ is equal to 1 if and only if $y^r \in N$ is selected and variable $x_{ij\ell}$ determines whether the relevant subbox $A_{ij\ell} \in \mathcal{A}$ is covered by some point in $\{y^r \in N : (i,j,\ell)^\top \leqq h(y^r)\}$ (cf. Lemma 6.24).

Remark 6.27:
Note that the scheme used to construct this IP formulation can also be used for more than three criteria. ◁

We denote with (LP_k^3) the LP relaxation of (IP_k^3) in which we relax the binary constraints (6.13) to $0 \leq x_{ij\ell} \leq 1$, $A_{ij\ell} \in \mathcal{A}$.

Remark 6.28:
For the bicriteria k-HSSP, we have shown the integrality of the corresponding LP relaxation (cf. Corollary 6.8). For the tricriteria case, the LP relaxation (LP_k^3) does not define an integral polyhedron in general and, thus, cannot be used to solve (IP_k^3). This can be observed, for example, with the following four points

$$y^1 = (1,2,3)^\top, \qquad y^2 = (2,1,3.1)^\top, \qquad y^3 = (2.1,2.1,2)^\top \quad \text{and} \quad y^4 = (2.2,3,1)^\top .$$

Here, the LP relaxation (w.r.t. $y^{ref} = 0$) has an optimal objective value of 11.31 and the IP an optimal objective value of 11.02. Nevertheless, the LP relaxation can be used to obtain an upper bound on the optimal hypervolume indicator which is utilized in the next subsection to obtain a branch-and-bound algorithm. ◁

6.3.2 A Branch-and-Bound Scheme

With an appropriate IP solver (e.g., CPLEX), the IP formulation (IP_k^3) can be used to solve the tricriteria k-HSSP. Internally, this solver already uses a LP-based branch-and-bound algorithm to solve general IPs. However, these branch-and-bound schemes are developed for general IPs and, in many cases, do not utilize special structures of a given problem. We present a description of a branch-and-bound scheme adjusted to the special structure of the tricriteria k-HSSP. For detailed information about branch-and-bound techniques, the reader is referred to the literature (see, e.g., Nemhauser and Wolsey, 1999). Without loss of generality, we assume in this context that $y^{ref} = 0$, which can be obtained by shifting the set N in direction $-y^{ref}$.

In an LP-based branch-and-bound scheme, upper bounds are provided by LP relaxations and branching is done by adding new constraints. A pseudocode description of a general LP-based branch-and-bound algorithm for a general IP formulation

$$\begin{aligned} (IP) \quad & \max \quad c^\top x \\ & \text{s.t.} \quad x \in X \subseteq \mathbb{Z}^p \end{aligned}$$

with $c \in \mathbb{R}^p$, is given in Algorithm 6.5.

In Algorithm 6.5, the upcoming problems in set $\mathcal{P}$ can be considered as nodes from a tree, where the root node corresponds to the initial problem (IP^0) and the child nodes are induced due to the *branching rule* described in a subroutine Branching. In each iteration of the algorithm, a node $(IP^i, X^i, \bar{z}^i)$ is selected due to the subroutine SelectProblem (cf. line 4). After the current node is processed, either the node is pruned (cf. lines 8, 10 and 14) or the branching rule is executed (cf. line 15). In the latter case, the subroutine Branching creates q new *branching nodes* $\{(IP^{ij}, X^{ij}, z_R^i)\}_{j=1}^q$, where $\bigcup_{j=1}^q X^{ij} = X$ and $X^{ij_1} \cap X^{ij_2} = \emptyset$ for $j_1, j_2 \in \{1, \ldots, q\}, j_1 \neq j_2$.

To get an appropriate branch-and-bound algorithm for the tricriteria k-HSSP, we have to specify the subroutines SelectProblem and Branching. Since we only have binary variables in (IP_k^3), the branching is usually done by selecting a *branching variable* $x_{ij\ell}$ and creating two new problems by adding the constraints $x_{ij\ell} = 0$ and $x_{ij\ell} = 1$, respectively. To quickly receive feasible integral solutions, we implement the subroutine SelectProblem due to a special depths first search in which always branches of the type $x_{ij\ell} = 1$ are preferred.

Due to the special structure of the problem, we get the following result concerning the integrality of the variables.

Lemma 6.29: *Let $(\widetilde{IP}_k^3)$ denote the IP corresponding to (IP_k^3) with an arbitrary number of additional constraints from the set $\{x_{ij\ell} = \varrho : A_{ij\ell} \in \mathcal{A}$ and $\varrho \in \{0, 1\}\}$. Let x be an optimal solution of the LP relaxation of $(\widetilde{IP}_k^3)$. If $x_{h(y^r)}$ is integral for all*

Algorithm 6.5 General LP-based Branch-and-Bound Algorithm (Nemhauser and Wolsey, 1999)

Input: An IP formulation (IP) with bounded optimal objective value.
Output: An optimal solution $\underline{x}_{IP}$ of (IP) with objective value $\underline{z}_{IP}$.

1: $IP^0 \leftarrow IP$, $X^0 \leftarrow X$, $\bar{z}^0 \leftarrow \infty$, $\underline{z}_{IP} \leftarrow -\infty$, $\underline{x}_{IP} \leftarrow NULL$
2: $\mathcal{P} \leftarrow \{(IP^0, X^0, \bar{z}^0)\}$
3: **while** $\mathcal{P} \neq \emptyset$ **do**
4: $(IP^i, X^i, \bar{z}^i) \leftarrow$ SelectProblem$(\mathcal{P})$
5: $\mathcal{P} \leftarrow \mathcal{P} \setminus \{(IP^i, X^i, \bar{z}^i)\}$
6: Solve the LP relaxation of IP^i and obtain an optimal solution x^i_R with objective value z^i_R.
7: **if** $x^i_R = NULL$ **then**
8: continue // *prune by infeasibility*
9: **else if** $z^i_R \leq \underline{z}_{IP}$ **then**
10: continue // *prune by bound*
11: **else if** $x^i_R \in X^i$ **and** $z^i_R > \underline{z}_{IP}$ **then**
12: $\underline{z}_{IP} \leftarrow z^i_R$, $\underline{x}_{IP} \leftarrow x^i_R$
13: Delete all problems $(IP^j, X^j, \bar{z}^j) \in \mathcal{P}$ with $\bar{z}^j \leq \underline{z}_{IP}$. // *prune by bound*
14: continue // *prune by optimality*
15: $\{(IP^{ij}, X^{ij}, z^i_R)\}_{j=1}^q \leftarrow$ Branching(X^i)
16: $\mathcal{P} \leftarrow \mathcal{P} \cup \{(IP^{ij}, X^{ij}, z^i_R)\}_{j=1}^q$
17: **return** $(\underline{x}_{IP}, \underline{z}_{IP})$

$r = 1, \ldots, m$, then there exists a solution $\hat{x}$ which is integral and produces the same objective value as x.

Proof:
The constraints (6.12) from (IP^3_k) induce integrality for all variables if the variables $x_{h(y^r)}$ are integral for all $r = 1, \ldots, m$ since we only have non-negative weights in the objective function (6.10). □

Hence, with this result, we only need to branch the m variables $x_{h(y^r)}$, $r = 1, \ldots, m$, which we call *parent variables*. Furthermore, in line 11 in Algorithm 6.5, integrality of the current solution can be checked by checking integrality of the m parent variables.

Suppose, we are in an arbitrary iteration of Algorithm 6.5 and a branching for the current node $(IP^i, X^i, \bar{z}^i)$ is executed (cf. line 15). For the optimal LP solution $x := x^i_R$, we define Γ as the set of all parent variable indices attaining the greatest non-integral

Algorithm 6.6 SelectBranchingVar

Input: x, Γ.
Output: Index of the branching variable.

1: $\Psi \leftarrow \{y^r \in N : r = 1, \ldots, m \text{ with } x_{h(y^r)} = 1\}$
2: $\bar{y} \leftarrow (\max_{y^r \in \Psi} y^r_1, \max_{y^r \in \Psi} y^r_2, \max_{y^r \in \Psi} y^r_3)^\top$
3: $\nu \leftarrow \min\{|\Gamma| - 1, k - |\Psi| - 1, 3\}$
4: **for** $h(y^r) \in \Gamma$ **do**
5: $\quad \delta^r \leftarrow \min\{S(\{y^r, \bar{y}, y^{s_1}, \ldots, y^{s_\nu}\}) - S(\{\bar{y}, y^{s_1}, \ldots, y^{s_\nu}\}) : h(y^{s_i}) \in \Gamma \setminus \{h(y^r)\}, i = 1, \ldots, \nu \text{ with } s_i \neq s_j \text{ for } i \neq j\}$
6: **return** $\arg\max_{h(y^r) \in \Gamma} \delta^r$

value, i.e.,

$$\Gamma = \{h(y^r) : r = 1, \ldots, m \text{ with } x_{h(y^r)} = \max\{x_{h(y^s)} : s = 1, \ldots, m, 0 < x_{h(y^s)} < 1\}\} .$$

If $|\Gamma| = 1$, we select the corresponding parent variable, say $x_{h(y^*)}$, as branching variable and, based on the current node, construct two new branching nodes corresponding to the two problems with additional constraints $x_{h(y^*)} = 0$ and $x_{h(y^*)} = 1$, respectively. In our experience, from solving different example problems, typically it holds $|\Gamma| > 1$. In this case, we have to select one variable from Γ as branching variable. This selection is done by the subroutine SelectBranchingVar described in Algorithm 6.6. For all candidates y^r with $h(y^r) \in \Gamma$, the subroutine SelectBranchingVar assigns a value δ^r to the candidate which quantifies the worst contribution of y^r to the hypervolume indicator for all sets constructed from a bounding box of all points in $\Psi = \{y^r \in N : r = 1, \ldots, m \text{ with } x_{h(y^r)} = 1\}$ and all combinations of $\nu = \min\{|\Gamma| - 1, k - |\Psi| - 1, 3\}$ candidate points which differ from y^r.

Remark 6.30:
In line 5 in Algorithm 6.6, for $\nu = 0$, it should return $\delta^r \leftarrow S(\{y^r, \bar{y}\}) - S(\{\bar{y}\})$. Furthermore, for each case $\nu \in \{0, 1, 2, 3\}$, a simple formula can be derived to compute the needed values for the hypervolume indicators. For $u \in \mathbb{R}^3$, let $B_u \coloneqq \mathrm{B}(0, u)$, then, e.g., for $\nu = 2$, we have:

$$\begin{aligned}
&S(\{y^r, \bar{y}, y^{s_1}, y^{s_2}\}) - S(\{\bar{y}, y^{s_1}, y^{s_2}\}) \\
&= \lambda^{\mathcal{L}}(B_{y^r}) - \lambda^{\mathcal{L}}(B_{y^r} \cap (B_{\bar{y}} \cup B_{y^{s_1}} \cup B_{y^{s_2}})) \\
&= \lambda^{\mathcal{L}}(B_{y^r}) - (\lambda^{\mathcal{L}}(B_{y^r} \cap B_{\bar{y}}) + \lambda^{\mathcal{L}}(B_{y^r} \cap B_{y^{s_1}}) + \lambda^{\mathcal{L}}(B_{y^r} \cap B_{y^{s_2}}) \\
&\quad - \lambda^{\mathcal{L}}(B_{y^r} \cap B_{y^{s_1}} \cap B_{y^{s_2}}) - \lambda^{\mathcal{L}}(B_{y^r} \cap B_{\bar{y}} \cap B_{y^{s_1}}) - \lambda^{\mathcal{L}}(B_{y^r} \cap B_{\bar{y}} \cap B_{y^{s_2}}) \\
&\quad + \lambda^{\mathcal{L}}(B_{y^r} \cap B_{\bar{y}} \cap B_{y^{s_1}} \cap B_{y^{s_2}}))
\end{aligned}$$

Moreover, for a set of points $\{p^1, \ldots, p^e\} \subseteq \mathbb{R}^3$, we get

$$\lambda^{\mathcal{L}}(B_{p^1} \cap \ldots \cap B_{p^e}) = \min\{p_1^1, \ldots, p_1^e\} \cdot \min\{p_2^1, \ldots, p_2^e\} \cdot \min\{p_3^1, \ldots, p_3^e\} \,. \qquad \lhd$$

Remark 6.31:
In line 2 in Algorithm 6.6, the point $\bar{y}$ is used to approximate the dominated region of Ψ by a bounding box. However, $\bar{y}$ can also be omitted or substituted with any other point suited to appropriately approximate the set Ψ, e.g., with

$$\bar{y} \leftarrow \frac{1}{|\Psi|} \left(\sum_{y^r \in \Psi} y_1^r, \sum_{y^r \in \Psi} y_2^r, \sum_{y^r \in \Psi} y_3^r \right)^\top . \qquad \lhd$$

6.3.3 Computational Results

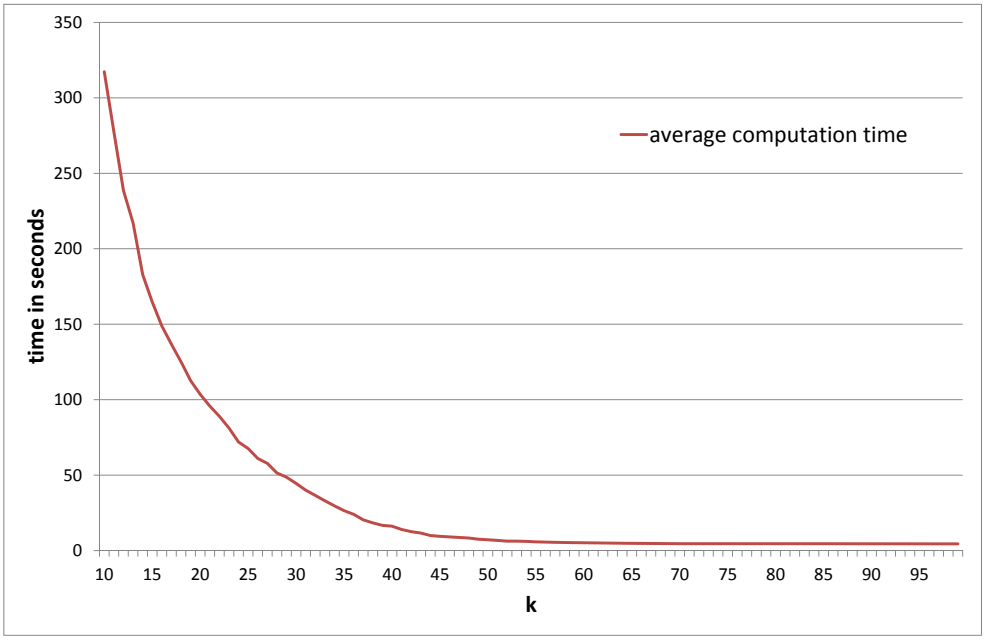
In this subsection, we provide computational results for the above introduced branch-and-bound scheme. For this task, we investigated several test instances. For each number of points $m \in \{50, 75\}$, we created twenty data sets, each of which with m points from $\mathbb{R}^3$ where each coordinate was randomly chosen between 0.1 and 10 (uniform distribution) and no point dominates another point from the corresponding data set. Moreover, for $m = 100$ points, we created one hundred random data sets. For each data set with $m \in \{50, 75, 100\}$ points, we constructed test instances for all $k \in \{\lceil 0.1 \cdot m \rceil, \ldots, m-1\}$.

All experiments were conducted on a compute server with a 16-core Intel Xeon E5-2670 processor, running at 2.60 GHz with 32 GB RAM and Ubuntu 12.04. We have implemented the described branch-and-bound scheme in Python 2.7 where we used CPLEX (v. 12.4 with 1 thread) to solve the upcoming LP problems. For comparison purposes, for each test instance, we also called CPLEX one time to solve the corresponding IP formulation (IP_k^3) using the standard branch-and-bound scheme from CPLEX.

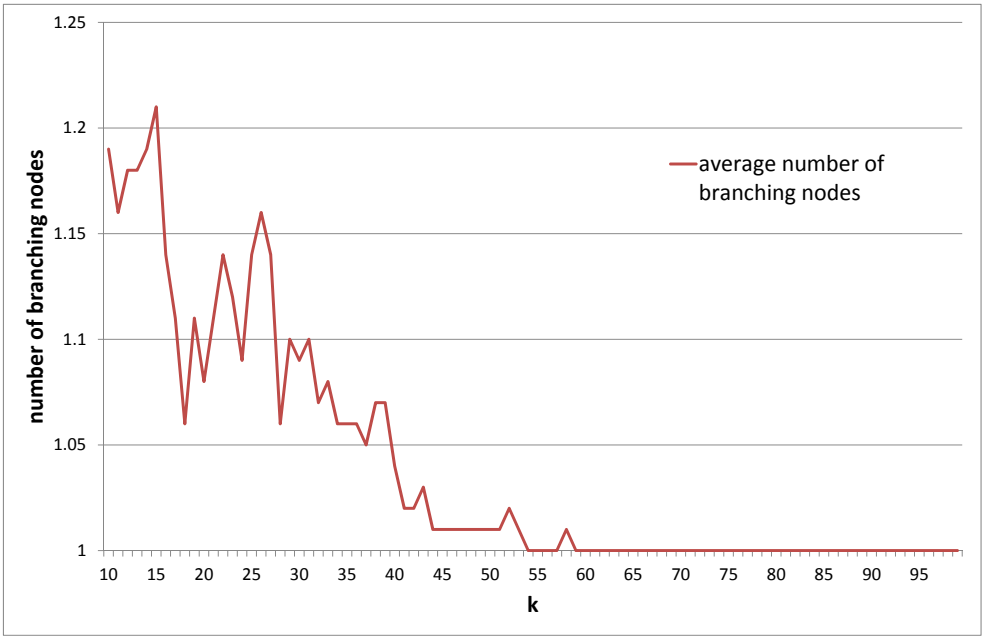
For the test instances with 50 points, both our branch-and-bound scheme and CPLEX solving the IP needed in the average less than 2 seconds to find an optimal solution. One reason of this is that all these problems were solved in the root node, i.e., the solution of the LP relaxation (LP_k^3) already delivered an integral solution. Also most of the test instances for $m = 75$ were solved right after the LP relaxation was solved. Only 19 from 1340 test instances, i.e., less than 1.42 %, needed further considerations in the branch-and-bound scheme but were then solved in the average after 40 seconds. In these test instances, our branch-and-bound scheme was in the average around 41 % faster than CPLEX solving the IP. For the test instances with 100 points, which is our

main concern in these computational study and for which we present the results more precisely, we got that 270 from 9000 instances, i.e., exactly 3 %, were not solved to optimality in the root node and needed further investigation in the branch-and-bound scheme. These 270 instances were solved in the average after around 160 seconds and the branch-and-bound scheme was in the average around 42 % faster than CPLEX solving the IP. In general, Figure 6.6a shows for all 9000 test instances the average computation time of our branch-and-bound scheme depending on the cardinality of the desired set k. The figure shows that the higher the value k the faster the algorithm obtains an optimal solution. If we consider the number of subsets for different cardinalities k, this is a very astonishing observation since, for a fixed k, there exists exactly the same number of subsets with cardinality k as well as with cardinality $m - k$. However, in the average, the linear programming solver from CPLEX is in our test instances more efficient if the right-hand side of the cardinality constraint gets greater. Moreover, Figures 6.6b and 6.6c show the average and maximal number of branching nodes, respectively, for all 9000 test instances during our scheme, depending on the value k. Hence, we explored at most 5 different branching nodes in our scheme before we have found an optimal solution. We can also observe that most of the test instances for which the scheme has to apply the branching rule have a low cardinality k, whereas for most test instances with larger values of k, the problem is solved in the root node, i.e., by the LP relaxation (LP_k^3).

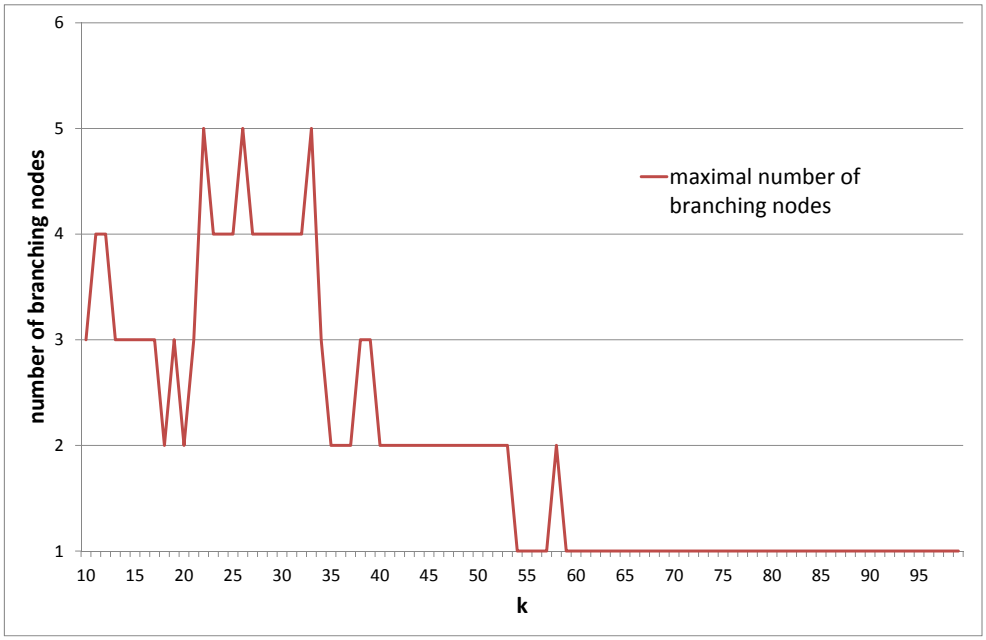
We now come to the big drawback we encountered during our computational tests. Before solving all instances corresponding to one data set, we first had to construct a CPLEX model corresponding to the LP relaxation (LP_k^3) of the root node which is then modified during the algorithm and for the different values of k. This construction took more time than solving a single instance, namely around 400 seconds for a data set with $m = 100$ points. Obviously, we divided the space into $\mathcal{O}(m^3)$ subboxes which leads to $\mathcal{O}(m^3)$ variables and constraints, i.e., for $m = 100$ points, we got around one million variables and constraints which justifies this observation. We also started testing data sets with $m = 200$ points, but the construction took around two hours and the initial CPLEX model had a size of 2 GB such that we had to abort these tests. Thus, the implemented branch-and-bound scheme at its current state is not practical for more than 100 points due to the enormous effort for the construction of the corresponding CPLEX model. However, after the model was constructed, the computational experiments for $m \in \{50, 75, 100\}$ yielded nice computational results. Hence, an appropriate column generation approach, which should be topic of future research, could overcome this drawback.



(a) average computation time



(b) average number of branching nodes



(c) maximal number of branching nodes

Figure 6.6: Computational results for all test instances with $m = 100$ points, depending on k.

7 Presentation to the Decision Makers

In this chapter, we assume the presence of an already computed representative system *Rep* for some MOP with p objective functions. We address the issue of how to present the whole set of computed solutions to the decision makers. Furthermore, we assume that no point in *Rep* dominates another point in *Rep*. We first list some general illustration examples from literature which can be used to present alternatives to the decision makers, provide a new illustration method, the extended bar charts, and propose two algorithms which aid the decision makers to find their desired solution(s).

7.1 Short Overview of Methods from Literature

With an example of three alternatives

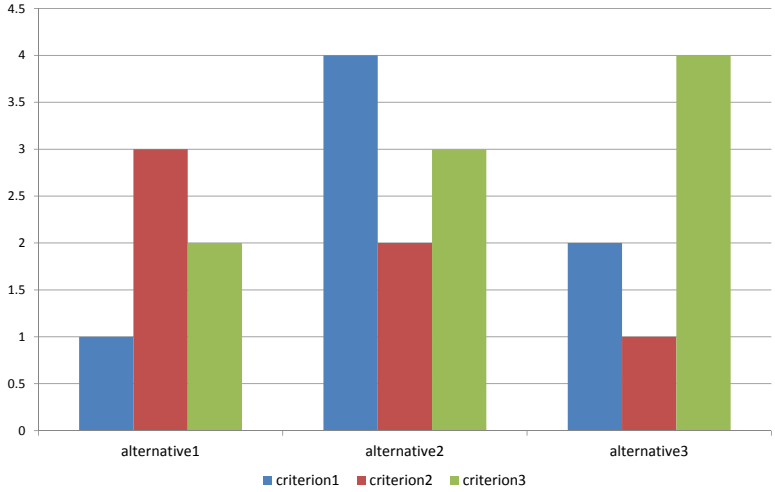
$$a^1 = (1,3,2)^\top, \quad a^2 = (4,2,3)^\top \quad \text{and} \quad a^3 = (2,1,4)^\top ,$$

we shortly present four interesting methods to illustrate different images of multicriteria solutions, called *alternatives* in this context, to the decision makers. For a detailed overview of methods used in the literature, we refer to the survey of Miettinen (2014) and references therein.

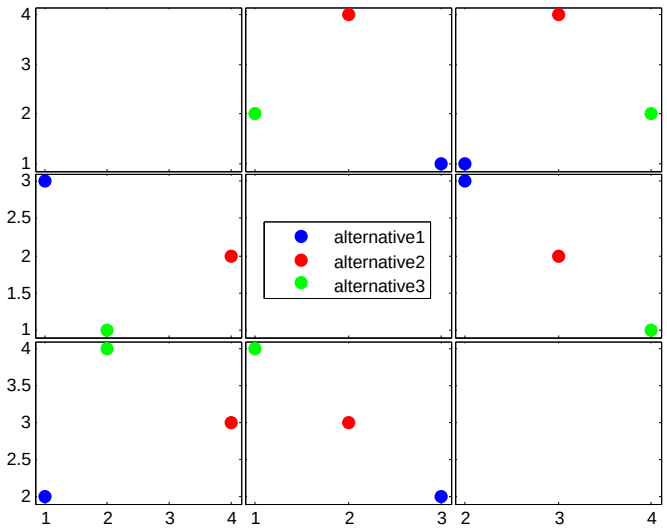
A natural method is the illustration of the alternatives in a *bar chart* (Figure 7.1a), in which the ith group of bars corresponds to the values for the ith alternative and bars of same color correspond to the same criterion. The *scatter plot matrix* (Figure 7.1b) consists of views from projections to all possible planes spanned by two coordinate axes. A bar in a *value path* (Figure 7.1c) corresponds to the range of one criterion and one polygonal curve corresponds to one alternative. In a *spider-web chart* (Figure 7.1d), each criterion describes one direction from the center to a corner point and an alternative corresponds to the polygon with extreme points at the corresponding corner directions due to the corresponding values. In a scaled version, the center corresponds to the ideal and the boundary to the nadir point w.r.t. all presented alternatives. Similar approaches are *petal diagrams* and *star-coordinate systems*.

