Numerical Methods for the Solution of Bi-Level Multi-Objective Optimization Problems

Von der Fakultät für Elektrotechnik,
Informatik und Mathematik
der Universität Paderborn
zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften
– Dr. rer. nat. –
genehmigte Dissertation

von
Alessandro Dell'Aere

Paderborn, 2008

Acknowledgements

First of all I want to thank my advisor Prof. Dr. Michael Dellnitz for giving me the opportunity to do my PhD on a very interesting field of research. It was a great pleasure to work with him and his group that has always provided a nice research environment to me. Then I want to thank the second reviewers Prof. Dr. Joachim Böcker and Prof. Dr. Stefan Schäffler, and the members of the examination board Prof. Dr. Norbert Köckler, Prof. Dr. Henning Krause, and Dr. habil. Martin Ziegler.

My working group has been like a family to me. Here I want to thank Tanja Bürger, Dr. Wang Fang, Kathrin Flaßkamp, Sebastian Hage-Packhäuser, Mirko Hessel-von Molo (imagine there is a vector space...), Christian Horenkamp, Prof. Dr. Oliver Junge, Marianne Kalle, Stefan Klus, Dr. Arvind Krishnamurty, Anna-Lena Meyer, Dr. Kathrin Padberg, Michael Petry, Dr. Marcus Post, Dr. Oliver Schütze, Marcel Schwalb, Stefan Sertl, Bianca Thiere, Katrin Witting, but special thanks are directed to Dr. Robert Preis, who has always been ready to help out in any case of problems and to Dr. Sina Ober-Blöbaum for the pleasurable ambiance in our office and for an exciting canoe ride.

Moreover I have to mention the students Maik Ringkamp, Albert Seifried, and Dominik Steenken. They have always been ready to support me, especially to do computational work.

During the last years, my work was partly supported by the German Research Foundation within the Collaborative Research Center 614: 'Self-optimizing Concepts and Structures in Mechanical Engineering' and therefore I want to thank the German Research Foundation and all the people involved in this project, especially Tobias Knoke, Eckehard Münch, and Henner Vöcking, who were always motivated for interesting interdisciplinary cooperations.

I also want to thank Dr. Gabriele Eichfelder from the University of Erlangen for visiting me in Paderborn and for fruitful discussions. Then I have to thank Dr. Kai Gehrs. During our study we have been able to solve many difficult exercises in cooperation and in our free time we had a lot of fun playing our guitars.

However, this list would not be complete without special thanks to all my friends. And last but not least, I want to thank my wife Heike and my daughter Pia Sophia for a number of things.

Abstract

Both *bi-level optimization* and *multi-objective optimization* are of crucial importance in many modern sciences and accordingly they have attracted great interest in many publications of the last decades. Since many applications, in particular in the field of self-optimizing systems, become more and more complex during the last years, in this thesis we go a step ahead and consider *bi-level multi-objective optimization problems*. Such problems can be understood as bi-level optimization problems, where the subproblems of both levels are given by multi-objective optimization problems. We develop the theoretical background and practical algorithms for the solution of these problems. Convergence of the algorithms is proved and their strength is demonstrated by academic example problems and real world applications.

Zusammenfassung

Sowohl *Bilevel-Optimierung* als auch *Mehrziel-Optimierung* sind von großer Bedeutung für viele moderne Wissenschaften und haben dementsprechend in vielen Publikationen der letzten Jahrzehnte großes Interesse erfahren. Da in den letzten Jahren viele Anwendungen, insbesondere im Bereich der selbstoptimierenden Systeme, immer komplexer werden, gehen wir in dieser Dissertation einen Schritt weiter und betrachten *Bilevel-Mehrzieloptimierungsprobleme*. Diese Probleme können als Bilevel-Optimierungsprobleme aufgefasst werden, bei denen die Teilprobleme der beiden Levels in Form von Mehrziel-Optimierungsproblemen vorliegen. Wir entwickeln die theoretische Basis und anwendbare Algorithmen zur Lösung dieser Probleme. Die Konvergenz der Algorithmen wird bewiesen und ihre Stärke wird an Hand von akademischen Beispielproblemen und realistischen Anwendungen gezeigt.

Contents

1 Introduction

Many problems in all areas of applied science, engineering, economics and statistics can be posed in terms of optimization. In particular, such problems can be characterized by the fact that several objective functions have to be optimized at the same time. For instance, for a perfect passenger car one wants to simultaneously *minimize energy consumption* and *maximize comfort*. As indicated by this example the different objectives typically contradict each other and therefore certainly do not have identical optima. Thus, the goal is to approximate the "optimal compromises" which, in mathematical terms, are called *Pareto points*. It turned out that in general there is an extensive set of Pareto points, the *Pareto-set*. Such optimization problems are known as *multi-objective optimization problems* (MOP).

There are two main directions in solving multi-objective optimization problems. On one hand, it might be sufficient to compute a single solution inside the Pareto set. In this case several methods like for instance the *weighted sums method* ([18, 34]) or the *ε-constraint method* ([18, 34]) are good choices. On the other hand, in many applications it is desired to know the entire Pareto set. In this case, *set-oriented methods* ([10, 11, 39]) or the *normal boundary intersection method* ([7]) work very efficiently. Certainly, if the entire Pareto set is available, the *decision making* has to be performed, that is, one particular Pareto point has to be selected for adjusting the system under consideration.

Very often the objectives of multi-objective optimization problems are outputs of a complex system, where interactions between the associated subsystems restrict the *feasible set* of the optimization problem. In particular the feasible set of a multi-objective optimization problem (the *higher level problem*) can itself be defined by the solution set of another multi-objective optimization problem (the *lower level problem*) within an underlying subsystem. Let us illustrate this fact by an example. Suppose one wants to minimize in a multi-objective sense both comfort and energy consumption of a vehicle, which forms a complex system. But for safety reasons, also optimality concerning the mechanical guidance both in the horizontal and the vertical direction of the undercarriage regarded as an underlying subsystem has to be guaranteed. Since this again is a multi-objective optimization problem, we are concerned with a *hierarchy* of multi-objective optimization problems. In many applications, the lower level problem is in addition parametrized by a part of the upper level variables, such that the resulting overall problem turns out to be very complex and hard to solve. Following the notion of classical *bi-level optimization*

problems, which are for instance considered in [1] and [13], the described hierarchy forms a *bi-level structure* of multi-objective optimization problems and therefore we call these problems *bi-level multi-objective optimization problems* (BLMOP).

The consideration of such problems was originally motivated by the author's work within the Collaborative Research Center 614 of the German Research Foundation, which is concerned with self-optimizing systems in mechanical engineering. To state the relevance of classical and bi-level multi-objective optimization in self-optimizing systems, we first point out that the process of self-optimization is performed repeatedly in the following three steps. In Step 1 the current state of the technical system is analyzed. Depending on the outcome of this analysis, a system of goals to be achieved is generated in Step 2. In Step 3 the behavior of the technical system has to be adapted due to the goals determined in Step 2. More details on self-optimizing systems can be found in [21]. In particular, the system of goals - or a part of it - can be made up not only by the fact that several performance properties have to meet certain values but also by an importance rating among these performance properties. Mathematically speaking, several possibly contradicting objective functions have to be optimized simultaneously in Step 3, that is, a multi-objective optimization problem has to be solved repeatedly during the self-optimizing process, while the weights of the objectives are varying. Since the objectives are outputs of a complex system, the corresponding feasible set can be restricted by the solution of another parametrized multi-objective optimization problem, that is, bi-level multi-objective optimization problems have to be solved in this case. In practice, instead of solving the BLMOP in every cycle of the self-optimization process, the entire Pareto set can be computed in advance, such that Step 3 is reduced to an inexpensive decision making process instead of solving the relatively expensive BLMOP. Therefore, in this work we concentrate on computing the Pareto set of a BLMOP, a task which is regarded as an important part of the complex self-optimizing process.

In the literature one can find many contributions dealing with classical bi-level optimization problems, which are characterized by the fact that every level is made up by a classical, that is, a scalar valued optimization problem. But there are only a few publications concerned with bi-level multi-objective optimization: first of all we want to cite [16], from where we have taken some examples to demonstrate the efficiency of our new algorithms. Similar problems with several parametrized lower level multi-objective problems and an upper level multi-objective problem are investigated in [32] and [42]. Statements on the existence of solutions for these problems are given in [32]. In [42] an algorithm for the solution of these problems is proposed,

but this algorithm is restricted to the computation of a single *preferred solution* out of the entire solution set. In [2] the authors consider problems with a parametrized lower level multi-objective problem and a scalar upper level problem. Problems with a scalar upper level problem and a linear lower level multi-objective problem are investigated in [44] and [8]. Such hierarchical problems with a non-parametric lower level multi-objective problem belong to the field of *optimization over the efficient set*.

In this work we propose different new approaches for the solution of bi-level multi-objective optimization problems. These methods are *set-oriented* in the sense that they allow to approximate the entire Pareto set (or a desired part of it) rather than just the computation of single points inside this set. The development of these methods was essentially influenced by two different concepts in technical applications. To see the difference between these directions, observe that there might be several solutions in a Pareto set for which the vectors of objective values are identical. On one hand it can be reasonable for many applications and it seems to be efficient to restrict the algorithms to the computation of a *complete set of alternatives* ([15]), that is, for every Pareto optimal point in the space of objective values (the *objective space* or *image space*) only one corresponding point in the space of parameters (the *parameter space* or *pre-image space*) has to be computed. But in other technical applications it can be very important to select Pareto optimal solutions which do not only correspond to a desired vector of objective values but also have additional properties like for example (in a certain sense) robustness. Moreover, for online applications as described in [46], small distances between certain solutions in parameter space are required. Thus, for these applications, the computation of the entire Pareto set provides the best basis for selecting solutions in order to adjust the technical system. According to the mentioned requirements corresponding to the two different views on applications, in this work we present both *parameter set-oriented methods* for the computation of entire Pareto sets and *image set-oriented methods* which are designed to compute a complete set of alternatives. Most of our algorithms are realized by the use of a *multi-level subdivision structure* in order to compute a tight covering formed by a collection of subsets of the parameter or image space, respectively, such that the union of these subsets contains the solution. Since these coverings are supposed to approximate the solution, they are computed in a way such that the standard Hausdorff distance between the solution and the computed covering is smaller than a given value. We have implemented

our algorithms in the software package GAIO[1], which was developed at the Chair of Applied Mathematics of the University of Paderborn and provides the required multi-level subdivision structure.

A more detailed outline of this thesis is as follows. In Section 2, we state more precisely, that is, from a mathematical point of view, how the bi-level multi-objective optimization problem is defined. To this end, we first recall the basic definitions and the theoretical background for classical multi-objective optimization problems in Section 2.1 ([15], [18], [34]). In particular we present the well-known Kuhn-Tucker necessary conditions ([31]) for a Pareto optimal solution, because these will provide the essential basis for the derivation of optimality conditions for BLMOP in Section 6.1. In Section 2.2 we consider classical bi-level optimization problems ([12], [13], [45]). We state the basic definition for these problems and we present both the *optimistic* and *pessimistic* formulation, which are the traditional concepts that have been introduced in order to obtain unique solutions. Based on the content of Sections 2.1 and 2.2, we are capable of giving the mathematical definition for BLMOP in Section 2.3. In particular, since this is the main interest of this thesis, we also state the corresponding optimistic formulation.

The methods for the solution of classical multi-objective optimization problems, from which we lend the basic concepts for the development of our new methods for the solution of BLMOP, are presented in Section 3. More detailed, in Section 3.1 we concentrate on the set-oriented methods ([11], [39]), which are capable of computing the entire Pareto set of MOP. In Section 3.2 we give a short introduction to reference point methods ([18]), which turned out to be the proper method for the solution of the particular subproblem, which has to be solved repeatedly in our image set-oriented methods.

Section 4 contains new results and forms one of the main parts of this thesis. Here, we are concerned with a new image set-oriented method for the generation of a complete set of alternatives of multi-objective optimization problems. Originally, we developed a variant of this method for the solution of BLMOP, but it turned out that this concept can also be used for the solution of more general constrained multi-objective optimization problems. In Section 4.1, the algorithm and its realization is described in detail. Since this method uses scalarizations based on reference point methods, in Section 4.2 we compare it to other scalarization methods, that is in particular the *weighted sums method*. In Section 4.3 we explain the advantages of the image set-oriented methods in the case of convex objective functions. Particularly,

[1] http://math-www.uni-paderborn.de/∼agdellnitz/gaio/

we show that in this case the complete set of alternatives corresponds to the entire Pareto set. Furthermore, an interactive hierarchical concept for the fast computation of a desired part of the solution based on the image set-oriented method is described in Section 4.4.

Some basic methods for the solution of classical bi-level optimization problems can be found in Section 5. Here, we mention very briefly several concepts presented in [12], but we concentrate on the Kuhn-Tucker based concepts that we use for the development of our algorithms for the solution of BLMOP. We also describe a problem arising in the presence of lower level inequality constraints, that is, the *violation of constraint qualifications*, see [12] and [33]. In Section 5.1, we present how such problems can be overcome by reformulations which use *merit functions* or *smoothing functions* and how *smoothing methods* help to solve these reformulations ([22], [28]).

Section 6 forms another main part of this thesis and is concerned with both theoretical and practical results for BLMOP without lower level inequality constraints and a convex lower level problem. This section is divided as follows. In Section 6.1 we derive – based on the theoretical background given in the preceding Sections – necessary optimality conditions for these problems. In Section 6.2 we propose several set-oriented methods for the solution of this particular subclass of BLMOP. These are on one hand *subdivision* algorithms, which are realized by the use of subsets of the parameter space, and on the other hand *recovering* algorithms, which work with subsets either in parameter space or in image space. We also prove convergence (in a sense to be stated precisely later on) for most of these algorithms. Thereafter, in Section 6.3 we concentrate on the particular case of a non-parametric lower level problem and derive a corresponding variant of the necessary optimality conditions. This is motivated by the fact that several applications fall into this family of problems, which are termed *Pareto set constrained multi-objective optimization problems* (PSCMOP). In Section 6.4, we present algorithms both of subdivision and recovering nature for the solution of PSCMOP. In particular, we present derivative-free algorithms, which can be efficiently implemented because of the simpler structure of PSCMOP.

For the sake of completeness, we included a short Section 6.5 on the solution of BLMOP with lower level inequality constraints. Here, we propose as an example an algorithm which was realized by extending one of the previously described algorithms using the smoothing concept mentioned in Section 5.1. In addition, we present an algorithm for the solution of BLMOP with non-convex and non-differentiable

objectives and constraints in Section 6.6.

In Section 7 we perform a sensitivity analysis for MOP and BLMOP, that is, we are interested in the variation of the Pareto set caused by the variation of an additional perturbation parameter. To this end, we first review the sensitivity analysis for classical optimization problems ([17]) in Section 7.1. Based on this background, in Section 7.2 we derive the desired sensitivity analysis for MOP and BLMOP. As an application, in Section 7.3 we use the results derived in Section 7.2 for the development of a concept for the adaptive choice of algorithmic parameters (the *targets*), which helps to control the spreading among the Pareto points computed by the image set-oriented methods described in Section 4.1 and Section 6.2.

In Section 8, the thesis closes with a summary of the results and a discussion about open problems and possible future directions.

2 Basic Definitions and Concepts

2.1 Multi-Objective Optimization

In this section we review the theoretical background on multi-objective optimization needed in this work. We briefly summarize the basic concepts including first order necessary optimality conditions (Kuhn-Tucker conditions) and a direction of descent. Then we describe the set-oriented methods for solving classical multi-objective optimization problems, from which we adapt the main concepts in order to develop an image set-oriented variant of these methods in Section 4 and set-oriented methods for the solution of bi-level multi-objective optimization problems in Section 6.

As mentioned in Section 1, the task of a *multi-objective optimization problem* MOP is to optimize several real valued objective functions $F_i : \mathbb{R}^n \to \mathbb{R}$ simultaneously. Before the MOP can be stated mathematically we have to define a partial order on $\mathbb{R}^k$.

DEFINITION 2.1 Let $v, w \in \mathbb{R}^k$. Then the vector v is *less than* w ($v <_p w$), if $v_i < w_i$ for all $i \in \{1, \ldots, k\}$. The relation $\leq_p$ is defined in an analogous way.

We collect k objective functions F_i in the vector valued function

$$F : \mathbb{R}^n \to \mathbb{R}^k, \qquad F(x) = (F_1(x), \ldots, F_k(x))^t.$$

Additionally, if there are $p \leq n$ equality constraints H_i and q inequality constraints G_i to be satisfied, that is, $H_i(x) = 0$ for $i = 1, \ldots, p$ and $G_i(x) \leq 0$ for $i = 1, \ldots, q$,

then we collect them analogously in vector valued functions

$$H : \mathbb{R}^n \to \mathbb{R}^p, \qquad H(x) = (H_1(x), \ldots, H_p(x))^t$$

and

$$G : \mathbb{R}^n \to \mathbb{R}^q, \qquad G(x) = (G_1(x), \ldots, G_q(x))^t$$

and denote the feasible set by

$$S = \{x \in \mathbb{R}^n : H(x) = 0 \text{ and } G(x) \leq_p 0\}.$$

With this notation we can state the multi-objective optimization problem as follows:

$$\min_{x \in S} F(x), \tag{MOP}$$

where minimization has to be understood in the sense of Definition 2.1. A point $\bar{x} \in S$ is a solution of MOP, if for any other $x \in S$ either $F(x) = F(\bar{x})$ or the value $F_i(x)$ of at least one objective F_i is greater than $F_i(\bar{x})$. To be more precise, we state the following

Definition 2.2 Consider the multi-objective optimization problem MOP. Then a point $\bar{x} \in S$ is called *(globally) Pareto optimal* or a *(global) Pareto point* if there is no $y \in S$ such that

$$F(y) \neq F(\bar{x}) \quad \text{and} \quad F(y) \leq_p F(\bar{x}). \tag{2.1}$$

A point $\bar{x} \in S$ is a *local Pareto point*, if there is a neighborhood $U(\bar{x})$ of $\bar{x}$ such that there is no $y \in U(\bar{x}) \cap S$ satisfying (2.1).

The following definition will be useful, because some of the algorithms described in this work use the partial order $\leq_p$ within certain sets of points.

Definition 2.3 (i) For $\mathcal{X} \subset \mathbb{R}^n$ we call a point $x \in \mathcal{X}$ *nondominated* with respect to F and $\mathcal{X}$ or *F-nondominated* with respect to $\mathcal{X}$ if there does not exist any point $y \in \mathcal{X}$ with $F(y) \neq F(x)$ and $F(y) \leq_p F(x)$.

(ii) For $\mathcal{X} \subset \mathbb{R}^n$ we call a point $x \in \mathcal{X}$ *nondominated* with respect to F, G, H and $\mathcal{X}$ or $F_{(G,H)}$*-nondominated* with respect to $\mathcal{X}$, if $G(x) \leq_p 0$, $H(x) = 0$ and there does not exist any point $y \in \mathcal{X}$ with $G(y) \leq_p 0$, $H(x) = 0$, $F(y) \neq F(x)$ and $F(y) \leq_p F(x)$.

Since $\leq_p$ just defines a partial order on $\mathbb{R}^k$, one cannot expect to find isolated Pareto points. Under certain assumptions the solution to an unconstrained MOP locally

forms a $(k-1)$-dimensional manifold, see [25]. On the other hand, if the constraints of a MOP define an r-dimensional manifold, we can expect that generically the solution to the MOP locally forms a manifold of dimension $\min\{k-1, r\}$.
The following theorem of Kuhn and Tucker ([31]) states a necessary condition for Pareto optimality.