Remark 7.1:
All these illustration methods can be used for minimization problems as well as for maximization problems. In the context of the meanings of the different criteria, in some methods, it could be helpful to interchange all axes or to negate all entries of each

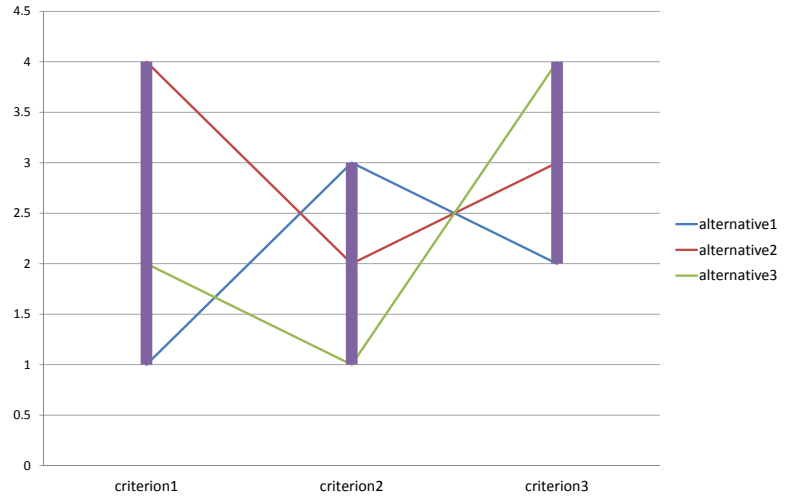
alternative. For example, in the spider-web chart: If we have solved a multicriteria minimization problem and if the decision makers assume that a polygon enclosing more area is better than a polygon enclosing a smaller area, we can negate all values of our alternatives before constructing the spider-web chart. ◁



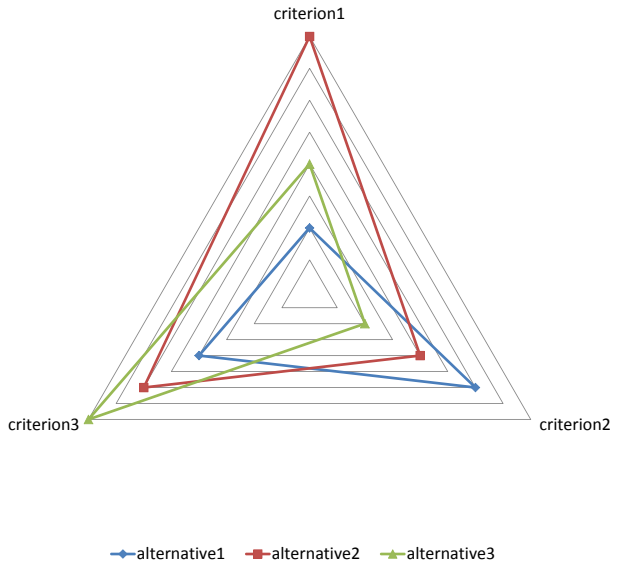
(a) Bar chart



(b) Scatter plot matrix



(c) Value path



(d) Spider-web chart

Figure 7.1: Different illustration methods

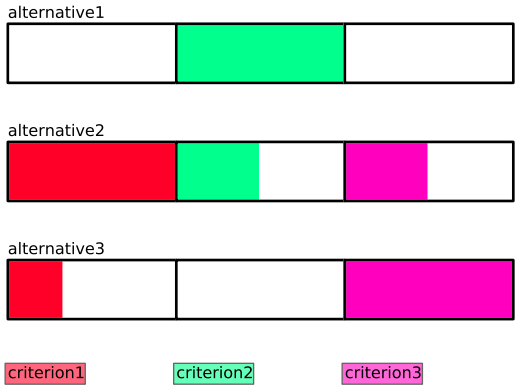


Figure 7.2: Extended bar chart

7.2 Extended Bar Chart

In this section, we present a new illustration method based on the common bar chart. We therefore call this new illustration *extended bar chart*. For each alternative, we concatenate different bars corresponding to the p criteria. We assume that the whole concatenation should have a fixed length $L > 0$ for all alternatives. The length of each bar is determined by predefined weight values $\lambda_i > 0$, $i = 1, \ldots, p$, with $\sum_{i=1}^{p} \lambda_i = 1$, i.e., the ith bar has length $\lambda_i \cdot L$. The weights can be chosen equally or due to the importance of the criteria which should be specified by the decision makers. Each bar is filled with color corresponding to the criterion values of the current alternative, i.e., for an alternative $y \in \mathbb{R}^p$ the ith bar is filled with length

$$\frac{y_i - y_i^{\min}}{y_i^{\max} - y_i^{\min}} \cdot \lambda_i \cdot L$$

where $y^{\min} \in \mathbb{R}^p$ and $y^{\max} \in \mathbb{R}^p$ denote in each coordinate some appropriate lower and upper bound, respectively, for all alternatives and $y_i^{min} < y_i^{max}$ for all $i = 1, \ldots, p$. The summed up (filled) lengths for an alternative are similar to the objective value of the normalized weighted sum method from Kim and de Weck (2006).

The extended bar chart corresponding to the three alternatives used in the previous section is depicted in Figure 7.2 with equally chosen weights, $y^{\min} = (1, 1, 2)^\top$ and $y^{\max} = (4, 3, 4)^\top$. A simple python program drawing such kind of extended bar charts for entered alternatives can be found in Listing A.1 in the appendix.

Obvious advantages of the extended bar chart are an easy pairwise comparison and an easy integration of a reference point for which we can just mark in each bar the corresponding value with, for instance, an arrow. Moreover, for a given alternative, we can directly recognize the (normalized) difference to $y^{\max}$ and $y^{\min}$ and we can

integrate preferences with different weights λ_i or an objective view with $\lambda_i = \lambda_j$ for $i, j = 1, \ldots, p$. However, as with the original bar chart, if we have too many criteria or too many alternatives, it could get confusing for the decision makers.

Remark 7.2:
As already mentioned in Remark 7.1, according to the optimization sense (minimization or maximization) and the preferences of the decision makers (filled bars are better than empty bars or vice versa), we can negate the alternatives to satisfy the needs of the decision makers. ◁

7.3 Guiding Algorithms

In this section, we propose an algorithm to guide the decision makers in choosing their preferred solution. We assume that the points of the representative system *Rep* are stored in a database such that the decision makers can be efficiently guided various times to their preferred solution (due to different opinions/viewpoints and consultations to other people; cf. Figure 1.2 from Introduction).

In each iteration of the algorithm, we have to be able to repeatedly present found points to the decision makers. In the previous two sections, we had mentioned some methods for the illustration of different alternatives. Here, we assume the use of spider-web charts since they are easy to understand and capable to visualize up to around 10 points simultaneously without confusing the decision makers (Miettinen, 2014); but also other methods can be used for which lower and upper bounds for each criterion can be incorporated.

Each time we show the spider-web chart to the decision makers, they are asked to put bounds on each criterion which corresponds to the decision makers' preferences and guides the search further. One possibility to set these bounds is the use of sliders on the corresponding spider-web axes (sliders in a spider-web chart for setting upper bounds were also used in Monz (2006) for an interactive multiobjective optimization method in radiotherapy planning). Another possibility are two dummy-alternatives corresponding to the vector of lower and upper bounds, respectively, and marking the currently forbidden area. In the context of the decision support system in the project DSS_Evac_Logistics, we decided to use the latter possibility to visualize bounds. Once we have given such bounds, we select a given number of representative points from *Rep* (less than 10 is recommended) obeying the new bounds and aiming at diversity. A pseudocode description of the algorithm which should guide the decision makers is depicted in Algorithm 7.1. For two given points $\ell, u \in \mathbb{R}^p$, we use the same notation for a box $\mathrm{B}(\ell, u) := \{y \in \mathbb{R}^p : \ell \leqq y \leqq u\}$ as in Chapter 5 (cf. Definition 5.1). The algorithm combines ideas from a filtering algorithm from Steuer and Harris (1980) and an iterative procedure from Sayın (2003).

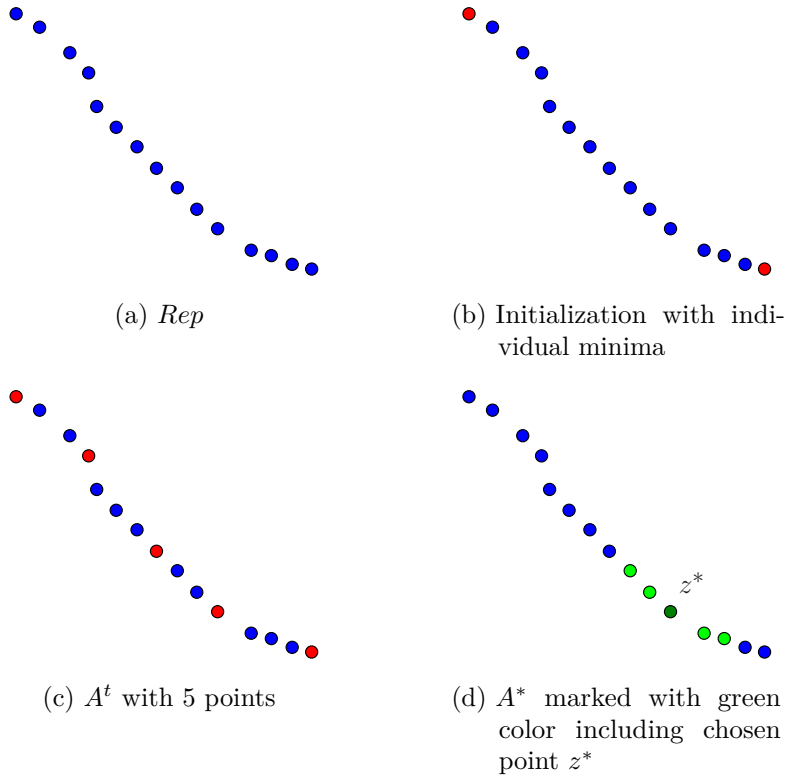
Algorithm 7.1 Guiding Algorithm (Hard Constraints)

Input: Rep, $k \in \mathbb{N}$, $1 \leq q < \infty$.

Output: $A^t \subseteq Rep$ with (at most) k points which should be presented to the decision makers (DM) at time t for all $t \geq 0$ until the DM are satisfied.

Initialize A^0 with at least one point (e.g., add all or a selection of the individual minima corresponding to the p different objectives to A^0).
$t \leftarrow 0$, $\ell_i \leftarrow \min\limits_{z \in Rep} z_i$, $u_i \leftarrow \max\limits_{z \in Rep} z_i$ for all $i = 1, \ldots, p$
$R_i \leftarrow u_i - \ell_i$ for all $i = 1, \ldots, p$
$\pi_i \leftarrow \pi_i^0 \leftarrow \left(R_i \sum\limits_{j=1}^{p} \frac{1}{R_j} \right)^{-1}$ for all $i = 1, \ldots, p$ // *w.l.o.g.* $R_j \neq 0 \; \forall j$
while true **do**
  while $|A^t| < k$ **and** $(Rep \cap \mathrm{B}(\ell, u)) \setminus A^t \neq \emptyset$ **do**
    $z^{new} \leftarrow \operatorname{arg\,max}\limits_{z^r \in (Rep \cap \mathrm{B}(\ell,u)) \setminus A^t} \min\limits_{z^s \in A^t} \left(\sum\limits_{i=1}^{p} (\pi_i |z_i^r - z_i^s|)^q \right)^{1/q}$
    $A^t \leftarrow A^t \cup \{z^{new}\}$
  Show A^t to the DM.
  if DM are satisfied **then**
    $z^* \leftarrow$ chosen point from DM, $A^* \leftarrow \{z^*\}$
    while $|A^*| < k$ **and** $Rep \setminus A^* \neq \emptyset$ **do** // *reverse filtering*
      $z^{new} \leftarrow \operatorname{arg\,min}\limits_{z^r \in Rep \setminus A^*} \left(\sum\limits_{i=1}^{p} (\pi_i^0 |z_i^r - z_i^*|)^q \right)^{1/q}$
      $A^* \leftarrow A^* \cup \{z^{new}\}$
    Show $A^{t+1} := A^*$ to the DM and **return** .
  else
    DM should input new bounds ℓ_i, u_i for all $i = 1, \ldots, p$.
    $R_i \leftarrow u_i - \ell_i$ for all $i = 1, \ldots, p$
    $\pi_i \leftarrow \left(R_i \sum\limits_{j=1}^{p} \frac{1}{R_j} \right)^{-1}$ for all $i = 1, \ldots, p$ // *w.l.o.g.* $R_j \neq 0 \; \forall j$
    $t \leftarrow t + 1$
    if $A^{t-1} \cap \mathrm{B}(\ell, u) \neq \emptyset$ **then**
      $A^t \leftarrow A^{t-1} \cap \mathrm{B}(\ell, u)$
    else
      Initialize A^t with at least one point from $Rep \cap \mathrm{B}(\ell, u)$ if possible.

In the following, we explain the idea behind Algorithm 7.1. Let the generated representative system Rep be of size m. At each time t, we only show $k < m$ alternatives/representative points, denoted with A^t, simultaneously to the decision makers. Similar to Definition 2.5, we call a point in Rep *individual minimum* w.r.t. the ith coordinate if



(a) *Rep*

(b) Initialization with individual minima

(c) A^t with 5 points

(d) A^* marked with green color including chosen point z^*

Figure 7.3: Example for Algorithm 7.1

its value in the ith coordinate is minimal in *Rep*.

For the explanation of the guiding algorithm, we give a simple example with two criteria and $k = 5$. The representative system for this example is given in Figure 7.3a. In the succeeding figures, we mark with red color all points contained in the current set A^t at time t. In a first step, in line 1 at time $t = 0$, we initialize A^0 with at least one point, i.e., with, for example, a randomly selected point or with a selection of the corresponding individual minima (Figure 7.3b). Then, we compute for each objective $i = 1, \ldots, p$ the lower and upper bounds $\ell_i := \min_{z \in Rep} z_i$ and $u_i := \max_{z \in Rep} z_i$, the corresponding range $R_i = u_i - \ell_i$ and a weight value

$$\pi_i = \left(R_i \sum_{j=1}^{p} \frac{1}{R_j} \right)^{-1}$$

which is used to normalize the range of the corresponding objective (see lines 2 to 4). In each iteration, we search for the point farthest away from the current set A^t (see

lines 6 to 8). Hence, we always add the following point to A^t

$$z^{new} := \operatorname*{arg\,max}_{z^r \in (Rep \cap \mathrm{B}(\ell,u)) \setminus A^t} \min_{z^s \in A^t} \left(\sum_{i=1}^{p} (\pi_i |z_i^r - z_i^s|)^q \right)^{1/q}$$

until we have reached a predefined number of points k in A^t or no points are available anymore. For our example, Figure 7.3c shows the set A^t after three iterations. In the next step, at line 9, we show all found alternatives from A^t to the decision makers who decide if there is some point satisfying their needs. If the decision makers are not satisfied, they should input new bounds ℓ_i and u_i for each objective $i = 1, \ldots, p$ (see line 17). In lines 18 to 24, we again compute the corresponding ranges and weight values, increase time t by one, initialize the new set A^t either with points from the old set obeying the new bounds or with at least one point from *Rep* obeying the new bounds, and start our iteration from beginning while respecting the new bounds. If at some point, the decision makers are satisfied with an illustrated alternative z^* (see line 11), we go over to the second phase in which we calculate in a similar fashion (at most) $k-1$ neighboring points of z^* (see lines 12 to 14). If A^* denotes the current set of neighboring points at some iteration, we calculate the next neighbor as

$$z^{new} = \operatorname*{arg\,min}_{z^r \in Rep \setminus A^*} \left(\sum_{i=1}^{p} (\pi_i^0 |z_i^r - z_i^*|)^q \right)^{1/q}.$$

Finally, we show the whole set A^* to the decision makers (see Figure 7.3d) who have to make their final choice from these alternatives (see line 15).

The first phase (lines 6 to 9) is intended to show the decision makers always a diverse set obeying the current bounds such that they can roughly imagine the structure and variety of the whole representative system. In the second phase (lines 11 to 15), the decision makers can refine their choice z^* since the algorithm computes the most similar alternatives to z^*.

Furthermore, whenever they find an accurate solution point, we give them a chance to use another tool which does not work with hard constraints (lower and upper bounds). Suppose that the decision makers have chosen one point z^* in some iteration from Algorithm 7.1 and they want to abort the current algorithm. Originating from the point z^*, they can use Algorithm 7.2 with soft constraints. First, they have to specify some direction vector $r \in \mathbb{R}^p \setminus (-\mathbb{R}^p_{\geqq})$ (assuming a minimization problem) such that they want to move from z^* in direction r, i.e., they aim at some vector similar to $z^* + r$. The selection of the direction vector r can be done for example with the help of minus- and plus-buttons at each axis in the spider-web chart such that one click on a button adds or subtracts a predefined amount to r (defined, e.g., by a given percentage times the remaining range in the current criterion). Algorithm 7.2 first searches for

Algorithm 7.2 Guiding Algorithm (Soft Constraints)

Input: Rep, $z^* \in Rep$, $r \in \mathbb{R}^p \setminus (-\mathbb{R}^p_{\geqq})$, $k \in \mathbb{N}$, $1 \leq q < \infty$.
Output: A^{res}

1: $\ell_i \leftarrow \min_{z \in Rep} z_i$, $u_i \leftarrow \max_{z \in Rep} z_i$ for all $i = 1, \ldots, p$
2: $R_i \leftarrow u_i - \ell_i$ for all $i = 1, \ldots, p$
3: $\pi_i^0 \leftarrow \left(R_i \sum_{j=1}^{p} \frac{1}{R_j} \right)^{-1}$ for all $i = 1, \ldots, p$ // *w.l.o.g.* $R_j \neq 0 \; \forall j$
4: **if** $\left((z^* + r) - \mathbb{R}^p_{\geqq}\right) \cap Rep \neq \emptyset$ **then**
5: Choose a subset A^{res} with at most k points from this set.
6: **else**
7: **while** $|A^{res}| < k$ **and** $Rep \setminus A^{res} \neq \emptyset$ **do**
8: $z^{new} \leftarrow \underset{z^s \in Rep \setminus A^{res}}{\arg\min} \left(\sum_{i=1}^{p} (\pi_i^0 | z_i^s - (z_i^* + r_i)|)^q \right)^{1/q}$
9: $A^{res} \leftarrow A^{res} \cup \{z^{new}\}$
10: Show A^{res} to the DM and **return** .

points in Rep dominating $z^* + r$. If there exist such points, we output some of them to the decision makers; otherwise, we use the reverse filtering from lines 12 to 15 in Algorithm 7.1. The algorithm can be repeated until the decision makers are satisfied. Moreover, instead an input based on an existing alternative z^*, also a pseudo-point can be used which is defined by constructing the desired alternative in the spider-web chart value for value. However, we have chosen to use a real alternative as input and forbid $r \in (-\mathbb{R}^p_{\geqq})$ since we assume that the decision makers have no complete knowledge of the structure of the representative system in the database such that we want to prevent the decision makers to only focus on pseudo-points near the ideal point.

We conclude this section with two remarks about the practical implementation of Algorithm 7.1.

Remark 7.3:
The most time-consuming operations in both algorithms are repetitious comparisons for finding minimal and maximal values from a given set. For a practical implementation, it would be worth to tweak these comparisons. Looking at the most crucial lines 6 to 8 in Algorithm 7.1, after one iteration, we add one element, say $\tilde{z}$, to A^t and in the next iteration, we perform similar comparisons but now with a slightly different set A^t. Hence, to calculate for an arbitrary $z^r \in (Rep \cap \mathrm{B}(l, u)) \setminus A^t$ the new value

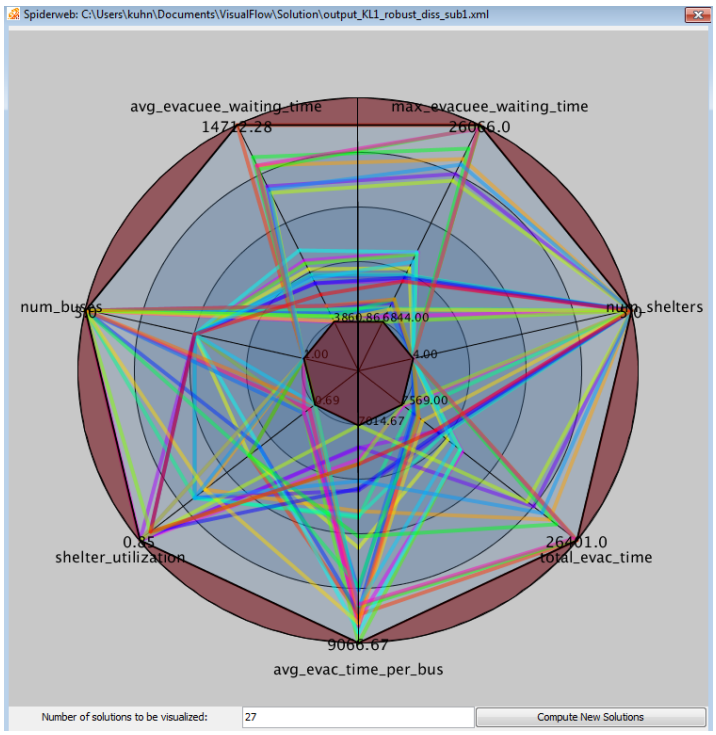
$$\min_{z^s \in A^t} \left(\sum_{i=1}^{p} (\pi_i | z_i^r - z_i^s|)^q \right)^{1/q},$$

we can compare the old value of this expression from previous iteration with the following value corresponding to the new point $\tilde{z}$

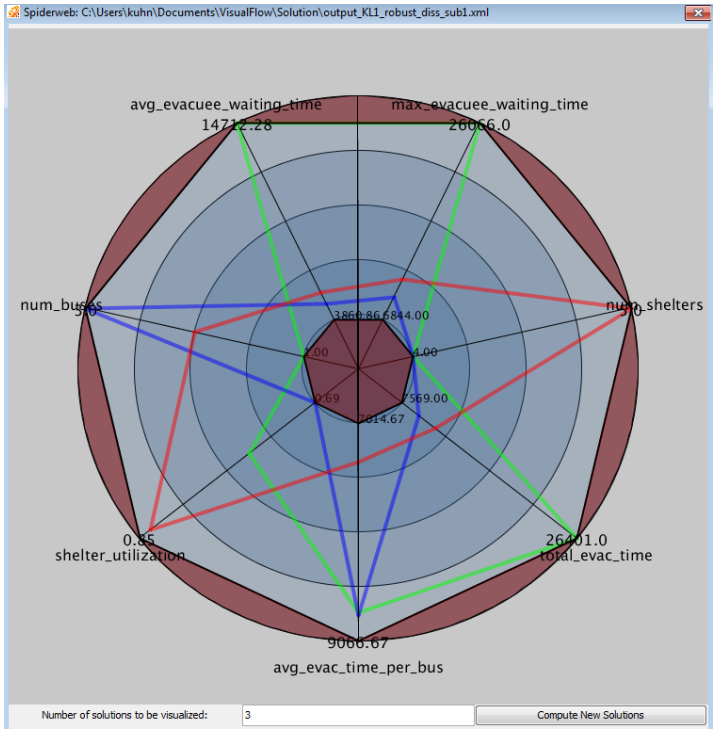
$$\left(\sum_{i=1}^{p}(\pi_i|z_i^r - \tilde{z}_i|)^q\right)^{1/q}$$

which saves a lot of comparisons during the run of the algorithm. ◁

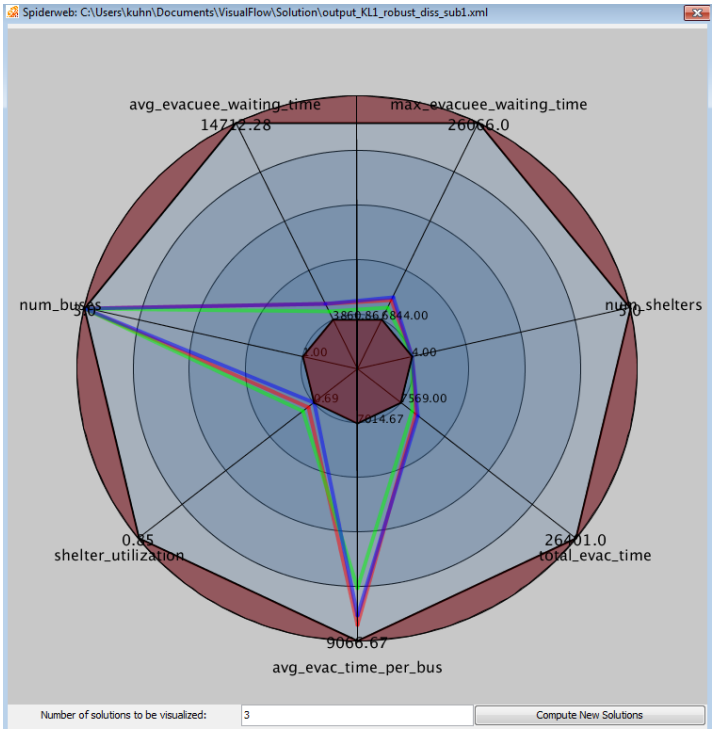
Remark 7.4:
In the context of the research project DSS_Evac_Logistics, Algorithm 7.1 was implemented in Java 7 such that our industrial partners D_Ind from INFORM GmbH could easily integrate it in their demonstrator, called VisualFlow, which was also written in Java. Therefore, we implemented a Java class GuidingHardCon, for which an object is initialized with all representative points (cf. lines 1 to 4). The initialized object possesses a method *oneIteration* corresponding to the first phase (lines 6 to 8) of the guiding algorithm. This method is repeatedly called by VisualFlow until the decision makers are satisfied (line 10). Thereafter, the method *lastIteration* is called which implements the second phase (reverse filtering, lines 12 to 14). For the visualization of the alternatives including the modification of the bounds (lines 9, 15 and 17), D_Ind implemented the above proposed spider-web chart and dummy-alternatives corresponding to the vector of lower and upper bounds which can be modified by the decision makers with appropriate mouse actions. After a corresponding mouse action ends, the new bounds are transmitted to our Java object (cf. lines 17 to 24) and the method *oneIteration* is called. The method *lastIteration* is called after a right click on an alternative is performed. Screenshots showing the spider-web chart from VisualFlow in an evacuation context can be found in Figure 7.4. ◁



(a) All 27 points from *Rep*



(b) *oneIteration* with $k = 3$



(c) *lastIteration* around the blue alternative

Figure 7.4: Screenshots from spider-web chart of VisualFlow

8 Combining Two Optimization Problems

8.1 Introduction

During our project DSS_Evac_Logistics, the question arises how to combine different optimization problems. For example, in large-scale evacuation modeling, two optimization problems arise: The first problem determines the collection points where the affected people have to go, which can be modeled for instance by a k-center location problem, and the second problem uses these collection points as starting points/sources and aims to transport the people to a save place out of the danger zone, modeled by for example some minimum cost flow problem. Certainly, the choice of the collection points and the routes for the transport interfere. Obviously, this complex task in this evacuation scenario consists of two subtasks and it suggests itself to treat them sequentially: First, collection points are determined where the affected people gather. Then, these meeting points are used to come up with a quick and save way of transporting the evacuees.