THEOREM 2.4 *Let x^* be a Pareto point of MOP and assume that the vectors $\nabla H_i(x^*)$, $i = 1, \ldots, p$ and $\nabla G_j(x^*)$, $j \in \{l : G_l(x^*) = 0\}$ are linearly independent.*
Then there exist scalars $\alpha_1, \ldots, \alpha_k, \mu_1, \ldots, \mu_q \geq 0, \lambda_1, \ldots, \lambda_p \in \mathbb{R}$, such that

$$\begin{aligned} \sum_{i=1}^{k} \alpha_i &= 1, \\ \mu_i G_i(x^*) &= 0 \ \textit{for} \ i = 1, \ldots, q \qquad \textit{and} \\ \sum_{i=1}^{k} \alpha_i \nabla F_i(x^*) + \sum_{i=1}^{p} \lambda_i \nabla H_i(x^*) &+ \sum_{i=1}^{q} \mu_i \nabla G_i(x^*) = 0. \end{aligned} \tag{2.2}$$

Points satisfying (2.2) are not necessarily Pareto optimal, but certainly "Pareto candidates" and thus we now emphasize their relevance by the following

DEFINITION 2.5 A point $x \in \mathbb{R}^n$ is called a *substationary point* of MOP if there exist scalars $\alpha_1, \ldots, \alpha_k, \mu_1, \ldots, \mu_q \geq 0$, $\lambda_1, \ldots, \lambda_p \in \mathbb{R}$ such that (2.2) is satisfied.

To find substationary points, one can use a direction of descent, that is, a direction in $\mathbb{R}^n$ in which all the k objectives are simultaneously non-increasing and at least one objective is decreasing. Several choices of descent directions can for instance be found in [3], [23], and [37] and references therein. As an example we review the descent direction given in [37]. To this end, we associate with $F : \mathbb{R}^n \to \mathbb{R}^k$, $F(x) = (F_1(x), \ldots, F_k(x))^t$, the following quadratic optimization problem for every fixed $x \in \mathbb{R}^n$:

$$\min_{\alpha \in \mathbb{R}^k} \left\{ \| \sum_{i=1}^{k} \alpha_i \nabla F_i(x) \|_2^2 ; \alpha_i \geq 0, i = 1, \ldots, k, \sum_{i=1}^{k} \alpha_i = 1 \right\}. \tag{QOP}$$

Then the weighted sum of gradients

$$-\sum_{i=1}^{k} \hat{\alpha}_i \nabla F_i(x),$$

where $\hat{\alpha}$ is a solution of QOP, either vanishes or is a descent direction for all objective functions $F_1, \ldots, F_k$ in x.

With this result an iteration step for finding substationary points can be defined by first computing $\hat{\alpha}$ for the current point x and then performing a line search along $-\sum_{i=1}^{k} \hat{\alpha}_i \nabla F_i(x)$ to find a new point $\tilde{x}$.

2.2 Bi-Level Optimization

In this section we give a brief introduction to *(classical) bi-level optimization*. In doing so we follow basically the definitions of [12] and concentrate on those contents required for the development of our methods for solving BLMOPs. Comprehensive overviews on bi-level optimization can be found in [1, 5, 12, 13, 45]. Bi-Level optimization (or bi-level programming) problems constitute a particular kind of hierarchical optimization problems, where a part of the constraints for the *upper (or higher) level problem* is defined by another parametric optimization problem, the *lower level problem*. To be more precise, denote by

$$F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$$

the upper level objective function. The r upper level equality constraints $H_i(x, y) = 0$ and s upper level inequality constraints $G_j(x, y) \leq 0$ are collected in the vector valued functions

$$H : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^r, \qquad H(x, y) = (H_1(x, y), \ldots, H_r(x, y))^t$$

and

$$G : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^s, \qquad G(x, y) = (G_1(x, y), \ldots, G_s(x, y))^t.$$

Analogously, denote by

$$f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$$

the lower level objective function. The p lower level equality constraints $h_i(x, y) = 0$ and q lower level inequality constraints $g_j(x, y) \leq 0$ are collected in the vector valued functions

$$h : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p, \qquad h(x, y) = (h_1(x, y), \ldots, h_p(x, y))^t$$

and

$$g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^q, \qquad g(x, y) = (g_1(x, y), \ldots, g_q(x, y))^t.$$

With these notations, the bi-level optimization problem can be written as

$$\begin{aligned} &''\min_{y}'' F(x(y), y) \\ &\quad s.t. \quad (x(y), y) \in S, \end{aligned} \tag{BLP}$$

where

$$S = \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^m : G(x, y) \leq_p 0, H(x, y) = 0, x \in \psi(y)\}$$

and $\psi(y)$ denotes the solution set of the following lower level problem:

$$\begin{aligned} \min_x \ & f(x, y) \\ s.t. \ & g(x, y) \leq_p 0, \\ & h(x, y) = 0. \end{aligned} \tag{LLP}$$

Observe that in the case of non-unique solutions $x(y) \in \psi(y)$ for the lower level problem, the notion of an optimal solution of the bi-level optimization problem is not necessarily obvious. This ambiguity is expressed by using the quotation marks in (BLP). One way out of this situation is given by the *optimistic* or *weak formulation*, where for every fixed $y \in \mathbb{R}^m$, $x(y) \in \psi(y)$ is chosen such that $F(x(y), y)$ is minimal. This optimistic formulation can be expressed as

$$\begin{aligned} \min_y \min_x \{F(x, y) : x \in \psi(y)\} & \\ s.t. \quad G(x, y) \ & \leq_p \ 0, \\ H(x, y) \ & = \ 0. \end{aligned} \tag{BLP-O}$$

Analogously, in the *pessimistic* or *strong formulation*, $x(y) \in \psi(y)$ is chosen such that $F(x(y), y)$ is maximal. This pessimistic formulation can be expressed as

$$\begin{aligned} \min_y \max_x \{F(x, y) : x \in \psi(y)\} & \\ s.t. \quad G(x, y) \ & \leq_p \ 0, \\ H(x, y) \ & = \ 0. \end{aligned} \tag{BLP-P}$$

Originally, the optimistic and pessimistic formulations were used in [43] for the description of real market situations, where different decision makers try to realize best decisions with respect to their individual aims while they are not able to realize their decisions independently but are forced to react according to a certain hierarchy. In particular, this hierarchy can be given by a bi-level structure as described above. Then, the optimistic or pessimistic formulations might be suitable, depending on the question whether the lower level decision maker (the *follower*) is willing to support the higher level decision maker (the *leader*) or not.

Non-uniqueness of lower level solutions will be particularly relevant in Section 6, where the lower level problem will be replaced by a MOP and therefore $\psi(y)$ is

given by a Pareto set which in general is an extensive set and not a singleton. In accordance to the structure of the applications we have in mind, we will concentrate on the optimistic formulation.

Let us now assume that the lower level problem LLP is convex, that is, for every fixed $y \in \mathbb{R}^m$, f and $g_i, i = 1, \ldots, q$ are convex and $h_i, i = 1, \ldots, p$ are affine-linear. Then the solution $x(y) \in \psi(y)$ is unique and we can write $x = x(y)$. Moreover, under the common regularity assumptions, LLP can be replaced by its Kuhn-Tucker conditions[2] which are necessary and sufficient in this case. In this way, with $\bar{\nabla} := \nabla_{(x)}$, an auxiliary problem equivalent to the original problem is obtained:

$$
\begin{aligned}
\min_{x,y,\zeta,\tau} \; & F(x,y) && \text{(BLP')}\\
s.t. \quad & G(x,y) \leq_p 0,\\
& H(x,y) = 0,\\
& \bar{\nabla} f(x,y) + \sum_{i=1}^{p} \zeta_i \bar{\nabla} h_i(x,y) + \sum_{i=1}^{q} \tau_i \bar{\nabla} g_i(x,y) = 0,\\
& g(x,y) \leq_p 0,\\
& h(x,y) = 0,\\
& \tau_i g_i(x,y) = 0 \quad \text{for} \quad i = 1, \ldots, q,\\
& \tau_i \geq 0 \quad \text{for} \quad i = 1, \ldots, q.
\end{aligned}
$$

There are several solution methods for (BLP), (see [12, 13, 1]), which are based on the solution of such auxiliary problems. In Section 6, we will borrow and extend this idea for the development of most of our algorithms for the solution of BLMOP.

2.3 Bi-Level Multi-Objective Optimization

In this section we introduce those problems, which make up the main problem class under consideration in this thesis, that is, the class of bi-level multi-objective optimization problems (BLMOP). This can be understood as a MOP, where some variables x have to be taken from the solution set of another MOP which is parametrized by variables y of the first MOP. In other words, BLMOP arises from BLP by replacing both the higher and lower level optimization problem by multi-objective optimization problems.

[2]The Kuhn-Tucker conditions for scalar valued optimization problems can be obtained from Theorem 2.4 by setting $k = 1$.

To express the BLMOP in mathematical terms, we use the same notations as in Section 2.2 with the exception that both the upper and lower level objectives are considered to be vector valued, that is

$$F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^k, \quad F(x, y) = (F_1(x, y), \ldots, F_k(x, y))^t$$

and

$$f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^l, \quad f(x, y) = (f_1(x, y), \ldots, f_l(x, y))^t.$$

For every $y \in \mathbb{R}^m$, let $\psi(y)$ denote the solution, that is, the Pareto set of the following lower level problem:

$$\begin{aligned} \min_x f(x, y), \qquad & \text{(BLMOP-LL)} \\ s.t. \quad g(x, y) &\leq_p 0, \\ h(x, y) &= 0, \end{aligned}$$

where minimization has to be understood in the sense of the partial order $\leq_p$. Then a BLMOP can be stated as follows:

$$\begin{aligned} {''}\min_y{''} F(x(y), y), \qquad & \text{(BLMOP)} \\ s.t. \quad (x(y), y) &\in S, \end{aligned}$$

where again minimization has to be understood in the sense of the partial order $\leq_p$ and the feasible set is defined by

$$S = \{(x, y) : G(x, y) \leq_p 0, H(x, y) = 0, x \in \psi(y)\}.$$

Motivated by applications and by the fact that for a BLMOP $\psi(y)$ is typically an extensive Pareto set, we will concentrate on a suitable variant of the optimistic formulation BLP-O. This variant differs from BLP-O in the way that the upper level problem and for every fixed y the lower level problem of BLMOP are minimized in a multi-objective sense. For this, we introduce the *Pareto-optimistic formulation* for BLMOP:

$$\begin{aligned} \min_y \min_x \{F(x, y) : x \in \psi(y)\} \qquad & \text{(BLMOP-O)} \\ s.t. \quad G(x, y) &\leq_p 0, \\ H(x, y) &= 0, \end{aligned}$$

where minimizations have to be understood in the sense of the partial order $\leq_p$. The solution of such problem makes up the main interest of this thesis and is considered in Section 6.

3 Basic Methods for Solving Multi-Objective Optimization Problems

In the last decades many methods for solving MOP have been developed. Comprehensive overviews on methods can be found for instance in [15] and [34]. There are some algorithms – e.g. the *weighted sums method* – that are only capable of computing single points of the Pareto set. Since many of these *single solution methods* can be realized by applying established optimization algorithms to scalar valued auxiliary functions, they are relatively fast and efficient. Although in some particular situations the user might be satisfied with a single Pareto point, in many applications it is desired to have an overview on the entire Pareto set. For this, algorithms for computing the entire Pareto set have been developed, see [11],[15],[20],[25], [34], and [39]. Once the entire Pareto set is available, the user has to select one of the solutions, e.g. for adjusting the technical system under consideration. On the one hand, this *decision making* problem is very challenging because the computed solutions are of equal worth from the mathematical point of view. On the other hand, having an overview on the entire Pareto set provides the best basis for selecting the right solution.

Of course, computing the entire Pareto set can be very time-consuming, particularly, if the dimension of the feasible set $S \subset \mathbb{R}^n$ is very high. In order to attenuate this drawback, other methods (e.g. the normal boundary intersection method, [7]) make a compromise and calculate only a certain subset of the Pareto set, which is sufficient for the decision making process in the sense that it forms a *complete set of alternatives* ([15]), that is, for every Pareto optimal point in image space, only one corresponding Pareto point in parameter space is computed. Our new image set-oriented method for the solution of MOP, see Section 4, belongs also to the latter class of methods and – as will be demonstrated later on – has some additional nice properties. Moreover, some of these properties gave reason to extend this algorithm, see Section 6, in order to solve BLMOP.

In the next sections we describe in more detail those methods from which we borrowed the basic ideas for the development of our methods for the solution of BLMOP. To be more precise, in Section 3.1 we review set-oriented methods which are originally tailored to compute the entire Pareto set of MOP in the absence of constraints, but – as we will see in Section 6 – can be extended in order to solve more general problems. In Section 3.2 we also give a short introduction to reference point methods, which turned out to be the proper method for finding individual

Pareto points, a subproblem, which has to be solved repeatedly as the new image set-oriented method described in Section 4 proceeds.

3.1 Set-Oriented Methods for Solving Unconstrained Multi-Objective Optimization Problems

We now briefly review the set-oriented algorithms for the approximation of the Pareto set or the set of substationary points of a given unconstrained MOP in a compact domain $Q \subset \mathbb{R}^n$. For a detailed exposition the reader is referred to [11]. Using a multi-level subdivision scheme each of these methods produces a sequence of sets $\mathcal{B}_0, \mathcal{B}_1, \mathcal{B}_2, \ldots$ where each $\mathcal{B}_j$ consists of finitely many subsets of Q. Generally, there are two different types of such methods: the methods of *subdivision* type and the methods of *recovering* type. In methods of subdivision type, every $\mathcal{B}_j$ covers the Pareto set or the set of substationary points, such that the *diameter* of the subsets shrinks as the index j increases. In methods of recovering type, the diameter of the subsets is fixed. Here, every $\mathcal{B}_j$ forms a partial covering of the Pareto set or the set of substationary points and is extended as the method proceeds.

The numerical realization of the set-oriented methods works as follows: the elements $B \in \mathcal{B}_j$ are boxes each of which is specified by a center in $\mathbb{R}^n$ and n radii. But rather than working explicitly with centers and radii of the boxes these are stored within a binary tree which leads to a significant reduction in the memory requirement. The boxes are discretized via *test points*. Strategies for the choice of test points are presented in [11].

In the following we will denote by $\mathcal{S}$ the set of substationary points within Q and we will call the elements B of $\mathcal{B}_j$ *boxes*. Let $\Phi : \mathbb{R}^n \to \mathbb{R}^n$ be an iteration scheme for finding substationary points of the MOP under consideration.

Subdivision Algorithm

The sequence $(\mathcal{B}_j)_{j\in\mathbb{N}}$ produced by the following algorithm has the property that the diameter

$$\operatorname{diam}(\mathcal{B}_j) := \max_{B\in\mathcal{B}_j} \operatorname{diam}(B)$$

tends to zero for $j \to \infty$. In fact it can be shown, see [11], that the box coverings which are created by the following algorithm converge towards $\mathcal{S}$.

Let $\mathcal{B}_0$ be an initial collection of finitely many subsets of the compact set Q such that $\cup_{B\in\mathcal{B}_0} B = Q$. Then $\mathcal{B}_j$ is inductively obtained from $\mathcal{B}_{j-1}$ in two steps:

(i) Subdivision Construct from $\mathcal{B}_{j-1}$ a new system $\hat{\mathcal{B}}_j$ of subsets such that

$$\bigcup_{B\in\hat{\mathcal{B}}_j} B = \bigcup_{B\in\mathcal{B}_{j-1}} B$$

and

$$\operatorname{diam}(\hat{\mathcal{B}}_j) = \theta_j \operatorname{diam}(\mathcal{B}_{j-1}),$$

where $0 < \theta_{min} \leq \theta_j \leq \theta_{max} < 1$.

(ii) Selection Define the new collection $\mathcal{B}_j$ by

$$\mathcal{B}_j = \left\{ B \in \hat{\mathcal{B}}_j : \text{there exists } \hat{B} \in \hat{\mathcal{B}}_j \text{ such that } \Phi^{-1}(B) \cap \hat{B} \neq \emptyset \right\}.$$

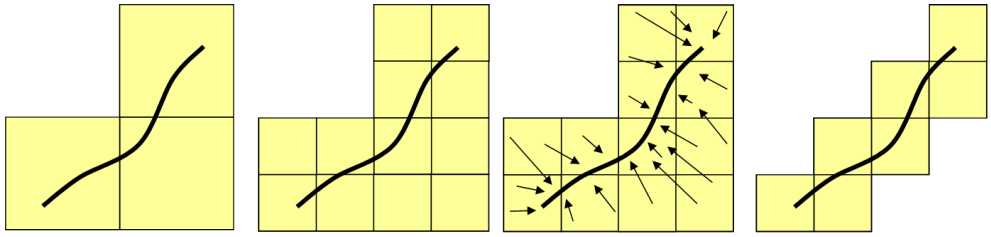

Figure 1: The idea of the Subdivision algorithm.

The described method is of global nature in the sense that the entire domain Q is explored, such that all connected components of $\mathcal{S}$ can be found, if the number of test points used for the discretization of the boxes is large enough. But it should be mentioned that particularly in the case of many test points this method is restricted to moderate dimensions of the parameter space, because the number of boxes $B \in \mathcal{B}_j$, which have to be explored in every iteration, might grow rapidly as the index j increases.

Recovering Algorithm

During the subdivision procedure boxes might get lost although they contain substationary points. This can be the case when the number of test points taken into account is not large enough. In the following we describe an algorithm which allows to recover boxes which have previously been lost, although they contain substationary points. But it should be mentioned that this algorithm is also a self-contained *continuation method*, that is, given at least one single Pareto point within the compact domain Q, further substationary points are generated successively in order to

obtain a representation of the entire Pareto set of the given MOP within Q. For formal reasons let us denote by $\mathcal{P}_d$ a *complete* partition[3] of the set Q into boxes of subdivision size – or *depth* – d, which are generated by successive bisection of Q. Then there exists for every point $x \in Q$ and every depth d exactly one box $B(x,d) \in \mathcal{P}_d$ with center c and radius r such that $c_i - r_i \leq x_i < c + r_i,\ \forall i = 1,\ldots,n$. Moreover, denote by $\Phi^q(s)$ the application of q steps of an iteration scheme for finding substationary points using an initial guess s. For a given initial box collection $\mathcal{B}_0$ the algorithm reads as follows:

(i) for all $B \in \mathcal{B}_0$

$B.active := TRUE$

(ii) for $j = 0,\ldots,MaxStep$

$\hat{\mathcal{B}}_j := \mathcal{B}_j$
for all $B \in \{B \in \mathcal{B}_j : B.active == TRUE\}$
choose starting points $\{s_i\}_{i=1,\ldots,l}$ near B
$\mathcal{X} := \{\Phi^q(s_i) | i = 1,\ldots,l\}$
$B.active := FALSE$
for all $x \in \mathcal{X}$:
if $B(x,d) \notin \hat{\mathcal{B}}_j$
$B(x,d).active := TRUE$
$\hat{\mathcal{B}}_j := \hat{\mathcal{B}}_j \cup B(x,d)$
if $\hat{\mathcal{B}}_j == \mathcal{B}_j$ **STOP**
$\mathcal{B}_{j+1} := \hat{\mathcal{B}}_j$

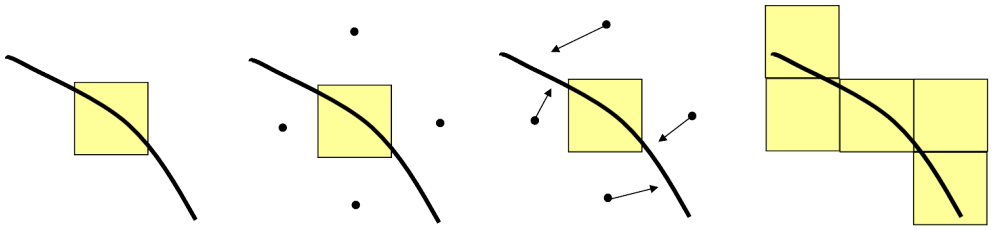

Figure 2: The idea of the Recovering algorithm.

Unfortunately, the Recovering algorithm may not perform adequately when a box does not contain part of $\mathcal{S}$ but is possibly far away. In this case many undesired

[3] $\mathcal{P}_d$ has *not* to be explicitly computed by our algorithms.

regions would be added to the box collection on the way towards $\mathcal{S}$ in the course of the iteration of test points. Strategies to overcome this problem can be found in [20].