The reader should be aware that it is by no means obvious—and in general not true—that an optimal solution obtained by this sequential approach leads to a best-possible solution for the overall task. Certainly, solving the two optimization problems in an integrative manner would be preferable. Yet, it is unclear how these two elementary optimization problems can be integrated or combined in order to adequately model the initial task. The goal of this chapter is to step back looking from a meta-level on the issue of how two given general problems can be combined and how the resulting combinations can be related to each other. In the literature, one can find quite a lot references dealing with bilevel optimization (see, e.g., Dempe (2003) and Colson et al. (2007) and references therein for an overview) or multicriteria optimization (see, e.g., Ehrgott, 2005). But, to the best of our knowledge, our approach of considering the integration of different optimization models is not addressed in the rich literature on bilevel or multicriteria optimization.

We consider two general (single-criterion) optimization problems with some special requirements which have to be solved in a greater framework, in which one of the problems depends on the other problem, i.e., one problem possesses an unspecified number of linking constraints influencing its feasible set by a feasible solution of the other problem. To get a better understanding of the term “integrative”, we analyze theoretically how the two general optimization problems can be combined and come

up for our easiest case with four different non-equivalent bilevel formulations and the combined (non-bilevel) bicriteria problem with joined objective function. We relate these different combinations with each other with respect to feasibility and optimality. We extend the models and results to the combination of two multicriteria optimization problems with respect to efficiency and look at further extensions, in which both problems possess linking constraints or at least one of the problems is extended by a combined objective function depending on the feasible solutions of both optimization problems. Thereafter, we give some ideas for the practical approach.

8.2 The Single-Criterion Case

In this section, we first give an illustrating example from evacuation planning which was already mentioned in the introduction. To introduce the idea of the different combined mathematical models under consideration, we give a single-criterion example combining a *k-center location problem* (k-CLP) (Daskin, 2013) and a subsequent *minimum cost flow problem* (MCFP) (Ahuja et al., 1993). The (k-CLP) determines the sources of the (MCFP) and the used (mixed-)integer programming formulation for the potential locations $L = \{v_1, \dots, v_q\}$ $(q > k)$ and the set of clients $\Gamma = \{\gamma_1, \dots, \gamma_r\}$ with associated non-negative weights w_i is given by the following

$$
\begin{aligned}
(k\text{-CLP}) \quad \min \quad & z && && (8.1a)\\
\text{s.t.} \quad & \sum_{j=1}^{q} \hat{x}_{ij} = 1 && i = 1, \dots, r && (8.1b)\\
& \hat{x}_{ij} \leq y_j && i = 1, \dots, r;\ j = 1, \dots, q && (8.1c)\\
& \sum_{j=1}^{q} y_j = k && && (8.1d)\\
& \sum_{j=1}^{q} d_{ij} \hat{x}_{ij} \leq z && i = 1, \dots, r && (8.1e)\\
& \hat{x}_{ij},\ y_j \in \{0, 1\} && i = 1, \dots, r;\ j = 1, \dots, q\,, && (8.1f)
\end{aligned}
$$

where $d_{ij} = w_i \cdot d(\gamma_i, v_j)$, $(i, j) \in \{1, \dots, r\} \times \{1, \dots, q\}$, and $d(\gamma_i, v_j)$ denotes the distance $\|\gamma_i - v_j\|$ between the client γ_i and the potential location v_j measured in a predefined norm $\|\cdot\|$. The variable y_j equals 1 if and only if location v_j is used and $\hat{x}_{ij}$ equals 1 if the client γ_i is assigned to location v_j.

In a real world scenario, the model could calculate the best *collection points*: The clients are places where many people are present and the task is to determine k collection points gathering nearby people and, thereafter, transporting them all to a safe place T.

The collection points are the starting points (*sources*) of the transportation problem (MCFP), where w_i measures the quantity of people in γ_i.

In a next step, we consider a (MCFP) on a directed graph $G = (V, E)$, for which we assume that $L \subseteq V$, i.e., the potential locations are represented by special vertices. To model the above stated task for a feasible solution $(\hat{x}^*, y^*, z^*)$ of the (k-CLP), we have to define the *supply* of vertex $v_j \in L$, $j = 1, \ldots, q$, the *demand* of the special *super sink* vertex $T \in V \setminus L$ and all other nodes as *transshipment nodes* by

$$b_{v_j} = \sum_{i=1}^{r} w_i \hat{x}_{ij}^* \quad j = 1, \ldots, q\,, \quad -b_T = \sum_{i=1}^{r} w_i\,, \quad b_v = 0 \quad \forall v \in V \setminus (L \cup \{T\})\,. \tag{8.2}$$

The consequential (MCFP) with non-negative arc-costs c_e and capacities u_e ($e \in E$) can then be stated as:

$$\begin{aligned}
(\text{MCFP}) \quad \min \quad & \sum_{e \in E} c_e x_e && && (8.3a)\\
\text{s.t.} \quad & \sum_{e=(v,\cdot)\in E} x_e - \sum_{e=(\cdot,v)\in E} x_e = b_v && \forall v \in V && (8.3b)\\
& x_e \leq u_e && \forall e \in E && (8.3c)\\
& x_e \in \mathbb{N}_0 && \forall e \in E && (8.3d)
\end{aligned}$$

In the remainder of this chapter, we refer to (k-CLP) and (MCFP) as the *special case problems* or also *special cases*.

In the following, we consider two general optimization problems which have to be solved in a greater framework, in which one of the problems depends on the other problem, i.e., one problem possesses an unspecified number of linking constraints influencing its feasible set by a feasible solution of the other problem.

In order to state the different combined mathematical models, we look at the following optimization problems (P^1) and $(Q^1_{s^*})$ as possible generalizations of the problems (k-CLP) and (MCFP).

Definition 8.1:
For $n_1, n_2, m_1, m_2, m_3 \in \mathbb{N}$, variables $s \in \mathbb{R}^{n_1}$ and $t \in \mathbb{R}^{n_2}$, constraint functions $g^1 : \mathbb{R}^{n_1} \to \mathbb{R}^{m_1}$, $g^2 : \mathbb{R}^{n_2} \to \mathbb{R}^{m_2}$ and $g^3 : \mathbb{R}^{n_1+n_2} \to \mathbb{R}^{m_3}$, objective functions $f^1 : \mathbb{R}^{n_1} \to \mathbb{R}$ and $f^2 : \mathbb{R}^{n_2} \to \mathbb{R}$, and a constant parameter $s^* \in \mathbb{R}^{n_1}$ with $g^1(s^*) \leqq 0$, we define the two optimization problems (P^1) and $(Q^1_{s^*})$ as

$$\begin{aligned}
(P^1) \quad \min_{s} \quad & f^1(s) && (8.4a)\\
\text{s.t.} \quad & g^1(s) \leqq 0 && (8.4b)
\end{aligned}$$

$$\begin{aligned}
(Q^1_{s^*}) \quad \min_{t} \quad & f^2(t) && (8.5a)\\
\text{s.t.} \quad & g^2(t) \leqq 0 && (8.5b)\\
& g^3(s^*, t) \leqq 0 && (8.5c)
\end{aligned}$$

◇

Remark 8.2:
In the special cases, the variables s and t correspond to the variables $(\hat{x}, y, z)$ of (k-CLP) and (x, b) of (MCFP), respectively. Constraints (8.4b) and (8.5b) are the general constraints (8.1b)–(8.1f) and (8.3b)–(8.3d), respectively, and constraints (8.5c) are referred to as *linking constraints* and correspond to the constraints (8.2).[1] ◁

In this section, we make the following assumptions:

A1 For each $s^* \in \mathbb{R}^{n_1}$ with $g^1(s^*) \leqq 0$, there exists a $t^* \in \mathbb{R}^{n_2}$ with $g^2(t^*) \leqq 0$ and $g^3(s^*, t^*) \leqq 0$.

A2 Every mathematical program which we consider is either infeasible or bounded (i.e., has an optimal solution).

A3 The mathematical program (P^1) is feasible.

These assumptions characterize a parameter set $\mathbb{P}$ determining the objectives (f^1 and f^2) and the constraints (g^1, g^2 and g^3) of (P^1) and $(Q^1_{s^*})$.

Moreover, if we look, for instance, at the following combination

$$\begin{aligned} \min_{s,t^*} \quad & f^1(s) \\ \text{s.t.} \quad & g^1(s) \leqq 0 \\ & g^3(s, t^*) \leqq 0 \\ & t^* \in \arg\min_t f^2(t) \\ & \qquad \text{s.t. } g^2(t) \leqq 0 \end{aligned}$$

we encounter a problem. We observe that this combination does not make much sense for the special case problems (k-CLP) and (MCFP). In the lower-level problem, the variables b_v are indefinite and, therefore, they can take arbitrary demand- and supply-values. Due to the non-negativity of the costs, the (MCFP) outputs a 0-flow with optimal value 0. In this case, no $s \in \mathbb{R}^{n_1}$ can be selected fulfilling the remaining constraints.

Reasonably, for the combinations of the two problems, the input of the objective functions should always adhere to the constraints of the corresponding feasible sets to avoid difficulties with the feasible domain of the objective functions:

A4 f^2 (and f^1) may only be evaluated at points $t \in \mathbb{R}^{n_2}$ with $g^2(t) \leqq 0$ and $g^3(s^*, t) \leqq 0$ for some $s^* \in \mathbb{R}^{n_1}$ with $g^1(s^*) \leqq 0$ (and at points $s \in \mathbb{R}^{n_1}$ fulfilling $g^1(s) \leqq 0$, respectively).

[1] Note that *binary constraints* with the shape $x_e \in \{0, 1\}$ are equivalent to $0 \leq x_e \leq 1$, $0 = \min\{x_e, 1 - x_e\}$ (Marcotte and Savard, 2005).

A5 We require the fulfillment of both constraints g^1 and g^2 before g^3 is checked.

To get a better understanding of the term "integrative", we combine the two general optimization problems, (P^1) and $(Q^1_{s^*})$, and come up with four different non-equivalent bilevel formulations (C_1)–(C_4) and the combined (non-bilevel) bicriteria optimization problem (C_5) with joined objective function.

Definition 8.3 (Combinations (C-problems)):
For the two problems (P^1) and $(Q^1_{s^*})$, we define the following five combinations.

$$
\begin{aligned}
(C_1)\ \min_{s^*,t}\ & f^2(t) && (8.6a)\\
\text{s.t.}\ & g^2(t) \leqq 0 && (8.6b)\\
& g^3(s^*,t) \leqq 0 && (8.6c)\\
& s^* \in \arg\min_s f^1(s) && (8.6d)\\
& \quad \text{s.t.}\ g^1(s) \leqq 0 && (8.6e)
\end{aligned}
\qquad
\begin{aligned}
(C_2)\ \min_{s^*,t}\ & f^2(t) && (8.7a)\\
\text{s.t.}\ & g^2(t) \leqq 0 && (8.7b)\\
& s^* \in \arg\min_s f^1(s) && (8.7c)\\
& \quad \text{s.t.}\ g^1(s) \leqq 0 && (8.7d)\\
& \quad g^3(s,t) \leqq 0 && (8.7e)
\end{aligned}
$$

$$
\begin{aligned}
(C_3)\ \min_{s,t^*}\ & f^1(s) && (8.8a)\\
\text{s.t.}\ & g^1(s) \leqq 0 && (8.8b)\\
& t^* \in \arg\min_t f^2(t) && (8.8c)\\
& \quad \text{s.t.}\ g^2(t) \leqq 0 && (8.8d)\\
& \quad g^3(s,t) \leqq 0 && (8.8e)
\end{aligned}
\qquad
\begin{aligned}
(C_4)\ \min_{s^*,t^*}\ & f^1(s^*) && (8.9a)\\
\text{s.t.}\ & (s^*,t^*) \in \arg\min_{s,t} f^2(t) && (8.9b)\\
& \quad \text{s.t.}\ g^1(s) \leqq 0 && (8.9c)\\
& \quad g^2(t) \leqq 0 && (8.9d)\\
& \quad g^3(s,t) \leqq 0 && (8.9e)
\end{aligned}
$$

$$
\begin{aligned}
(C_5)\ \min_{s,t}\ & (f^1(s), f^2(t)) && (8.10a)\\
\text{s.t.}\ & g^1(s) \leqq 0 && (8.10b)\\
& g^2(t) \leqq 0 && (8.10c)\\
& g^3(s,t) \leqq 0 && (8.10d)
\end{aligned}
$$

◇

Observation 8.4:
The assumptions **A1** to **A3** ensure feasibility of all combinations (C_i), $i = 1, \ldots, 5$. ◁

Remark 8.5:
In the following, we give some interpretations regarding the special cases, i.e., from the viewpoint of the programs (k-CLP) and (MCFP).

(C_1): The optimality of (k-CLP) should be guaranteed and from all the optimal locations, we choose the location inducing the cheapest optimal flow w.r.t. (MCFP).

(C_2): Locations are determined by feasible flows with arbitrary "b-values", therefore the lower-level problem is often infeasible or consists of a single choice of k locations. Hence, the model determines the flow with minimal cost such that the corresponding (k-CLP) is feasible, i.e., a minimum cost flow subject to k sources with appropriate supply- and demand-values.

(C_3): The optimality of (k-CLP) should be guaranteed but the corresponding optimal flow w.r.t. (MCFP) depends on the chosen optimal location.

(C_4): The optimality of (MCFP) under all possible/feasible locations of (k-CLP) should be guaranteed and from all the optimal flows, we choose the flow-location-tuple with the best location w.r.t. objective function f^1 of (k-CLP).

(C_5) *Multicriteria approach*: We seek all possible combinations of feasible location-assignments and feasible flows which cannot be dominated by any other combination. ◁

Remark 8.6:
All other possible bilevel combinations of the two programs (P^1) and ($Q^1_{s^*}$) are equivalent to one of the combinations from Definition 8.3, reduce to simple optimization problems with only one objective function, or are not relevant due to **A1** to **A5**. Moreover, assumption **A3** is only needed to guarantee feasibility of the combined problems and is not relevant for the next results. ◁

In the following, we derive relations between these combinations (for an arbitrary but fixed parameter in $\mathbb{P}$) with respect to feasibility and optimality (or efficiency). Two relations are trivially obtained due to definition and assumption **A1**:

Observation 8.7:
Combinations (C_1) and (C_4) define the two lexicographic programs for (C_5):

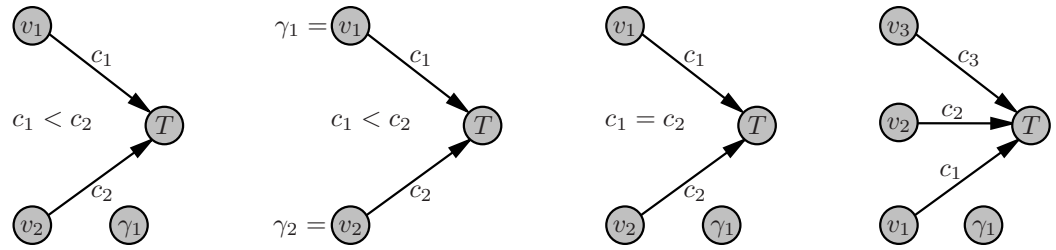
- $(\tilde{s}, \tilde{t})$ optimal for (C_1) $\Longleftrightarrow$ $(\tilde{s}, \tilde{t})$ defines the lexicographic minimum of (C_5) in normal order
- $(\tilde{s}, \tilde{t})$ optimal for (C_4) $\Longleftrightarrow$ $(\tilde{s}, \tilde{t})$ defines the lexicographic minimum of (C_5) in reversed order

In particular, feasible solutions for (C_1) or (C_4) are weakly efficient for (C_5). ◁

Theorem 8.8:

(i) $(\tilde{s}, \tilde{t})$ *feasible for* (C_1) $\Longrightarrow$ $(\tilde{s}, \tilde{t})$ *feasible for* (C_2)

(ii) $(\tilde{s}, \tilde{t})$ *feasible for* (C_4) $\Longrightarrow$ $(\tilde{s}, \tilde{t})$ *feasible for* (C_3)



(a) Remark 8.9 (b) Theorem 8.10(ii) (c) Theorem 8.10(iv) (d) Example 8.13

Figure 8.1: Special cases (counterexamples)

Proof:
We only prove (i); the proof for (ii) works analogously.
Let $(\tilde{s}, \tilde{t}) \in \mathbb{R}^{n_1+n_2}$ be feasible for (C_1). Then, $\tilde{s}$ is feasible for the lower-level problem (8.7c)–(8.7e) w.r.t. $\tilde{t}$ of (C_2).
Suppose there exists a $\bar{s} \in \mathbb{R}^{n_1}$ such that $g^1(\bar{s}) \leqq 0$, $g^3(\bar{s}, \tilde{t}) \leqq 0$ and $f^1(\bar{s}) < f^1(\tilde{s})$. Then, $\bar{s}$ is feasible for the lower-level problem of (C_1) which is a contradiction. □

Remark 8.9:
The reverse directions of Theorem 8.8 are not true in general. Consider the special case problems given in Figure 8.1a, where $k = 1$, $w_1 = 1$, cost c_1 is less than c_2, and γ_1 is closer to v_2 than to v_1. Both induced flow-location-tuples corresponding to the assignments $y_1 = 1$ and $y_2 = 1$, respectively, are feasible for (C_2) and (C_3). In addition, $y_1 = 1$ and $y_2 = 1$ induce optimal solutions for (C_2) and (C_3), respectively. However, only the second assignment ($y_2 = 1$) produces a feasible solution of (C_1) and only the first assignment ($y_1 = 1$) produces a feasible solution of (C_4). ◁

Theorem 8.10:

(i) $(\tilde{s}, \tilde{t})$ *optimal for* $(C_1) \Longrightarrow (\tilde{s}, \tilde{t})$ *optimal for* (C_3)

(ii) $(\tilde{s}, \tilde{t})$ *optimal for* $(C_3) \Longrightarrow (\tilde{s}, \tilde{t})$ *feasible for* (C_1) *but in general not optimal*

(iii) $(\tilde{s}, \tilde{t})$ *optimal for* $(C_4) \Longrightarrow (\tilde{s}, \tilde{t})$ *optimal for* (C_2)

(iv) $(\tilde{s}, \tilde{t})$ *optimal for* $(C_2) \Longrightarrow (\tilde{s}, \tilde{t})$ *feasible for* (C_4) *but in general not optimal*

Proof:
(i)+(iii): We only prove (iii) since the proof for (i) works analogously.
Let $(\tilde{s}, \tilde{t}) \in \mathbb{R}^{n_1+n_2}$ be optimal for (C_4).

Claim: $\tilde{s}$ is feasible and optimal for the lower-level problem (8.7c)–(8.7e) w.r.t. $\tilde{t}$ of (C_2).

Proof:
Clearly, $\tilde{s}$ is feasible. Suppose there is some $\bar{s} \in \mathbb{R}^{n_1}$ such that $g^1(\bar{s}) \leqq 0$, $g^3(\bar{s}, \tilde{t}) \leqq 0$ and $f^1(\bar{s}) < f^1(\tilde{s})$. Then, $(\bar{s}, \tilde{t})$ is feasible for (C_4) with objective value $f^1(\bar{s}) < f^1(\tilde{s})$, which is a contradiction. ♢

Hence, $(\tilde{s}, \tilde{t})$ is feasible for problem (C_2). Suppose there is some $(\bar{s}, \bar{t}) \in \mathbb{R}^{n_1+n_2}$ feasible for (C_2) with $f^2(\bar{t}) < f^2(\tilde{t})$. Thus, $(\bar{s}, \bar{t})$ is also feasible for the lower-level problem (8.9b)–(8.9e) of (C_4) with objective value $f^2(\bar{t}) < f^2(\tilde{t})$ which is a contradiction to the feasibility of $(\tilde{s}, \tilde{t})$ for (C_4).

(ii)+(iv): We only prove (ii) since the proof for (iv) works analogously.
Let $(\tilde{s}, \tilde{t}) \in \mathbb{R}^{n_1+n_2}$ be optimal for (C_3). Suppose there exists some $\bar{s} \in \mathbb{R}^{n_1}$ with $g^1(\bar{s}) \leqq 0$ and $f^1(\bar{s}) < f^1(\tilde{s})$. Then, due to **A1**, there is some $t' \in \mathbb{R}^{n_2}$ such that $g^2(t') \leqq 0$ and $g^3(\bar{s}, t') \leqq 0$. As a consequence of assumption **A2**, we can find a $\bar{t} \in \mathbb{R}^{n_2}$ with $g^2(\bar{t}) \leqq 0$, $g^3(\bar{s}, \bar{t}) \leqq 0$ and with minimal value $f^2(\bar{t})$ under these constraints. Thus, $(\bar{s}, \bar{t})$ is feasible for (C_3) with objective value $f^1(\bar{s}) < f^1(\tilde{s})$ which is a contradiction. Hence, $(\tilde{s}, \tilde{t})$ is feasible for (C_1).

For (ii) and (iv), it remains to show that, in general, $(\tilde{s}, \tilde{t})$ is not an optimal solution of (C_1) and (C_4), respectively. Consider, for instance, the special case problems given in Figure 8.1b and Figure 8.1c, respectively.
In Figure 8.1b, the clients coincide with the potential locations, $k = 1$, $w_1 = w_2$, and cost c_1 is smaller than c_2. Both possible assignments, $y_1 = 1$ and $y_2 = 1$, induce optimal solutions for the lower-level problem (8.6d)–(8.6e) of (C_1), but only the associated flow of the first assignment $(y_1 = 1)$ produces an optimal solution of (C_1). However, both assignments with their obviously associated flows are feasible and optimal for (C_3).
In Figure 8.1c, $k = 1$, $w_1 = 1$, $c_1 = c_2$, and γ_1 is closer to v_2 than to v_1. Both possible assignments, $y_1 = 1$ and $y_2 = 1$, induce optimal solutions for the lower-level problem (8.9b)–(8.9e) of (C_4), but only the second assignment $(y_2 = 1)$ produces an optimal solution of (C_4). However, both assignments with their obviously associated flows are feasible and optimal for (C_2). □

The combination of the statements (ii) and (iv) from Theorem 8.10 with Observation 8.7 and Theorem 8.8, respectively, yields the following two corollaries:

Corollary 8.11: *If $(\tilde{s}, \tilde{t})$ is optimal for (C_3) or (C_2), then $(\tilde{s}, \tilde{t})$ defines a weakly efficient solution of (C_5).*

Corollary 8.12:

(i) $(\tilde{s}, \tilde{t})$ *optimal for* $(C_3) \Longrightarrow (\tilde{s}, \tilde{t})$ *feasible for* (C_2) *but in general not optimal*

(ii) $(\tilde{s}, \tilde{t})$ *optimal for* $(C_2) \Longrightarrow (\tilde{s}, \tilde{t})$ *feasible for* (C_3) *but in general not optimal*

opt. / feas.	(C_1)	(C_2)	(C_3)	(C_4)
(C_1)	•	-	✓	-
(C_2)	✓′	•	✓	✓*
(C_3)	✓*	✓	•	✓′
(C_4)	-	✓	-	•

Table 8.1: This table should be read from the columns to the rows, e.g., the *checkmark* ✓ in the cell induced by column (C_2) and row (C_4) says: "An optimal solution of (C_2) is also feasible for (C_4)." The *star* * remarks that this solution is also optimal for the row-problem and the *apostrophe* ′ that we only need feasibility for the column-problem instead of optimality.

Proof:
To get the two results, combine Theorem 8.8 with the statements (ii) and (iv) from Theorem 8.10, respectively. The example of Remark 8.9 shows that an optimal solution of one problem is not optimal for the other problem, in general, since only the first assignment ($y_1 = 1$) produces an optimal solution for (C_2) and only the second assignment ($y_2 = 1$) produces an optimal solution for (C_3). □

The detected relations between the different models are summarized in Table 8.1. For the cells without a checkmark or with only a checkmark and without a star, we have seen counterexamples which prove that there are instances which are optimal for the column-problem but in general not feasible or optimal for the row-problem.

Consider now the bicriteria optimization problem (C_5). Since the optimal solutions of the problems (C_1) and (C_4) are characterized by means of lexicographic optima in Observation 8.7, we focus on the remaining problems (C_2) and (C_3). Corollary 8.11 implies that optimal solutions of (C_2) or (C_3) induce weakly efficient solutions of (C_5). The following example shows that the reverse direction, i.e., an efficient solution of (C_5) induces an optimal solution of another program under consideration, is not true in general.

Example 8.13:
Let us consider the special case problems given in Figure 8.1d with $k = 1$, $w_1 = 1$, $c_1 = 2$, $c_2 = 1$ and $c_3 = \frac{1}{2}$. The distances are given by $d(\gamma_1, v_1) = \frac{1}{2}$, $d(\gamma_1, v_2) = 1$ and $d(\gamma_1, v_3) = 2$. For problem (C_5), we obtain three different feasible solutions x^1, x^2 and x^3 with corresponding outcomes $y^1 := (\frac{1}{2}, 2)$, $y^2 := (1, 1)$ and $y^3 := (2, \frac{1}{2})$, respectively. Thus, all three solutions are efficient for (C_5), but only x^1 and x^3 are feasible (and, thus, optimal) for (C_1) and (C_4), respectively. x^1, x^2 and x^3 are all feasible for (C_2) and (C_3), but only x^1 and x^3 are optimal for (C_3) and (C_2), respectively. ◁

Nevertheless, if we have an efficient solution of (C_5), we can state that this is a feasible solution for the problems (C_2) and (C_3).

Theorem 8.14: $(\tilde{s}, \tilde{t})$ *efficient for* $(C_5) \Longrightarrow (\tilde{s}, \tilde{t})$ *feasible for* (C_2) *and* (C_3)

Proof:
Let $(\tilde{s}, \tilde{t}) \in \mathbb{R}^{n_1+n_2}$ be an efficient solution for (C_5). In the following, we only proof feasibility for (C_2) since, due to the similarities of both problems, feasibility for (C_3) can be proven analogously.

Claim: $\tilde{s}$ is feasible and optimal for the lower-level problem (8.7c)–(8.7e) w.r.t. $\tilde{t}$ of (C_2).

Proof:
Clearly, $\tilde{s}$ is feasible. Suppose there is some $\bar{s} \in \mathbb{R}^{n_1}$ such that $g^1(\bar{s}) \leqq 0$, $g^3(\bar{s}, \tilde{t}) \leqq 0$ and $f^1(\bar{s}) < f^1(\tilde{s})$. Then, $(\bar{s}, \tilde{t})$ is feasible for (C_5) dominating $(\tilde{s}, \tilde{t})$, which is a contradiction. ♢

Hence, $(\tilde{s}, \tilde{t})$ is feasible for (C_2). □

Remark 8.15:
The results including the bicriteria problem (C_5) can be summarized as follows:

- $(\tilde{s}, \tilde{t})$ *optimal* for (C_1) or $(C_4) \Longrightarrow (\tilde{s}, \tilde{t})$ *efficient* for (C_5)
  (see Observation 8.7)
- $(\tilde{s}, \tilde{t})$ *optimal* for (C_2) or $(C_3) \Longrightarrow (\tilde{s}, \tilde{t})$ *weakly efficient* for (C_5)
  (see Corollary 8.11)
- $(\tilde{s}, \tilde{t})$ *feasible* for (C_1) or $(C_4) \Longrightarrow (\tilde{s}, \tilde{t})$ *weakly efficient* for (C_5)
  (see Observation 8.7)
- $(\tilde{s}, \tilde{t})$ *efficient* for $(C_5) \Longrightarrow (\tilde{s}, \tilde{t})$ *feasible* for (C_2) and (C_3) but in general not *optimal* for (C_2) or (C_3) and not *feasible* for (C_1) or (C_4)
  (see Theorem 8.14 and Example 8.13) ◁

8.3 The Multicriteria Case

In this section, optimality is relaxed to efficiency and, thus, multicriteria objective functions are considered in all problems.

Definition 8.16:
With the same assumptions as in Definition 8.1, except for the two objective functions f^1 and f^2 which should fulfill $f^1 : \mathbb{R}^{n_1} \to \mathbb{R}^{p_1}$ and $f^2 : \mathbb{R}^{n_2} \to \mathbb{R}^{p_2}$ for $p_1, p_2 \in \mathbb{N}$, we

define the two multicriteria optimization problems (P^1) and $(Q^1_{s^*})$ as

$$(P^1) \quad \min_s \ (f^1_1(s), \ldots, f^1_{p_1}(s)) \tag{8.11a}$$
$$\text{s.t.} \ \ g^1(s) \leqq 0 \tag{8.11b}$$

$$(Q^1_{s^*}) \quad \min_t \ (f^2_1(t), \ldots, f^2_{p_2}(t)) \tag{8.12a}$$
$$\text{s.t.} \ \ g^2(t) \leqq 0 \tag{8.12b}$$
$$g^3(s^*, t) \leqq 0 \tag{8.12c}$$

◇

We make the same assumptions **A1**–**A5** as in the preceding section except for **A2**, it should be substituted with:

A2′ Every mathematical program which we consider is either infeasible or bounded (i.e., the *ideal point* y^I exists) and externally stable.

Definition 8.17 (Combinations (*C*-problems)):
Analogous to Definition 8.3, we define the optimization problems (C_1) to (C_5) with

$$f^1(s) \coloneqq (f^1_1(s), \ldots, f^1_{p_1}(s)) \qquad \text{and} \qquad f^2(t) \coloneqq (f^2_1(t), \ldots, f^2_{p_2}(t)) \ . \qquad \diamond$$

For (C_1) to (C_4), with nearly similar proofs as for the single-criterion case, all relations detected in the previous section can be established again:

Corollary 8.18: *All relations mentioned in Table 8.1, yet substituting optimality with efficiency, are also valid in the multicriteria context.*

In the following, we want to prove relations between (C_5) and the other combinations. First, we define the notion of "piecewise efficient" solutions for some MOP according to Definition 2.1.

Definition 8.19 (Piecewise Efficiency):
A weakly efficient solution $\hat{x} \in X$ of MOP with outcome $\hat{y} \in Y \subseteq \mathbb{R}^p$ is called *piecewise efficient according to the coordinates* $I \coloneqq \{i_1, \ldots, i_\xi\} \subseteq \{1, \ldots, p\}$, $\xi > 0$, if the projection of its outcome to these coordinates $\hat{y}_I$ is nondominated in the corresponding projected outcome set, i.e., there is no other $y \in Y$ with

$$y_j \leq \hat{y}_j \ \forall j \in I \quad \text{and} \quad y_I \neq \hat{y}_I \ ,$$

where z_I denotes the subvector of a vector $z \in \mathbb{R}^p$ corresponding to the coordinates I in a fixed order. ◇

The following Lemma shows that lexicographic minima are a subset of piecewise efficient solutions according to some coordinates.

Lemma 8.20: *For* MOP*, let* $I = \{i_1, \ldots, i_\ell\} \subseteq \{1, \ldots, p\}$ *be given with* $|I| = \ell$*. Let* $x \in X$ *be the lexicographic minimum w.r.t. an order* $(i_1, \ldots, i_\ell, i_{\ell+1}, \ldots, i_p)$ *with* $(i_{\ell+1}, \ldots, i_p) \in \{1, \ldots, p\}^{p-\ell}$ *and* $i_j \neq i_k$ *for* $j \neq k$*. Then,* x *is piecewise efficient according to the coordinates* I*.*

Proof:
Suppose x is not piecewise efficient according to I. Then, there exists some $\bar{x} \in X$ with $f_I(\bar{x}) \leq f_I(x)$. This is a contradiction to the lexicographic optimality. □

Theorem 8.21: $(\tilde{s}, \tilde{t})$ *efficient for* (C_1) *or* $(C_4) \Longrightarrow (\tilde{s}, \tilde{t})$ *efficient for* (C_5)

Proof:
Due to the obvious similarity between the two problems, we only consider problem (C_4); the proof for problem (C_1) works analogously.
Let $(\tilde{s}, \tilde{t}) \in \mathbb{R}^{n_1+n_2}$ be efficient for (C_4).

We immediately get feasibility for (C_5). Suppose there exists some $(\bar{s}, \bar{t}) \in \mathbb{R}^{n_1+n_2}$ feasible for (C_5) with $\left(f^1(\bar{s}), f^2(\bar{t})\right) \leq \left(f^1(\tilde{s}), f^2(\tilde{t})\right)$.

Case 1 $\left(f^2(\bar{t}) \leq f^2(\tilde{t})\right)$:
This is a contradiction to the efficiency of $\tilde{s}$ for the lower-level problem (8.9b)–(8.9e) of (C_4).

Case 2 $\left(f^2(\bar{t}) = f^2(\tilde{t}) \text{ and } f^1(\bar{s}) \leq f^1(\tilde{s})\right)$:
It follows that $(\bar{s}, \bar{t})$ is feasible for (C_4) with objective value $f^1(\bar{s}) \leq f^1(\tilde{s})$, which is a contradiction. □

Remark 8.22:
As we have seen in Example 8.13, the inverse direction of Theorem 8.21 is not true in general. ◁

An analogous version of Theorem 8.14 can be immediately obtained with an analogous proof.

Corollary 8.23: $(\tilde{s}, \tilde{t})$ *efficient for* $(C_5) \Longrightarrow (\tilde{s}, \tilde{t})$ *feasible for* (C_2) *and* (C_3)

In the single-criterion case, we have seen in Observation 8.7 that a lexicographic minimum (in the right order) defines an optimal solution for (C_1) and (C_4), respectively. In the multicriteria case, we have noted in Remark 8.22 that an efficient solution of (C_5) is in general not feasible for the problems (C_1) or (C_4). But what about the lexicographic minima? From Lemma 8.20, we know that all lexicographic minima are piecewise efficient for (C_5). In the following, we see that the piecewise efficient solutions of (C_5) according to special coordinates correspond to the feasible solutions of the problems (C_1) or (C_4).

Theorem 8.24: *For $(\tilde{s},\tilde{t}) \in \mathbb{R}^{n_1+n_2}$, it holds: $(\tilde{s},\tilde{t})$ is piecewise efficient for (C_5) according to the coordinates $I \coloneqq \{1,\dots,p_1\}$ or $I \coloneqq \{p_1+1,\dots,p_1+p_2\}$, respectively, if and only if $(\tilde{s},\tilde{t})$ is feasible for (C_1) or (C_4), respectively.*

Proof:
We only consider the case for problem (C_4) since the proof for (C_1) works analogously.
$\Rightarrow$:
For (C_5), let $(\tilde{s},\tilde{t}) \in \mathbb{R}^{n_1+n_2}$ be piecewise efficient according to $I \coloneqq \{p_1+1,\dots,p_1+p_2\}$.

Claim: $(\tilde{s},\tilde{t})$ is efficient for the lower-level problem (8.9b)–(8.9e) of (C_4).

Proof:
Clearly, $(\tilde{s},\tilde{t})$ is feasible for the lower-level problem. Suppose there exists some $(\bar{s},\bar{t}) \in \mathbb{R}^{n_1+n_2}$ feasible for the lower-level problem with $f^2(\bar{t}) \leq f^2(\tilde{t})$, but this is a contradiction to the definition of the piecewise efficiency since, in this case, $(\bar{s},\bar{t})$ would dominate $(\tilde{s},\tilde{t})$ in the coordinates I. ♢

Hence, $(\tilde{s},\tilde{t})$ is feasible for (C_4).

$\Leftarrow$:
Let $(\tilde{s},\tilde{t}) \in \mathbb{R}^{n_1+n_2}$ be feasible for (C_4). Suppose there exists some $(\bar{s},\bar{t}) \in \mathbb{R}^{n_1+n_2}$ feasible for (C_5) with $f^2(\bar{t}) \leq f^2(\tilde{t})$. Then, $(\bar{s},\bar{t})$ is also feasible for the lower-level problem of (C_4) with $f^2(\bar{t}) \leq f^2(\tilde{t})$, which is a contradiction to the feasibility of $(\tilde{s},\tilde{t})$ for (C_4). □

Remark 8.25:
In the following, we sum up all detected relations including problem (C_5) and incorporating the results from Corollary 8.18. We denote by X^{C_i}, $i = 1,\dots,5$, the feasible set of problem (C_i), by $X_E^{C_i}$ the set of all efficient solutions of problem (C_i), and by X_E^I the set of all piecewise efficient solutions of problem (C_5) according to the coordinates I. Then, we get:

- $X_E^{C_1} \cup X_E^{C_4} \subseteq X_E^{C_5} \subseteq X^{C_2} \cap X^{C_3}$
- $X_E^{C_1} \subseteq X_E^{C_3} \subseteq X_E^{\{1,\dots,p_1\}} = X^{C_1} \subseteq X^{C_2}$
- $X_E^{C_4} \subseteq X_E^{C_2} \subseteq X_E^{\{p_1+1,\dots,p_1+p_2\}} = X^{C_4} \subseteq X^{C_3}$ ◁

8.4 Extensions of the Two Optimization Problems

In this section, we point out ideas and basic results for some possible extensions of the two problems (P^1) and $(Q^1_{s^*})$ to provide more general statements and to get more general combinations matching more problems. We concentrate on the multicriteria case but it is also mentioned if a result only holds for the single-criterion case.

8.4.1 Combined Objective Functions

First, we consider one combined objective function depending on both parts of the variables $s \in \mathbb{R}^{n_1}$ and $t \in \mathbb{R}^{n_2}$. Since, in the previous sections, the problems (P^1) and $(Q^1_{s^*})$ are only depending on one part of the variables, either on $s \in \mathbb{R}^{n_1}$ or on $t \in \mathbb{R}^{n_2}$, we initially have to transform the second not considered part of the variables in the problem with the future combined objective function to real variables; otherwise, the combined objective function could be considered as a simple objective function depending only on one part with a constant second part.

Analogously as in Section 8.3, in this first part of this subsection, we consider the following two optimization problems.

Definition 8.26:
With the same assumptions as in Definition 8.16, except for the objective function f^1 which should fulfill $f^1 : \mathbb{R}^{n_1+n_2} \to \mathbb{R}^{p_1}$, we define the two multicriteria optimization problems (P^2) and $(Q^2_{s^*})$ as

$$
\begin{aligned}
(P^2) \quad \min_{s,t} \quad & f^1(s,t) && (8.13a)\\
\text{s. t.} \quad & g^1(s) \leqq 0 && (8.13b)\\
& g^2(t) \leqq 0 && (8.13c)
\end{aligned}
\qquad
\begin{aligned}
(Q^2_{s^*}) \quad \min_{t} \quad & f^2(t) && (8.14a)\\
\text{s. t.} \quad & g^2(t) \leqq 0 && (8.14b)\\
& g^3(s^*,t) \leqq 0 && (8.14c)
\end{aligned}
$$

◇

Remark 8.27:
In a practical scenario, it can also make sense to consider the following two problems:

$$
\begin{aligned}
(\hat{P}^2) \quad \min_{s,t} \quad & f^1(s,t) && (8.15a)\\
\text{s. t.} \quad & g^1(s) \leqq 0 && (8.15b)\\
& g^2(t) \leqq 0 && (8.15c)\\
& g^3(s,t) \leqq 0 && (8.15d)
\end{aligned}
\qquad
\begin{aligned}
(\hat{Q}^2_{s^*}) \quad \min_{t} \quad & f^2(t) && (8.16a)\\
\text{s. t.} \quad & g^2(t) \leqq 0 && (8.16b)\\
& g^3(s^*,t) \leqq 0 && (8.16c)
\end{aligned}
$$

However, since the considerations of these two problems result in the same combinations (see below), this variant can be treated in the same way as the upper variant. Here, we only assume that for the interpretation and evaluation of objective function f^1 the input has to fulfill the constraints g^1 and g^2 (see **A4′**). ◁

As in Section 8.3, we assume analogous conditions as **A1**, **A2′**, **A3**, **A5** and additionally:

A4′ f^2 (and f^1) may only be evaluated at points $t \in \mathbb{R}^{n_2}$ with $g^2(t) \leqq 0$ and $g^3(s^*,t) \leqq 0$ for some $s^* \in \mathbb{R}^{n_1}$ with $g^1(s^*) \leqq 0$ (and at points $(s,t) \in \mathbb{R}^{n_1+n_2}$ fulfilling $g^1(s) \leqq 0$ and $g^2(t) \leqq 0$, respectively).

Definition 8.28 (Combinations (*D*-problems)):
Analogous to Definition 8.17, we define the optimization problems (D_1) to (D_5) obtained by substituting $f^1(s)$ with $f^1(s,t)$ in (C_1) to (C_5) and further at the combinations

$$
\begin{aligned}
(D_6)\ \min_{s^*,t^*}\ & f^2(t^*) && (8.17a)\\
\text{s.t.}\ & \\
& g^3(s^*,t^*) \leqq 0 && (8.17b)\\
& (s^*,t^*) \in \arg\min_{s,t} f^1(s,t) && (8.17c)\\
& \qquad \text{s.t. } g^1(s) \leqq 0 && (8.17d)\\
& \qquad\quad g^2(t) \leqq 0 && (8.17e)
\end{aligned}
$$

$$
\begin{aligned}
(D_7)\ \min_{s,t^*}\ & f^2(t^*) && (8.18a)\\
& \text{s.t. } g^1(s) \leqq 0 && (8.18b)\\
& t^* \in \arg\min_{t} f^1(s,t) && (8.18c)\\
& \qquad \text{s.t. } g^2(t) \leqq 0 && (8.18d)\\
& \qquad\quad g^3(s,t) \leqq 0 && (8.18e)
\end{aligned}
$$

$$
\begin{aligned}
(D_8)\ \min_{s^*,t^*}\ & f^2(t^*) && (8.19a)\\
\text{s.t.}\ & \\
& (s^*,t^*) \in \arg\min_{s,t} f^1(s,t) && (8.19b)\\
& \qquad \text{s.t. } g^1(s) \leqq 0 && (8.19c)\\
& \qquad\quad g^2(t) \leqq 0 && (8.19d)\\
& \qquad\quad g^3(s,t) \leqq 0 && (8.19e)
\end{aligned}
$$

◇

Remark 8.29:
The combination induced in (D_6) and (D_7) would be redundant or not relevant for us in the previous sections. Equivalence between (D_1) and (D_8) (as in the previous sections; cf. Observation 8.7) cannot be proven since the objective function of the lower-level problem of (D_1) depends on t. ◁

Observing that the combinations (C_1) to (C_5) are special cases of the combinations (D_1) to (D_5), we only consider the relations between those D-problems for which a relation was found for the corresponding C-problems. Since the corresponding C-problem of (D_8) was equivalent to (C_1), we can also exclude relations for (D_8) considering relations for (C_1). We omit the proofs for relations which are similar proven as in the previous sections and only show the relations in detail which are not obviously obtained and two additional results for some extended assumptions. We refer to Table 8.2 for the relations found, except the two results which need additional assumptions (see Theorems 8.32 and 8.34).

Since, in general, (D_8) is not equivalent to (D_1), we have to check the feasible-relation between (D_2) and (D_8).

feas. \ eff.	(D_1)	(D_2)	(D_4)	(D_6)	(D_8)
(D_1)	•	-	-	✓′	-
(D_2)	✓′	•	✓*	✓′	✓′
(D_3)	-	✓	✓′	-	-
(D_4)	-	✓	•	-	-
(D_7)	-	-	-	✓′	✓′
(D_8)	-	-	-	✓′	•

Table 8.2: This table should be read in the same way as Table 8.1. Rows and columns without checkmarks have been omitted.

Theorem 8.30: $(\tilde{s}, \tilde{t})$ *feasible for* $(D_8) \Longrightarrow (\tilde{s}, \tilde{t})$ *feasible for* (D_2)

Proof:
Let $(\tilde{s}, \tilde{t}) \in \mathbb{R}^{n_1+n_2}$ be feasible for (D_8). Then, $\tilde{s}$ is feasible for the lower-level problem w.r.t. $\tilde{t}$ of (D_2).
Suppose there exists some $\bar{s} \in \mathbb{R}^{n_1}$ with $g^1(\bar{s}) \leqq 0,\ g^3(\bar{s}, \tilde{t}) \leqq 0$ and $f^1(\bar{s}, \tilde{t}) \leq f^1(\tilde{s}, \tilde{t})$. Thus, $(\bar{s}, \tilde{t})$ is feasible for the lower-level problem of (D_8), which is a contradiction. □

The relation between (D_1) and (D_3) could not be proven with our above stated assumptions alone. However, we can give an additional assumption such that we can prove a relation between them:

A6 f^1 is *nondominance-preserving* in the first part, i.e., for all $s^1, s^2 \in \mathbb{R}^{n_1}$, $t^1, t^2 \in \mathbb{R}^{n_2}$ with $g^1(s^1) \leqq 0$, $g^1(s^2) \leqq 0$, $g^2(t^1) \leqq 0$, and $g^2(t^2) \leqq 0$, it holds:

$$f^1(s^1, t^1) \not\leqq f^1(s^2, t^1) \Longrightarrow f^1(s^1, t^2) \not\leqq f^1(s^2, t^2)$$

Remark 8.31:

- An equivalent formulation of **A6** is:
  For all $s^1, s^2 \in \mathbb{R}^{n_1}$, $t^1, t^2 \in \mathbb{R}^{n_2}$ with $g^1(s^1) \leqq 0$, $g^1(s^2) \leqq 0$, $g^2(t^1) \leqq 0$, and $g^2(t^2) \leqq 0$, it holds:

  $$f^1(s^1, t^1) \leq f^1(s^2, t^1) \Longrightarrow f^1(s^1, t^2) \leq f^1(s^2, t^2)$$

- A sufficient condition for f^1 to fulfill **A6** is $f^1(s, t) = f^{1,1}(s) + f^{1,2}(t)$ with $f^{1,1} : \mathbb{R}^{n_1} \to \mathbb{R}^{p_1}$ and $f^{1,2} : \mathbb{R}^{n_2} \to \mathbb{R}^{p_1}$, which can for example be achieved by linear objective functions: $f^1(s, t) = Cs + Dt$ with $C \in \mathbb{R}^{p_1 \times n_1}$ and $D \in \mathbb{R}^{p_1 \times n_2}$.

◁

Theorem 8.32: *Let assumption **A6** be satisfied. Then, we have:*

$$(\tilde{s},\tilde{t}) \textit{ efficient for } (D_1) \Longrightarrow (\tilde{s},\tilde{t}) \textit{ feasible for } (D_3)$$

Proof:
Let $(\tilde{s},\tilde{t}) \in \mathbb{R}^{n_1+n_2}$ be efficient for (D_1). Then, $(\tilde{s},\tilde{t})$ is feasible for the lower-level problem of (D_3). Suppose there is a $\bar{t} \in \mathbb{R}^{n_2}$ such that $g^2(\bar{t}) \leqq 0,\ g^3(\tilde{s},\bar{t}) \leqq 0$ and $f^2(\bar{t}) \leq f^2(\tilde{t})$.

Claim: $(\tilde{s},\bar{t})$ is feasible for (D_1).

Proof:
Note that $\tilde{s}$ is feasible for the lower-level problem w.r.t. $\bar{t}$ of (D_1). Suppose there is a $\bar{s} \in \mathbb{R}^{n_1}$ with $g^1(\bar{s}) \leqq 0$ and $f^1(\bar{s},\bar{t}) \leq f^1(\tilde{s},\bar{t})$. Then, $\tilde{s},\bar{s},\tilde{t}$ and $\bar{t}$ fulfill the requirements of **A6** which implies that $f^1(\bar{s},\tilde{t}) \leq f^1(\tilde{s},\tilde{t})$. This is a contradiction to the feasibility of $(\tilde{s},\tilde{t})$ for (D_1). ♢

Thus, $(\tilde{s},\bar{t})$ is feasible for (D_1) with objective value $f^2(\bar{t}) \leq f^2(\tilde{t})$, which is a contradiction. □

In a next step, we consider the new combination (D_6). Note that feasibility of combination (D_6) (as well as (D_1)) cannot be induced from the assumptions **A1** to **A3** since constraint g^3 is checked after the lower-level problem is considered.

Theorem 8.33: $(\tilde{s},\tilde{t})$ *feasible for* $(D_6) \Longrightarrow (\tilde{s},\tilde{t})$ *feasible for* (D_1) *and* (D_8)

Proof:
We only prove the feasibility for (D_1) since the proof for (D_8) works analogously.
Let $(\tilde{s},\tilde{t}) \in \mathbb{R}^{n_1+n_2}$ be feasible for (D_6). Then, $\tilde{s}$ is feasible for the lower-level problem of (D_1) w.r.t. $\tilde{t}$. Suppose there exists some $\bar{s} \in \mathbb{R}^{n_1}$ with $g^1(\bar{s}) \leqq 0$ and $f^1(\bar{s},\tilde{t}) \leq f^1(\tilde{s},\tilde{t})$. Thus, $(\bar{s},\tilde{t})$ is feasible for the lower-level problem of (D_6), which is a contradiction to the feasibility of $(\tilde{s},\tilde{t})$ for (D_6) □

We could not prove the reverse direction in general, but if we look at the single-criterion case, we can state the following theorem:

Theorem 8.34: *Let* (D_6) *and* (D_8) *be feasible and with single-criterion objective function* f^1*. Then, it holds:*

$$(\tilde{s},\tilde{t}) \textit{ feasible for } (D_8) \Longrightarrow (\tilde{s},\tilde{t}) \textit{ feasible for } (D_6)$$

In particular, both problems are equivalent (cf. Theorem 8.33).

Proof:
Let $(\tilde{s}, \tilde{t}) \in \mathbb{R}^{n_1+n_2}$ be feasible for (D_8). From the assumptions, we have that there is a feasible solution $(s', t') \in \mathbb{R}^{n_1+n_2}$ for (D_6). Since $g^1(\tilde{s}) \leqq 0$ and $g^2(\tilde{t}) \leqq 0$, we have $f^1(s', t') \leq f^1(\tilde{s}, \tilde{t})$.
On the other hand, we know that all three constraints are fulfilled by (s', t') giving us feasibility of (s', t') for the lower-level problem of (D_8). Then, we get $f^1(\tilde{s}, \tilde{t}) \leq f^1(s', t')$ implying $f^1(s', t') = f^1(\tilde{s}, \tilde{t})$. Hence, $(\tilde{s}, \tilde{t})$ is optimal for the lower-level problem and, thus, feasible for (D_6). □

Theorem 8.35: $(\tilde{s}, \tilde{t})$ *efficient for* $(D_6) \Longrightarrow (\tilde{s}, \tilde{t})$ *efficient for* (D_5)

Proof:
Let $(\tilde{s}, \tilde{t}) \in \mathbb{R}^{n_1+n_2}$ be efficient for (D_6). Then, $(\tilde{s}, \tilde{t})$ is feasible for (D_5). Suppose there exists some $(\bar{s}, \bar{t}) \in \mathbb{R}^{n_1+n_2}$ feasible for (D_5) with $(f^1(\bar{s}, \bar{t}), f^2(\bar{t})) \leq (f^1(\tilde{s}, \tilde{t}), f^2(\tilde{t}))$.

Case 1 $\left(f^1(\bar{s}, \bar{t}) \leq f^1(\tilde{s}, \tilde{t})\right)$:
This is a contradiction to the feasibility of $(\tilde{s}, \tilde{t})$ for (D_6).

Case 2 $\left(f^1(\bar{s}, \bar{t}) = f^1(\tilde{s}, \tilde{t}) \text{ and } f^2(\bar{t}) \leq f^2(\tilde{t})\right)$:
It follows that $(\bar{s}, \bar{t})$ is feasible for (D_6) with objective value $f^2(\bar{t}) \leq f^2(\tilde{t})$, which is a contradiction. □

The following observation summarizes our results concerning problem (D_5).

Observation 8.36:
We use the same notation as in Remark 8.25.

- $X_E^{D_4} \cup X_E^{D_6} \cup X_E^{D_8} \subseteq X_E^{D_5} \subseteq X^{D_2}$
- $X^{D_6} \subseteq X_E^{\{1,\dots,p_1\}} = X^{D_8} \subseteq X^{D_2} \cap X^{D_7}$
- $X_E^{D_4} \subseteq X_E^{D_2} \subseteq X_E^{\{p_1+1,\dots,p_1+p_2\}} = X^{D_4} \subseteq X^{D_3}$ ◁

Furthermore, we now shortly look at the situation in which the second optimization problem has a combined objective function.