Sampling Algorithm

There are some potential drawbacks of the algorithms described above: for many iteration schemes Φ, the gradients of the objectives are needed. But in general, gradients might not exist at all or the calculation of gradients might be very expensive from the computational point of view. Moreover, the set $\mathcal{S}$ is generally a strict superset of the Pareto set for several reasons. On the one hand, there might be local Pareto points, which are not Pareto optimal from a global point of view. On the other hand, in some applications, e.g. if the objectives are not defined outside the domain Q, penalization strategies have to be used to avoid that the iteration scheme Φ leads to points outside Q. In this case, the algorithms are capable of finding points on the boundary ∂Q of Q, which are not necessarily Pareto optimal for the given MOP. These problems can be avoided using the following sampling algorithm, which takes only the function values of the objectives into account. An outline of an iteration of the algorithm is as follows. Given a box collection $\mathcal{B}_{j-1}$ the collection $\mathcal{B}_j$ is obtained by:

(i) Subdivision Construct from $\mathcal{B}_{j-1}$ a new system $\hat{\mathcal{B}}_j$ of subsets such that

$$\bigcup_{B\in\hat{\mathcal{B}}_j} B = \bigcup_{B\in\mathcal{B}_{j-1}} B$$

and

$$\operatorname{diam}(\hat{\mathcal{B}}_j) = \theta_j \operatorname{diam}(\mathcal{B}_{j-1}),$$

where $0 < \theta_{min} \leq \theta_j \leq \theta_{max} < 1$.

(ii) Selection

for all $B \in \hat{\mathcal{B}}_j$

choose a set of **test points** $\mathcal{X}_B \subset B$

$N :=$ nondominated points of $\bigcup_{B\in\hat{\mathcal{B}}_j} \mathcal{X}_B$

$\mathcal{B}_j := \left\{ B \in \hat{\mathcal{B}}_j : \exists\, x \in \mathcal{X}_B \cap N \right\}$

Observe that, caused by the discretization of the boxes and, where required, also by the application of an iteration scheme Φ, the described set-oriented methods

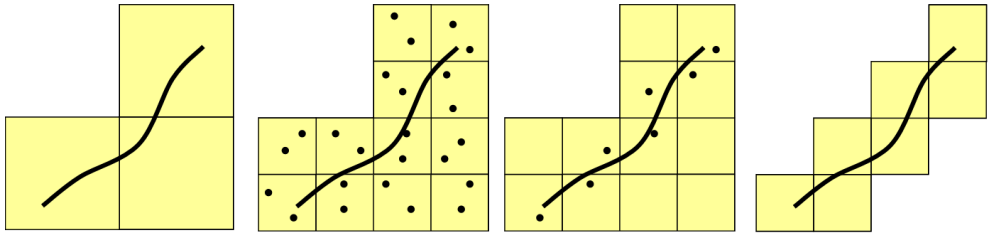

Figure 3: The idea of the Sampling algorithm.

(Subdivision, Recovering, Sampling) are capable of generating not only boxes but also substationary points within these boxes. Typically, these points can be saved in an archive and the globally nondominated points of this archive can be filtered out by nondominance tests. These points are well-distributed in parameter space in the sense that in every generated box there is at least one of these points. Such a representation of the Pareto set forms an advantageous basis for the respective decision making process in many (but not all) real world applications.

3.2 Reference Point Methods

A *reference point* $t \in \mathbb{R}^k$ can be regarded as a vector of desirable objective values called *aspiration levels* or *targets*, $t_i, i = 1, \ldots, k$. Reference point methods use feasible or infeasible reference points for the construction of scalar valued auxiliary functions. For an overview on different types of reference point methods the reader is referred to [18]. In the following we will focus on *distance function based approaches*, which are relevant for our new algorithm presented in Section 4. As indicated by its notation, distance function based approaches use a *distance function*, which is typically based on a norm, to measure the distance between a reference point and a given point in image space. To state the auxiliary problem corresponding to a target vector $t \in \mathbb{R}^k$, let $\delta : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}_+$ be a distance function derived from a norm, i.e., $\delta(a, b) = ||a - b||$ for some norm $|| \cdot || : \mathbb{R}^k \to \mathbb{R}_+$. Then the auxiliary problem to be solved is

$$\min_{x \in S} \delta(F(x), t). \tag{RPP}$$

If we have $\delta(F(x^\star), t) > 0$, where $x^\star$ is a solution to RPP, then we know that $F(x^\star)$ is on the boundary of the image $F(S) = \{F(x) : x \in S \subset \mathbb{R}^n\}$. Moreover, if in addition $t <_p F(x^\star)$ we can expect that $x^\star$ is (at least a local) Pareto point. Thus, local Pareto points can be found by first choosing proper targets and then solving RPP. Indeed, Theorem 3.1, which was taken from [15], guarantees that, under

certain assumptions, $x^\star$ is a Pareto point. For this, recall that a norm $||\cdot|| : \mathbb{R}^k \to \mathbb{R}_+$ is called *strictly monotonically increasing*, if $||y^1|| < ||y^2||$ for all $y^1, y^2 \in \mathbb{R}^k$ with $|y_j^1| \leq |y_j^2|, j = 1, 2, \ldots, k$ and $|y_j^1| \neq |y_j^2|$ for some j.

THEOREM 3.1 *Let $||\cdot||$ be a strictly monotonically increasing norm and assume $t_i = \min\{F_i(x) : x \in S\}$ for $i = 1, 2, \ldots, k$. If $x^\star$ is an optimal solution of RPP, then $x^\star$ is a solution of MOP.*

Proof: See [15]. □

In order to compute a representation of the entire Pareto set, our new image set-oriented algorithm presented in Section 4 repeatedly solves a variant of RPP while the targets are varying. To state a corollary which guarantees that the corresponding solutions are at least locally Pareto optimal we denote $T = (T_1, \ldots, T_k), T_i = \min\{F_i(x) : x \in S\}$ for $i = 1, 2, \ldots, k$ and define for a given target vector $t \in \mathbb{R}^k$ the modified feasible set

$$S_t = \{x \in S : F_i(x) \geq t_i,\ i = 1, \ldots, k\}.$$

Furthermore, we define both a modified multi-objective optimization problem and a modified auxiliary problem by replacing S by S_t:

$$\min_{x \in S_t} F(x), \tag{MOP '}$$

$$\min_{x \in S_t} \delta(F(x), t). \tag{RPP '}$$

Now, with these notations we can state the following

COROLLARY 3.2 *Let F be continuous on the compact domain S. Moreover, let $||\cdot||$ be a strictly monotonically increasing norm and assume that $T <_p t <_p F(x^\star)$, where $x^\star$ is an optimal solution of RPP'. Then $x^\star$ is a local solution of MOP.*

Proof: Since F_i is continuous and since there are $\bar{x}^i, x^\star \in S$ with $F_i(\bar{x}^i) = T_i < t_i < F_i(x^\star)$, there exist $x^i \in S$ such that $F_i(x^i) = t_i$ for all $i = 1, 2, \ldots, k$. From construction of S_t it is obvious, that $x^\star \in S_t$ and $t_i = \min\{F_i(x) : x \in S_t\}$. Thus, Theorem 3.1 guarantees, that $x^\star$ solves MOP '. Since S_t is constructed from S just by constraining the image of F, such that $F(S_t)$ contains a part of a local Pareto optimal set in image space, $x^\star$ is a local solution of MOP. □

4 A New Image Set-Oriented Recovering Method for Solving Multi-Objective Optimization Problems

As mentioned in Section 3.1, the (classical) Recovering algorithm uses a multi-level subdivision scheme in order to discretize the parameter space and to achieve diversity of the computed solution set. For our new algorithm presented in this section, this set-oriented strategy was adapted in order to operate adequately in image space, that is, we use boxes in image space rather than in parameter space. This concept is motivated by the fact that in many applications there are just a few objectives which depend on a relatively high number of parameters. Another reason for the development of this image set-oriented variant is given by the decision makers requirement that the calculated Pareto points should be well-distributed in image space rather than in parameter space. Later on in Section 6.2, we will adapt the concept of this new algorithm for the development of an image set-oriented algorithm for the solution of BLMOPs.

4.1 Recovering-IS Algorithm

The crucial difference between our new algorithm and the classical Recovering algorithm is the fact that a randomly chosen point $t \in \mathbb{R}^k$ does not necessarily belong to the image $F(S) = \{F(x) : x \in S\}$, that is, we do not know whether there is any $x \in S$ such that $F(x) = t$. Moreover, if $F(x) = t$ for some $x \in S$, we do not know whether x is Pareto optimal. To get an answer to these questions, we solve the auxiliary problem RPP' and – as mentioned in Section 3.2 – if $t <_p F(x^\star)$ for a solution $x^\star$ of RPP', then we know that – under suitable assumptions – $x^\star$ is at least locally Pareto optimal. Otherwise, if $t = F(x^\star)$, then we repeatedly have to vary t and solve RPP' until $t <_p F(x^\star)$. A strategy for the choice and variation of the targets t can be found later on in this section.

With respect to the computational effort, the described technique for evaluating a box in image space is similar to the technique for evaluating a box in parameter space as used by the classical Recovering algorithm (provided that the dimensions of both spaces are the same): for both techniques, a sufficiently number of targets or starting points, respectively, has to be chosen and consequently the same number of executions of the respective iteration scheme has to be applied. Thus, we expect that the computational effort has the same order of magnitude for both evaluation

techniques. But due to the fact that in many applications the dimension of the parameter space is much higher than the dimension of the image space, our new Recovering-IS algorithm (with boxes in image space) turns out to be more efficient in this case, because the number of points needed to represent a box grows rapidly as the dimension of the considered space grows. For example, p^m points are required for generating a consistent grid of points in an m-dimensional box using p positions in every coordinate.

In the algorithm described below and for the remainder of this work the distance function δ is based on the norm $||\cdot||_2$ that is, $\delta(a,b) = ||a-b||_2$ for all $a, b \in \mathbb{R}^k$. Observe, that for every point $F \in \mathbb{R}^k$ there is exactly one box $B(F,d)$ (in image space) of depth d containing F. Starting with a given box collection $\mathcal{B}_0$ in image space and associating a solution (x_B, F_B) to every box B, the *Recovering-IS* algorithm reads as follows:

(i) for all $B \in \mathcal{B}_0$

$B.active := TRUE$

(ii) for $j = 0, \ldots, MaxStep$

$\hat{\mathcal{B}}_j := \mathcal{B}_j$

for all $B \in \{B \in \mathcal{B}_j : B.active == TRUE\}$

choose target vectors $\{t_i\}_{i=1,\ldots,l}$ near B with $t_i <_p F_B$

$x_i^\star := \arg\min_{x \in \mathbb{R}^n} \delta(F(x), t_i), i = 1, \ldots, l$

$F_i^\star := F(x_i^\star), i = 1, \ldots, l$

$B.active := FALSE$

for all $i = 1, \ldots, l$:

if $B(F_i^\star, d) \notin \hat{\mathcal{B}}_j$

$\tilde{B} := B(F_i^\star, d),\ x_{\tilde{B}} := x_i^\star,\ F_{\tilde{B}} := F_i^\star$

$\tilde{B}.active := TRUE$

$\hat{\mathcal{B}}_j := \hat{\mathcal{B}}_j \cup \tilde{B}$

if $\hat{\mathcal{B}}_j == \mathcal{B}_j$ **STOP**

$\mathcal{B}_{j+1} := \hat{\mathcal{B}}_j$

As desired, this algorithm computes Pareto points that are well-distributed in image space in the sense that in every generated box there is one of the computed Pareto point. Consequently, these points make up a suitable basis for the decision making process related to many applications.

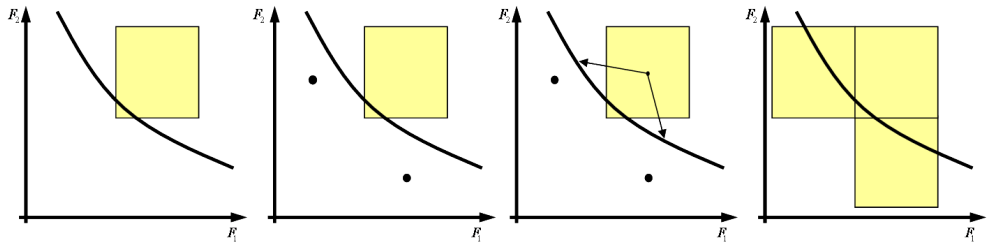

Figure 4: The idea of the Recovering-IS algorithm.

There remains the question, how to choose the target vectors $t_i, i = 1, 2, \ldots, l$ near a current box B in order to obtain a complete set of alternatives. Efficient strategies for the choice of target vectors can be defined, particularly by using local information on the Pareto set, e.g. orientation or curvature, which can be calculated via objective derivatives (or numerical approximations of the derivatives). Moreover, as presented in Section 7, sensitivity analysis can be used as an additional tool to control the distance among the computed Pareto points. In the following we will focus on a particular strategy for the choice of the targets which was originally designed for problems with smooth objectives, but is also applicable and works satisfactorily in the case of more general objectives. Let us assume that the image $F(P) \subset \mathbb{R}^k$ of the Pareto set $P \subset \mathbb{R}^n$ is smooth and forms a $(k-1)$-dimensional manifold in a neighborhood $N_\varepsilon(y^\star)$ of a given Pareto optimal point $y^\star = F(x^\star) \in F(P)$ in image space. Since an approximation of $F(P)$ at $y^\star$ is given by the tangent space $T_{y^\star}F(P)$, there are certainly further Pareto points near $T_{y^\star}F(P) \cap N_\varepsilon(y^\star)$. Consequently, we can expect that there are $\lambda \in \mathbb{R}$ and $p \in T_{y^\star}F(P) \cap N_\varepsilon(y^\star)$, such that suitable targets needed for the computation of further Pareto points can be expressed by $p + \lambda d$, where $d \leq_p 0$ denotes a basis vector of the 1-dimensional space $(T_{y^\star}F(P))^\perp$. Thus, to apply this idea in practice, we first have to construct d and a basis $V := \{b_1, b_2, \ldots, b_{k-1}\}$ of $T_{y^\star}F(P)$ and then to specify targets

$$t_i = y_i^\star + \sum_{j=1}^{k-1} \alpha_{i,j}\, b_j + \lambda_i\, d, \quad i = 1, 2, \ldots, l$$

by determining the coefficients $\alpha_{i,j}$ and λ_i. For example, the construction of two different targets in the neighborhood of a Pareto Point $y^\star$ in image space for the case $k = 2$ is depicted in Figure 5. Fortunately, if $y^\star$ was found in a previous step by solving RPP' for a given target $t^\star, t^\star <_p y^\star$, then d can be obtained without any additional effort by $d := t^\star - y^\star$. Once d is available, any standard method for the construction of an orthogonal basis, e.g. the Grahm-Schmidt method can be

used to obtain the required basis V. For all $i = 1, 2, \ldots, l$, the coefficients $\alpha_{i,j}$ are chosen such that $p_i := \sum_{j=1}^{k-1} \alpha_{i,j}\, b_j$ is located inside a neighbor box of the current box. Moreover, the p_i should be well-distributed around $y^\star$. With this heuristic, it is very likely to find new boxes containing Pareto points. For the choice of λ_i an adaptive concept has to be applied, because a computed solution $\bar{y}$ of RPP' can only be accepted, if $t_i <_p \bar{y}$ is satisfied. Such an adaptive concept should be guided by the fact that $t_i <_p \bar{y}$ certainly holds if λ_i is sufficiently large, but it should also be considered that RPP' is ill-conditioned if λ_i is too large.

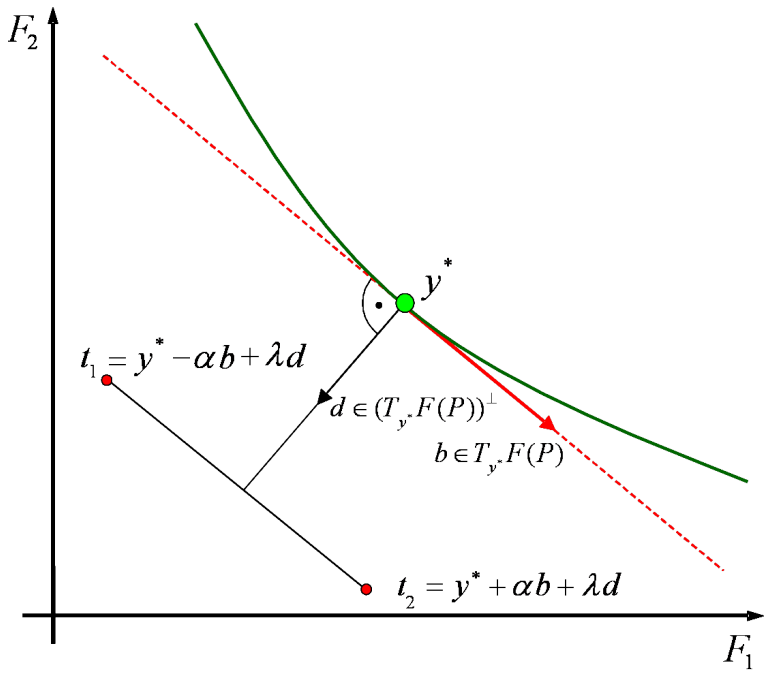

Figure 5: The construction of targets based on $b \in T_{y^\star}F(P)$ and $d \in (T_{y^\star}F(P))^\perp$.

EXAMPLE 4.1 For our first example let $F = (F_1, F_2)^t : \mathbb{R}^3 \to \mathbb{R}^2$ be defined by

$$F(x_1, x_2, x_3) = \begin{pmatrix} (x_1 - 1)^2 + (x_2 - 1)^2 + (x_3 - 1)^4 \\ (x_1 + 1)^4 + (x_2 + 1)^2 + (x_3 + 1)^2 \end{pmatrix} \tag{4.1}$$

We assume that the decision maker is only interested in solutions for which both objective values are located within the interval $I := [0, 20]$ and therefore we define $S := \{x \in \mathbb{R}^3 : F_i(x) \in I, i = 1, 2\}$ and consider the MOP

$$\min_{x \in S} F(x). \tag{4.2}$$

Figure 6 shows the solutions generated by the Recovering-IS algorithm using different box sizes (depths). Here, the reader can get an impression of how the density of the computed representation can be controlled by choosing the box size.

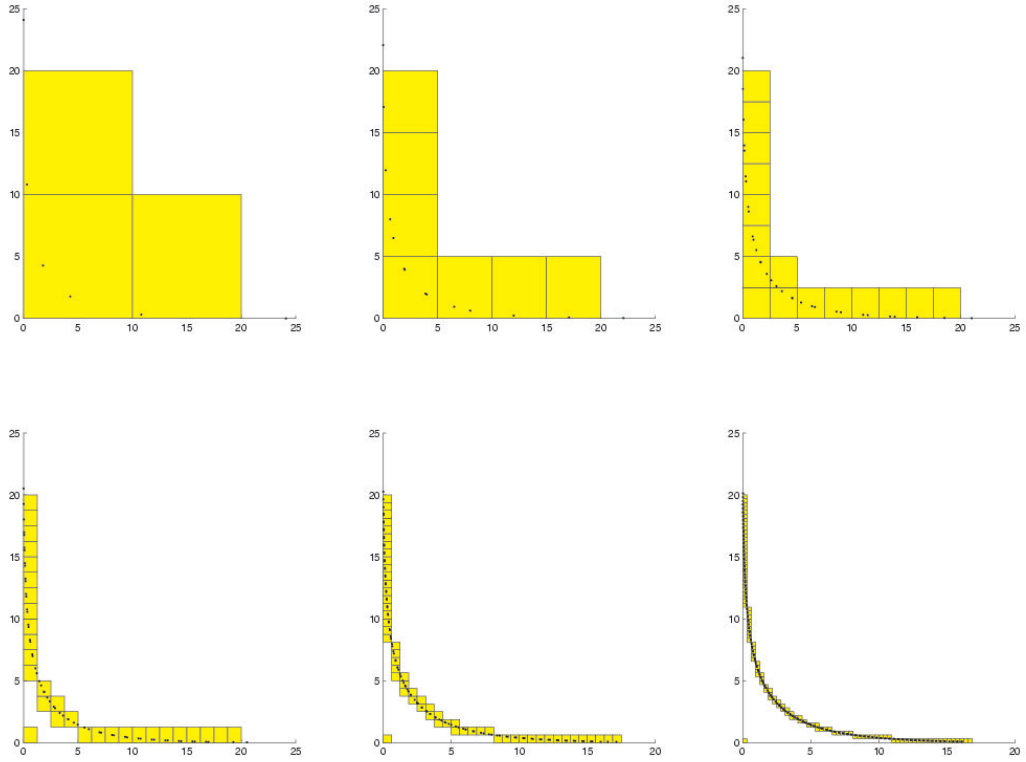

Figure 6: The solution of Example 4.1 computed by the image set-oriented recovering algorithm using different box sizes in image space.