Definition 8.37:
With the same assumptions as in Definition 8.16, except for the objective function f^2 which should fulfill $f^2 : \mathbb{R}^{n_1+n_2} \to \mathbb{R}^{p_2}$, we define the two multicriteria optimization problems (P^3) and (Q^3) as

$$(P^3) \quad \min_{s} \; f^1(s) \tag{8.20a}$$
$$\text{s. t.} \; g^1(s) \leqq 0 \tag{8.20b}$$

$$(Q^3) \quad \min_{s,t} \; f^2(s,t) \tag{8.21a}$$
$$\text{s. t.} \; g^1(s) \leqq 0 \tag{8.21b}$$
$$g^2(t) \leqq 0 \tag{8.21c}$$
$$g^3(s,t) \leqq 0 \tag{8.21d}$$

◇

Here, we assume analogous conditions as **A1**, **A2′**, **A3**, **A5** and additionally:

A4″ f^2 (and f^1) may only be evaluated at points $(s,t) \in \mathbb{R}^{n_1+n_2}$ with $g^1(s) \leqq 0$, $g^2(t) \leqq 0$ and $g^3(s,t) \leqq 0$ (and at points $s \in \mathbb{R}^{n_1}$ fulfilling $g^1(s) \leqq 0$, respectively).

Definition 8.38 (Combinations (*E*-problems)):
Analogous to Definition 8.17, we define the optimization problems (E_1) to (E_5) obtained by substituting $f^2(t)$ with $f^2(s,t)$ in (C_1) to (C_5) and further at the combination

$$
\begin{aligned}
(E_9) \quad \min_{s^*,t} \quad & f^1(s^*) && \text{(8.22a)}\\
\text{s.t.} \quad & g^2(t) \leqq 0 && \text{(8.22b)}\\
& s^* \in \arg\min_{s} f^2(s,t) && \text{(8.22c)}\\
& \qquad \text{s.t.} \quad g^1(s) \leqq 0 && \text{(8.22d)}\\
& \qquad\quad g^3(s,t) \leqq 0 && \text{(8.22e)}
\end{aligned}
$$

◇

Remark 8.39:
The combination induced in (E_9) would be redundant in the previous sections. An analogue E-problem as (D_6), but with switched objective functions, would not fulfill condition **A4″** since f^2 should not be evaluated before g^3 is checked. A similar combination as (D_7) would not be relevant for us since the objective function of the corresponding lower-level problem would be constant and, thus, the whole E-problem would reduce to a normal optimization problem with one objective function. Moreover, from previous sections, we can imply that (E_1) is equivalent to the analogue E-problem of (D_8) since they describe the same feasible region such that we do not examine the analogue of (D_8) separately. ◁

Also here, the proofs for the relations which can be easily obtained with similar proofs as in the previous sections are omitted. For the new combination $(E9)$, we get the following result.

Theorem 8.40: $(\tilde{s},\tilde{t})$ *feasible for* $(E_4) \implies (\tilde{s},\tilde{t})$ *feasible for* (E_9)

Proof:
Let $(\tilde{s},\tilde{t}) \in \mathbb{R}^{n_1+n_2}$ be feasible for (E_4). Then, $\tilde{s}$ is feasible and optimal for the lower-level problem w.r.t. $\tilde{t}$ of (E_9). □

Table 8.3 summarizes the found relations without problem (E_5), and the following observation our results concerning problem (E_5).

feas. \ eff.	(E_1)	(E_3)	(E_4)
(E_1)	•	✓	-
(E_2)	✓′	✓	-
(E_3)	✓*	•	✓′
(E_9)	-	-	✓′

Table 8.3: This table should be read in the same way as Table 8.1. Rows and columns without checkmarks have been omitted.

Observation 8.41:
We use the same notation as in Remark 8.25.

- $X_E^{E_1} \cup X_E^{E_4} \subseteq X_E^{E_5} \subseteq X^{E_3}$
- $X_E^{E_1} \subseteq X_E^{E_3} \subseteq X_E^{\{1,\dots,p_1\}} = X^{E_1} \subseteq X^{E_2}$
- $X_E^{\{p_1+1,\dots,p_1+p_2\}} = X^{E_4} \subseteq X^{E_3} \cap X^{E_9}$ ◁

An obvious further extension of the two optimization problems is the dependency on both variable-parts for both objective functions.

Definition 8.42:
With the same assumptions as in Definition 8.16, except for the objective functions f^1 and f^2 which should fulfill $f^1 : \mathbb{R}^{n_1+n_2} \to \mathbb{R}^{p_1}$ and $f^2 : \mathbb{R}^{n_1+n_2} \to \mathbb{R}^{p_2}$, we define the two multicriteria optimization problems (P^4) and (Q^4) as

$$(P^4) \quad \min_{s,t} \; f^1(s,t) \tag{8.23a}$$
$$\text{s. t.} \; g^1(s) \leqq 0 \tag{8.23b}$$
$$g^2(t) \leqq 0 \tag{8.23c}$$

$$(Q^4) \quad \min_{s,t} \; f^2(s,t) \tag{8.24a}$$
$$\text{s. t.} \; g^1(s) \leqq 0 \tag{8.24b}$$
$$g^2(t) \leqq 0 \tag{8.24c}$$
$$g^3(s,t) \leqq 0 \tag{8.24d}$$

◇

For this setting, we assume analogous conditions as **A1**, **A2′**, **A3**, **A5** and additionally:

A4‴ f^2 (and f^1) may only be evaluated at points $(s,t) \in \mathbb{R}^{n_1+n_2}$ with $g^1(s) \leqq 0$, $g^2(t) \leqq 0$ and $g^3(s,t) \leqq 0$ (and at points $(s,t) \in \mathbb{R}^{n_1+n_2}$ fulfilling $g^1(s) \leqq 0$ and $g^2(t) \leqq 0$, respectively).

feas. / feas.	(F_1)	(F_4)	(F_6)	(F_8)
(F_1)	•	-	✓	-
(F_2)	✓	-	✓	✓
(F_3)	-	✓	-	-
(F_7)	-	-	✓	✓
(F_8)	-	-	✓	•
(F_9)	-	✓	-	-

Table 8.4: This table should be read from the columns to the rows, e.g., the checkmark ✓in the cell induced by column (F_1) and row (F_2) says: "A feasible solution of (F_1) is also feasible for (F_2)." Rows and columns without checkmarks have been omitted.

Definition 8.43 (Combinations (F-problems)):
Analogous to Definition 8.28 and Definition 8.38, we define the optimization problems (F_1) to (F_9) obtained by substituting $f^2(t)$ with $f^2(s,t)$ in (D_1) to (D_8) and $f^1(s)$ with $f^1(s,t)$ in (E_9), respectively. ◇

Since the previous two extensions are special cases of this extension, we expect to lose a lot of relations since we have to merge the relations of the previous two modifications. Therefore, for (F_1) to (F_5), it is sufficient to only consider the relations which are fulfilled for both D- and E-problems. With similar proofs as for the former combinations, we easily obtain the relations given in Table 8.4.

Including problem (F_5), we get analogous to the previous considerations and in compliance with the results from Table 8.4 the following results.

Observation 8.44:
We use the same notation as in Remark 8.25.

- $X_E^{F_8} \cup X_E^{F_4} \cup X_E^{F_6} \subseteq X_E^{F_5}$
- $X^{F_6} \subseteq X_E^{\{1,\dots,p_1\}} = X^{F_8} \subseteq X^{F_2} \cap X^{F_7}$
- $X_E^{\{p_1+1,\dots,p_1+p_2\}} = X^{F_4} \subseteq X^{F_3} \cap X^{F_9}$ ◁

8.4.2 Additional Linking Constraints

Up to now, if we look at the two investigated problems (P^1) and $(Q^1_{s^*})$ from Definition 8.16, only the second problem was depending on the linking constraints g^3 with some

fixed feasible solution $s^* \in \mathbb{R}^{n_1}$ fulfilling $g^1(s^*) \leqq 0$. A natural extension are additional linking constraints in the first problem. As in Subsection 8.4.1, we concentrate on the multicriteria case combining the following two general optimization problems.

Definition 8.45:
With the same assumptions as in Definition 8.16, with additional constant parameter $t^* \in \mathbb{R}^{n_2}$ with $g^2(t^*) \leqq 0$ and constraint function $g^4 : \mathbb{R}^{n_1+n_3} \to \mathbb{R}^{m_4}$ for $m_4 \in \mathbb{N}$, we define the two multicriteria optimization problems $(P^5_{t^*})$ and $(Q^5_{s^*})$ as

$$
\begin{aligned}
(P^5_{t^*}) \quad \min_{s} \quad & f^1(s) && (8.25a)\\
\text{s.t.} \quad & g^1(s) \leqq 0 && (8.25b)\\
& g^4(s, t^*) \leqq 0 && (8.25c)
\end{aligned}
\qquad
\begin{aligned}
(Q^5_{s^*}) \quad \min_{t} \quad & f^2(t) && (8.26a)\\
\text{s.t.} \quad & g^2(t) \leqq 0 && (8.26b)\\
& g^3(s^*, t) \leqq 0 && (8.26c)
\end{aligned}
$$

◇

In addition to analogous assumptions as **A1** and **A2′**, we make the assumption:

A3′ There exists some $(s^*, t^*) \in \mathbb{R}^{n_1+n_2}$ fulfilling $g^1(s^*) \leqq 0$, $g^2(t^*) \leqq 0$, $g^3(s^*, t^*) \leqq 0$, $g^4(s^*, t^*) \leqq 0$.

This assumption is part of a guarantee for an almost well-defined problem; otherwise, we cannot guarantee feasibility of some problems. Moreover, we require:

A4′′′′ f^2 (and f^1) may only be evaluated at points $t \in \mathbb{R}^{n_2}$ with $g^2(t) \leqq 0$ and $g^3(s^*, t) \leqq 0$ for some $s^* \in \mathbb{R}^{n_1}$ with $g^1(s^*) \leqq 0$ (and at points $s \in \mathbb{R}^{n_1}$ fulfilling $g^1(s) \leqq 0$ and $g^4(s, t^*) \leqq 0$ for some $t^* \in \mathbb{R}^{n_2}$ with $g^2(t^*) \leqq 0$, respectively).

A5′ We require the fulfillment of both constraints g^1 and g^2 before g^3 or g^4 is checked.

Hence, for evaluating f^1, we require g^1, g^2 and g^4 and for evaluating f^2, we need g^1, g^2 and g^3. If we pay attention to these requirements, the following relevant combinations remain.

Definition 8.46 (Combinations (*G*-problems)):
For the two problems (P^5) and $(Q^5_{s^*})$, we define the following nine combinations.

$$
\begin{aligned}
(G_1) \quad \min_{s^*,t} \quad & f^2(t) && (8.27a)\\
\text{s.t.} \quad & g^2(t) \leqq 0 && (8.27b)\\
& g^3(s^*, t) \leqq 0 && (8.27c)\\
& s^* \in \arg\min_{s} f^1(s) && (8.27d)\\
& \quad \text{s.t.} \quad g^1(s) \leqq 0 && (8.27e)\\
& \qquad g^4(s, t) \leqq 0 && (8.27f)
\end{aligned}
\qquad
\begin{aligned}
(G_2) \quad \min_{s^*,t} \quad & f^2(t) && (8.28a)\\
\text{s.t.} \quad & g^2(t) \leqq 0 && (8.28b)\\
& s^* \in \arg\min_{s} f^1(s) && (8.28c)\\
& \quad \text{s.t.} \quad g^1(s) \leqq 0 && (8.28d)\\
& \qquad g^3(s, t) \leqq 0 && (8.28e)\\
& \qquad g^4(s, t) \leqq 0 && (8.28f)
\end{aligned}
$$

$$
\begin{aligned}
(G_3)\ \min_{s,t^*}\ & f^1(s) && (8.29a)\\
\text{s.t. }\ & g^1(s) \leqq 0 && (8.29b)\\
& g^4(s,t^*) \leqq 0 && (8.29c)\\
& t^* \in \arg\min_t f^2(t) && (8.29d)\\
& \quad \text{s.t. } g^2(t) \leqq 0 && (8.29e)\\
& \qquad g^3(s,t) \leqq 0 && (8.29f)
\end{aligned}
$$

$$
\begin{aligned}
(G_4)\ \min_{s^*,t^*}\ & f^1(s^*) && (8.30a)\\
\text{s.t. }\ & (s^*,t^*) \in \arg\min_{s,t} f^2(t) && (8.30b)\\
& \quad \text{s.t. } g^1(s) \leqq 0 && (8.30c)\\
& \qquad g^2(t) \leqq 0 && (8.30d)\\
& \qquad g^3(s,t) \leqq 0 && (8.30e)\\
& \qquad g^4(s,t) \leqq 0 && (8.30f)
\end{aligned}
$$

$$
\begin{aligned}
(G_5)\ \min_{s,t}\ & (f^1(s), f^2(t)) && (8.31a)\\
\text{s.t. }\ & g^1(s) \leqq 0 && (8.31b)\\
& g^2(t) \leqq 0 && (8.31c)\\
& g^3(s,t) \leqq 0 && (8.31d)\\
& g^4(s,t) \leqq 0 && (8.31e)
\end{aligned}
$$

$$
\begin{aligned}
(G_6)\ \min_{s^*,t^*}\ & f^2(t^*) && (8.32a)\\
\text{s.t. }\ & g^3(s^*,t^*) \leqq 0 && (8.32b)\\
& (s^*,t^*) \in \arg\min_{s,t} f^1(s) && (8.32c)\\
& \quad \text{s.t. } g^1(s) \leqq 0 && (8.32d)\\
& \qquad g^2(t) \leqq 0 && (8.32e)\\
& \qquad g^4(s,t) \leqq 0 && (8.32f)
\end{aligned}
$$

$$
\begin{aligned}
(G_8)\ \min_{s^*,t^*}\ & f^2(t^*) && (8.33a)\\
\text{s.t. }\ & (s^*,t^*) \in \arg\min_{s,t} f^1(s) && (8.33b)\\
& \quad \text{s.t. } g^1(s) \leqq 0 && (8.33c)\\
& \qquad g^2(t) \leqq 0 && (8.33d)\\
& \qquad g^3(s,t) \leqq 0 && (8.33e)\\
& \qquad g^4(s,t) \leqq 0 && (8.33f)
\end{aligned}
$$

$$
\begin{aligned}
(G_{10})\ \min_{s,t^*}\ & f^1(s) && (8.34a)\\
\text{s.t. }\ & g^1(s) \leqq 0 && (8.34b)\\
& t^* \in \arg\min_t f^2(t) && (8.34c)\\
& \quad \text{s.t. } g^2(t) \leqq 0 && (8.34d)\\
& \qquad g^3(s,t) \leqq 0 && (8.34e)\\
& \qquad g^4(s,t) \leqq 0 && (8.34f)
\end{aligned}
$$

$$
\begin{aligned}
(G_{11})\ \min_{s^*,t^*}\ & f^1(s^*) && (8.35a)\\
\text{s.t. }\ & g^4(s^*,t^*) \leqq 0 && (8.35b)\\
& (s^*,t^*) \in \arg\min_{s,t} f^2(t) && (8.35c)\\
& \quad \text{s.t. } g^1(s) \leqq 0 && (8.35d)\\
& \qquad g^2(t) \leqq 0 && (8.35e)\\
& \qquad g^3(s,t) \leqq 0 && (8.35f)
\end{aligned}
$$

◇

feas. \ eff.	(G_1)	(G_2)	(G_3)	(G_4)	(G_6)	(G_8)	(G_{10})	(G_{11})
(G_1)	•	-	-	-	✓′	-	-	-
(G_2)	✓′	•	-	✓*	✓′	✓′	✓	✓*
(G_3)	-	-	•	-	-	-	-	✓′
(G_4)	-	✓	-	•	-	-	-	✓′
(G_8)	-	-	-	-	✓′	•	✓	-
(G_{10})	-	✓	✓′	✓′	✓*	✓*	•	✓′

Table 8.5: This table should be read in the same way as Table 8.1. Rows without checkmarks have been omitted.

Observation 8.47:
From assumptions **A2′** and **A3′**, we immediately get feasibility for all combinations except (G_1), (G_3), (G_6) and (G_{11}) since in these combinations an efficient solution of the corresponding lower-level problem does not need to fulfill g^3 or g^4 from the upper level looking only at the given assumptions. ◁

Remark 8.48:
Here, the numbering of the combinations is similar to the numbering in the previous sections. However, due to the additional constraint function, we cannot immediately conclude that (G_i) is a generalization of (C_i), but if we consider $g^4 \equiv 0$ (i.e., is always fulfilled), we can say: (C_i) is a special case of (G_i) for $i = 1, \ldots, 5$. Equivalence between (G_1) and (G_8) cannot be proven as in previous sections since the lower-level problem of (G_1) and, thus, the objective value is depending on t, but we can still say that (C_1) is a special case of (G_8). In addition, also (C_1), (C_3) and (C_4) are special cases of (G_6), (G_{10}) and (G_{11}), respectively.
For this reason, we only investigated the relations which are valid for the corresponding C-problems. The proofs are omitted due to obvious similarities as in the previous sections. The results are presented in Table 8.5 and Observation 8.49. Combinations similar to (D_7) or (E_9) are not relevant in this setting since, here, the objective functions of the lower-level problem would be constant. ◁

Observation 8.49:
We use the same notation as in Remark 8.25.

- $X_E^{G_4} \cup X_E^{G_6} \cup X_E^{G_8} \cup X_E^{G_{11}} \subseteq X_E^{G_5} \subseteq X^{G_2} \cap X^{G_{10}}$
- $X_E^{G_8} \subseteq X^{G_6} \cup X_E^{G_{10}} \subseteq X_E^{\{1,\ldots,p_1\}} = X^{G_8} \subseteq X^{G_2}$
- $X_E^{G_4} \subseteq X_E^{G_2} \cup X^{G_{11}} \subseteq X_E^{\{p_1+1,\ldots,p_1+p_2\}} = X^{G_4} \subseteq X^{G_{10}}$ ◁

8.5 Ideas for the Practical Approach

This section summarizes some ideas for the practical approach. We first propose a naive approach treating the bicriteria optimization problem (C_5) from Definition 8.3 and, thereafter, develop conditions under which we can approximate the lower-level problem of a multicriteria combination with quality guarantees for the upper-level problem.

8.5.1 Naive Approach

In this subsection, we are interested in nondominated solutions of the bicriteria optimization problem (C_5) from Definition 8.3. These nondominated solutions should be computed with the help of the corresponding combined problem (C_1) since we assume that it is easier to find efficient algorithms (combinations of efficient algorithms for (P^1) and $(Q^1_{s^*})$) to solve this problem than to deal with a general multicriteria approach like, for instance, a weighted sum method in which we optimize over the feasible set with all constraints. We make the following assumptions:

- We have given problems (P^1) and $(Q^1_{s^*})$ from Definition 8.1 fulfilling **A1** to **A3**.
- (C_5) is externally stable and the nondominated set of (C_5) is finite.
- The feasible set of (P^1) is discrete.
- There exists some ranking-/k-best-method for (P^1) (see, e.g., Eppstein (1998) for the shortest path problem).
- We know the f^1-value of the lexicographic solution $(s^{\text{lexrev}}, t^{\text{lexrev}})$ in the reversed order of (C_5) or can estimate it, i.e., we have given some value f^1_{lr} with

$$f^1_{lr} \geq f^1(s^{\text{lexrev}}) \, .$$

Definition 8.50:
For the combination (C_1) and some $k \in \mathbb{N}$, we define the problem (C_1^k) as

$$\begin{aligned} (C_1^k) \quad & \min_{s^*,t} \quad f^2(t) \\ & \text{s.t.} \quad g^2(t) \leqq 0 \\ & \qquad g^3(s^*,t) \leqq 0 \\ & \qquad s^* \in \arg\min_{s}{}^{k} \, f^1(s) \\ & \qquad\qquad \text{s.t.} \; g^1(s) \leqq 0 \end{aligned}$$

where the symbol "$\arg\min_{s}{}^{k}$" denotes the determination of all solutions with the k-best optimal objective value (strictly better than the $(k-1)$-best objective value). $\diamond$

Algorithm 8.1 Naive Approach

Input: Two problems (P^1) and $(Q^1_{s^*})$ from Definition 8.1 fulfilling **A1** to **A3**.
Output: A minimal complete set X^* of efficient solutions for (C_5) (cf. Theorem 8.52).

1: $(P^{\text{curr}}) \leftarrow (C_1)$ // *initialize the current problem with* (C_1)
2: $i \leftarrow 1$
3: Solve problem (P^{curr}) with optimal solution (s^*, t^*).
4: $X^* \leftarrow \{(s^*, t^*)\}$
5: $f^1_{\text{curr}} \leftarrow f^1(s^*)$
6: $f^2_{\text{curr}} \leftarrow f^2(t^*)$
7: **while** $f^1_{\text{curr}} \leq f^1_{lr}$ **do** // *repeat until the lex-bound is exceeded*
8: $(P^{\text{curr}}) \leftarrow (C_1^{i+1})$ // *create the new problem*
9: **if** (P^{curr}) is infeasible **then** // *Have we enumerated all solutions?*
10: **return** X^*
11: Solve problem (P^{curr}) with optimal solution (s^*, t^*).
12: **if** $f^2(t^*) < f^2_{\text{curr}}$ **then** // *Is the new point nondominated?*
13: $X^* \leftarrow X^* \cup \{(s^*, t^*)\}$
14: $f^2_{\text{curr}} \leftarrow f^2(t^*)$
15: $f^1_{\text{curr}} \leftarrow f^1(s^*)$
16: $i \leftarrow i + 1$
17: **return** X^*

In the following, we describe our approach to obtain all nondominated solutions of (C_5). First, we solve problem (C_1), which leads to the lexicographic solution in normal order (cf. Observation 8.7) and, thus, determines the nondominated point with the smallest f^1-value and the greatest f^2-value, which is a special property of the bicriteria case (see, e.g., Ehrgott, 2005). We store the current f^2-value.
Thereafter, we solve problem (C_1^2) and compare the obtained f^2-value with the stored f^2-value. If the current value is smaller, then this point is a new nondominated point; otherwise, it is dominated by the previously calculated point.
We repeat this procedure with appropriate modifications until we have enumerated all solutions from the lower-level problem or we have reached the bound for the lexicographic problem in reversed order.

A pseudocode description of the suggested algorithm is depicted in Algorithm 8.1.

Lemma 8.51: *The output X^* from Algorithm 8.1 contains only efficient solutions of (C_5).*

Proof:
We first note that all solutions found are feasible for (C_5) since all modifications of the original problem (C_1) preserve all constraints from (P^1) and $(Q^1_{s^*})$.

$d(\gamma_i, v_j)$	v_1	v_2	v_3	v_4
γ_1	350	**850**	250	650
γ_2	400	100	**800**	**900**
γ_3	**450**	800	500	100

Table 8.6: Distances for Example 8.53. The bold numbers represent the maximum in the respective column.

Suppose there exists a $(s,t) \in X^*$ which is dominated by some $(\bar{s},\bar{t}) \in \mathbb{R}^{n_1+n_2}$ feasible for (C_5). Without loss of generality, let $(\bar{s},\bar{t})$ be efficient. We know that there must exist a $k \in \mathbb{N}$ such that $\bar{s}$ is an optimal solution of the lower-level problem of (C_1^k). Thus, due to efficiency, $(\bar{s},\bar{t})$ is optimal for the whole problem (C_1^k). Since $(\bar{s},\bar{t})$ dominates (s,t), $(\bar{s},\bar{t})$ or some feasible solution with same outcome was considered in an earlier iteration than the corresponding iteration in which (s,t) was considered. After this iteration, it held $f^2_{curr} \leq f^2(\bar{t})$. Since f^2_{curr} is non-increasing in every iteration, (s,t) was unable to fulfill the if-condition in line 12 and, thus, there was no possibility to add (s,t) to X^*, which is a contradiction. □

Theorem 8.52: *The output X^* from Algorithm 8.1 defines a minimal complete set of efficient solutions for (C_5).*

Proof:
Due to lines 12 to 14 in Algorithm 8.1, the set X^* does not contain equivalent solutions with same outcome. Moreover, due to Lemma 8.51, we only have to show that for each nondominated point there exists a preimage in X^*.

Let $(s,t) \in \mathbb{R}^{n_1+n_2}$ be efficient for (C_5). Again, we know that there must exist a $k \in \mathbb{N}$ such that s is an optimal solution of the lower-level problem of (C_1^k). Due to the efficiency, there does not exist a feasible solution of (C_5) having the same f^1-value as (s,t) with smaller f^2-value. Hence, (s,t) is optimal for (C_1^k) and a feasible solution with the same outcome is considered in iteration k. Suppose $f^2(t) \geq f^2_{\text{curr}}$, but then there exists some feasible point $(\bar{s},\bar{t})$ with $f^2(\bar{t}) = f^2_{\text{curr}} \leq f^2(t)$ and, due to the modifications of (P^{curr}) during the running time, $f^1(\bar{s}) < f^1(s)$ implying that (s,t) cannot be efficient, which is a contradiction. Therefore, (s,t) or a feasible solution with the same outcome fulfills the condition in line 12 and is added to X^*. □

Example 8.53:
We look at the special case problem given by the graph in Figure 8.2, the distances from Table 8.6, $k = 1$, and the weights $w_1 = w_2 = w_3 = 100$. As mentioned at the beginning of Section 8.2, the weights could be interpreted as the amount of people who have to be evacuated and the goal is to find one collection point and an appropriate route to transport the people from the selected collection point to the safe place T.

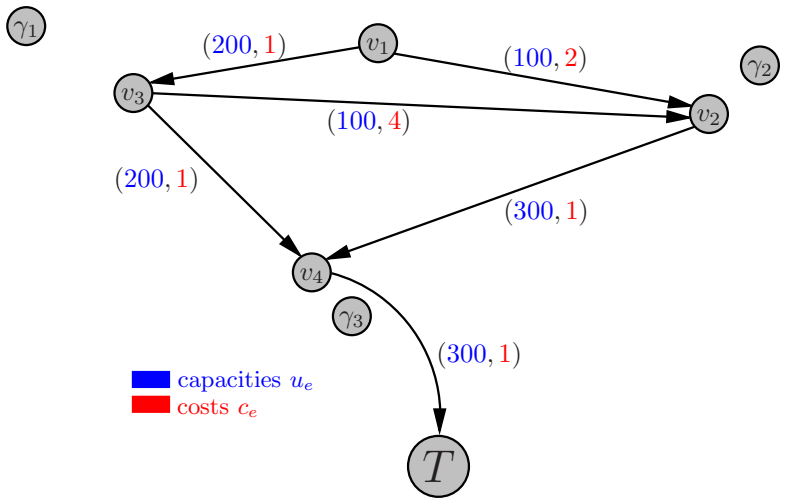


Figure 8.2: Graph from Example 8.53

The distances in meters could represent the length of some path which the people have to walk to get to the possible collection point.

Obviously, we have only four feasible possibilities to choose the location for the first part, the (1-CLP). Since we have uniform weights, in Table 8.6, we have highlighted the best objective value z for each selection/column divided by 100:

$$y_1 = 1 \Longrightarrow z = 45\,000 \qquad y_2 = 1 \Longrightarrow z = 85\,000$$
$$y_3 = 1 \Longrightarrow z = 80\,000 \qquad y_4 = 1 \Longrightarrow z = 90\,000$$

Algorithm 8.1 calculates the best integral flow for each selection of a location. Obviously, for each location, there exists a unique optimal integral flow for the subsequent problem (MCFP) transporting the people to the safe place T. The costs (i.e., objective values) ν of these flows are listed below:

$$y_1 = 1 \Longrightarrow \nu = 200 \cdot 3 + 100 \cdot 4 = 1\,000 \qquad y_2 = 1 \Longrightarrow \nu = 300 \cdot 2 = 600$$
$$y_3 = 1 \Longrightarrow \nu = 200 \cdot 2 + 100 \cdot 6 = 1\,000 \qquad y_4 = 1 \Longrightarrow \nu = 300$$

If we look at the objective values of the four possibilities w.r.t. the bicriteria problem (C_5), we get the four outcomes $(45\,000, 1\,000)^\top$, $(85\,000, 600)^\top$, $(80\,000, 1\,000)^\top$ and $(90\,000, 300)^\top$.

The algorithm now behaves as follows: First, it identifies the lexicographic outcome $(45\,000, 1\,000)^\top$ and before we examine the next problem in line 8 of Algorithm 8.1,

the current state is as follows

$$Y^* := f(X^*) = \{(45\,000, 1\,000)^\top\}, \quad f^1_{\text{curr}} = 45\,000, \quad f^2_{\text{curr}} = 1\,000\,.$$

The second best location selection is induced by $y_3 = 1$ which delivers the point $(80\,000, 1\,000)^\top$. This point does not pass the nondominance check at line 12. Thereafter, we have two more iterations in which we find the two points $(85\,000, 600)^\top$ and $(90\,000, 300)^\top$ which are added to Y^*. The final state is

$$Y^* = \{(45\,000, 1\,000)^\top, (85\,000, 600)^\top, (90\,000, 300)^\top\},\ f^1_{\text{curr}} = 90\,000,\ f^2_{\text{curr}} = 300\,.$$

From Theorem 8.52, we can conclude that the nondominated set for (C_5) is given by $Y^* = \{(45\,000, 1\,000)^\top,\ (85\,000, 600)^\top,\ (90\,000, 300)^\top\}$. ◁

8.5.2 Approximation of the Lower-Level Problem

In this subsection, we propose an idea to tackle a combined multicriteria optimization problem. Under the assumption that the lower-level problem is sufficiently easy to handle, we investigate an approximation of the lower-level problem by a small collection of sets of discrete points with low cardinality. This collection can be used to get an approximation of the nondominated set for the combined problem. However, since the lower-level problem often depends on parameters/variables from the upper-level, we would need to approximate a lot of different lower-level problems. Therefore, approximating the lower-level problem is only profitable if this level is independent from the upper-level or if the parameter-induced range of different lower-level problems remains low, e.g., if we have a final parameter-set for the lower-level problem, which is assumed in this subsection.

One option is to approximate the efficient set of the lower-level problem with "near points" with respect to some distance measure induced by some norm $\|\cdot\|$.

Independently of the assumptions from the previous sections, we assume the following given problem similar to (E_2) (analogously, with a shape similar to (E_1)).

Definition 8.54:
For $n_1, n_2, m_1, m_2, m_3, p_1, p_2 \in \mathbb{N}$, variables $s, s^* \in \mathbb{R}^{n_1}$ and $t \in \mathbb{R}^{n_2}$, constraint functions $g^1 : \mathbb{R}^{n_1} \to \mathbb{R}^{m_1}$, $g^2 : \mathbb{R}^{n_2} \to \mathbb{R}^{m_2}$ and $g^3 : \mathbb{R}^{n_1+n_2} \to \mathbb{R}^{m_3}$, and objective functions $f^1 : \mathbb{R}^{n_1} \to \mathbb{R}^{p_1}$ and $f^2 : \mathbb{R}^{n_1+n_2} \to \mathbb{R}^{p_2}$, we define the optimization problem $(\tilde{E})$ as

$$\begin{aligned}
(\tilde{E}) \quad \min_{s^*,t} \quad & f^2(s^*, t) \\
\text{s.t.} \quad & g^2(t) \leqq 0 \\
& \left.\begin{aligned} s^* \in \arg\min_{s} \ & f^1(s) \\ \text{s.t.} \ & g^1(s) \leqq 0 \\ & g^3(s,t) \leqq 0 \end{aligned}\right\} \tilde{E}^{\text{sub}}_t
\end{aligned}$$

with feasible set $X^{\tilde{E}}$, efficient set $X_E^{\tilde{E}}$, lower-level problem $\tilde{E}_t^{\text{sub}}$ (depending on $t \in \mathbb{R}^{n_2}$ with $g^2(t) \leqq 0$) with feasible set $X^{\tilde{E}_t^{\text{sub}}}$ and efficient set $X_E^{\tilde{E}_t^{\text{sub}}}$. ◇

Then, for all $t \in \mathbb{R}^{n_2}$ with $g^2(t) \leqq 0$ and given $\varepsilon_t > 0$, we are interested in a set $X_{\varepsilon_t}^{\tilde{E}_t^{\text{sub}}} \subseteq X^{\tilde{E}_t^{\text{sub}}}$ with the property

$$\forall \bar{s} \in X_E^{\tilde{E}_t^{\text{sub}}} \; \exists s' \in X_{\varepsilon_t}^{\tilde{E}_t^{\text{sub}}} : \|f^1(\bar{s}) - f^1(s')\| \leq \varepsilon_t \,. \tag{8.36}$$

Remark 8.55:
Looking at Definition 2.9, parameter ε_t from property (8.36) describes an upper bound on the coverage error for the approximation of the lower-level problem for a fixed $t \in \mathbb{R}^{n_2}$. Hence, for $p_1 \in \{2, 3\}$ the coverage-based Box-Algorithm treated in Chapters 4 and 5 can be used to obtain the desired approximations. In this case, for a fixed t, Theorems 4.17 and 5.15 show that we get an approximation of the lower-level problem fulfilling property (8.36). ◁

Since we do not want to relate $(\tilde{E})$ to other combinations, we do not force assumptions **A1** to **A3** to be true unless specifically stated otherwise.

Suppose now that we have access to a method (as a black-box) which can calculate such a set $X_{\varepsilon_t}^{\tilde{E}_t^{\text{sub}}}$ for each given feasible $t \in \mathbb{R}^{n_2}$ and $\varepsilon_t > 0$. We refer to this special approximation-scenario around property (8.36) as the *"approximation of the 1st variant"*.

In the following, we give two sufficient conditions such that by the use of these sets (depending on t), we can approximate the feasible set of the whole problem $(\tilde{E})$ appropriately, i.e., by solving the following problem.

Definition 8.56:
For problem $(\tilde{E})$ from Definition 8.54 and sets $X_{\varepsilon_t}^{\tilde{E}_t^{\text{sub}}}$ fulfilling property (8.36) for all $t \in \mathbb{R}^{n_2}$ with $g^2(t) \leqq 0$ and given $\varepsilon_t > 0$, we define the optimization problem $(\tilde{E}^{\text{approx}})$ as

$$\begin{aligned} (\tilde{E}^{\text{approx}}) \quad \min_{s^*,t} \quad & f^2(s^*, t) \\ \text{s.t.} \quad & g^2(t) \leqq 0 \\ & s^* \in X_{\varepsilon_t}^{\tilde{E}_t^{\text{sub}}} \end{aligned}$$

with feasible set $X^{\tilde{E}^{\text{approx}}}$ and efficient set $X_E^{\tilde{E}^{\text{approx}}}$. ◇

The first condition relates the two objective functions directly:

$$\forall t \in \mathbb{R}^{n_2} \text{ with } g^2(t) \leqq 0 \; \exists L_t > 0 : \tag{8.37a}$$

$$\left(\forall s^1, s^2 \in X^{\tilde{E}_t^{\text{sub}}} : \|f^2(s^1, t) - f^2(s^2, t)\| \leq L_t \cdot \|f^1(s^1) - f^1(s^2)\|\right) \tag{8.37b}$$

Observation 8.57:
Let $\delta > 0$ be given, condition (8.37) be satisfied and for each $t \in \mathbb{R}^{n_2}$ with $g^2(t) \leqq 0$, let $\varepsilon_t \leq \frac{\delta}{L_t}$. Then, for all feasible solutions $(\bar{s}, \bar{t}) \in X^{\tilde{E}}$ of $(\tilde{E})$, there exists a $s' \in X^{\tilde{E}^{\text{sub}}}_{\varepsilon_{\bar{t}}}$ such that $(s', \bar{t}) \in X^{\tilde{E}^{\text{approx}}}$ and it holds

$$\|f^2(\bar{s}, \bar{t}) - f^2(s', \bar{t})\| \leq L_{\bar{t}} \cdot \|f^1(\bar{s}) - f^1(s')\| \leq L_{\bar{t}} \cdot \varepsilon_{\bar{t}} \leq \delta \, . \qquad \lhd$$

This means, we can look at the simpler problem $(\tilde{E}^{\text{approx}})$ instead of $(\tilde{E})$ without changing the outcome set too much, i.e., up to a predefined approximation ratio δ.

The second condition combines a Lipschitz-condition in the first variable-part for the upper-level objective function with some kind of "inverse Lipschitz-condition" for the lower-level objective function:

$$\forall t \in \mathbb{R}^{n_2} \text{ with } g^2(t) \leqq 0 \; \exists L_t > 0 :$$

$$\left(\forall s^1, s^2 \in X^{\tilde{E}^{\text{sub}}_t} : \|f^2(s^1, t) - f^2(s^2, t)\| \leq L_t \cdot \underbrace{\|(s^1, t) - (s^2, t)\|}_{= \|s^1 - s^2\|} \right) \tag{8.38a}$$

and

$$\exists M_t > 0 :$$

$$\left(\forall s^1, s^2 \in X^{\tilde{E}^{\text{sub}}_t} : \|f^1(s^1) - f^1(s^2)\| \geq M_t \cdot \|s^1 - s^2\| \right) \tag{8.38b}$$

Observation 8.58:
Let $\delta > 0$ be given, condition (8.38) be satisfied and for each $t \in \mathbb{R}^{n_2}$ with $g^2(t) \leqq 0$, let $\varepsilon_t \leq \frac{\delta \cdot M_t}{L_t}$. Then, for all feasible solutions $(\bar{s}, \bar{t}) \in X^{\tilde{E}}$ of $(\tilde{E})$, there exists a $s' \in X^{\tilde{E}^{\text{sub}}}_{\varepsilon_{\bar{t}}}$ such that $(s', \bar{t}) \in X^{\tilde{E}^{\text{approx}}}$ and it holds

$$\|f^2(\bar{s}, \bar{t}) - f^2(s', \bar{t})\| \leq \frac{L_{\bar{t}}}{M_{\bar{t}}} \cdot \|f^1(\bar{s}) - f^1(s')\| \leq \frac{L_{\bar{t}}}{M_{\bar{t}}} \cdot \varepsilon_{\bar{t}} \leq \delta \, . \qquad \lhd$$

Remark 8.59:
Clearly, condition (8.38) is a special case of condition (8.37); however, the second condition is more conspicuous and can often be checked more quickly. Assuming sufficient smoothness, it is well known that the Lipschitz-condition (8.38a) can, for instance, be checked with the help of the mean value theorem (see, e.g., Forster, 2011). More precisely, the Lipschitz-condition is fulfilled if the partial derivatives of f^2 are bounded on each line segment between two distinct points from $X^{\tilde{E}^{\text{sub}}_t}$. For instance, in s affine linear objective functions, say $f^2(s, t) = d(t) + C(t) \cdot s$ with matrix $C(t) \in \mathbb{R}^{p_2 \times n_1}$ and vector $d(t) \in \mathbb{R}^{p_2}$ depending on $t \in \mathbb{R}^{n_2}$, fulfill the

Lipschitz-condition in s with Lipschitz-constant $L_t \coloneqq \|C(t)\|$, where here $\|\cdot\|$ refers to the well-known matrix norm, i.e., for a matrix $A \in \mathbb{R}^{m\times n}$ its norm is defined by $\|A\| = \max\{\|Ax\| : x \in \mathbb{R}^n \text{ with } \|x\| = 1\} = \max\limits_{x\neq 0}\left\{\frac{\|Ax\|}{\|x\|}\right\}$ (see, e.g., Schwarz and Köckler, 2011).
On the other hand, a sufficient condition for the second part (8.38b) of this condition is affine linearity of f^1, say $f^1(s) = A\cdot s + b$ with $A \in \mathbb{R}^{p_1\times n_1}$, $b \in \mathbb{R}^{p_1}$, $p_1 = n_1$, and A invertible. Then, we can calculate the chain

$$\begin{aligned}\|s^1 - s^2\| &= \|A^{-1}\Big(A\cdot s^1\Big) - A^{-1}\Big(A\cdot s^2\Big)\| \\ &= \|A^{-1}\Big(A\cdot s^1 - A\cdot s^2\Big)\| \\ &\leq \|A^{-1}\|\cdot\|A\cdot s^1 - A\cdot s^2\| \\ \Longrightarrow \|f^1(s^1) - f^1(s^2)\| &\geq \frac{1}{\|A^{-1}\|}\cdot\|s^1 - s^2\|\end{aligned}$$

and set M_t to $\frac{1}{\|A^{-1}\|}$. ◁

In general, having an approximation method for the feasible set for arbitrary accuracy similar to property (8.36) does not automatically imply an approximation of the outcome set in a similar sense. To point out this issue, we consider a very simplified example.

Example 8.60:
Consider the bicriteria optimization problem with feasible set $X = (0,1) \subseteq \mathbb{R}$ and objective function $f : X \longrightarrow \mathbb{R}^2$ with $f(x) \coloneqq (x, 1/x)^\top$. Moreover, let $0 < \varepsilon < 1$ be arbitrary but fixed.
We can give a method for calculating an approximation for $X_E = X$ similar to property (8.36). For this issue, we suggest the following set

$$X_\varepsilon \coloneqq \left\{ i\cdot\varepsilon : 1 \leq i \leq \begin{cases}\left\lfloor\frac{1}{\varepsilon}\right\rfloor, & \text{if } \frac{1}{\varepsilon}\notin\mathbb{N} \\ \frac{1}{\varepsilon} - 1, & \text{if } \frac{1}{\varepsilon}\in\mathbb{N}\end{cases}\right\}$$

which is also a subset of X.

This set approximates X in the following sense:

$$\forall x \in X\ \exists \hat{x} \in X_\varepsilon : \|x - \hat{x}\|_\infty = |x - \hat{x}| < \varepsilon$$

where $\|\cdot\|_\infty$ denotes the maximum norm.

We claim that we cannot find an accuracy parameter $\delta > 0$ such that this set approximates the outcome set in the following way:

$$\forall x \in X\ \exists \hat{x} \in X_\varepsilon : \|f(x) - f(\hat{x})\|_\infty < \delta \tag{8.39}$$

To prove this, we assume the contrary, namely that we have found a particular δ fulfilling (8.39). Next, we look at the special point $x_0 := {}^1/_{(2\delta+1/\varepsilon)}$. Clearly, $x_0 < \varepsilon < i \cdot \varepsilon$ and ${}^1/_{x_0} > {}^1/_{\varepsilon} > {}^1/_{i\cdot\varepsilon}$ for all $i \geq 2$; therefore, $\|f(x_0) - f(i \cdot \varepsilon)\|_\infty > \|f(x_0) - f(\varepsilon)\|_\infty$ for all $i \geq 2$.

Hence, we only consider the distance to the point $f(\varepsilon)$:

$$\|f(x_0) - f(\varepsilon)\|_\infty = \|(x_0, {}^1/_{x_0})^\top - (\varepsilon, {}^1/_{\varepsilon})^\top\|_\infty \geq \frac{1}{x_0} - \frac{1}{\varepsilon} = 2\delta + \frac{1}{\varepsilon} - \frac{1}{\varepsilon} = 2\delta > \delta$$

which is a contradiction to the assumption.
This contradiction can also be obtained if we substitute the maximum norm with an arbitrary p-norm (see also Lemma 8.67 below). ◁

Observation 8.61:
If we take a closer look at the conditions for the approximation of the 1st variant and, in particular, at property (8.36), a natural question arises: Are there situations in which an efficient point $(\bar{s}, \bar{t}) \in X_E^{\tilde{E}}$ is approximated by a point $(s', \bar{t}) \in X^{\tilde{E}^{\text{approx}}}$, i.e.,

$$\|f^2(\bar{s}, \bar{t}) - f^2(s', \bar{t})\| \leq \delta$$

but no point in $X_E^{\tilde{E}^{\text{approx}}}$ exists fulfilling this approximation property, i.e.,

$$\nexists (\tilde{s}, \tilde{t}) \in X_E^{\tilde{E}^{\text{approx}}} : \|f^2(\bar{s}, \bar{t}) - f^2(\tilde{s}, \tilde{t})\| \leq \delta \quad ?$$

Suppose $(\tilde{E}^{\text{approx}})$ is externally stable. Since $(s', \bar{t})$ is feasible for $(\tilde{E}^{\text{approx}})$, there must exist a point $(\tilde{s}, \tilde{t}) \in X_E^{\tilde{E}^{\text{approx}}}$ dominating $(s', \bar{t})$ but not approximating $(\bar{s}, \bar{t})$, in this situation. Obviously, this is a possible scenario and is illustrated in Figure 8.3.

Nevertheless, if δ is chosen sufficiently small, it can be argued that this is not an issue since point $f(\tilde{s}, \tilde{t})$ from the illustrating figure would then be almost better than point $f(\bar{s}, \bar{t})$ since it dominates the approximated red point which is very close to $f(\bar{s}, \bar{t})$. ◁

Up to now, we have concentrated on the coverage error for the approximation of the lower-level problem (cf. Remark 8.55); however, many other articles dealing with approximations for the efficient or nondominated set are using $(1+\varepsilon)$-Pareto sets (cf. Definition 2.10).

In this context, we make the assumption that for each feasible t, the set $f^1(X^{\tilde{E}_t^{\text{sub}}})$ is non-negative (or can be easily shifted to obtain non-negativity).

Suppose now that for all $t \in \mathbb{R}^{n_2}$ with $g^2(t) \leqq 0$ and given $\varepsilon_t > 0$, we have given a method calculating a $(1+\varepsilon_t)$-Pareto set $P_{\varepsilon_t}^{\tilde{E}_t^{\text{sub}}} \subseteq X^{\tilde{E}_t^{\text{sub}}}$ for the lower-level problem $(\tilde{E}_t^{\text{sub}})$, i.e., in analogy to (8.36)

$$\forall \bar{s} \in X_E^{\tilde{E}_t^{\text{sub}}} \exists s' \in P_{\varepsilon_t}^{\tilde{E}_t^{\text{sub}}} : f^1(s') \leqq (1+\varepsilon_t) \cdot f^1(\bar{s}) \,. \tag{8.40}$$

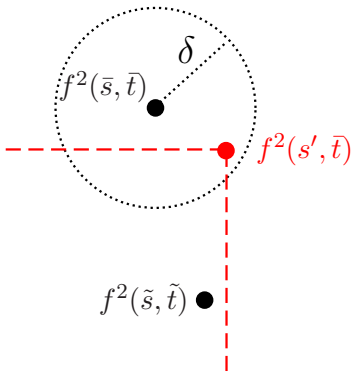


Figure 8.3: The red point approximates point $f^2(\bar{s},\bar{t})$, i.e., lies in the δ-neighborhood of $f^2(\bar{s},\bar{t})$ indicated by the dotted circle. However, this point is dominated by $f^2(\tilde{s},\tilde{t})$ which lies outside the circle.

In the following, we denote now with $(\tilde{E}^{\text{approx}})$ the same problem as introduced in Definition 8.56 but substituting $X_{\varepsilon_t}^{\tilde{E}_t^{\text{sub}}}$ with $P_{\varepsilon_t}^{\tilde{E}_t^{\text{sub}}}$. Furthermore, we refer to this special approximation-scenario around property (8.40) as the *"approximation of the 2nd variant"*.

Here, we also give two special sufficient conditions under which the feasible set of the problem $(\tilde{E})$ can be approximated appropriately.

For the first condition, we assume $p_1 = p_2$ (which in particular includes the single-criterion case) and the enclosure of the upper-level objective function in s within some special multiples of the lower-level objective function:

$$\exists \sigma > 0 : \forall t \in \mathbb{R}^{n_2} \text{ with } g^2(t) \leqq 0 \; \exists L_t > 0 \text{ with} \tag{8.41a}$$

$$\left(\forall s \in X^{\tilde{E}_t^{\text{sub}}} : \; \frac{L_t}{1+\sigma} \cdot f^1(s) \leqq f^2(s,t) \leqq L_t \cdot f^1(s) \right) \tag{8.41b}$$

Observation 8.62:
Let $\delta_0 > 0$ be given, condition (8.41) be satisfied and for each $t \in \mathbb{R}^{n_2}$ with $g^2(t) \leqq 0$, let $\varepsilon_t = \delta_0$. Define $\delta := \delta_0 + \sigma \cdot (1+\delta_0) > \sigma$. Then, for all feasible solutions $(\bar{s},\bar{t}) \in X^{\tilde{E}}$ of $(\tilde{E})$, there exists a $s' \in P_{\varepsilon_{\bar{t}}}^{\tilde{E}_{\bar{t}}^{\text{sub}}}$ such that $(s',\bar{t}) \in X^{\tilde{E}^{\text{approx}}}$ and it holds

$$f^2(s',\bar{t}) \leqq L_{\bar{t}} \cdot f^1(s') \leqq L_{\bar{t}} \cdot (1+\delta_0) f^1(\bar{s}) \leqq \frac{1+\sigma}{L_{\bar{t}}} L_{\bar{t}} \cdot (1+\delta_0) f^2(\bar{s},\bar{t}) = (1+\delta) f^2(\bar{s},\bar{t}) \; .$$

◁

For the second condition, we assume $p_2 \mid p_1$. Let $f^{1,1}(s)$ to $f^{1,p_1/p_2}(s)$ be the corresponding p_1/p_2 parts of the lower-level objective function f^1 of length p_2, i.e., for

$i \in \{1, \ldots, \frac{p_1}{p_2}\}$, let $f^{1,i}(s)$ correspond to the function consisting of the p_2 objective functions from coordinate $(i-1)p_2+1$ up to ip_2 of $f^1(s)$. Furthermore, for $a, b \in \mathbb{R}^{p_2}$, we denote by $a \diamond b$ the vector resulting by component-wise multiplication, i.e., $a \diamond b = (a_1, \ldots, a_{p_2})^\top \diamond (b_1, \ldots, b_{p_2})^\top = (a_1 \cdot b_1, \ldots, a_{p_2} \cdot b_{p_2})^\top$.

Then, for each $t \in \mathbb{R}^{n_2}$ with $g^2(t) \leqq 0$, the second condition is given by

$$f^2(s,t) = \sum_{i=1}^{p_1/p_2} w_i^t \diamond f^{1,i}(s) \text{ with } w_i^t \in \mathbb{R}^{p_2}_{\geqq} \text{ for } s \in X^{\tilde{E}_t^{\text{sub}}}. \tag{8.42}$$

Observation 8.63:
Let $\delta > 0$ be given, condition (8.42) be satisfied and for each $t \in \mathbb{R}^{n_2}$ with $g^2(t) \leqq 0$, let $\varepsilon_t = \delta$. Then, for all feasible solutions $(\bar{s}, \bar{t}) \in X^{\tilde{E}}$ of $(\tilde{E})$, there exists a $s' \in P_{\varepsilon_{\bar{t}}}^{\tilde{E}_{\bar{t}}^{\text{sub}}}$ such that $(s', \bar{t}) \in X^{\tilde{E}^{\text{approx}}}$ and it holds

$$f^2(s', \bar{t}) = \sum_{i=1}^{p_1/p_2} w_i^{\bar{t}} \diamond f^{1,i}(s') \leqq \sum_{i=1}^{p_1/p_2} w_i^{\bar{t}} \diamond \Big((1+\delta) \cdot f^{1,i}(\bar{s})\Big) = (1+\delta) \cdot f^2(\bar{s}, \bar{t}) \,. \qquad \lhd$$

We note here that if we are in a setting of one of the above observations and calculate all efficient solutions of problem $(\tilde{E}^{\text{approx}})$, we get an approximation in a special sense of $(\tilde{E})$ which is stated in the following theorem (cf. Observation 8.61).

Theorem 8.64: *Let property (8.40) and the requirements described in Observation 8.62 or 8.63 be satisfied. Additionally, let $(\tilde{E}^{approx})$ be externally stable.*
Then, the set $X_E^{\tilde{E}^{approx}}$ fulfills

$$\forall (\bar{s}, \bar{t}) \in X_E^{\tilde{E}} \; \exists (\hat{s}, \hat{t}) \in X_E^{\tilde{E}^{approx}} : f^2(\hat{s}, \hat{t}) \leqq (1+\delta) \cdot f^2(\bar{s}, \bar{t}) \,. \tag{8.43}$$

Proof:
Let $(\bar{s}, \bar{t}) \in X_E^{\tilde{E}}$. Then, by Observation 8.62 or 8.63, we get that there exists some $s' \in \mathbb{R}^{n_1}$ with $(s', \bar{t}) \in X^{\tilde{E}^{\text{approx}}}$ and $f^2(s', \bar{t}) \leqq (1+\delta) \cdot f^2(\bar{s}, \bar{t})$. Since problem $(\tilde{E}^{\text{approx}})$ is externally stable, there exists some $(\hat{s}, \hat{t}) \in X_E^{\tilde{E}^{\text{approx}}}$ which fulfills $f^2(\hat{s}, \hat{t}) \leqq f^2(s', \bar{t}) \leqq (1+\delta) \cdot f^2(\bar{s}, \bar{t})$. $\square$

Remark 8.65:
Note that $X_E^{\tilde{E}^{\text{approx}}}$ from Theorem 8.64 does not necessarily define a $(1+\delta)$-Pareto set for problem $(\tilde{E})$ since some elements of $X_E^{\tilde{E}^{\text{approx}}}$ may be infeasible for $(\tilde{E})$. $\lhd$

For condition (8.42), we can give an example extending a problem from a diploma thesis (Erfurth, 2012).