EXAMPLE 4.2 We use the next example to demonstrate the nice property of the Recovering-IS algorithm to generate a well-distributed representation of the Pareto set. To this end let $F = (F_1, F_2, F_3)^t : \mathbb{R}^3 \to \mathbb{R}^3$ be defined by

$$F(x_1, x_2, x_3) = \begin{pmatrix} (x_1 - 1)^4 + (x_2 - 1)^2 + (x_3 - 1)^2 \\ (x_1 + 1)^2 + (x_2 + 1)^4 + (x_3 + 1)^2 \\ (x_1 - 1)^2 + (x_2 + 1)^2 + (x_3 - 1)^4 \end{pmatrix} \tag{4.3}$$

and consider the MOP

$$\min_{x \in \mathbb{R}^3} F(x). \tag{4.4}$$

Figure 7 depicts the solution computed by the Recovering-IS algorithm and Figure 8 depicts the solution computed by the classical Recovering algorithm.[4] Observe that

[4]The generated boxes are omitted in these figures in order to obtain more transparency.

the Pareto points computed by the Recovering-IS algorithm are well-distributed over the entire Pareto set (in image space). On the other hand, the distribution obtained by the classical Recovering algorithm varys drastically. Consequently, there are too many unneeded points in one region, whereas the density is possibly too sparse for the decision makers' consideration in an essential part of the Pareto set.

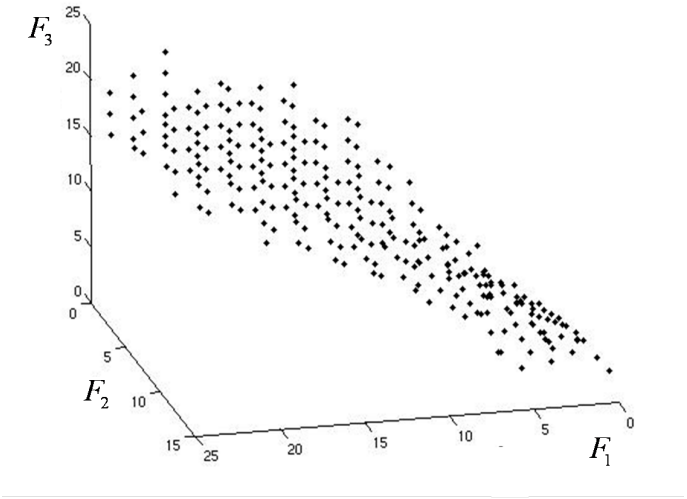

Figure 7: The solution of Example 4.2 computed by the Recovering-IS algorithm (image space).

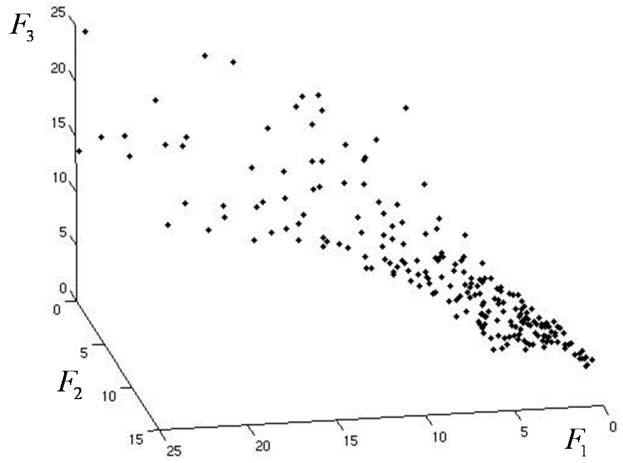

Figure 8: The solution of Example 4.2 computed by the classical Recovering algorithm (image space).

Example 4.3 Application to Mechatronics

In the following we review the main contents of [10] where the efficiency of the Recovering-IS algorithm was demonstrated on a technical application taken from [30]. Here we are concerned with an energy management problem for an onboard storage system of an overhead line powered tram as shown in Figure 9. Here the aim is to reduce both overhead line peak power and energy consumption simultaneously. The storage device accumulates the energy which is released while the vehicle is

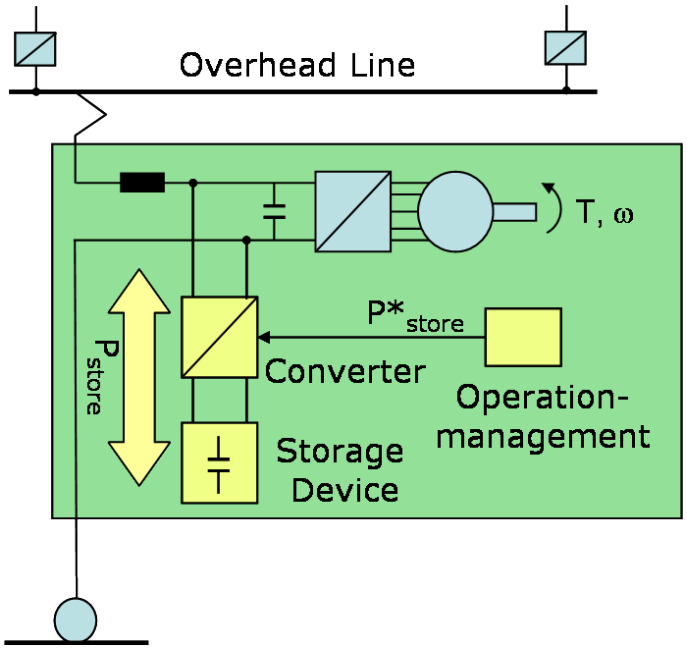

Figure 9: Block diagram of the tram system with onboard storage and operation management system.

braking and, as shown in Figure 10, this stored energy can be used later on at a suitable point of time to power the drive. Our task is to design the operation management that is, to manage this charging and discharging, such that overhead line peak power and energy consumption are minimized in a multi-objective sense. For a more detailed description of this application see [30].

Since the drive cycle of the tram is divided into 1241 track sections and the energy management system has to assign a reference value to each of these sections, we have to solve a MOP with 2 objective functions and 1241 parameters. From the engineer's point of view a solution with a linear or at least monotone behavior of the series of reference values corresponding to certain connected blocks of track sections seemed to be reasonable. Motivated by this fact, the optimization process

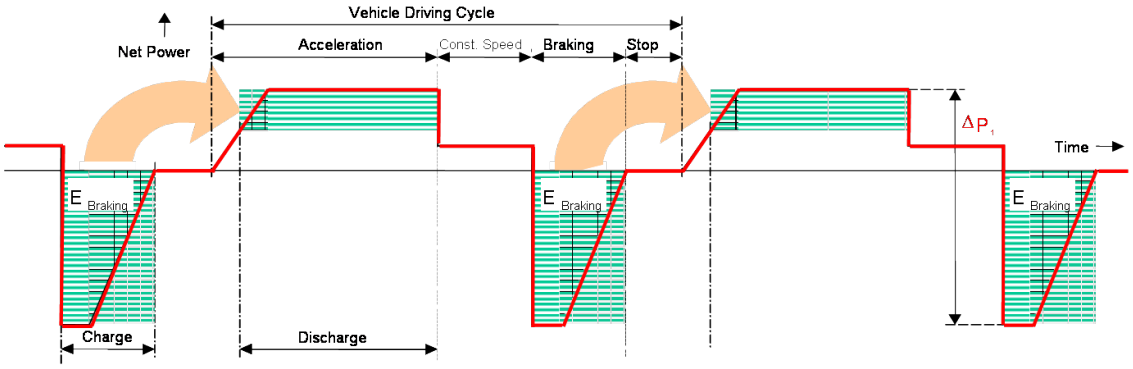

Figure 10: The operation principle of the onboard storage system.

was performed in two steps. For a *pre-optimization* step, a reduced model was constructed by grouping the parameters within each block and representing the entire block by just two parameters i.e., by a linear auxiliary function. The resulting model turned out to have only 184 parameters. Thus, it was possible to compute very quickly a first approximation to the solution of the original problem as follows. First, the Pareto set of the reduced model was computed using the Recovering-IS algorithm. Due to the smaller number of parameters of the reduced model, this step was executed relatively quickly. Then the higher-dimensional points of the desired first approximation were calculated from the Pareto points of the reduced model simply by linear interpolation. In the second step of the optimization process the Recovering-IS algorithm was applied to the original problem using the first approximation computed by the pre-optimization step as a starting iteration. This principle of first performing a pre-optimization on a simplified lower-dimensional model and taking the result for generating a starting iteration for solving the original problem turned out to be a helpful step in decreasing the necessary computational effort.

The results of the optimization procedure are shown in Figure 12. The stars are the results of the pre-optimization and the circles are the results obtained from the subsequent optimization on the original model using the results of the pre-optimization as starting points. In order to compare the objectives, they are normalized in the sense that they have the value 1 when no storage device is used.

For the purpose of comparison the reduced model has again been optimized both with the classical Recovering algorithm in parameter space and with the new Recovering-IS algorithm. To keep the results comparable, both algorithms have been initialized with the same starting point and the provided computation time was

the same for both algorithms. The archived points associated with the computed boxes are shown in Figure 11. Observe that the objective values computed by the Recovering-IS algorithm are considerably better than those computed by the Recovering algorithm in parameter space. Moreover, there is a crucial difference between the results concerning diversity. This is caused by the fact that - as in many applications - the map from the Pareto set in parameter space to the corresponding points in image space is not injective, such that the classical Recovering algorithm computes many Pareto points which are evenly distributed in the high-dimensional parameter space but possibly close together in the low-dimensional image space. On the other hand, due to the box-oriented concept in image space, the Recovering-IS algorithm produces Pareto points which are evenly distributed in image space. This shows the previously mentioned advantages of the Recovering-IS algorithm: the time-consuming but unnecessary computation of too many points is avoided by restricting to a few points which are evenly distributed in image space. These points form a sufficient subset of the Pareto set and provide a manageable basis for selecting the right solutions, particularly when this selection has to be done repeatedly in the course of a self-optimizing process.

Let us make a remark on the choice of starting points. First observe that at least one Pareto point has to be provided as a starting point for applying the Recovering or the Recovering-IS algorithm. Obviously, such starting points can be generated by just optimizing the individual objectives or a weighted sum of all objectives using a standard optimization technique for scalar valued problems. Fortunately, in the application presented here, a starting point was given because of the engineers' knowledge on the technical system, such that this preliminary step was not necessary.

4.2 Comparison with Other Methods

In this section we will first compare the (image set-oriented) Recovering-IS algorithm with the classical (parameter set-oriented) Recovering algorithm. In doing so, we will point out some advantages of the Recovering-IS algorithm, but we will also give some guidelines for combining the classical and the new method. Then we will illustrate how the solutions of the Recovering-IS algorithm are related to the solutions of the well-known weighted sums methods. We will give an example to

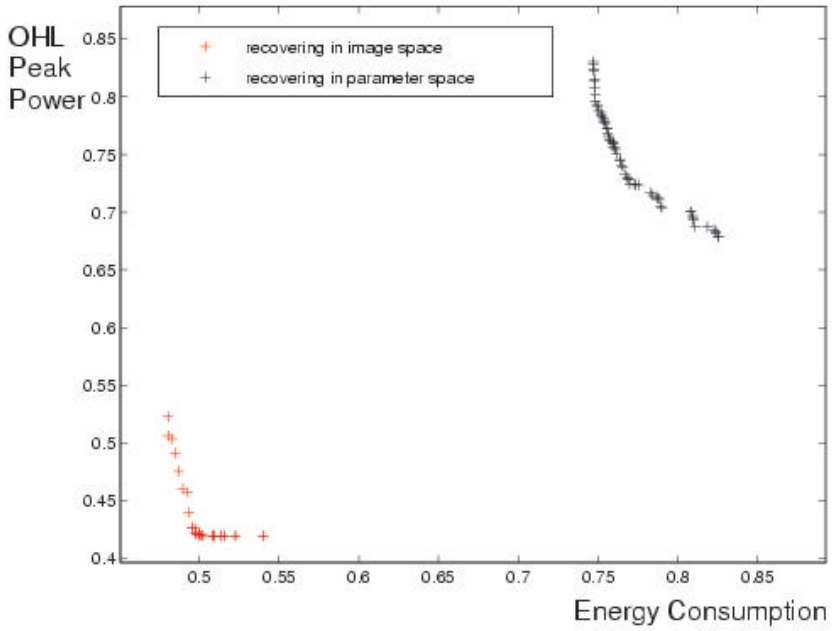

Figure 11: The solution of the tram problem computed by the IS-Recovering algorithm and by the Recovering algorithm (image space).

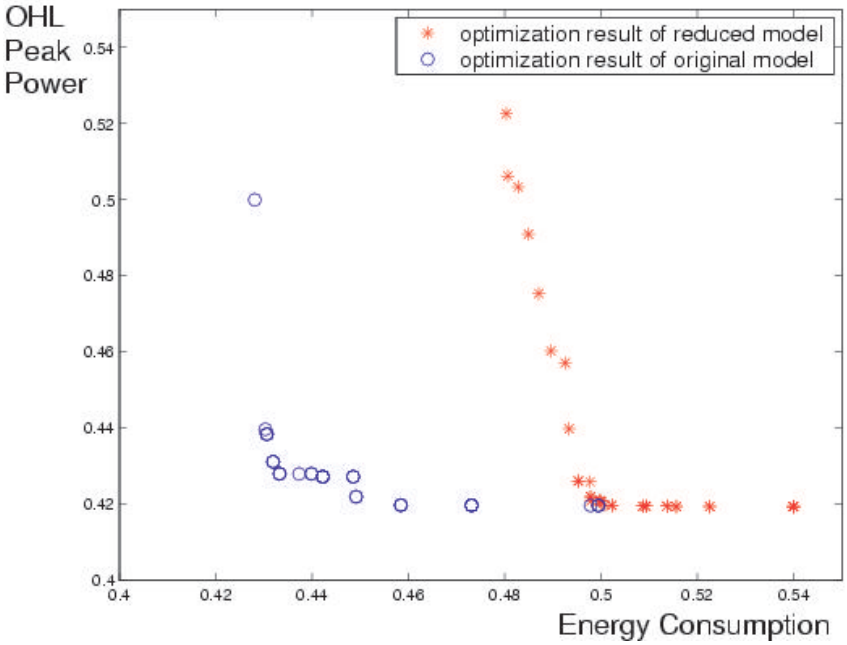

Figure 12: Optimization results for the tram model (image space).

see that the Recovering-IS algorithm is capable of finding Pareto points for which the weighted sums method is inapplicable. Moreover, we will prove that at least every solution of the weighted sums method can also be found by the reference point method, i.e., by solving RPP ', which is the particular technique used to find individual Pareto points within the Recovering-IS algorithm. Thereafter, we will state some advantages of the Recovering-IS algorithm over the normal boundary intersection method.

Comparison with the Classical Recovering Algorithm As stated in Section 4.1, the Recovering-IS algorithm is particularly advantageous in the case where the dimension of the parameter space is significantly higher than the dimension of the image space or, the number of objectives to be optimized. Moreover, the computed points are well-distributed in image space, which can be particularly advantageous from the decision maker's point of view. But there is another advantage which is independent of the dimensions. To see this, remember that in the Recovering-IS algorithm a basis of the tangent space $T_{y^\star}F(P)$, which is used for determining the targets t_i, can be obtained by some very simple calculations based on $y^\star$ and $t^\star$. But, for the classical Recovering algorithm, if one wants to construct the corresponding tangent space $T_{x^\star}P$ in parameter space in order to determine starting points (or predictors) close to the Pareto set P, the gradients $\nabla F_i(x^\star), i = 1, \ldots, k$ of all objectives are required and a QR-decomposition of a matrix at least of dimension $(n+k)\times(n+1)$ (in the unconstrained case) has to be calculated, see [25]. Moreover, for a suitable iteration scheme for finding further points on P and for a termination criterion gradients must be available, too. Of course, there are variants of the classical Recovering method which do not require derivatives, but these methods require many objective evaluations and consequently are not as efficient as gradient based methods, particularly when the parameter space is high-dimensional.

Comparison with the Weighted Sums Method In the following we derive some relationships between the solutions of the weighted sums method and RPP '. We assume that all objectives $F_i, i = 1, 2, \ldots, k$, are differentiable up to second order.

For given weights $w_i \geq 0$, $i = 1, 2, \ldots, k$, satisfying $\sum_{i=1}^{k} w_i = 1$ the weighted sums problem, which forms a scalarized auxiliary problem for MOP, is given by:

$$\min_{x\in S} \sum_{i=1}^{k} w_i F_i(x). \tag{WS}$$

To obtain a representation of the entire Pareto set, WS has to be solved repeatedly for different weighting vectors $w = (w_1, \ldots, w_k)^t$. One difficulty in this approach is, that it is not always clear a priori how to choose these weighting vectors in order to obtain a well-distributed representation of the Pareto set, see [6]. Moreover, as demonstrated by the following example, the weighted sums method may fail to compute the entire Pareto set.

EXAMPLE 4.4 Let $F : \mathbb{R}_+ \to \mathbb{R}^2$ be defined by $F(x) = (F_1(x), F_2(x))^t$, where $F_1(x) = 1 - x$ and $F_2(x) = \sqrt{x}$, and consider the following MOP:

$$\min_{x>0} F(x).$$

Observe that $F_1'(x) = -1 < 0$ and $F_2'(x) = (2\sqrt{x})^{-1} > 0$ for all $x > 0$ that is, every $x > 0$ is Pareto optimal. Assume we want to find the Pareto point $x^\star = \frac{1}{4}$. Keeping in mind that $w_2 = 1 - w_1$ we can write down the necessary condition for $x^\star$ to solve WS:

$$w_1 F_1'(x^\star) + (1 - w_1) F_2'(x^\star) = 1 - 2w_1 = 0.$$

Consequently, if there is a weighted sums expression with a minimum at $x^\star$, then it corresponds to $w_1 = w_2 = \frac{1}{2}$ and thus must be given by

$$F_w(x) := \frac{1}{2} F_1(x) + \frac{1}{2} F_2(x) = \frac{1 - x + \sqrt{x}}{2}.$$

Unfortunately, with

$$F_w'(x) = \frac{1}{4\sqrt{x}} - \frac{1}{2}$$

we have $F_w'(x) > 0$ for $0 < x < x^\star$ and $F_w'(x) < 0$ for $x > x^\star$ that is, $x^\star$ is a maximum of F_w and a minimization of F_w cannot lead to $x^\star$, unless in the unlikely case when $x^\star$ is chosen to be the initial guess and a gradient based method is used. On the other hand, using the target vector $t = (\frac{1}{4}, 0)^t$, minimization of the expression

$$F_t(x) := ||F(x) - t||_2 = \sqrt{x^2 - \frac{1}{2}x + \frac{9}{16}}$$

leads to $x^\star$, because with

$$F_t'(x) = \frac{x - \frac{1}{4}}{\sqrt{x^2 - \frac{1}{2}x + \frac{9}{16}}}$$

we have $F_t'(x) < 0$ for $0 < x < x^\star$ and $F_t'(x) > 0$ for $x > x^\star$. In other words, we have detected a solution $x^\star$ of the given MOP which can be found using RPP ', whereas WS fails to find this solution.

Now, for the unconstrained case, i.e., $S = \mathbb{R}^n$, we want to state this relationship in a more general context. In a first step we prove that any solution of WS fulfills the necessary optimality conditions for being a solution of RPP', if the target vector is chosen properly.