Example 8.66:
We consider the newly introduced path-median network location problem on a directed network with multicriteria edge cost vectors c and multicriteria node weight vectors w (cf. Erfurth, 2012, Definition 6.3). For convenience, we use the same notation as in the diploma thesis. Since there are three different models, w.l.o.g., we only look at the *Path-Out median network location problem* (with satisfied vertex dominance) on a graph $G = (\mathcal{V} = \{v_1, \dots, v_N\}, \mathcal{E})$ with cost function $c = (c^1, \dots, c^Q)^\top : \mathcal{E} \longrightarrow \mathbb{R}^Q_>$:

$$\begin{aligned} \min \quad & \sum_{i=1}^{N} \begin{pmatrix} w_i^1 \cdot c^1(P(x, v_i)) \\ \vdots \\ w_i^Q \cdot c^Q(P(x, v_i)) \end{pmatrix} \\ \text{s.t.} \quad & x \in \mathcal{V} \\ & (P(x, v_1), \dots, P(x, v_N))^\top \in \arg\min \left(c(P(x, v_1)), \dots, c(P(x, v_N))\right)^\top \\ & \qquad\qquad \text{s.t. } P(x, v_i) \in \mathcal{P}(x, v_i) \quad i = 1, \dots, N \end{aligned}$$

where $\mathcal{P}(x, v_i)$ denotes the set of all paths from node x to v_i, $i = 1, \dots, N$.

The lower-level problem defines a multicriteria shortest path problem and, thus, can be solved by calculating the efficient shortest paths from x to v_i separately for each $i = 1, \dots, N$. Hence, for each given node $x \in \mathcal{V}$, the lower-level problem can be solved with a *multiobjective label setting algorithm* (see, e.g., Ehrgott, 2005) which outputs, after one call, all efficient paths from x to all other nodes in $\mathcal{V}$. The running time of the algorithm developed in Erfurth (2012) for solving this location problem heavily depends on the maximal cardinality of the set of efficient paths of the lower-level problem. Since, in general, the multicriteria shortest path problem is NP-hard and intractable (see, e.g., Ehrgott, 2005), this maximal cardinality can be very large and may not be polynomially bounded in the input size.
Therefore, we suggest using an approximation, a $(1 + \delta)$-Pareto set, for each lower-level problem. In the literature, one can find many approximation methods for the multicriteria shortest path problem. An algorithm suitable for our particular needs can be found in Tsaggouris and Zaroliagis (2009) outputting a set matching property (8.40) and having an adequate size polynomially bounded in the input parameters.
All requirements from Theorem 8.64 (in particular, the requirements of Observation 8.63) are fulfilled. Hence, for an arbitrary $\delta > 0$, we can apply Theorem 8.64 by adapting the algorithm from Erfurth (2012) to get an approximation of the above location problem with accuracy δ (in the sense of Theorem 8.64). ◁

In Observation 8.61, it was illustrated that solving $(\tilde{E}^{\text{approx}})$ in the context of an approximation of the 1st variant may not lead to an approximation of $(\tilde{E})$ with a property similar to (8.36). Nevertheless, for the obtained set $X^{\tilde{E}^{\text{approx}}}$, we can show a

relation to an approximation of $(\tilde{E})$ of the 2nd variant fulfilling property (8.43) from Theorem 8.64 for an appropriately chosen δ.

For this result, we first need a (well-known) Lemma linking the maximum norm $\|\cdot\|_\infty$ to all other p-norms $\|\cdot\|_p$:

Lemma 8.67: *Let $1 \leq p < \infty$ and $n \in \mathbb{N}$. Then, it holds:*

$$\|x\|_\infty \leq \|x\|_p \qquad \text{for all } x \in \mathbb{R}^n$$

In particular, we have for a $x_0 \in \mathbb{R}^n$ and a fixed $\varepsilon > 0$

$$B_\varepsilon^p(x_0) \subseteq B_\varepsilon^\infty(x_0)$$

where $B_\varepsilon^\ell(x_0) \coloneqq \{x \in \mathbb{R}^n : \|x - x_0\|_\ell \leq \varepsilon\}$ for $1 \leq \ell \leq \infty$.

Proof:
Let $i_0 \coloneqq \underset{i=1,\ldots,n}{\arg\max} |x_i|$, then we have

$$\|x\|_p = \left(|x_{i_0}|^p + \underbrace{\sum_{i \neq i_0} |x_i|^p}_{\geq 0} \right)^{1/p} \overset{\text{mon. incr.}}{\geq} (|x_{i_0}|^p)^{1/p} = |x_{i_0}| = \|x\|_\infty \, .$$

The inclusion of an arbitrary neighborhood $B_\varepsilon^p(x_0)$ into the corresponding neighborhood $B_\varepsilon^\infty(x_0)$ follows directly from the proven inequality. □

Theorem 8.68: *Let $\delta_1 > 0$ be given, the assumptions of Observation 8.57 or 8.58 with $\delta \coloneqq \delta_1$ and $\|\cdot\| \coloneqq \|\cdot\|_\ell$ ($1 \leq \ell \leq \infty$ fixed) be satisfied and $X_E^{\tilde{E}^{approx}}$ denote the efficient set of $(\tilde{E}^{approx})$ in the context of an approximation of the 1st variant (cf. Definition 8.56). Moreover, let $(\tilde{E}^{approx})$ be externally stable and $\inf\limits_{(s,t) \in X^{\tilde{E}}} f_i^2(s,t) > 0$ for all $i = 1, \ldots, p_2$.*
Then, $X_E^{\tilde{E}^{approx}}$ fulfills the following property:

$$\forall (\bar{s}, \bar{t}) \in X_E^{\tilde{E}} \; \exists (\hat{s}, \hat{t}) \in X_E^{\tilde{E}^{approx}} : f^2(\hat{s}, \hat{t}) \leqq (1 + \delta_2) \cdot f^2(\bar{s}, \bar{t}) \tag{8.44}$$

with $\delta_2 \coloneqq \dfrac{\delta_1}{\min\limits_{j=1,\ldots,p_2} \inf\limits_{(s,t) \in X^{\tilde{E}}} f_j^2(s,t)}$

Proof:
Let $(\bar{s}, \bar{t}) \in X_E^{\tilde{E}}$. Then, due to Observation 8.57 or 8.58, there exists some $s' \in \mathbb{R}^{n_1}$ with $(s', \bar{t}) \in X^{\tilde{E}^{\text{approx}}}$ and $\|f^2(\bar{s}, \bar{t}) - f^2(s', \bar{t})\|_\ell \leq \delta_1$, in particular $f^2(s', \bar{t}) \in B_{\delta_1}^\ell(f^2(\bar{s}, \bar{t}))$.

Since problem $(\tilde{E}^{\text{approx}})$ is externally stable, there exists some $(\hat{s},\hat{t}) \in X_E^{\tilde{E}^{\text{approx}}}$ with $f^2(\hat{s},\hat{t}) \leqq f^2(s',\bar{t})$. Moreover, we have $f^2(s',\bar{t}) \leqq f^2(\bar{s},\bar{t}) + (\delta_1,\ldots,\delta_1)^\top$ since $f^2(\bar{s},\bar{t}) + (\delta_1,\ldots,\delta_1)^\top$ is dominated by all points in $B_{\delta_1}^{\infty}(f^2(\bar{s},\bar{t}))$ and, thus, from Lemma 8.67, it is dominated by all points in $B_{\delta_1}^{\ell}(f^2(\bar{s},\bar{t}))$.

Claim: $(1+\delta_2)\cdot f^2(\bar{s},\bar{t}) \geqq f^2(\bar{s},\bar{t}) + (\delta_1,\ldots,\delta_1)^\top$

Proof:
Consider the ith coordinate (fixed but arbitrary):

$$\begin{aligned}(1+\delta_2)\cdot f_i^2(\bar{s},\bar{t}) &\overset{\text{def. } \delta_2}{=} f_i^2(\bar{s},\bar{t}) + \frac{f_i^2(\bar{s},\bar{t})}{\min\limits_{j=1,\ldots,p_2}\ \inf\limits_{(s,t)\in X^{\tilde{E}}} f_j^2(s,t)}\cdot\delta_1 \\ &\overset{\text{all pos.}}{\geq} f_i^2(\bar{s},\bar{t}) + \underbrace{\frac{f_i^2(\bar{s},\bar{t})}{\inf\limits_{(s,t)\in X^{\tilde{E}}} f_i^2(s,t)}}_{\geq 1}\cdot\delta_1 \\ &\geq f_i^2(\bar{s},\bar{t}) + \delta_1 \qquad \diamondsuit\end{aligned}$$

Hence, we have

$$f^2(\hat{s},\hat{t}) \leqq f^2(s',\bar{t}) \leqq f^2(\bar{s},\bar{t}) + (\delta_1,\ldots,\delta_1)^\top \leqq (1+\delta_2)\cdot f^2(\bar{s},\bar{t})\ . \qquad \square$$

Remark 8.69:
Since the accuracy δ_2 in Theorem 8.68 depends on δ_1, it can be refined by refining the approximation with accuracy δ_1. Therefore, if the approximation of the 1st variant can be made arbitrary fine, then also the accuracy δ_2 can be shrunk to an arbitrary small number. $\triangleleft$

9 Conclusion

In this thesis, we investigated several upcoming issues occurring in the context of conceiving and building a decision support system (cf. Figure 1.2). We elaborated new algorithms for computing representative systems with special quality guarantees, provided concepts for supporting the decision makers after a representative system was computed, and considered a methodology of combining two optimization problems.

In Chapter 3, we gave an extension of the survey article of Ruzika and Wiecek (2005) and categorized several existing methods for constructing approximations for a multicriteria optimization problem. We classified these methods with respect to the number of objectives in the addressed optimization problem, the utilized scalarization technique and the resulting approximation structure. For bicriteria approaches, most articles considered an approximation of the 0^{th} order whereas more authors focused on approximations of the 1^{st} order in multicriteria approaches. This overview reflected the continuous development and the importance of quality guarantees in this field.

Chapter 4 was dedicated to the Box-Algorithm for two objectives by Hamacher et al. (2007). There, we first reviewed the original Box-Algorithm relying on an area-based quality guarantee and proved two important properties of this algorithm, the distinct-rectangle-property and the correctness property, which are crucial for the correct functioning. In a next step, we incorporated other quality guarantees into the Box-Algorithm. We first aimed at a low coverage error such that the whole nondominated set is well represented. For this, we had to switch between a horizontal and a vertical lexicographic ε-constraint scalarization during the run of the algorithm. Secondly, we investigated the incorporation of the uniformity to get a diverse representative system while fulfilling the original area-based quality guarantee. Thereafter, we addressed the issue of a complete enumeration of the whole nondominated set. There, it turned out that bisecting the shortest side of any upcoming rectangle can lead to a smaller number of iterations than bisecting the longest side as it is done in the coverage-based Box-Algorithm. The last extension dealt with a practical issue, namely the situation in which the used single-criterion solver cannot solve upcoming subproblems to optimality, but up to a positive gap. We provided ideas how the original Box-Algorithm can be modified to treat this situation appropriately.

In Chapter 5, we extended the original Box-Algorithm to cope with tricriteria optimization problems. The new Box-Algorithm was constructed to obtain both a collection of non-overlapping boxes subdividing the whole nondominated set and a representative

system fulfilling a desired coverage error. For the algorithm, we adaptively chose the appropriate lexicographic ε-constraint scalarization in each iteration to efficiently reduce the unexplored space. We proved correctness of the algorithm and an iteration bound to achieve the desired accuracy. In contrast to the original Box-Algorithm, the new algorithm had to rely on lower bounds on the objectives. Hence, the representation error of the obtained representative system could be positive. We proved two appropriate upper bounds for the representation error. We further investigated the incorporation of a volume-based quality guarantee similar to the quality guarantee used in the original Box-Algorithm. We also gave several ideas for extensions of the new algorithm concerning the incorporation of dominance during the algorithm, other quality measures, and the implementation of two different selection rules with their advantages. A direction for future research was presented in Section 5.9. There, we investigated a theoretical scalarization problem which is capable to either find a nondominated point in a given box or to prove that there does not exist such a point. Since this scalarization problem has a bilevel structure, it is hard to solve in practice. Yet, this bilevel idea is a promising approach for future research. Finally, we applied the coverage-based Box-Algorithm to several instances of the Bus Evacuation Problem to compare different parameter-settings with respect to the incorporation of dominance, different selection rules and feasible-tests for appropriate boxes. An open question and material for future research is the generalization of the Box-Algorithm and the corresponding results to an arbitrary number of objectives greater than three.

In Chapter 6, we addressed the hypervolume subset selection problem (HSSP), where we aimed at a subset with desired cardinality from an already computed representative system. We provided two new formulations for the bicriteria variant of this problem. The first is an integer programming formulation that can be solved by solving its linear programming relaxation. The second formulation is a k-link shortest path formulation on a special digraph with the Monge property that can be solved by dynamic programming. With the latter algorithm, we obtained the currently best known complexity bound for solving the bicriteria HSSP. For the tricriteria HSSP, we extended this integer programming formulation and utilized its linear programming relaxation to state a simple branch-and-bound algorithm. For this algorithm, we provided computational results for several data sets. There, for 200 representative points, we reached the limit of the used linear programming solver since the size of the constructed model was cubic in the number of the points. However, the computational experiments for less than 200 representative points revealed the potential of this approach. Hence, the development of an appropriate column generation approach to circumvent reaching the limit of the linear programming solver is motivation for further research in this field.

Chapter 7 dealt with the presentation of a computed representative system to the decision makers. There, we first presented four common illustration methods from

literature and, thereafter, proposed a new illustration method, the extended bar chart, in which the length of each bar was defined by given weight values. The bars were filled with color corresponding to the outcome values of the considered representative point. Based on such an illustration method, we elaborated an algorithm guiding the decision makers in choosing their preferred solution. This algorithm consists of two phases. The first phase is intended to show a diverse set in each iteration until the decision makers are satisfied with some representative point z^*. In the second phase, the algorithm computes the most similar representative points to z^* such that the decision makers can refine their last choice.

Finally, in Chapter 8 we analyzed theoretically how two (multicriteria) optimization problems can be combined. Under different assumptions, we came up with several combined (multicriteria) bilevel formulations and a combined (non-bilevel) multicriteria optimization problem. We related these different combinations with each other with respect to feasibility and optimality/efficiency. Moreover, we gave some ideas for the practical approach. Firstly, we proposed a naive approach treating the (non-bilevel) bicriteria optimization problem and, secondly, we proposed different conditions under which the lower-level problem of a fixed multicriteria bilevel combination can be approximated inducing quality guarantees for the upper-level problem. In the latter, we considered the coverage error as well as $(1+\varepsilon)$-Pareto sets as quality guarantees. The investigation of other extensions regarding the combination of two or even more (multicriteria) optimization problems as well as efficient solution methods combining algorithms specialized to each single optimization problem is substance for future research.

A Appendix

A.1 Subdivisions for the Box-Algorithm with Three Objective Functions

Following Section 5.3, the different quarters of the current box $\mathrm{B}(\ell, u)$ and the resulting boxes after solving $(P^2_{\varepsilon_1,\varepsilon_3})$ or $(P^3_{\varepsilon_2,\varepsilon_3})$ are analogously defined as for $(P^1_{\varepsilon_1,\varepsilon_2})$ since all subproblems describe the same procedure but with exchanged axes (cf. Definitions 5.7 and 5.8).

For problem $(P^2_{\varepsilon_1,\varepsilon_3})$, we get the quarters

$$Q^{2,1} \coloneqq \mathrm{B}\Big(\ell, (\varepsilon_1, u_2, \varepsilon_3)^\top\Big), \quad Q^{2,2} \coloneqq \mathrm{B}\Big((\varepsilon_1, \ell_2, \ell_3)^\top, (u_1, u_2, \varepsilon_3)^\top\Big),$$

$$Q^{2,3} \coloneqq \mathrm{B}\Big((\varepsilon_1, \ell_2, \varepsilon_3)^\top, u\Big), \quad Q^{2,4} \coloneqq \mathrm{B}\Big((\ell_1, \ell_2, \varepsilon_3)^\top, (\varepsilon_1, u_2, u_3)^\top\Big),$$

with boxes

$$B^{2,1} \coloneqq \mathrm{B}\Big((z_1^*, z_2^*, \ell_3)^\top, (\varepsilon_1, u_2, z_3^*)^\top\Big), \quad B^{2,2} \coloneqq \mathrm{B}\Big((\ell_1, z_2^*, \ell_3)^\top, (z_1^*, u_2, \varepsilon_3)^\top\Big)$$

for $Q^{2,1}$,

$$B^{2,3} \coloneqq \mathrm{B}\Big((\varepsilon_1, \ell_2, \ell_3)^\top, (u_1, z_2^*, \varepsilon_3)^\top\Big), \quad B^{2,4} \coloneqq \mathrm{B}\Big((\varepsilon_1, z_2^*, \ell_3)^\top, (u_1, u_2, z_3^*)^\top\Big)$$

for $Q^{2,2}$,

$$B^{2,5} \coloneqq \mathrm{B}\Big((\varepsilon_1, \ell_2, \varepsilon_3)^\top, (u_1, z_2^*, u_3)^\top\Big)$$

for $Q^{2,3}$ and

$$B^{2,6} \coloneqq \mathrm{B}\Big((\ell_1, \ell_2, \varepsilon_3)^\top, (\varepsilon_1, z_2^*, u_3)^\top\Big), \quad B^{2,7} \coloneqq \mathrm{B}\Big((\ell_1, z_2^*, \varepsilon_3)^\top, (z_1^*, u_2, u_3)^\top\Big)$$

for $Q^{2,4}$.

For problem $(P^3_{\varepsilon_2,\varepsilon_3})$, we get the quarters

$$Q^{3,1} \coloneqq \mathrm{B}\Big(\ell, (u_1, \varepsilon_2, \varepsilon_3)^\top\Big), \quad Q^{3,2} \coloneqq \mathrm{B}\Big((\ell_1, \ell_2, \varepsilon_3)^\top, (u_1, \varepsilon_2, u_3)^\top\Big),$$

$$Q^{3,3} \coloneqq \mathrm{B}\Big((\ell_1, \varepsilon_2, \varepsilon_3)^\top, u\Big), \quad Q^{3,4} \coloneqq \mathrm{B}\Big((\ell_1, \varepsilon_2, \ell_3)^\top, (u_1, u_2, \varepsilon_3)^\top\Big),$$

with boxes

$$B^{3,1} \coloneqq \mathrm{B}\Big((z_1^*, \ell_2, z_3^*)^\top, (u_1, z_2^*, \varepsilon_3)^\top\Big), \quad B^{3,2} \coloneqq \mathrm{B}\Big((z_1^*, \ell_2, \ell_3)^\top, (u_1, \varepsilon_2, z_3^*)^\top\Big)$$

for $Q^{3,1}$,

$$B^{3,3} \coloneqq \mathrm{B}\Big((\ell_1, \ell_2, \varepsilon_3)^\top, (z_1^*, \varepsilon_2, u_3)^\top\Big), \quad B^{3,4} \coloneqq \mathrm{B}\Big((z_1^*, \ell_2, \varepsilon_3)^\top, (u_1, z_2^*, u_3)^\top\Big)$$

for $Q^{3,2}$,

$$B^{3,5} \coloneqq \mathrm{B}\Big((\ell_1, \varepsilon_2, \varepsilon_3)^\top, (z_1^*, u_2, u_3)^\top\Big)$$

for $Q^{3,3}$ and

$$B^{3,6} \coloneqq \mathrm{B}\Big((\ell_1, \varepsilon_2, \ell_3)^\top, (z_1^*, u_2, \varepsilon_3)^\top\Big), \quad B^{3,7} \coloneqq \mathrm{B}\Big((z_1^*, \varepsilon_2, \ell_3)^\top, (u_1, u_2, z_3^*)^\top\Big)$$

for $Q^{3,4}$.

A.2 A Box-Gap in the Practical Implementation of the Box-Algorithm

We have seen in Chapter 5 that it would be useful for the practical implementation of the Box-Algorithm for three objectives to introduce a box-gap between neighbored boxes in order to prevent upcoming issues regarding a specific selection rule (cf. Subsection 5.8.3), to exclude found solutions from *Rep* in future subproblems, in particular for needed feasible-tests (cf. Subsection 5.8.1), or to get disjoint boxes. Moreover, with a box-gap also the boundary of the regions described in Proposition 5.5 (cf. Figure 5.1) can be excluded. If the underlying single-criterion solver supports strict inequalities, it is also possible to use strict inequalities instead of box-gaps to solve these issues.

Since the following modifications are similar for all three subproblems, we again limit our discussion to the first subproblem $(P^1_{\varepsilon_1,\varepsilon_2})$.

Suppose that (depending on the problem) a small box-gap $\beta > 0$ (e.g., $\beta = 10^{-3}$) is given. To overcome the above depicted issues, we propose the following modified subdivision after we have found an outcome z^* in a box $\mathrm{B}(\ell, u)$ (cf. Definition 5.8):

$$B^{1,1} \coloneqq \mathrm{B}\Big((\ell_1, z_2^*, z_3^*)^\top, (\max\{z_1^* - \beta, \ell_1\}, \max\{\varepsilon_2 - \beta, z_2^*\}, u_3)^\top\Big),$$

$$B^{1,2} \coloneqq \mathrm{B}\Big((\ell_1, \ell_2, \min\{z_3^* + \beta, u_3\})^\top, (\varepsilon_1, \max\{z_2^* - \beta, \ell_2\}, u_3)^\top\Big),$$

$$B^{1,3} \coloneqq \mathrm{B}\Big((\min\{\varepsilon_1 + \beta, u_1\}, \ell_2, \ell_3)^\top, (u_1, \max\{\varepsilon_2 - \beta, \ell_2\}, \max\{z_3^* - \beta, \ell_3\})^\top\Big),$$

$$B^{1,4} \coloneqq \mathrm{B}\Big((\min\{\varepsilon_1 + \beta, u_1\}, \ell_2, z_3^*)^\top, (u_1, \max\{z_2^* - \beta, \ell_2\}, u_3)^\top\Big),$$

$$B^{1,5} \coloneqq \mathrm{B}\Big((\varepsilon_1, \varepsilon_2, \ell_3)^\top, (u_1, u_2, \max\{z_3^* - \beta, \ell_3\})^\top\Big),$$

$$B^{1,6} \coloneqq \mathrm{B}\Big((\ell_1, \min\{\varepsilon_2 + \beta, u_2\}, \ell_3)^\top, (\max\{\varepsilon_1 - \beta, \ell_1\}, u_2, \max\{z_3^* - \beta, \ell_3\})^\top\Big),$$

$$B^{1,7} \coloneqq \mathrm{B}\Big((\ell_1, \varepsilon_2, z_3^*)^\top, (\max\{z_1^* - \beta, \ell_1\}, u_2, u_3)^\top\Big).$$

With these modifications, the Box-Algorithm works analogously but without the depicted issues. We conclude this section with two remarks.

Remark A.1:
Using a box-gap in the practical implementation, Lemma 5.14 and all results based on it do then not necessarily hold true since points between two neighbored boxes could be excluded from further investigations. However, we assume that the box-gap is chosen sufficiently small such that no point is missed during the algorithm. ◁

Remark A.2:
If 3OP has three integer-valued objective functions, an additional box-gap is not needed since, in this case, the regions between two boxes for which no integer outcome is possible can be excluded. ◁

A.3 Pseudocode

Algorithm A.1 Box-Algorithm (for three objectives, volume)

Input: A multiple objective optimization problem with three objectives, $\Delta > 0$.
Output: A representative system *Rep* fulfilling the volume-based Δ-accuracy (cf. Theorem 5.28).