LEMMA 4.5 *For $S = \mathbb{R}^n$ let $x^\star$ be a solution of WS. Then there is a target vector $t = (t_1, t_2, \ldots, t_k)^t \in \mathbb{R}^k$, $t \leq_p F(x^\star)$, such that $x^\star$ fulfills the necessary optimality conditions for being a solution of RPP'.*

Proof: For $i = 1, 2, \ldots, k$ let $t_i = F_i(x^\star) - w_i$. Since $x^\star$ is a solution of WS, we have

$$\nabla \sum_{i=1}^{k} w_i F_i(x^\star) = \sum_{i=1}^{k} w_i \nabla F_i(x^\star) = 0$$

and consequently

$$\begin{aligned} \nabla ||F(x^\star) - t||_2 &= \nabla \left(\sqrt{\sum_{i=1}^{k} (F_i(x^\star) - t_i)^2} \right) \\ &= \frac{\sum_{i=1}^{k} 2(F_i(x^\star) - t_i) \nabla F_i(x^\star)}{2\sqrt{\sum_{i=1}^{k} (F_i(x^\star) - t_i)^2}} \\ &= \frac{1}{||F(x^\star) - t||_2} \sum_{i=1}^{k} w_i \nabla F_i(x^\star) = 0. \end{aligned}$$

□

Under certain conditions a solution of WS is indeed also a solution of RPP'. To prove this fact, which is formulated later on in Corollary 4.7, we have to do some preliminary work. To decide whether a point is a solution of an unconstrained minimization problem, in addition to the fact that the gradient vanishes, a statement on the definiteness of the corresponding Hessian has to be made. For this we denote by $\nabla^2 F(x)$ the Hessian of a function $F : \mathbb{R}^n \to \mathbb{R}$ at x and consider the following

THEOREM 4.6 *For $S = \mathbb{R}^n$ let $x^\star$ be a solution of WS, such that the Hessian*

$$\nabla^2 \sum_{i=1}^{k} w_i F_i(x^\star) = \sum_{i=1}^{k} w_i \nabla^2 F_i(x^\star)$$

is positive (semi-)definite. Then there is a target vector $t = (t_1, t_2, \ldots, t_k)^t \in \mathbb{R}^k$, $t \leq_p F(x^\star)$, such that

$$\nabla^2 ||F(x^\star) - t||_2$$

is positive (semi-)definite.

Proof: For $i = 1, 2, \ldots, k$ let $t_i = F_i(x^\star) - w_i$. Then $||F(x^\star) - t||_2 \neq 0$ and

$$\nabla^2 ||F(x^\star) - t||_2 = \nabla \left(\nabla ||F(x^\star) - t||_2\right)^t = \nabla \left(\frac{\sum_{i=1}^k (F_i(x^\star) - t_i) \nabla F_i(x^\star)}{||F(x^\star) - t||_2} \right)^t$$

$$= \frac{\sum_{i=1}^k \nabla \left(F_i(x^\star) - t_i\right) \left(\nabla F_i(x^\star)\right)^t}{||F(x^\star) - t||_2} + \frac{\sum_{i=1}^k \left(F_i(x^\star) - t_i\right) \nabla^2 F_i(x^\star)}{||F(x^\star) - t||_2} + \nabla \left(\frac{1}{||F(x^\star) - t||_2} \right) \left(\sum_{i=1}^k (F_i(x^\star) - t_i) \nabla F_i(x^\star) \right)^t$$

$$= \frac{1}{||F(x^\star) - t||_2} \left(\sum_{i=1}^k \nabla F_i(x^\star) \left(\nabla F_i(x^\star)\right)^t + \sum_{i=1}^k w_i \nabla^2 F_i(x^\star) \right) + \nabla \left(\frac{1}{||F(x^\star) - t||_2} \right) \left(\underbrace{\sum_{i=1}^k w_i \nabla F_i(x^\star)}_{=0} \right)^t .$$

Here, $\sum_{i=1}^k w_i \nabla F_i(x^\star) = 0$, because $x^\star$ is a solution of WS. The dyadic product

$$\nabla F_i(x^\star) \left(\nabla F_i(x^\star)\right)^t$$

is positive semidefinite for all $i = 1, 2, \ldots, k$, and since

$$\nabla^2 \sum_{i=1}^k w_i F_i(x^\star) = \sum_{i=1}^k w_i \nabla^2 F_i(x^\star)$$

is assumed to be positive (semi-)definite we can conclude that $\nabla^2 ||F(x^\star) - t||_2$ is also positive (semi-)definite. □

COROLLARY 4.7 *For $S = \mathbb{R}^n$ let $x^\star$ be an isolated solution of WS. Then there is a target vector $t = (t_1, t_2, \ldots, t_k)^t \in \mathbb{R}^k$, $t \leq_p F(x^\star)$, such that $x^\star$ is an isolated solution of RPP'.*

Proof: For $i = 1, 2, \ldots, k$ let $t_i = F_i(x^\star) - w_i$. Since $x^\star$ is an isolated solution of WS, $\nabla^2 \sum_{i=1}^k w_i F_i(x^\star)$ is positive definite and $\nabla \sum_{i=1}^k w_i F_i(x^\star) = 0$. From Lemma 4.5 it follows that $\nabla ||F(x^\star) - t||_2 = 0$. Moreover, Theorem 4.6 guarantees that

$\nabla^2 ||F(x^\star) - t||_2$ is positive definite. Thus, the necessary and sufficient conditions for $x^\star$ being an isolated solution of RPP ' are satisfied. □

Comparison with Normal Boundary Intersection As mentioned in Section 1, the Recovering-IS algorithm concentrates on the computation of a complete set of alternatives ([15]). Another method following this concept is the normal boundary intersection (NBI) method described in [7]. To apply this method, the individual minima $F_i^\star$ of all objectives F_i have to be computed in order to construct the *pay-off matrix* $\Phi \in \mathbb{R}^{k\times k}$ given by $\Phi_{ij} = F_j^\star - F_i^\star$. The set of points in $\mathbb{R}^k$ that are convex combinations of the columns of Φ, i.e., $\{\Phi\beta : \beta \in \mathbb{R}^k_+, \sum_{i=1}^k \beta_i = 1\}$ is called the *convex hull of individual minima* (CHIM). Let $\hat{n}$ denote the unit normal to the CHIM pointing towards the origin. The main idea of NBI is, that for every $\beta \in \{\beta \in \mathbb{R}^k_+, \sum_{i=1}^k \beta_i = 1\}$, the intersection of the line $\{\Phi\beta + t\hat{n} : t \in \mathbb{R}\}$ with the boundary $\partial F(S)$ of $F(S) = \{F(x), x \in S\}$ is accepted to be a local Pareto point. Although the NBI method works as satisfactorily as the Recovering-IS algorithm, we want to point out that there is an essential difference from the designers point of view. In many applications, the decision making process is based on the *trade-offs* associated with every Pareto point $x^\star$ that is, information on how much the value of one objective F_j increases when another objective F_i is improved. The corresponding trade-off can be expressed by $\frac{\alpha_j}{\alpha_i}$, where α_j and α_i are the multipliers occuring in the Kuhn-Tucker conditions associated with the given MOP: there are $\alpha_l \in \mathbb{R}, l = 1, \ldots, k$, such that

$$\sum_{l=1}^{k} \alpha_l \nabla F_l(x^\star) = 0, \tag{4.5}$$

$$\sum_{l=1}^{k} \alpha_l = 1, \tag{4.6}$$

$$\alpha_l \geq 0, \quad l = 1, \ldots, k. \tag{4.7}$$

Certainly, $\alpha = (\alpha_1, \ldots, \alpha_k)^t$ can be obtained by first calculating the gradients of all objectives (nk derivatives!) and then solving (4.5), (4.6), (4.7) for α. But the Recovering-IS algorithm has the advantage that for every calculated Pareto point $x^\star$ the vector $F(x^\star) - t >_p 0$ is available, where $t \in \mathbb{R}^k$ denotes the respective target used to obtain $x^\star$. Multiplying the Kuhn-Tucker equation

$$\nabla||F(x^\star) - t||_2 = \frac{1}{||F(x^\star) - t||_2} \sum_{l=1}^{k} (F_l(x^\star) - t_l)\nabla F_l(x^\star) = 0$$

associated with RPP' by

$$\frac{||F(x^\star) - t||_2}{\sum_{l=1}^{k} F_l(x^\star) - t_l} \in \mathbb{R}$$

makes clear that α, given by

$$\alpha_i = \frac{F_i(x^\star) - t_i}{\sum_{l=1}^{k} F_l(x^\star) - t_l}, \quad i = 1, \ldots, k,$$

already solves (4.5), (4.6), (4.7). In other words, if the Pareto set is computed by the Recovering-IS algorithm, α can be simply obtained by scaling the vector $F(x^\star) - t$ instead of calculating all derivatives, which can be relatively expensive, and then solving (4.5), (4.6), (4.7).

4.3 Convex Objectives

The Recovering-IS algorithm has some crucial advantages in the case where the MOP is unconstrained and where all objectives are strictly convex. In Theorem 4.11 we will show that in this case, given a solution $x^\star$ and $F^\star := F(x^\star)$, suitable targets for finding further solutions in the neighborhood of $x^\star$ can always be chosen on the tangent space $T_{F^\star}F(P)$. Moreover, we will show in Theorem 4.9 that the complete set of alternatives in image space as generated by the Recovering-IS algorithm implies that the corresponding Pareto set in parameter space is also complete. To this end, denote by $< \cdot, \cdot >$ the standard scalar product in $\mathbb{R}^n$ and consider the following

LEMMA 4.8 *Let $F \in C^1(\mathbb{R}^n, \mathbb{R})$ be strictly convex and let $x_1, x_2 \in \mathbb{R}^n, x_1 \neq x_2$, such that $F(x_1) = F(x_2)$. Then we have*

$$\langle x_2 - x_1, \nabla F(x_1) \rangle < 0.$$

Proof: Since F is strictly convex and $F(x_1) = F(x_2)$ we have

$$F(\lambda\, x_2 + (1 - \lambda)\, x_1) < \lambda F(x_2) + (1 - \lambda) F(x_1) = F(x_1)$$

for all $\lambda \in (0, 1)$. In particular, for $\bar{x} = \frac{1}{2}x_1 + \frac{1}{2}x_2$ we can write

$$F(\bar{x}) < F(x_1),$$

or

$$z := F(\bar{x}) - F(x_1) < 0.$$

Again using the strict convexity of F, we have

$$\begin{aligned} F(\lambda\, \bar{x} + (1 - \lambda)\, x_1) &< \lambda F(\bar{x}) + (1 - \lambda) F(x_1) \\ &= F(x_1) + \lambda (F(\bar{x}) - F(x_1)), \end{aligned}$$

or

$$\frac{F(\lambda\,\bar{x}+(1-\lambda)\,x_1)-F(x_1)}{\lambda} < z < 0$$

for all $\lambda \in (0,1)$. Finally, since $\langle \bar{x}-x_1, \nabla F(x_1)\rangle$ is the directional derivative of F at x_1 corresponding to the direction $\bar{x}-x_1$, we can conclude

$$\begin{aligned}\langle x_2-x_1, \nabla F(x_1)\rangle &= 2\,\langle \bar{x}-x_1, \nabla F(x_1)\rangle \\ &= 2\lim_{\lambda\searrow 0}\frac{F(\lambda\bar{x}+(1-\lambda)\,x_1)-F(x_1)}{\lambda} \;<\; 2z \;<0.\end{aligned}$$

□

Now we are in the position to prove the following

THEOREM 4.9 *Let $F_i \in C^1(\mathbb{R}^n,\mathbb{R})$ be strictly convex for $i=1,\ldots,k$.*
Define $F(x) := (F_1(x),\ldots,F_k(x))^t$ and consider the multi-objective optimization problem

$$\min_{x\in\mathbb{R}^n} F(x).$$

Let $P\subset\mathbb{R}^n$ be the corresponding Pareto set. Then the mapping

$$\Phi : P \to F(P), x \mapsto F(x)\ \forall x\in P$$

is one-to-one.

Proof: Since Φ is obviously surjective it remains to show that Φ is injective. Assume that there are $x_1,x_2\in P, x_1\neq x_2$, such that $F(x_1)=F(x_2)$. Then there are $\alpha_1,\ldots,\alpha_k\geq 0$ with $\sum_{i=1}^{k}\alpha_i = 1$ such that

$$\sum_{i=1}^{k}\alpha_i\nabla F_i(x_1) = 0. \tag{4.8}$$

Since the objectives F_i are strictly convex and $F_i(x_1)=F_i(x_2)$ for $i=1,\ldots,k$, it follows from Lemma 4.8, that

$$\langle x_2-x_1, \nabla F_i(x_1)\rangle < 0 \quad \text{for all } i=1,\ldots,k. \tag{4.9}$$

W.l.o.g. let $\alpha_1>0$. Then we have from (4.8)

$$\nabla F_1(x_1) = -\sum_{i=2}^{k}\frac{\alpha_i}{\alpha_1}\,\nabla F_i(x_1)$$

and consequently

$$\langle x_2 - x_1, \nabla F_1(x_1)\rangle = -\sum_{i=2}^{k} \frac{\alpha_i}{\alpha_1} \langle x_2 - x_1, \nabla F_i(x_1)\rangle \geq 0,$$

which is a contradiction to (4.9). Thus, $F(x_1) \neq F(x_2)$ that is, Φ is injective. This completes the proof. □

As mentioned at the beginning of this section, the choice of targets is very easy in the case of convex objectives. Before we show this fact, in the following lemma we first state that the vector $F^\star - t^\star$ given by the difference between a target $t^\star$ and $F^\star = F(x^\star)$, where $x^\star$ is the Pareto optimal point found by solving RPP', is orthogonal to the tangent space $T_{F^\star}F(P)$ with respect to $< \cdot, \cdot >$.

LEMMA 4.10 *Let $F_i \in C^1(\mathbb{R}^n, \mathbb{R})$ for $i = 1, \ldots, k$, and consider the multi-objective optimization problem*

$$\min_{x \in \mathbb{R}^n} F(x).$$

Denote by P the corresponding Pareto set and let $F^\star := F(x^\star)$, where $x^\star \in \mathbb{R}^n$ is the unique solution of RPP' associated with the target $t^\star <_p F^\star$. Then

$$F^\star - t^\star \in T_{F^\star}F(P)^\perp.$$

Proof: Let ∂F denote the boundary of $\{F(x) : x \in \mathbb{R}^n\}$. Since $F(P)$ locally forms a differentiable manifold (see [25]), there exists a differentiable curve $\alpha : [-1, 1] \to \partial F$ with $\alpha(0) = F^\star$, $\alpha'(0) \in T_{F^\star}F(P)$ and $\alpha(\lambda) \in F(P)$ for all $\lambda \in [0, 1]$. Then, since $x^\star$ is a solution of RPP', $\lambda = 0$ is a solution of

$$\min_{\lambda \in [-1,1]} ||\alpha(\lambda) - t^\star||^2$$

and therefore

$$\frac{d}{d\lambda}||\alpha(\lambda) - t^\star||^2 = \frac{d}{d\lambda} \langle \alpha(\lambda) - t^\star, \alpha(\lambda) - t^\star\rangle = 2 \langle \alpha(\lambda) - t^\star, \alpha'(\lambda)\rangle = 0$$

for $\lambda = 0$. With $\alpha(0) = F^\star$ we obtain

$$\langle F^\star - t^\star, \alpha'(0)\rangle = 0,$$

that is, $F^\star - t^\star \in T_{F^\star}F(P)^\perp$. □

THEOREM 4.11 *Denote by P the Pareto set of an unconstrained MOP and assume that the objectives $F_1, \ldots, F_k$ are strictly convex . Let $x^\star \in P$ be obtained by solving RPP' with target $t^\star <_p F^\star := F(x^\star)$. Then for every $\bar{F} \in F(P) \setminus \{F^\star\}$ there is a target $\tilde{t} \in T_{F^\star}F(P)$, $\tilde{t} <_p \bar{F}$ such that for the solution $\tilde{x}$ of RPP' with target $\tilde{t}$ we have $F(\tilde{x}) = \bar{F}$.*

Proof: First observe, that Theorem 4.9 guarantees that there is a unique solution $\bar{x}$ of MOP with $F(\bar{x}) = \bar{F}$. Then

$$F_i(\lambda\bar{x} + (1-\lambda)x^\star) < \lambda\bar{F}_i + (1-\lambda)F_i^\star$$

and therefore

$$F_i(\lambda\bar{x} + (1-\lambda)x^\star) - F_i^\star < \lambda(\bar{F}_i - F_i^\star) \tag{4.10}$$

for every $\lambda \in (0,1)$ and for every $i = 1, \ldots, k$. Since $F(P)$ forms a smooth manifold, see [25], there exists a curve $\alpha : [0,1] \to F(P)$ with $\alpha(0) = F^\star$ and

$$\alpha_i(\lambda) \leq F_i(\lambda\bar{x} + (1-\lambda)x^\star) \tag{4.11}$$

for all $i = 1, \ldots, k$ and $\lambda \in [0,1]$. Thus, from (4.10) and (4.11) it follows that

$$\alpha_i(\lambda) - F_i^\star < \lambda(\bar{F}_i - F_i^\star)$$

for $\lambda \in (0,1)$ and $i = 1, \ldots, k$. Multiplication by $(t_i^\star - F_i^\star) < 0$ yields

$$(t_i^\star - F_i^\star)(\alpha_i(\lambda) - F_i^\star) > \lambda(t_i^\star - F_i^\star)(\bar{F}_i - F_i^\star)$$

for $\lambda \in (0,1)$ and $i = 1, \ldots, k$. With $F^\star = \alpha(0)$, after summation of these k equations and division by $\lambda \neq 0$, we have

$$\left\langle t^\star - F^\star, \frac{\alpha_i(\lambda) - \alpha(0)}{\lambda} \right\rangle > \left\langle t^\star - F^\star, \bar{F} - F^\star \right\rangle$$

for $\lambda \in (0,1)$. The left hand side of this inequality vanishes for $\lambda \searrow 0$, because

$$\lim_{\lambda \searrow 0} \frac{\alpha_i(\lambda) - \alpha(0)}{\lambda} \in T_{F^\star}F(P)$$

and, as stated in Lemma 4.10,

$$t^\star - F^\star \in T_{F^\star}F(P)^\perp.$$

Thus we have

$$\left\langle t^\star - F^\star, \bar{F} - F^\star \right\rangle < 0.$$

Now, for any vector $v \in \mathbb{R}^k$ with $\langle v, t^\star - F^\star \rangle > 0$ define

$$\mu_v := -\frac{\left\langle t^\star - F^\star, \bar{F} - F^\star \right\rangle}{\langle v, t^\star - F^\star \rangle}.$$

Then $\mu_v > 0$ and

$$\left\langle (\bar{F} + \mu_v v) - F^\star, t^\star - F^\star \right\rangle = \left\langle \bar{F} - F^\star, t^\star - F^\star \right\rangle + \mu_v \langle v, t^\star - F^\star \rangle = 0,$$

that is, $\bar{F} + \mu_v v \in T_{F^\star} F(P)$.

Since $\bar{x}$ is a solution of MOP, there are $\bar{w}_i \geq 0, i = 1, \ldots, k$, such that

$$\sum_{i=1}^{k} \bar{w}_i \nabla F_i(\bar{x}) = 0$$

and since the objectives F_i are strictly convex we know that $\bar{x}$ solves WS. Moreover, the Hessians $\nabla^2 F_i$ are positive definite and therefore we can conclude from Lemma 4.5 and Theorem 4.6, that there is a target $\hat{t} <_p \bar{F}$ such that $\tilde{x} := \bar{x}$ is a solution of RPP'. Obviously, $\tilde{x}$ is also a solution of RPP' for all targets $t \in T := \{\bar{F} + \mu(\hat{t} - \bar{F}) : \mu > 0\}$. Moreover, since $t^\star <_p F^\star$ and $t <_p \bar{F}$ for all $t \in T$, we have $\left\langle t - \bar{F}, t^\star - F^\star \right\rangle > 0$ for all $t \in T$. Finally, for the choice $\tilde{t} := \bar{F} + \mu_{\hat{t}-\bar{F}}(\hat{t} - \bar{F}) \in T$ we have $\tilde{t} \in T_{F^\star} F(P)$. □

4.4 A Hierarchical Concept for Computing the Desired Part of the Pareto Set

In many applications, after considering a discrete representation of the entire Pareto set, the decision maker finds out that the best Pareto points e.g. for adjusting the technical system must be somewhere in a particular part of the Pareto set. To make a more concrete decision, a more detailed representation of that part is desired that is, more Pareto points must be available such that their density is significantly higher within that particular part. Of course, one could use boxes of suitable depth in the Recovering-IS algorithm, such that the computed representation is sufficiently dense. But since in the end only an a priori unknown part of the Pareto set is of special interest, this proceeding is not very efficient. Here, too many unneeded points are computed. On the other hand, an initial representation of suitable density is required in order to analyze which part of the Pareto set one wants to concentrate on. Here, again the question arises how to choose the depth of the boxes, such that the required density is obtained, whereas the computational effort has to be adequate.

In the following we propose a hierarchical concept based on a multi-level subdivision scheme, which helps to successively extend the computed set of Pareto points within those boxes which are selected by the decision maker for further consideration. The extensions are obtained by first subdividing the selected boxes and then applying the Recovering-IS algorithm exclusively to those "subboxes" containing Pareto points found so far. Consequently, the points of the computed extensions are well-distributed within the selected boxes and the density among these points increases as the iteration proceeds, whereas the computation of too many undesired Pareto points is avoided.

In Figure 13 the solution for Example 4.1 as obtained by our Recovering-IS algorithm in combination with the interactive hierarchical concept described above is depicted. After two steps of subdivision and execution of our algorithm, the decision maker decided to concentrate on a certain region (marked box) of the image space. Thus, further subdivisions and executions of our algorithm were restricted to this region. These computations were continued until the decision maker was satisfied with the density of the generated Pareto points within this region.

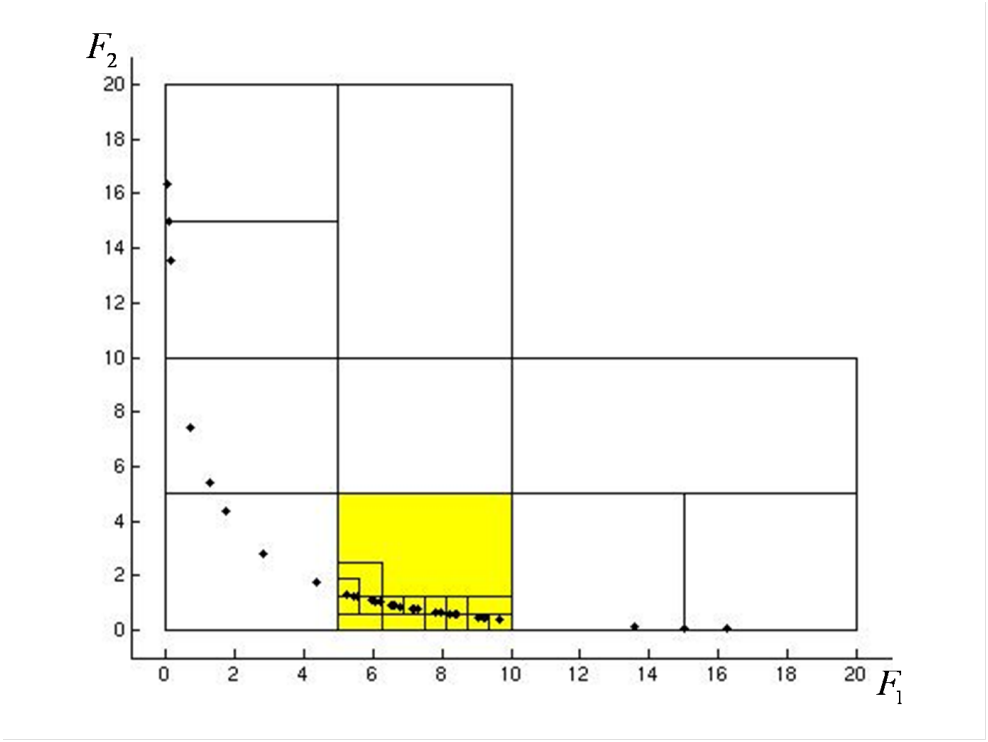

Figure 13: The solution (in image space) of Example 4.1 computed by the Recovering-IS algorithm in combination with the described hierarchical concept.

5 Basic Methods for Solving Bi-Level Optimization Problems

Many different approaches for solving (classical) bi-level optimization problems have been proposed in the past, as there are for example descent algorithms, bundle algorithms, penalty methods, trust region methods, smoothing methods and branch-and-bound methods. Many of these approaches are based on the conversion of the bi-level problem to an ordinary (or classical) optimization problem (a one-level problem). One possibility is to replace the lower level objective f by an additional non-differentiable equation $f(x,y) = \varphi(y)$, where $\varphi(y) = \min_x\{f(x,y) : g(x,y) \leq 0, h(x,y) = 0\}$. Other approaches use the implicit function theorem to derive a local description of the function $x(y) : \mathbb{R}^m \to \mathbb{R}^n$, which is then inserted into the higher level problem. As mentioned in Section 2.2, another concept is to replace the lower level problem by its Kuhn-Tucker conditions. In general, the resulting one-level problem BLP', which is a *mathematical program with equilibrium constraints* or MPEC, see [33], is not equivalent to the original problem, but the desired equivalence is ensured in the particular case where the lower level problem is a convex one. Since in this work we will basically concentrate on optimistic formulations of BLMOP with convex lower level problems, we will follow this concept based on lower level Kuhn-Tucker conditions. Here, one essential difficulty to overcome in the presence of lower level inequality constraints $g_i(x,y) \leq 0$ is given by the fact that in this case the usual constraint qualifications like linear independence or Mangasarian-Fromowitz constraint qualifications for the higher level problem are violated at every feasible point. A proof on this can be found for example in [12]. We will give an alternative proof concerning linear independence constraint qualifications in the more general context of BLMOP in Section 6.1. In order to overcome the mentioned difficulties concerning constraint qualifications when solving the respective auxiliary problems corresponding to BLMOP with lower level inequality constraints, we will solve reformulations constructed by the use of *merit functions* ([47, 19]) as described in the next section. We will also give a review on *smoothing methods* ([28]), which help to improve the numerical solvability of the reformulated problems.

5.1 Merit Functions and Smoothing Methods

As we will see in the proof of Theorem 6.3, the mentioned violation of constraint qualifications originates from the inequalities $g_i(x,y) \leq 0, \tau_i \geq 0, i = 1,\ldots,q$ and the complementarity equations $\tau_i g_i(x,y) = 0, i = 1,\ldots,q$ in BLP'. A way out is

to replace these inequalities and equations, which act as constraints in BLP', by q equivalent constraints of the form $\Phi(\tau_i, -g_i(x,y)) = 0$, where $\Phi : \mathbb{R}^2 \to \mathbb{R}$ is a so-called *merit function*. With this replacement and the notation $\bar{\nabla} := \nabla_{(x)}$, BLP' is reformulated as

$$\begin{aligned} \min_{x,y,\tau,\zeta} \; & F(x,y) && \text{(BLP")} \\ s.t. \quad & G(x,y) \leq_p 0, \\ & H(x,y) = 0, \\ & \bar{\nabla} f(x,y) + \sum_{i=1}^{p} \zeta_i \bar{\nabla} h_i(x,y) + \sum_{i=1}^{q} \tau_i \bar{\nabla} g_i(x,y) = 0, \\ & h(x,y) = 0, \\ & \Phi(\tau_i, -g_i(x,y)) = 0 \quad \text{for} \quad i = 1, \ldots, q. \end{aligned}$$

Several merit functions have been proposed in the past, see [47, 19]. Unfortunately, it turned out that these merit functions are in general non-differentiable. Surely, non-smooth methods can be used for the solution of merit function based reformulations, but it is worth looking for differentiability conserving alternatives in order to benefit from the advantages of gradient based optimization methods, that is in particular, the higher rate of convergence.

The idea of smoothing methods is to use a family of *smoothing functions* $\Phi_\varepsilon : \mathbb{R}^2 \to \mathbb{R}$, which are smooth and use a *smoothing parameter* ε in order to approximate the merit function Φ in the sense that

$$\lim_{\varepsilon \searrow 0} \Phi_\varepsilon(a,b) = \Phi(a,b) \quad \forall a, b \in \mathbb{R}.$$

In the following we will sometimes use synonymously the notation $\Phi(a,b,\varepsilon) := \Phi_\varepsilon(a,b)$. Examples of smoothing functions can for instance be found in [28]. The method proposed in this work is particularly based on one of these smoothing function, the *perturbed Fischer-Burmeister function*.

DEFINITION 5.1 The *Fischer-Burmeister function* $\Phi : \mathbb{R}^2 \to \mathbb{R}$ is defined by

$$\Phi(a,b) = a + b - \sqrt{a^2 + b^2}.$$

For $\varepsilon \geq 0$, the *perturbed Fischer-Burmeister function* $\Phi : \mathbb{R}^3 \to \mathbb{R}$ is defined by

$$\Phi(a,b,\varepsilon) = a + b - \sqrt{a^2 + b^2 + \varepsilon}.$$

Observe that the perturbed Fischer-Burmeister function Φ has the following properties:

(i) for $\varepsilon > 0$, Φ is continuously differentiable with respect to (a, b) on $\mathbb{R}^2$,

(ii) for $\varepsilon = 0$, Φ is continuously differentiable with respect to (a, b) on $\mathbb{R}^2 \backslash (0, 0)^t$,

(iii) $\Phi(a, b, 0) = 0 \Leftrightarrow a \geq 0, b \geq 0$, and $ab = 0$.

Consequently, BLP" is equivalent to BLP', but for every $i = 1, \ldots, q$, BLP" includes just one single equality constraint $\Phi(\tau_i, -g_i(x, y)) = 0$ instead of the constraints $g_i(x, y) \leq 0, \tau_i \geq 0$ and $\tau_i\, g_i(x, y) = 0$, that is, the sources for the mentioned violation of the constraint qualifications are circumvented. Another difference is, that BLP" is not everywhere, but anyway almost everywhere differentiable, such that one might generically expect to find solutions of BLP" using derivative based algorithms. But as mentioned in [28], well-posedness can be improved in the sense that feasibility and constraint qualifications, hence stability, are often more likely to be satisfied, if $\Phi(\tau_i, -g_i(x, y))$ is replaced by $\Phi(\tau_i, -g_i(x, y), \varepsilon)$ for some small $\varepsilon > 0$. Moreover, solvability of quadratic approximation problems is improved in this case, which opens the way to use sequential quadratic programming methods (SQP) from classical nonlinear optimization.

REMARK 5.2 (i) In general, it is not clear whether a solution of a perturbation of BLP" according to $\varepsilon > 0$ approximates the solution of the original problem. To see that under certain assumptions this desired approximation property is indeed guaranteed, let

$$\mathbf{\Phi}(\tau, x, y, \varepsilon) := (\Phi_\varepsilon(\tau_1, -g_1(x, y)), \ldots, \Phi_\varepsilon(\tau_q, -g_q(x, y)))^t$$

and

$$\tilde{H}(x, y, \tau, \zeta, \varepsilon) := \begin{pmatrix} H(x, y) \\ \bar{\nabla} f(x, y) + \sum_{i=1}^{p} \zeta_i \bar{\nabla} h_i(x, y) + \sum_{i=1}^{q} \tau_i \bar{\nabla} g_i(x, y) \\ h(x, y) \\ \mathbf{\Phi}(\tau, x, y, \varepsilon) \end{pmatrix}.$$

Then, the perturbed variant of BLP" can be written as

$$\begin{aligned} \min_{x,y,\tau,\zeta} \; & F(x, y) && \text{(BLP}(\varepsilon)\text{)} \\ s.t. \quad & G(x, y) \leq_p 0, \\ & \tilde{H}(x, y, \tau, \zeta, \varepsilon) = 0, \end{aligned}$$

which makes up a parametric optimization problem as considered in [17]. Then, denoting by $X(\varepsilon) := (x(\varepsilon), y(\varepsilon), \tau(\varepsilon), \zeta(\varepsilon))$ the solution of BLP(ε) for every ε, in [17] it is shown that – under suitable assumptions – the mapping $\varepsilon \rightarrow X(\varepsilon)$ is continuous in a neighborhood of $\varepsilon = 0$. Consequently, we can expect that for a sufficiently small perturbation parameter $\varepsilon > 0$, the solution $X(\varepsilon)$ approximates $X(0)$, that is, the solution of BLP".

(ii) Moreover, if $X(\bar{\varepsilon})$ has been computed for some $\bar{\varepsilon} > 0$, the results of [17] constitute a linear approximation for the mapping $\varepsilon \rightarrow X(\varepsilon)$ at $\bar{\varepsilon}$, which can be used to derive different a posteriori error estimations, e.g., an estimation of the form $||(x(\bar{\varepsilon}), y(\bar{\varepsilon})) - (x(0), y(0))||$.

There remains the question – which can in general hardly be answered a priori – how to choose $\varepsilon > 0$, such that the corresponding reformulation is sufficiently well-posed and the optimization results in a point which is somehow "close enough" to the solution of the original reformulation, that is, for the case $\varepsilon = 0$. Consequently, a common technique is to start with a relatively large perturbation parameter $\varepsilon_0 > 0$, which has to converge towards 0 during the optimization process. Such techniques are well-known in the field of *Mathematical Programs with Complementarity Constraints* and can be divided into *implicit* and *explicit* methods, see for instance [28].

In implicit smoothing methods, the smoothing parameter ε is included as one of the optimization variables of the problem reformulation and is updated at each iteration just like the other variables. To this end, an additional equality constraint, e.g. of the form $e^{\varepsilon} - 1 = 0$, which is satisfied if and only if $\varepsilon = 0$, has to be included in the reformulation (here, e denotes Euler's constant).

In explicit smoothing methods the smoothing parameter is updated separately, that is, after obtaining a solution for a fixed perturbation parameter $\varepsilon > 0$, ε has to be decreased successively and in every cycle a subproblem associated with a new perturbation parameter has to be solved, starting from the solution of the respective previous cycle. Here, a strategy for decreasing the perturbation parameter is needed. Furthermore, a termination criterion, i.e., an a posteriori estimation is required in order to decide at the end of every cycle whether the solution is satisfying in the sense that it is "close enough" to the solution of the original reformulation, or not.

6 Bi-Level Multi-Objective Optimization Problems (BLMOP)

In this section, we present the main topic of this thesis. First we derive some theoretical results including optimality conditions for a certain class of BLMOP, that is in particular, BLMOP without lower level inequality constraints. Then we present some algorithms for the solution of these problems, which are based on the mentioned optimality conditions and we prove convergence for such algorithms. We also consider a special subclass of BLMOP, the *Pareto set constrained multi-objective optimization problems* (PSCMOP), which are characterized by the fact that the lower level problem is not a parametrized one. For the sake of completeness, we give some basic ideas for the solution of BLMOP with lower level inequality constraints in Section 6.5 and for non-convex and non-smooth BLMOPs in Section 6.6. Moreover, in Section 6.4 we propose methods which are tailored to solve non-convex and non-smooth PSCMOPs.

6.1 An Optimality Condition for BLMOPs without Lower Level Inequality Constraints

For the case where lower level inequality constraints are absent, that is $q = 0$, we want to state a necessary condition for a solution of BLMOP on the basis of the Kuhn-Tucker conditions (2.2) for MOP. We restrict ourselves to this particular case, because, as we will see later on, lower level inequality constraints lead necessarily to the violation of constraint qualifications which have to be satisfied for the desired optimality condition for BLMOP stated in Theorem 6.1. Later on in Section 6.5 we will apply the results of this section to reformulations based on the Fischer-Burmeister function in order to handle the general BLMOP with both equality and inequality constraints for both the higher and lower level.

We rewrite the optimistic formulation BLMOP-O for the case without lower level inequality constraints as

$$\begin{aligned} \min_{x\in\mathbb{R}^n, y\in\mathbb{R}^m} \; & F(x,y), && && \text{(BLMOP-O')}\\ \text{s.t.} \quad & G(x,y) && \leq_p \; 0,\\ & H(x,y) && = \; 0,\\ \text{and } x \text{ solves:} \quad & \min_{x\in\mathbb{R}^n} f(x,y),\\ \text{s.t.} \quad & h(x,y) && = \; 0. \end{aligned}$$

In the following let $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^l \times \mathbb{R}^p \to \mathbb{R}$,

$$\mathcal{L}(x, y, \alpha, \zeta) := \sum_{i=1}^{l} \alpha_i \, f_i(x, y) + \sum_{i=1}^{p} \zeta_i \, h_i(x, y)$$

be the *lower level Lagrangian*. Denote by $\mathcal{L}_{x_i}$ the derivative of $\mathcal{L}$ with respect to x_i, $i = 1, \ldots, n$. Let $I_\alpha := \{i : \alpha_i = 0\} \subset \{1, \ldots, l\}$ and for fixed $x \in \mathbb{R}^n, y \in \mathbb{R}^m$ let $I_G(x, y) := \{i : G_i(x, y) = 0\} \subset \{1, \ldots, s\}$. The gradient $\nabla f(x)$ of a function $f : D \subset \mathbb{R}^n \to \mathbb{R}$ is understood to be a row vector and the Hessian of f is denoted by $\nabla^2 f(x)$. Accordingly, the $i-$th row of the Jacobian $\nabla g(x)$ of a vector valued function $g : D \subset \mathbb{R}^n \to \mathbb{R}^k$ is given by the gradient $\nabla g_i(x)$ of the $i-$th component of g. Furthermore, denote by e_i the $i-$th vector of the standard basis in $\mathbb{R}^{n+m+l+p}$ and let $\bar{\nabla} := \nabla_{(x)}$, $\tilde{\nabla} := \nabla_{(y)}$, $\hat{\nabla} := \nabla_{(x,y)}$, and $\nabla := \nabla_{(x,y,\alpha,\zeta)}$.