Rep $\leftarrow \emptyset$, $\mathcal{B} \leftarrow \emptyset$
$\mathrm{B}(\ell^0, u^0) \leftarrow$ InitialBox()
$S \leftarrow \{\mathrm{B}(\ell^0, u^0)\}$
while $S \neq \emptyset$ **do** // *termination condition*
  $\mathrm{B}(\ell, u) \leftarrow$ SelectBox(S) // *selection rule*
  $S \leftarrow S \backslash \{\mathrm{B}(\ell, u)\}$
  if $\mathrm{Vol}(\mathrm{B}(\ell, u)) \leq \Delta$ **then** // *accuracy condition*
    Solve $(P^1_{u_1,u_2})$ and obtain optimal outcome $\hat{z}$. //*completion step*
    if $\hat{z} \neq NULL$ **then**
      Rep $\leftarrow$ *Rep* $\cup \{\hat{z}\}$
      $\mathcal{B} \leftarrow \mathcal{B} \cup \{\mathrm{B}(\ell, u)\}$
  else // *update step*
    Solve $(P^1_{\varepsilon_1,\varepsilon_2})$ with $\varepsilon_j = \dfrac{\ell_j + u_j}{2}$, $j = 1, 2$, and obtain optimal outcome z^*.
    if $z^* \neq NULL$ **then** // $z^* \in Y$
      Rep $\leftarrow$ *Rep* $\cup \{z^*\}$
      for $i = 1, \ldots, 7$ **do**
        $S \leftarrow S \cup \{B^{1,i}\}$
    else // $z^* = NULL$
      for $i = 2, 3, 4$ **do**
        $S \leftarrow S \cup \{Q^{1,i}\}$
return (*Rep*, $\mathcal{B}$)

A.4 Listings

Listing A.1: Python code to draw extended bar charts for some given *points*

```
import matplotlib
import matplotlib.pyplot as plt
from numpy import *

def find_min_and_maxpoint(points,n):
    minpoint = array([float('Inf')]*n)
```

```
    maxpoint = -minpoint
    for p in points:
        for i in xrange(n):
            if p[i] < minpoint[i]:
                minpoint[i]=p[i]
            if p[i] > maxpoint[i]:
                maxpoint[i]=p[i]
    return (minpoint,maxpoint)

def create_rectangle(start, length, edgecolor='black',
   facecolor='white', fill=True):
    return matplotlib.patches.Rectangle(start, length, 1,
       facecolor=facecolor, edgecolor=edgecolor, linewidth=2,
        fill=fill)

# points is a matrix/doubled list where each row corresponds
   to a point, e.g. points = array([[1,3,2],[4,2,3],[2,1,4]])
def plot_extended_charts(points,color_map=plt.get_cmap('
   gist_rainbow')):
    L = 100.0
    height = 1
    start_x = 0
    start_y = 0

    (m,n) = shape(points)
    lambda_weights = [1.0/n]*n

    if m>50:
        raise IOError('No more than 50 alternatives')


    # Create colors
    colors = [];
    for i in xrange(n):
        colors.append(color_map(float(i)/(n-1)))

    (minpoint,maxpoint) = find_min_and_maxpoint(points,n)

    fig = plt.figure()
    ax = fig.add_subplot(111)
```

```
for i in xrange(m):
    current_point = points[m-1-i]
    curr_x = start_x
    curr_y = start_y + (height+1)*i
    plt.text(curr_x, curr_y + height +0.1, 'alternative'+
        str(m-i))
    for j in xrange(n):
        curr_length = (current_point[j] - minpoint[j])/(
            maxpoint[j]-minpoint[j])*lambda_weights[j]*L
        remainder_length = lambda_weights[j]*L -
            curr_length
        curr_x_bound = curr_x
        lower_left_corner_point = (curr_x,curr_y)
        rect = create_rectangle(lower_left_corner_point,
            curr_length,'white',colors[j])
        curr_x = curr_x + curr_length
        lower_left_corner_point = (curr_x, curr_y)
        rect_remainder = create_rectangle(
            lower_left_corner_point, remainder_length, '
            white','white')
        curr_x = curr_x + remainder_length
        rect_bound = create_rectangle((curr_x_bound,curr_y
            ), lambda_weights[j]*L, 'black', fill=False)
        ax.add_patch(rect)
        ax.add_patch(rect_remainder)
        ax.add_patch(rect_bound)



for j in xrange(n):
    plt.text(start_x + j*(L*lambda_weights[j]),start_y-1,'
        criterion'+str(j+1), bbox=dict(facecolor=colors[j],
        alpha=0.6))

ax.set_xticks([])
ax.set_yticks([])
plt.xlim([-10, L+10])
plt.ylim([-2, curr_y + height + 1])
plt.show()
```

Notation

$\mathbb{N}$	set of natural numbers $\{1, 2, \ldots\}$
$\mathbb{N}_0$	$\mathbb{N} \cup \{0\}$
$\mathbb{Z}$	set of integer numbers $\{0, \pm 1, \pm 2, \ldots\}$
$\mathbb{R}$	set of real numbers
$\leqq$	$y^1 \leqq y^2 :\Leftrightarrow y_i^1 \leq y_i^2 \ \forall i = 1, \ldots, p$
$\leq$	$y^1 \leq y^2 :\Leftrightarrow y_i^1 \leqq y_i^2$ but $y^1 \neq y^2$
$<$	$y^1 < y^2 :\Leftrightarrow y_i^1 < y_i^2 \ \forall i = 1, \ldots, p$
$\mathbb{R}^p_{\leqq}$	$\{y \in \mathbb{R}^p : y \leqq 0\}$
$\mathbb{R}^p_{\leq}$	$\{y \in \mathbb{R}^p : y \leq 0\}$
$\mathbb{R}^p_{<}$	$\{y \in \mathbb{R}^p : y < 0\}$
X	feasible set of a MOP
Y	outcome set of a MOP
X_E	efficient set of a MOP
Y_N	nondominated set of a MOP
y^I	ideal point
y^{AI}	anti-ideal point
y^N	nadir point
Rep	representative system for some MOP
$\|\cdot\|_\infty$	maximum norm, $\|x\|_\infty \coloneqq \max\{\|x_1\|, \ldots, \|x_n\|\}$ for $x \in \mathbb{R}^n$
$\|\cdot\|_p$	p-norm, $\|x\|_p \coloneqq \left(\sum_{i=1}^n \|x_i\|^p\right)^{1/p}$ for $x \in \mathbb{R}^n$ and $1 \leq p < \infty$
$\lambda^{\mathcal{L}}(\cdot)$	Lebesgue measure
A_{ij}	entry of a matrix $A \in \mathbb{R}^{m \times n}$ corresponding to the ith row and jth column

Bibliography

Aggarwal, A., Klawe, M. M., Moran, S., Shor, P., and Wilber, R. (1987). Geometric applications of a matrix-searching algorithm. *Algorithmica*, 2(1-4):195–208.

Aggarwal, A. and Park, J. (1988). Notes on searching in multidimensional monotone arrays. In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science*, pages 497–512, White Plains, NY, USA.

Aggarwal, A. and Park, J. (1989). Sequential searching in multidimensional monotone arrays. Technical report, Massachusetts Institute of Technology.

Aggarwal, A., Schieber, B., and Tokuyama, T. (1994). Finding a minimum-weight k-link path in graphs with the concave monge property and applications. *Discrete & Computational Geometry*, 12(1):263–280.

Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Network Flows*. Prentice Hall, NJ.

Alves, M. J. and Costa, J. P. (2009). An exact method for computing the nadir values in multiple objective linear programming. *European Journal of Operational Research*, 198(2):637–646.

Bader, J. (2009). *Hypervolume-Based Search for Multiobjective Optimization: Theory and Methods*. PhD thesis, Eidgenössische Technische Hochschule Zürich.

Bader, J. and Zitzler, E. (2011). HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1):45–76.

Barichard, V. and Hao, J.-K. (2003). A population and interval constraint propagation algorithm. In Fonseca, C., Fleming, P., Zitzler, E., Thiele, L., and Deb, K., editors, *Evolutionary Multi-Criterion Optimization*, volume 2632 of *Lecture Notes in Computer Science*, pages 88–101. Springer Berlin / Heidelberg.

BBC News Online (2013). Derailed train explodes in Lac-Megantic, Quebec. `http://www.bbc.co.uk/news/world-us-canada-23213284`.

Belenky, A. (1997). A 2-person game on a polyhedral set of connected strategies. *Computers & Mathematics with Applications*, 33(6):99–125.

Benson, H. (1995). A geometrical analysis of the efficient outcome set in multiple objective convex programs with linear criterion functions. *Journal of Global Optimization*, 6(3):231–251.

Benson, H. (1998). An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple objective linear programming problem. *Journal of Global Optimization*, 13(1):1–24.

Benson, H. P. and Sayın, S. (1997). Towards finding global representations of the efficient set in multiple objective mathematical programming. *Naval Research Logistics (NRL)*, 44(1):47–67.

Beume, N., Naujoks, B., and Emmerich, M. (2007). SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669.

Boland, N., Charkhgard, H., and Savelsbergh, M. (2014a). The L-shape search method for triobjective integer programming. *Optimization Online, preprint*.

Boland, N., Charkhgard, H., and Savelsbergh, M. (2014b). A simple and efficient algorithm for solving three objective integer programs. *Optimization Online, preprint*.

Bringmann, K. and Friedrich, T. (2010). An efficient algorithm for computing hypervolume contributions. *Evolutionary Computation*, 18(3):383–402.

Bringmann, K., Friedrich, T., and Klitzke, P. (2014). Two-dimensional subset selection for hypervolume and epsilon-indicator. In *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation*, GECCO '14, pages 589–596, Vancouver, BC, Canada.

Colson, B., Marcotte, P., and Savard, G. (2007). An overview of bilevel optimization. *Annals of Operations Research*, 153(1):235–256.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms*. The MIT Press, Cambridge, Massachusetts, USA / London, England, 3rd edition.

Dächert, K. and Klamroth, K. (2013). A linear bound on the number of tricriteria optimization problems. Technical report, Bergische Universität Wuppertal.

Das, I. and Dennis, J. (1998). Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657.

Daskin, M. S. (2013). *Network and Discrete Location*. Wiley, Hoboken, NJ, USA, 2nd edition.

Deb, K., Miettinen, K., and Chaudhuri, S. (2010). Toward an estimation of nadir objective vector using a hybrid of evolutionary and local search approaches. *IEEE Transactions on Evolutionary Computation*, 14(6):821–841.

Dempe, S. (2003). Bilevel programming - a survey. Technical Report 2003-11, Fakultät für Mathematik und Informatik, TU Bergakademie Freiberg.

Dhaenens, C., Lemesre, J., and Talbi, E. G. (2010). K-PPM: A new exact method to solve multi-objective combinatorial optimization problems. *European Journal of Operational Research*, 200(1):45–53.

Diakonikolas, I. and Yannakakis, M. (2009). Small approximate pareto sets for biobjective shortest paths and other problems. *SIAM Journal on Computing*, 39(4):1340–1371.

Die Rheinpfalz (2014a). Bombe explodiert: Ernstweiler am Abend evakuiert. `http://www.rheinpfalz.de/nachrichten/titelseite/artikel/bombe-explodiert-ernstweiler-am-abend-evakuiert/`.

Die Rheinpfalz (2014b). Bombenfund: Entschärfung auf Rhein Höhe Petersau. `http://www.rheinpfalz.de/nachrichten/titelseite/artikel/bombenfund-entschaerfung-auf-rhein-hoehe-petersau/`.

Die Rheinpfalz (2014c). Flakgranate gesprengt: Anwohner können zurück in Häuser. `http://www.rheinpfalz.de/nachrichten/titelseite/artikel/flakgranate-gesprengt-anwohner-koennen-zurueck-in-haeuser/`.

Die Rheinpfalz (2014d). Gefahr am Rheinufer. `http://www.rheinpfalz.de/nachrichten/titelseite/artikel/gefahr-am-rheinufer/`.

Die Rheinpfalz (2014e). Landau: Wieder Bombenfund auf Gartenschaugelände. `http://www.rheinpfalz.de/nachrichten/titelseite/artikel/landau-wieder-bombenfund-auf-gartenschaugelaende/`.

Die Rheinpfalz (2014f). Weltkriegsbombe in Mainz entdeckt: Größere Evakuierung geplant. `http://www.rheinpfalz.de/nachrichten/titelseite/artikel/weltkriegsbombe-in-mainz-entdeckt-groessere-evakuierung-geplant/`.

Dutta, J. and Kaya, C. Y. (2011). A new scalarization and numerical method for constructing the weak pareto front of multi-objective optimization problems. *Optimization*, 60(8-9):1091–1104.

Ehrgott, M. (2005). *Multicriteria Optimization*. Springer, Berlin, 2nd edition.

Ehrgott, M. and Gandibleux, X. (2000). A survey and annotated bibliography of multiobjective combinatorial optimization. *OR Spectrum*, 22(4):425–460.

Ehrgott, M. and Gandibleux, X. (2004). Approximative solution methods for multiobjective combinatorial optimization. *Top*, 12(1):1–63.

Ehrgott, M., Löhne, A., and Shao, L. (2012). A dual variant of Benson's "outer approximation algorithm" for multiple objective linear programming. *Journal of Global Optimization*, 52(4):757–778.

Ehrgott, M., Shao, L., and Schöbel, A. (2011). An approximation algorithm for convex multi-objective programming problems. *Journal of Global Optimization*, 50(3):397–416.

Ehrgott, M. and Tenfelde-Podehl, D. (2003). Computation of ideal and nadir values and implications for their use in MCDM methods. *European Journal of Operational Research*, 151(1):119–139.

Ehrgott, M. and Wiecek, M. (2005). Multiobjective programming. In *Multiple Criteria Decision Analysis: State of the Art Surveys*, volume 78 of *International Series in Operations Research & Management Science*, pages 667–708. Springer New York.

Eichfelder, G. (2009a). An adaptive scalarization method in multiobjective optimization. *SIAM Journal on Optimization*, 19(4):1694–1718.

Eichfelder, G. (2009b). Scalarizations for adaptively solving multi-objective optimization problems. *Computational Optimization and Applications*, 44(2):249–273.

Eppstein, D. (1998). Finding the k shortest paths. *SIAM Journal on Computing*, 28(2):652–673.

Erfurth, S. (2012). Multicriteria network location with vector-valued edge costs. Diploma thesis, Department of Mathematics, University of Kaiserslautern.

Falk, J. E. (1973). A linear max-min problem. *Mathematical Programming*, 5(347):169–188.

Faulkenberg, S. and Wiecek, M. (2012). Generating equidistant representations in biobjective programming. *Computational Optimization and Applications*, 51(3):1173–1210.

Fernández, J. and Tóth, B. (2007). Obtaining an outer approximation of the efficient set of nonlinear biobjective problems. *Journal of Global Optimization*, 38(2):315–331.

Fernández, J. and Tóth, B. (2009). Obtaining the efficient set of nonlinear biobjective optimization problems via interval branch-and-bound methods. *Computational Optimization and Applications*, 42(3):393–419.

Fliege, J. (2006). An efficient interior-point method for convex multicriteria optimization problems. *Mathematics of Operations Research*, 31(4):825–845.

Forster, O. (2011). *Analysis 2: Differentialrechnung im $\mathbb{R}^n$, gewöhnliche Differentialgleichungen*. Vieweg+Teubner, Wiesbaden.

Galperin, E. A. (2004). Set contraction algorithm for computing pareto set in nonconvex nonsmooth multiobjective optimization. *Mathematical and Computer Modelling*, 40(7–8):847–859.

Goerigk, M., Grün, B., and Heßler, P. (2013). Branch and bound algorithms for the bus evacuation problem. *Computers & Operations Research*, 40(12):3010–3020.

Goerigk, M. and Grün, B. (2012). The robust bus evacuation problem. *KLUEDO, University of Kaiserslautern, preprint*.

Gourion, D. and Luc, D. T. (2008). Generating the weakly efficient set of nonconvex multiobjective problems. *Journal of Global Optimization*, 41(4):517–538.

Gourion, D. and Luc, D. T. (2010). Finding efficient solutions by free disposal outer approximation. *SIAM Journal on Optimization*, 20(6):2939–2958.

Hamacher, H. W. and Klamroth, K. (2006). *Lineare Optimierung und Netzwerkoptimierung*. Vieweg, Wiesbaden.

Hamacher, H. W., Ky, V. K., and Grün, B. (2015). Perimeter-based box algorithm for bi-criteria optimization with application to multi-commodity flows. *in preparation*.

Hamacher, H. W., Pedersen, C. R., and Ruzika, S. (2007). Finding representative systems for discrete bicriterion optimization problems. *Operations Research Letters*, 35(3):336–344.

Heermann, C., Weyers, B., and Fliege, J. (2005). A new adaptive algorithm for convex quadratic multicriteria optimization. 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.

Heyde, F. and Löhne, A. (2008). Geometric duality in multiple objective linear programming. *SIAM Journal on Optimization*, 19(2):836–845.

Huband, S., Hingston, P., While, L., and Barone, L. (2003). An evolution strategy with probabilistic mutation for multi-objective optimisation. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC '03)*, volume 4, pages 2284–2291, Canberra, Australia.

IFRC (2015). What is a disaster? `https://www.ifrc.org/en/what-we-do/disaster-management/about-disasters/what-is-a-disaster/`.

Igel, C., Hansen, N., and Roth, S. (2007). Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation*, 15(1):1–28.

Karasakal, E. and Köksalan, M. (2009). Generating a representative subset of the nondominated frontier in multiple criteria decision making. *Operations Research*, 57(1):187–199.

Kim, I. Y. and de Weck, O. L. (2005). Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and Multidisciplinary Optimization*, 29(2):149–158.

Kim, I. Y. and de Weck, O. L. (2006). Adaptive weighted sum method for multiobjective optimization: A new method for pareto front generation. *Structural and Multidisciplinary Optimization*, 31(2):105–116.

Kirlik, G. and Sayın, S. (2014). A new algorithm for generating all nondominated solutions of multiobjective discrete optimization problems. *European Journal of Operational Research*, 232(3):479–488.

Kirlik, G. and Sayın, S. (2015). Computing the nadir point for multiobjective discrete optimization problems. *Journal of Global Optimization*, 62(1):79–99.

Köksalan, M. and Lokman, B. (2015). Finding nadir points in multi-objective integer programs. *Journal of Global Optimization*, 62(1):55–77.

Korhonen, P., Salo, S., and Steuer, R. E. (1997). A heuristic for estimating nadir criterion values in multiple objective linear programming. *Operations Research*, 45(5):751–757.

Kouvelis, P. and Sayın, S. (2006). Algorithm robust for the bicriteria discrete optimization problem. *Annals of Operations Research*, 147(1):71–85.

Kuhn, T., Fonseca, C. M., Paquete, L., Ruzika, S., Duarte, M. M., and Figueira, J. R. (2015). Hypervolume subset selection in two dimensions: Formulations and algorithms. *Evolutionary Computation*.

Kuhn, T., Fonseca, C. M., Paquete, L., Ruzika, S., and Figueira, J. R. (2014). Hypervolume subset selection in two dimensions: Formulations and algorithms. *KLUEDO, University of Kaiserslautern, preprint*.

Kuhn, T. and Ruzika, S. (2014a). A coverage-based box-algorithm to compute a representation for optimization problems with three objective functions. Report in Wirtschaftsmathematik 152, University of Kaiserslautern.

Kuhn, T. and Ruzika, S. (2014b). Toward a methodology of combining optimization models. In Langton, S., Morton, A., Geiger, M. J., and Siebert, J., editors, *Decision Analysis and Multiple Criteria Decision Making: Proceedings of the Joint GOR- and DASIG-Conference 2013*, Operations Research, pages 25–44, Hamburg, Germany.

Ky, V. K. (2012). Change of variable methods and representative systems for multiobjective multicommodity network flows. Master's thesis, Department of Mathematics, University of Kaiserslautern.

Laumanns, M., Thiele, L., and Zitzler, E. (2006). An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method. *European Journal of Operational Research*, 169(3):932–942.

Legriel, J., Le Guernic, C., Cotton, S., and Maler, O. (2010). Approximating the pareto front of multi-criteria optimization problems. In Esparza, J. and Majumdar, R., editors, *Tools and Algorithms for the Construction and Analysis of Systems*, volume 6015 of *Lecture Notes in Computer Science*, pages 69–83. Springer Berlin / Heidelberg.

Lehtonen, T. and Pulkkinen, P. (2010). A simple method for approximating a general pareto surface. *Journal of Multi-Criteria Decision Analysis*, 17(3-4):115–124.

Lotov, A. and Maiskaya, T. (2012). Nonadaptive methods for polyhedral approximation of the edgeworth-pareto hull using suboptimal coverings on the direction sphere. *Computational Mathematics and Mathematical Physics*, 52(1):31–42.

Luc, D. T., Phong, T. Q., and Volle, M. (2005). Scalarizing functions for generating the weakly efficient solution set in convex multiobjective problems. *SIAM Journal on Optimization*, 15(4):987–1001.

Marcotte, P. and Savard, G. (2005). Bilevel programming: A combinatorial perspective. In Avis, D., Hertz, A., and Marcotte, O., editors, *Graph Theory and Combinatorial Optimization*, pages 191–217. Springer US.

Masin, M. and Bukchin, Y. (2008). Diversity maximization approach for multiobjective optimization. *Operations Research*, 56(2):411–424.

Messac, A. (1996). Physical programming: Effective optimization for computational design. *AIAA Journal*, 34(1):149–158.

Messac, A., Ismail-Yahaya, A., and Mattson, C. A. (2003). The normalized normal constraint method for generating the pareto frontier. *Structural and Multidisciplinary Optimization*, 25(2):86–98.

Messac, A. and Mattson, C. A. (2004). Normal constraint method with guarantee of even representation of complete pareto frontier. *AIAA Journal*, 42(10):2101–2111.

Miettinen, K. (2014). Survey of methods to visualize alternatives in multiple criteria decision making problems. *OR Spectrum*, 36(1):3–37.

Monz, M. (2006). *Pareto Navigation–interactive multiobjective optimisation and its application in radiotherapy planning*. PhD thesis, Department of Mathematics, University of Kaiserslautern.

Nemhauser, G. L. and Wolsey, L. A. (1999). *Integer and Combinatorial Optimization*. Wiley, New York.

Özlen, M. and Azizoğlu, M. (2009). Multi-objective integer programming: A general approach for generating all non-dominated solutions. *European Journal of Operational Research*, 199(1):25–35.

Papadimitriou, C. H. and Yannakakis, M. (2000). On the approximability of trade-offs and optimal access of web sources. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 86–92, Redondo Beach, California, USA.

Pascoletti, A. and Serafini, P. (1984). Scalarizing vector optimization problems. *Journal of Optimization Theory and Applications*, 42(4):499–524.

Pospelov, A. (2009). Approximating the convex edgeworth-pareto hull in integer multi-objective problems with monotone criteria. *Computational Mathematics and Mathematical Physics*, 49(10):1686–1699.

Ralphs, T., Saltzman, M., and Wiecek, M. (2006). An improved algorithm for solving biobjective integer programs. *Annals of Operations Research*, 147(1):43–70.

Rennen, G., van Dam, E. R., and den Hertog, D. (2011). Enhancement of sandwich algorithms for approximating higher-dimensional convex pareto sets. *INFORMS Journal on Computing*, 23(4):493–517.

Ruzika, S. (2007). *On Multiple Objective Combinatorial Optimization*. PhD thesis, Department of Mathematics, University of Kaiserslautern.

Ruzika, S. and Wiecek, M. (2005). Approximation methods in multiobjective programming. *Journal of Optimization Theory and Applications*, 126(3):473–501.

Sayın, S. (1996). An algorithm based on facial decomposition for finding the efficient set in multiple objective linear programming. *Operations Research Letters*, 19(2):87 – 94.

Sayın, S. (2000). Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. *Mathematical Programming*, 87(3):543–560.

Sayın, S. (2003). A procedure to find discrete representations of the efficient set with specified coverage errors. *Operations Research*, 51(3):427–436.

Sayın, S. and Kouvelis, P. (2005). The multiobjective discrete optimization problem: A weighted min-max two-stage optimization approach and a bicriteria algorithm. *Management Science*, 51(10):1572–1581.

Schwarz, H. R. and Köckler, N. (2011). *Numerische Mathematik*. Vieweg+Teubner, Wiesbaden, 8th edition.

Shao, L. and Ehrgott, M. (2007). Finding representative nondominated points in multiobjective linear programming. In *Proceedings of the IEEE Symposium on Computational Intelligence in Multicriteria Decision Making*, pages 245–252, Honolulu, Hawaii, USA.

Shao, L. and Ehrgott, M. (2008a). Approximately solving multiobjective linear programmes in objective space and an application in radiotherapy treatment planning. *Mathematical Methods of Operations Research*, 68(2):257–276.

Shao, L. and Ehrgott, M. (2008b). Approximating the nondominated set of an MOLP by approximately solving its dual problem. *Mathematical Methods of Operations Research*, 68(3):469–492.

Shimizu, K. and Aiyoshi, E. (1981). A new computational method for stackelberg and min-max problems by use of a penalty method. *IEEE Transactions on Automatic Control*, 26(2):460–466.

Solanki, R. S., Appino, P. A., and Cohon, J. L. (1993). Approximating the noninferior set in multiobjective linear programming problems. *European Journal of Operational Research*, 68(3):356–373.

Steuer, R. E. and Choo, E.-U. (1983). An interactive weighted tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26(3):326–344.

Steuer, R. E. and Harris, F. W. (1980). Intra-set point generation and filtering in decision and criterion space. *Computers & Operations Research*, 7(1–2):41–53.

Sylva, J. and Crema, A. (2007). A method for finding well-dispersed subsets of nondominated vectors for multiple objective mixed integer linear programs. *European Journal of Operational Research*, 180(3):1011–1027.

The Guardian (2011). Japan earthquake forces thousands to evacuate in nuclear plant emergency. `http://www.theguardian.com/world/2011/mar/11/japan-earthquake-evacuate-nuclear-plant`.

The Guardian (2014). Indonesia volcano eruption prompts mass evacuation. `http://www.theguardian.com/world/2014/feb/14/indonesia-volcano-mount-kelud-java-evacuation`.

Tsaggouris, G. and Zaroliagis, C. (2009). Multiobjective optimization: Improved FPTAS for shortest paths and non-linear objectives with applications. *Theory of Computing Systems*, 45(1):162–186.

Utyuzhnikov, S. V., Fantini, P., and Guenov, M. D. (2009). A method for generating a well-distributed pareto set in nonlinear multiobjective optimization. *Journal of Computational and Applied Mathematics*, 223(2):820–841.

Vassilvitskii, S. and Yannakakis, M. (2004). Efficiently computing succinct trade-off curves. In Díaz, J., Karhumäki, J., Lepistö, A., and Sannella, D., editors, *Automata, Languages and Programming*, volume 3142 of *Lecture Notes in Computer Science*, pages 1201–1213. Springer Berlin / Heidelberg.

Zhang, W. and Gao, T. (2006). A min-max method with adaptive weightings for uniformly spaced pareto optimum points. *Computers & Structures*, 84(28):1760 – 1769.

Zitzler, E. and Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms - A comparative case study. In Eiben, A. E., Bäck, T., Schoenauer, M., and Schwefel, H.-P., editors, *Conference on Parallel Problem Solving from Nature (PPSN V)*, volume 1498 of *Lecture Notes in Computer Science*, pages 292–301. Springer Berlin / Heidelberg.

Index

Curriculum Vitae

03/2007	Graduation (Abitur) at Hofenfels-Gymnasium Zweibrücken
04/2007-09/2011	Studies in mathematics and computer science (minor) at the University of Kaiserslautern
	Specialization Optimization
	Student Assistant at the University of Kaiserslautern
09/2011	Diplom degree in mathematics from the University of Kaiserslautern
10/2011-03/2012	Student Assistant at the University of Kaiserslautern
since 04/2012	Doctorate in mathematics at the University of Kaiserslautern
since 05/2012	Research Assistant at the Optimization Research Group of the Department of Mathematics at the University of Kaiserslautern
	Project DSS_Evac_Logistics (Decision Support System for Large-Scale Evacuation Logistics)

Wissenschaftlicher und Beruflicher Werdegang

03/2007	Abitur am Hofenfels-Gymnasium Zweibrücken
04/2007-09/2011	Studium der Mathematik und Informatik (Nebenfach) an der Technischen Universität Kaiserslautern
	Schwerpunkt Optimierung
	Wissenschaftliche Hilfskraft an der Technischen Universität Kaiserslautern
09/2011	Diplom in Mathematik, verliehen von der Technischen Universität Kaiserslautern
10/2011-03/2012	Wissenschaftliche Hilfskraft an der Technischen Universität Kaiserslautern
seit 04/2012	Promotion in Mathematik an der Technischen Universität Kaiserslautern
seit 05/2012	Wissenschaftlicher Mitarbeiter in der Arbeitsgruppe Optimierung des Fachbereichs Mathematik der Technischen Universität Kaiserslautern
	Projekt DSS_Evac_Logistics (Decision Support System for Large-Scale Evacuation Logistics)