THEOREM 6.1 *Let (x^*, y^*) be a solution of BLMOP. If the gradients*

$$\bar{\nabla} h_i(x^*, y^*), \qquad i = 1, \ldots, p,$$

are linearly independent, then there exist $\alpha_1, \ldots, \alpha_l \geq 0$, $\zeta_1, \ldots, \zeta_p \in \mathbb{R}$ such that

$$(i) \qquad \bar{\nabla} \mathcal{L}(x^*, y^*, \alpha, \zeta) = 0, \tag{6.1}$$

$$(ii) \qquad \sum_{i=1}^{l} \alpha_i = 1. \tag{6.2}$$

Assume that in addition all (x, y, α, ζ) satisfying (6.1), (6.2), $\alpha_i \geq 0$, and $h(x, y) = 0$ correspond to Pareto points for the lower level problem. Furthermore assume that the first n rows of the Hessian

$$\hat{\nabla}^2 \mathcal{L}(x^*, y^*, \alpha, \zeta)$$

of the lower level Lagrangian with respect to (x, y) and the gradients

$$\begin{aligned}
\hat{\nabla} h_i(x^*, y^*) &= \left(\bar{\nabla} h_i(x^*, y^*), \tilde{\nabla} h_i(x^*, y^*) \right), \qquad i = 1, \ldots, p, \\
\hat{\nabla} H_i(x^*, y^*) &= \left(\bar{\nabla} H_i(x^*, y^*), \tilde{\nabla} H_i(x^*, y^*) \right), \qquad i = 1, \ldots, r, \\
\hat{\nabla} G_i(x^*, y^*) &= \left(\bar{\nabla} G_i(x^*, y^*), \tilde{\nabla} G_i(x^*, y^*) \right), \qquad i \in I_G(x^*, y^*)
\end{aligned}$$

of the active higher and lower level constraints with respect to (x, y), are linearly independent. Then there exist $\beta_1, \ldots, \beta_k$, $\mu_1, \ldots, \mu_l$, $\delta_1, \ldots, \delta_s \geq 0$, $\omega_1, \ldots, \omega_p$,

$\nu, \lambda_1, \ldots, \lambda_n,\ \rho_1 \ldots, \rho_r \in \mathbb{R}$ *such that*

$$
\begin{aligned}
(iii) \qquad & \sum_{i=1}^{k} \beta_i \nabla F_i(x^*, y^*) + \sum_{i=1}^{n} \lambda_i \nabla \mathcal{L}_{x_i}(x^*, y^*, \alpha, \zeta) \\
& + \sum_{i=1}^{r} \rho_i \nabla H_i(x^*, y^*) + \sum_{i=1}^{s} \delta_i \nabla G_i(x^*, y^*) \\
& + \sum_{i=1}^{p} \omega_i \nabla h_i(x^*, y^*) + \sum_{i=1}^{l} (\nu - \mu_i)\, e_{n+m+i} = 0,
\end{aligned} \tag{6.3}
$$

$$
(iv) \qquad \sum_{i=1}^{k} \beta_i = 1, \tag{6.4}
$$

$$
(v) \qquad \mu_i\, \alpha_i = 0 \qquad \text{for } i = 1, \ldots, l, \tag{6.5}
$$

$$
(vi) \qquad \delta_i\, G_i(x^*, y^*) = 0 \qquad \text{for } i = 1, \ldots, s. \tag{6.6}
$$

Proof: Let (x^*, y^*) be a solution to BLMOP. Then x^* is a solution to the lower level problem parametrized by y^*. Since $\bar{\nabla} h_i(x^*, y^*), i = 1, \ldots, p$ are linearly independent, the Kuhn-Tucker condition 2.2 holds: there are $\alpha_1, \ldots, \alpha_l, \zeta_1, \ldots, \zeta_p \in \mathbb{R}$ such that

$$
\bar{\nabla} \mathcal{L}(x^*, y^*, \alpha, \zeta) = 0, \tag{6.7}
$$

$$
\sum_{i=1}^{l} \alpha_i = 1 \tag{6.8}
$$

and

$$
\alpha_i \geq 0,\ i = 1, \ldots, l. \tag{6.9}
$$

Denote by S_L the set of those solutions of (6.7),(6.8),(6.9) for which $h(x, y) = 0$ holds. Observe that – according to the assumptions of the theorem – for any $(x, y, \alpha, \zeta) \in S_L$, x is Pareto optimal for the lower level problem corresponding to y. Thus, we can define the following auxiliary problem, which is equivalent to the given BLMOP:

$$
\begin{aligned}
& \min_{x,y,\alpha,\zeta} F(x, y), \\
& \text{s.t} \quad G(x, y) \leq_p 0, \\
& \qquad H(x, y) = 0, \\
& \qquad (x, y, \alpha, \zeta) \in S_L.
\end{aligned} \tag{6.10}
$$

In order to apply the Kuhn-Tucker conditions (2.2) to this auxiliary problem, we have to ensure that the gradients with respect to (x, y, α, ζ) of the active constraints,

that is,

$$\nabla h_i(x^*, y^*), \qquad i = 1, \dots, p, \tag{6.11}$$

$$\nabla H_i(x^*, y^*), \qquad i = 1, \dots, r, \tag{6.12}$$

$$\nabla G_i(x^*, y^*), \qquad i \in I_G(x^*, y^*), \tag{6.13}$$

$$\nabla \mathcal{L}_{x_i}(x^*, y^*, \alpha, \zeta), \qquad i = 1, \dots, n, \tag{6.14}$$

$$\nabla \sum_{i=1}^{l} \alpha_i = \sum_{i=1}^{l} e_{n+m+i}, \tag{6.15}$$

$$\nabla(-\alpha_i) = -e_{n+m+i}, \qquad i \in I_\alpha, \tag{6.16}$$

are linearly independent or equivalently, that the matrix $M \in \mathbb{R}^{p+r+s+n+|I_\alpha|+1 \times n+m+l+p}$ defined by

$$M := \begin{pmatrix} \bar{\nabla} h(x^*, y^*) & \tilde{\nabla} h(x^*, y^*) & \mathbf{0} & \mathbf{0} \\ \bar{\nabla} H(x^*, y^*) & \tilde{\nabla} H(x^*, y^*) & \mathbf{0} & \mathbf{0} \\ \bar{\nabla} G(x^*, y^*) & \tilde{\nabla} G(x^*, y^*) & \mathbf{0} & \mathbf{0} \\ \bar{\nabla}^2 \mathcal{L}(x^*, y^*, \alpha, \zeta) & \tilde{\nabla} \left(\bar{\nabla} \mathcal{L}(x^*, y^*, \alpha, \zeta)\right)^t & \left(\bar{\nabla} f(x^*, y^*)\right)^t & \left(\bar{\nabla} h(x^*, y^*)\right)^t \\ \mathbf{0} & \mathbf{0} & -\tilde{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 1 \dots 1 & \mathbf{0} \end{pmatrix}$$

has maximal rank, where the rows of $\tilde{I}$ are given by the transposed of distinct standard basis vectors of $\mathbb{R}^l$ according to (6.16). Since

$$\begin{pmatrix} \bar{\nabla}^2 \mathcal{L}(x^*, y^*, \alpha, \zeta) & \tilde{\nabla} \left(\bar{\nabla} \mathcal{L}(x^*, y^*, \alpha, \zeta)\right)^t \end{pmatrix}$$

complies with the first n (linearly independent) rows of the Hessian $\hat{\nabla}^2 \mathcal{L}(x^*, y^*, \alpha, \zeta)$, the rows of the upper left block

$$\begin{pmatrix} \bar{\nabla} h(x^*, y^*) & \tilde{\nabla} h(x^*, y^*) \\ \bar{\nabla} H(x^*, y^*) & \tilde{\nabla} H(x^*, y^*) \\ \bar{\nabla} G(x^*, y^*) & \tilde{\nabla} G(x^*, y^*) \\ \bar{\nabla}^2 \mathcal{L}(x^*, y^*, \alpha, \zeta) & \tilde{\nabla} \left(\bar{\nabla} \mathcal{L}(x^*, y^*, \alpha, \zeta)\right)^t \end{pmatrix}$$

of M are linearly independent and since M has a certain triangular block structure, it remains to show that the submatrix

$$\begin{pmatrix} -\tilde{I} \\ 1 \dots 1 \end{pmatrix} \quad \in \mathbb{R}^{|I_\alpha|+1 \times l} \tag{6.17}$$

has maximal rank. Indeed, with (6.8) and (6.9) we have $|I_\alpha| < l$ that is, (6.17) has at most l rows. Obviously, these rows are linearly independent and thus both (6.17) and also M have maximal rank. Consequently, (6.11), (6.12), (6.13), (6.14),

(6.15) and (6.16) are linearly independent and we can write down the Kuhn-Tucker conditions (2.2) for the auxiliary problem (6.10): there are $\beta_1, \ldots, \beta_k$, $\mu_1, \ldots, \mu_l$, $\delta_1, \ldots, \delta_s \geq 0$, $\omega_1, \ldots, \omega_p$, ν, $\lambda_1, \ldots, \lambda_n$, $\rho_1, \ldots, \rho_r \in \mathbb{R}$ such that

$$\begin{aligned} &\sum_{i=1}^{k} \beta_i \nabla F_i(x^*, y^*) + \sum_{i=1}^{n} \lambda_i \nabla \mathcal{L}_{x_i}(x^*, y^*, \alpha, \zeta) \\ &+ \nu \sum_{i=1}^{l} \nabla \alpha_i - \sum_{i=1}^{l} \mu_i \nabla \alpha_i + \sum_{i=1}^{p} \omega_i \nabla h_i(x^*, y^*) \\ &+ \sum_{i=1}^{r} \rho_i \nabla H_i(x^*, y^*) + \sum_{i=1}^{s} \delta_i \nabla G_i(x^*, y^*) = 0, \end{aligned} \tag{6.18}$$

$$\sum_{i=1}^{k} \beta_i = 1, \tag{6.19}$$

$$\mu_i \, \alpha_i = 0 \qquad \text{for} \quad i = 1, \ldots, l, \tag{6.20}$$

$$\delta_i \, G_i(x^*, y^*) = 0 \qquad \text{for} \quad i = 1, \ldots, s. \tag{6.21}$$

Observing that $\nabla \alpha_i = e_{n+m+i}$ for $i = 1, \ldots, l$, we obtain from (6.18)

$$\begin{aligned} &\sum_{i=1}^{k} \beta_i \nabla F_i(x^*, y^*) + \sum_{i=1}^{n} \lambda_i \nabla \mathcal{L}_{x_i}(x^*, y^*, \alpha, \zeta) + \sum_{i=1}^{l} (\nu - \mu_i) \, e_{n+m+i} \\ &+ \sum_{i=1}^{p} \omega_i \nabla h_i(x^*, y^*) + \sum_{i=1}^{r} \rho_i \nabla H_i(x^*, y^*) + \sum_{i=1}^{s} \delta_i \nabla G_i(x^*, y^*) = 0. \end{aligned} \tag{6.22}$$

□

REMARK 6.2 (i) In the case without explicitly given constraints, that is $r = s = p = 0$, the maximal rank condition for the matrix M is already satisfied, if the Hessian $\bar{\nabla}^2 \mathcal{L}(x^*, y^*, \alpha, \zeta)$ (with respect to x) is regular.

(ii) After introducing slack variables to express the involved inequalities as equations, Theorem 6.1 along with the given constraints corresponds to a system of nonlinear equations that has $(k-1)$ more variables, than it has equations, if $k \geq 2$. If $k = 1$, that is, if we have one single higher level objective F_1, then

(6.3) simplifies to

$$\nabla F_1(x^*, y^*) + \sum_{i=1}^{n} \lambda_i \nabla \mathcal{L}_{x_i}(x^*, y^*, \alpha, \zeta)$$
$$+ \sum_{i=1}^{p} \omega_i \nabla h_i(x^*, y^*) + \sum_{i=1}^{r} \rho_i \nabla H_i(x^*, y^*) + \sum_{i=1}^{s} \delta_i \nabla G_i(x^*, y^*)$$
$$+ \sum_{i=1}^{l} (\nu - \mu_i)\, e_{n+m+i} = 0$$

and (6.4) simplifies to $\beta_1 = 1$. In this case the system turns out to be quadratic in the sense that the number of variables equals the number of equations.

(iii) If (x^*, y^*) is an inner point of the solution set with respect to G and α, that is, $I_G(x^*, y^*) = \emptyset$ and $I_\alpha = \emptyset$, then $\mu_i = 0$ for $i = 1, \ldots, l$, $\delta_i = 0$ for $i = 1, \ldots, s$, and (6.3) simplifies to

$$\sum_{i=1}^{k} \beta_i \nabla F_i(x^*, y^*) + \sum_{i=1}^{n} \lambda_i \nabla \mathcal{L}_{x_i}(x^*, y^*, \alpha, \zeta)$$
$$+ \sum_{i=1}^{p} \omega_i \nabla h_i(x^*, y^*) + \sum_{i=1}^{r} \rho_i \nabla H_i(x^*, y^*) + \nu \sum_{i=1}^{l} e_{n+m+i} = 0.$$

In the next sections we will propose algorithms based on Theorem 6.1 which are tailored to solve BLMOPs without lower level inequality constraints. In Section 6.5 we will consider the more general BLMOPs, where lower level inequality constraints are involved. Here, we will approximate the original problems by related BLMOPs where the lower level inequality constraints are replaced by certain equality constraints which are constructed by using the perturbed Fischer-Burmeister function. In this way, Theorem 6.1 will also be the basis for our algorithms for the solution of BLMOPs with lower level inequality constraints.

There remains the question why Theorem 6.1 is stated only for BLMOPs without lower level inequality constraints. For this, we state the following

THEOREM 6.3 *Let the Kuhn-Tucker conditions of an inequality constrained MOP act as constraints for a higher level optimization problem. Then the linear independence constraint qualifications for the higher level problem are violated at every feasible point.*

Proof: W.l.o.g., we assume that there are no higher level constraints and no lower level equality constraints. In this case, with a lower level problem involving

objectives $f : \mathbb{R}^{n+m} \to \mathbb{R}^l$ and inequality constraints $g : \mathbb{R}^{n+m} \to \mathbb{R}^q$, the lower level Lagrangian is given by

$$\mathcal{L}(x, y, \alpha, \tau) := \sum_{i=1}^{l} \alpha_i \, f_i(x, y) + \sum_{i=1}^{q} \tau_i \, g_i(x, y).$$

Let $\nabla := \nabla_{(x,y,\alpha,\tau)}$ and denote by $\mathcal{L}_{x_i}$ the derivative of $\mathcal{L}$ with respect to x_i, $i = 1, \ldots, n$. Then, the Kuhn-Tucker reformulation associated with the lower level is given by

$$\begin{aligned}
\bar{\nabla}\mathcal{L}(x, y, \alpha, \tau) &= 0, \\
\sum_{i=1}^{l} \alpha_i &= 1, \\
\alpha_i &\geq 0, \quad i = 1, \ldots, l, \\
g_i(x, y)\, \tau_i &= 0, \quad i = 1, \ldots, q, \\
\tau_i &\geq 0, \quad i = 1, \ldots, q.
\end{aligned}$$

Consequently, the linear independence condition for applying the Kuhn-Tucker conditions to the higher level problem means that the gradients of the active constraints (which are defined by the lower level constraints and the lower level Kuhn-Tucker conditions) with respect to (x, y, α, τ), that is

$$\begin{aligned}
&\nabla \mathcal{L}_{x_i}(x, y, \alpha, \tau), \quad i = 1, \ldots, n, \\
&\nabla (g_i(x, y)\, \tau_i), \quad i = 1, \ldots, q, \\
&\nabla \tau_i = e_{n+m+l+i}, \qquad i \in I_\tau = \{i : \tau_i = 0\}, \\
&\nabla g_i(x, y), \qquad i \in I_g = \{i : g_i(x, y) = 0\}, \\
&\nabla \sum_{i=1}^{l} \alpha_i, \\
&\nabla \alpha_i = e_{n+m+i}, \qquad i \in I_\alpha = \{i : \alpha_i = 0\}
\end{aligned}$$

have to be linearly independent. But for $i \in I_g$ we have

$$\begin{aligned}
\nabla (g_i(x, y)\, \tau_i) &= \begin{pmatrix} \tau_i \bar{\nabla} g_i(x, y) & \tau_i \tilde{\nabla} g_i(x, y) & \mathbf{0} & \mathbf{0} \end{pmatrix}, \\
\nabla g_i(x, y) &= \begin{pmatrix} \bar{\nabla} g_i(x, y) & \tilde{\nabla} g_i(x, y) & \mathbf{0} & \mathbf{0} \end{pmatrix},
\end{aligned}$$

and for $i \in I_\tau$, we have

$$\begin{aligned}
\nabla (g_i(x, y)\, \tau_i) &= \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & e_i \, g_i(x, y) \end{pmatrix}, \\
\nabla \tau_i &= \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & e_i \end{pmatrix}.
\end{aligned}$$

Consequently, since the condition $g_i(x,y)\,\tau_i = 0$ implies that $i \in I_g \cup I_\tau$ for all $i = 1,\ldots,q$, any lower level inequality constraint violates the desired linear independence conditions. □

6.2 Methods for Solving BLMOPs without Lower Level Inequality Constraints

In this section we propose new algorithms for the computation of the Pareto set of a given BLMOP without lower level inequality constraints. We first consider how to solve the BLMOP in the particular case $k = 1$, where the Pareto set turns out to be a singleton and then we present the *BL-Subdivision*, the *BL-Recovering-PS*, and the *BL-Recovering-IS* algorithms for the computation of both tight coverings and discrete representations of the typically extensive Pareto sets in the case $k > 1$.

Here, we concentrate on derivative based methods, which are based on the solution of a system of equations derived from Theorem 6.1. Therefore, we have to assume that for any fixed $y \in \mathbb{R}^m$ all feasible substationary points $x^* \in \mathbb{R}^n$ of the corresponding lower level problem are Pareto optimal, e.g., if the lower level problem is a convex one. In this case the algorithms to be described below compute a set of points, which certainly solve the corresponding lower level problems, but in general this set is a superset of the Pareto set of the given BLMOP. Consequently, a nondominance and feasibility test with respect to the higher level has to be performed among these points in order to filter out the desired Pareto points. Efficient strategies for nondominance tests can be found in [27].

Obviously, such nondominance tests can be omitted, if it is ensured that all substationary points of the higher level problem are necessarily Pareto optimal for the given BLMOP. This is for instance the case, if – in addition to the mentioned assumptions for the lower level – also the higher level along with the higher level constraints and the constraints implicitly given by the lower level solutions turns out to be a convex problem.

Algorithms for the solution of BLMOPs which also include lower level inequality constraints are presented in Section 6.5. Moreover, in Section 6.4 and Section 6.6 we present algorithms, which do not require derivatives or convexity of the involved functions.

There are several possibilities to derive from Theorem 6.1 the system of equations which has to be solved in order to compute individual Pareto points. Here, we want to present a variant which uses slack variables s, t, u, v, w in order to incorporate the

higher level inequality constraints and the requirement that some multipliers have to be non-negative. For this, let $\tilde{z} := (x, y, \alpha, \zeta, \beta, \mu, \omega, \delta, \nu, \lambda, \rho, s, t, u, v, w)$ and denote by $a \circ b \in \mathbb{R}^d$ the *component wise* or *Hadamard product* of vectors $a, b \in \mathbb{R}^d$, that is, $(a \circ b)_i = a_i b_i$ for $i = 1, \ldots, d$. Then, with the notations of the preceding sections, we propose to solve the following system:

$$\tilde{F}(\tilde{z}) = \begin{pmatrix} \sum_{i=1}^{k} \beta_i \nabla F_i(x,y) + \sum_{i=1}^{n} \lambda_i \nabla \mathcal{L}_{x_i}(x,y,\alpha,\zeta) + \sum_{i=1}^{l} (\nu - \mu_i)\, e_{n+m+i} \\ + \sum_{i=1}^{p} \omega_i \nabla h_i(x,y) + \sum_{i=1}^{r} \rho_i \nabla H_i(x,y) + \sum_{i=1}^{s} \delta_i \nabla G_i(x,y) \\ h(x,y) \\ H(x,y) \\ G(x,y) + w \circ w \\ \bar{\nabla} \mathcal{L}(x,y,\alpha,\zeta) \\ \sum_{i=1}^{k} \beta_i - 1 \\ \sum_{i=1}^{l} \alpha_i - 1 \\ \mu \circ \alpha \\ \delta \circ G(x,y) \\ \alpha - s \circ s \\ \beta - t \circ t \\ \mu - u \circ u \\ \delta - v \circ v \end{pmatrix} = \mathbf{0}.$$

Observe that the $\mathcal{L}_{x_i}(x, y, \alpha, \zeta)$ include first derivatives and so the $\nabla \mathcal{L}_{x_i}(x, y, \alpha, \zeta)$ appearing in the first components of $\tilde{F}$ include second derivatives. Moreover, derivatives of all components of $\tilde{F}$ are required if one wants to solve the system with a derivative based, e.g., a Gauß-Newton method. Thus, in this case derivatives up to third order are needed for approaches based on the solution of $\tilde{F}(\tilde{z}) = 0$.

In some applications, which are affected by uncertainties, it might be required to exclude those points, which are located close to the boundary of the solution set with respect to G and α. This can be achieved by a simplification of the system according to Remark 6.2. For this, we use some sufficiently small $\varepsilon > 0$ to guarantee that all components of α are strictly positive and that all components of $G(x, y)$ are strictly negative. Let $\bar{z} := (x, y, \alpha, \zeta, \beta, \omega, \nu, \lambda, \rho, s, t, w)$ and denote by $\mathbf{1}_d$ the vector $(1, \ldots, 1) \in \mathbb{R}^d$. Then, for obtaining the 'inner' part of the solution, the following simplified system can be solved:

$$\bar{F}(\bar{z}) = \begin{pmatrix} \sum_{i=1}^{k} \beta_i \nabla F_i(x,y) + \sum_{i=1}^{n} \lambda_i \nabla \mathcal{L}_{x_i}(x,y,\alpha,\zeta) + \nu \sum_{i=1}^{l} e_{n+m+i} \\ + \sum_{i=1}^{p} \omega_i \nabla h_i(x,y) + \sum_{i=1}^{r} \rho_i \nabla H_i(x,y) \\ h(x,y) \\ H(x,y) \\ G(x,y) + w \circ w + \varepsilon \, \mathbf{1}_s \\ \bar{\nabla} \mathcal{L}(x,y,\alpha,\zeta) \\ \sum_{i=1}^{k} \beta_i - 1 \\ \sum_{i=1}^{l} \alpha_i - 1 \\ \alpha - s \circ s - \varepsilon \\ \beta - t \circ t \end{pmatrix} = \mathbf{0}$$

Later on in this section, the difference between the solutions of $\tilde{F}(\tilde{z}) = 0$ and $\bar{F}(\bar{z}) = 0$ corresponding to Example 6.6 will be shown (Figures 20 and 21). It should be pointed out that the application of the simplified system $\bar{F}(\bar{z}) = 0$ can be meaningful for the recovering methods presented later on in this section. These methods are based on continuation techniques which require for every connected component of the Pareto set at least one single solution out of this component as a starting point. According to the simpler structure, if it is known that $\{\bar{z} : \bar{F}(\bar{z}) = 0\} \neq \emptyset$, we propose to compute a solution of the system $\bar{F}(\bar{z}) = 0$ for finding such starting points. Moreover, the simplification from $\tilde{F}$ to $\bar{F}$ might be an acceptable compromise for those applications, where the corresponding part of the solution is satisfactory to the decision maker while computation has to be fast. This applies particularly for the Subdivision-BL algorithm, since the computational effort of subdivision methods increases drastically with the dimension of the search space.

For the remainder of this work we will always write $\tilde{F}(\tilde{z})$, but we emphasize that in practice a simplification according to Remark 6.2, e.g., of the form $\bar{F}(\bar{z})$, can be used, if desired.

The Case $k = 1$

Before we describe the set-oriented methods for the solution of the typically extensive Pareto sets in the case $k > 1$, we consider the special case $k = 1$, that is, if there

is only one upper level objective. In this case each connected component of the Pareto set is typically given by a single Pareto point. Here, the task of computing a solution of the BLMOP can be regarded as computing a *preferred solution* out of the Pareto set of the lower level problem, where the selection of this preferred solution is performed due to the upper level problem. Such problems are strongly related to the field of *optimization over the efficient set*, see for instance [8], [2], [44] and references therein. As mentioned in Remark 6.2, for $k = 1$ the system $\tilde{F}$ has the same number of variables as it has equations. Consequently, a standard Newton method is suitable for finding a solution of this particular BLMOP.

A BLMOP Formulation for Finding Robust Pareto Points

As a special application for the case $k = 1$, we consider the problem of finding a Pareto point $x^\star$ of a MOP with the additional property that it is *robust* or *insensible* in the sense that the variation of the objective values is as small as possible in a neighborhood $U_\varepsilon(x^\star)$ of $x^\star$. According to the requirements associated with different applications, there are several possibilities to model robustness or sensitivity. Here we use the expression

$$\sigma_i(x) := ||\nabla f_i(x)||_2^2$$

to measure the sensitivity corresponding to the i-th (lower level) objective and we assume that the user is interested in a solution $x^\star$ for which

$$\max_{i \in L} \{\sigma_i(x^\star)\}$$

is minimal ($L := \{1, \ldots, l\}$). Now, with $f(x) = (f_1(x), \ldots, f_l(x))^t$ we can write down the mathematical formulation of the described problem:

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} \max_{i \in L} \sigma_i(x), \\ \text{where } x \text{ solves:} \quad &\min_{x \in \mathbb{R}^n} f(x). \end{aligned} \tag{RMOP}$$

Observe that only the lower level problem, i.e., the minimization of f, has to be understood in the multi-objective sense, whereas minimization and maximization in the upper level problem have to be understood in the classical sense, i.e., minimization and maximization of a scalar valued expression. In order to obtain differentiability we consider the equivalent problem

$$\begin{aligned} &\min_{x \in \mathbb{R}^n, \gamma \in \mathbb{R}} \gamma, \\ \text{where} \quad &\sigma_i(x) - \gamma \leq 0 \quad \forall i \in L, \\ \text{and } x \text{ solves:} \quad &\min_{x \in \mathbb{R}^n} f(x), \end{aligned} \tag{6.24}$$

which has the form of BLMOP.

EXAMPLE 6.4 We consider the RMOP associated with the MOP

$$\min_{x\in\mathbb{R}^2} f(x) = \begin{pmatrix} 2(x_1-1)^4 + 3(x_2-1)^2 \\ (x_1+1)^2 + (x_2+1)^2 \end{pmatrix}. \tag{6.25}$$

The corresponding sensitivity expressions are given by

$$\sigma_1(x) = 64(x_1-1)^6 + 36(x_2-1)^2$$

and

$$\sigma_2(x) = 4(x_1+1)^2 + 4(x_2+1)^2.$$

The solution was computed by solving the system $\tilde{F}(\tilde{z}) = 0$ associated with the corresponding reformulation of the form (6.24). For the purpose of visualization we have also computed both the entire Pareto set of (6.25) and the corresponding sensitivity values. The results are shown in Figure 14 and Figure 15. The Pareto optimal objective values (blue) of (6.25), the respective sensitivity values (red) and, the solution (black) of the RMOP are shown in Figure 14. Here one can see that the maximum of the sensitivities is minimal at the solution. The Pareto set together with the solution of the RMOP in image space is shown in Figure 15.

BL-Subdivision Algorithm

We define different variants of the BL-Subdivision algorithm by taking the classical Subdivision algorithm presented in Section 3.1, changing the search space and replacing the iteration scheme by an iteration scheme for solving the system (or subsystems of) $\tilde{F}(\tilde{z}) = 0$.

To define the basic BL-Subdivision algorithm, denote by $GN^s(z_0)$ the application of s steps of the iteration scheme, e.g., a Gauß-Newton method for solving $\tilde{F}(\tilde{z}) = 0$ with starting point z_0. Let N be the dimension of $z = \tilde{z}$. Then, for a given domain $Q = \mathcal{B}_0 \in \mathbb{R}^N$ an iteration of the basic BL-Subdivision algorithm reads as follows:

(i) Construct from $\mathcal{B}_{j-1}$ a new system $\hat{\mathcal{B}}_j$ of subsets such that

$$\bigcup_{B\in\hat{\mathcal{B}}_j} B = \bigcup_{B\in\mathcal{B}_{j-1}} B$$

and

$$\operatorname{diam}(\hat{\mathcal{B}}_j) = \theta_j \operatorname{diam}(\mathcal{B}_{j-1}),$$

where $0 < \theta_{min} \leq \theta_j \leq \theta_{max} < 1$.

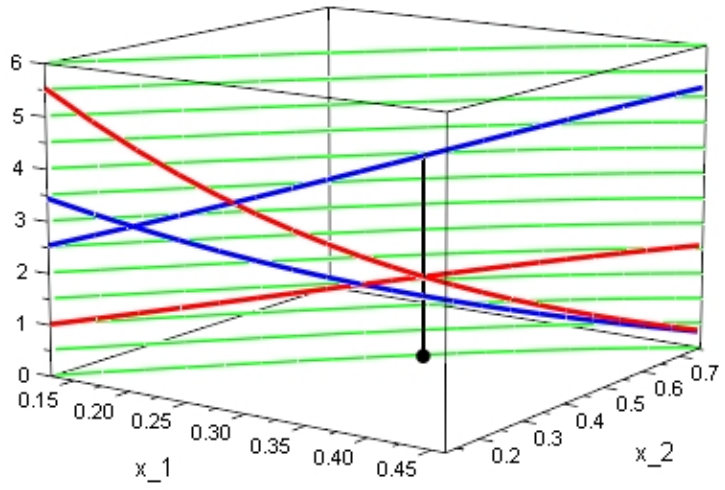

Figure 14: The Pareto optimal objective values (blue), the corresponding sensitivities (red), and the solution (black) of Problem (6.25).

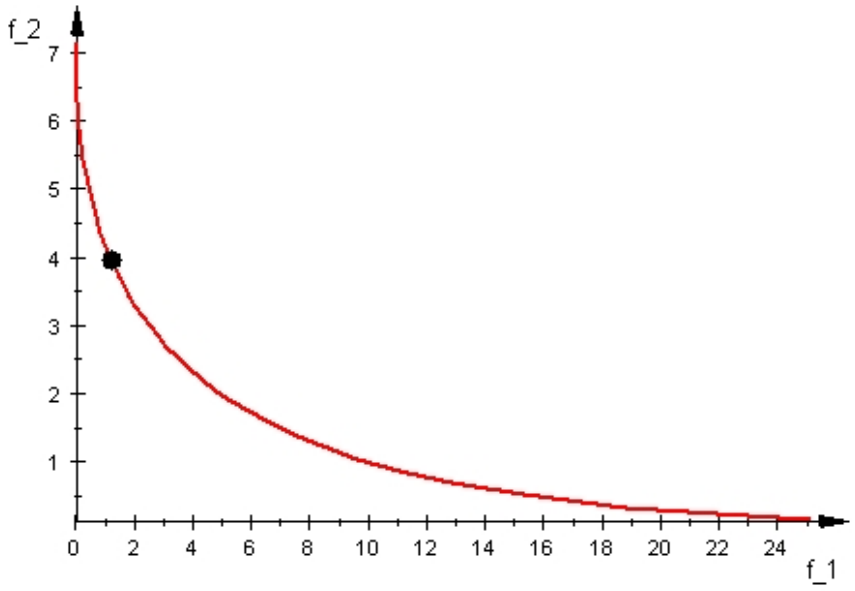

Figure 15: The Pareto set of (6.25) in image space and the solution to the corresponding RMOP.

(ii) For all $B \in \hat{\mathcal{B}}_j$:

choose a set of starting points $\mathcal{X} \subset B$

$\mathcal{B}_j = \mathcal{B}_j \cup \{\bar{B} : \exists z \in \mathcal{X} \text{ with } GN^s(z) \in \bar{B}\}$.

Observe that for every starting point $z \in \mathcal{X}$ only a small number s of steps of the iteration scheme for solving $\tilde{F}(\tilde{z}) = 0$ is performed and so the generated point $GN^s(z)$ is just one point of the trajectory defined by all iterates $GN^i(z), i \in \mathbb{N}$. Moreover, for $i \to \infty$ the trajectory might lead to a Pareto point far away from the 'starting box' $B \ni z$ regardless of the fact that there might also be Pareto points close to or even within B. The resulting inconvenience is, that the box B and with it a potential part of the Pareto set might get lost while the box containing $GN^s(z)$ is kept even if it does not contain any Pareto points. A way out is to force the iterates to converge towards a Pareto point located as close as possible to a reference point within the starting box. Motivated by these considerations, we propose an alternative way to realize the above algorithm. To this end, we replace the zero finding method by a suitable iteration scheme for the solution of the constrained minimization problem

$$\begin{aligned} &\min_{\tilde{z}} ||\tilde{z} - r_B||, \\ &s.t.\ \ \tilde{F}(\tilde{z}) = 0, \end{aligned} \tag{6.26}$$

where r_B denotes a reference point within the starting box B, e.g., the center of B or the used starting point. For an illustration of this strategy see Figure 16.

Example 6.5 To demonstrate how the Subdivision-BL algorithm works, we consider the following BLMOP with two higher level objectives F_1, F_2, three lower level objectives f_1, f_2, f_3, a higher level inequality constraint G_1 and a lower level equality constraint h_1:

$$\min_{x \in \mathbb{R}^3,\, y \in \mathbb{R}} F(x,y) = \begin{pmatrix} (x_0+1)^2 + (x_1 - 1 - y)^4 + x_2^2 \\ (x_0-1)^2 + (x_1 + 1 - y)^2 + (x_2 - 0.5)^4 \end{pmatrix},$$

such that $\quad G_1(x,y) = -0.5\, x_0^2 + x_1 + \sin y - 0.5 \leq 0,$

and x solves:

$$\min_{x \in \mathbb{R}^3} f(x,y) = \begin{pmatrix} (x_0-1)^2 + 0.5(x_1+y)^2 + (x_2-1)^4 \\ (x_0+1)^2 + 0.5(x_1+y)^2 + (x_2+1)^2 \\ x_0^2 + x_1^2 + (x_2+1)^2 \end{pmatrix}$$

such that $h_1 = x_0 - x_1 y = 0.$

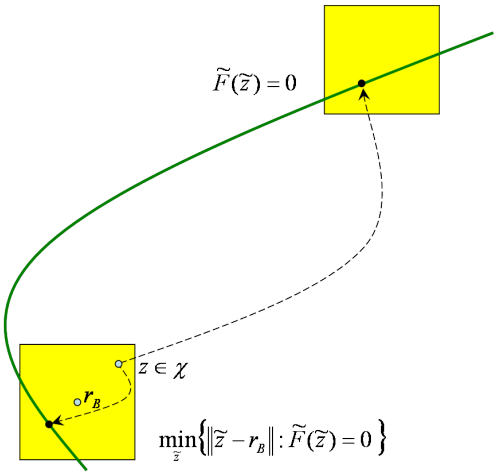

Figure 16: $\min_{\tilde{z}}\{||\tilde{z} - r_B|| : \tilde{F}(\tilde{z}) = 0\}$ versus $\tilde{F}(\tilde{z}) = 0$.

The results are shown in Figures 17, 18 and 19. In particular, Figure 17 shows a projection of the generated box collection to the x-space. The individual solutions $GN^s(z)$ of the subproblems of the last iteration have been stored in an archive. Figure 18 and 19 show these solutions in lower and higher level image space, respectively.

Let us now consider another variant of the BL-Subdivision algorithm, which uses a strategy described in [38]. Instead of using an iteration scheme for solving the entire system of equations, a scheme for solving just one of the components $\tilde{F}_l(\tilde{z}) = 0$, $l \in \{1, \ldots, N - k + 1\}$, e.g., a damped Newton method, is used in every step of the subdivision procedure. In order to guarantee convergence, every component has to be applied infinitely often as the algorithm proceeds. In [38] this concept was originally used for the location of zeros of a function $f : \mathbb{R}^n \to \mathbb{R}$ using a finite set of iteration schemes each of them characterized by an individual step length parameter. In this way it can be avoided that the iterates run into undesired oscillations such that the algorithm would keep the corresponding boxes which possibly do not contain zeros of f.

For the development of the variant of the BL-Subdivision algorithm, we extended this idea in the sense that the different iteration schemes correspond to Newton methods applied to the individual components $\tilde{F}_l$, $l \in \{1, \ldots, N - k + 1\}$. For the description of this variant let $\{u_j\}_{k=j}^{\infty}$ be a sequence with $u_j \in \{1, \ldots, N-k+1\}$ for all $j \in \mathbb{N}$ and $|\{j : u_j = l\}| = \infty$ for all $l = 1, \ldots, N - k + 1$. Furthermore, denote

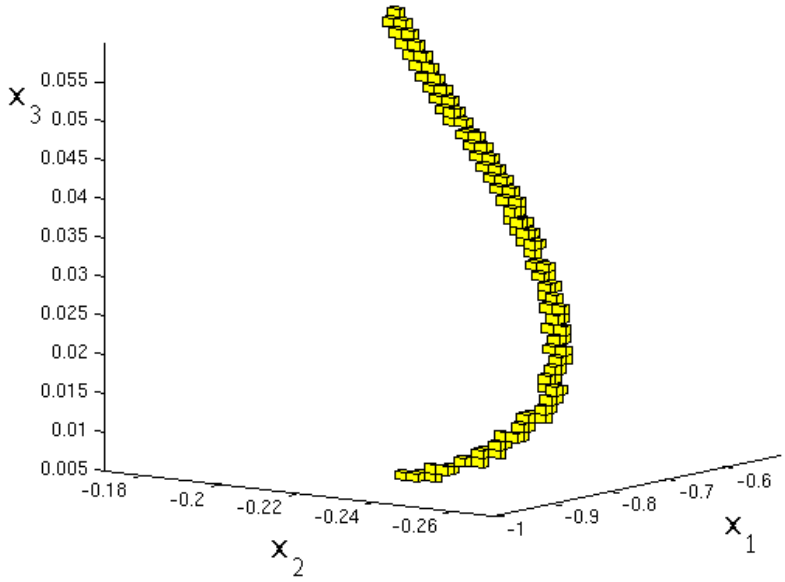

Figure 17: Projection of the generated box collection to the x-space.

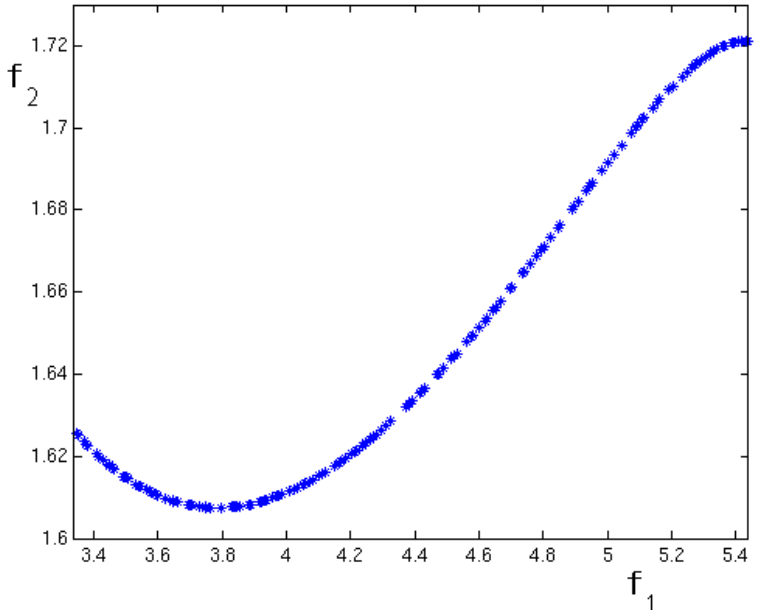

Figure 18: The solution in lower level image space.

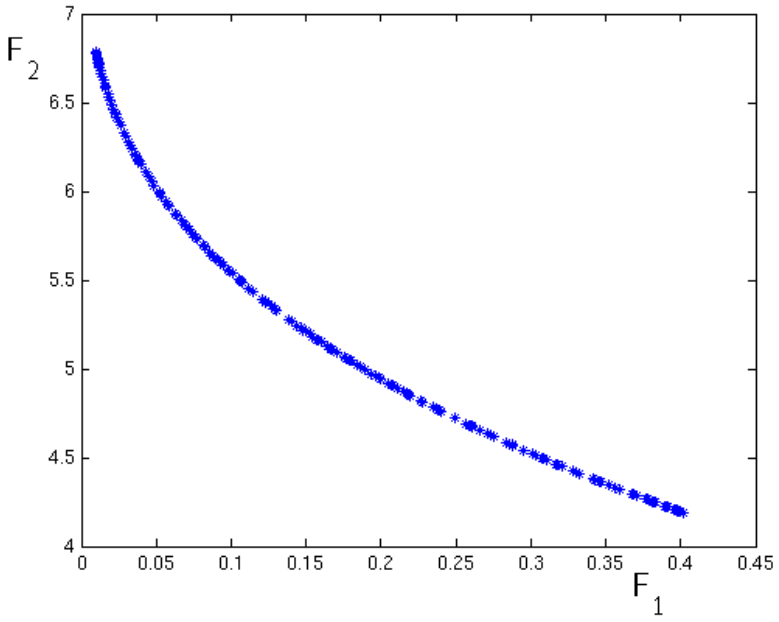

Figure 19: The solution in higher level image space.

by $N_l^s(z_0)$ the application of s steps of an iteration scheme, e.g., a Newton method, for solving $\tilde{F}_l(\tilde{z}) = 0$ with starting point z_0. Then, for a given domain $Q = \mathcal{B}_0 \in \mathbb{R}^N$ an iteration of the variant of the BL-Subdivision algorithm reads as follows:

(i) Construct from $\mathcal{B}_{j-1}$ a new system $\hat{\mathcal{B}}_j$ of subsets such that

$$\bigcup_{B \in \hat{\mathcal{B}}_j} B = \bigcup_{B \in \mathcal{B}_{j-1}} B$$

and

$$\operatorname{diam}(\hat{\mathcal{B}}_j) = \theta_j \operatorname{diam}(\mathcal{B}_{j-1}),$$

where $0 < \theta_{min} \leq \theta_j \leq \theta_{max} < 1$.

(ii) For all $B \in \hat{\mathcal{B}}_j$:

choose a set of starting points $\mathcal{X} \subset B$

$\mathcal{B}_j = \mathcal{B}_j \cup \{\bar{B} : \exists z \in \mathcal{X}$ with $N_{u_j}^s(z) \in \bar{B}\}$.

BL-Recovering-PS Algorithm

If in addition to the previous assumptions at least one box containing points of the solution has been found so far, a variant of the Recovering-PS algorithm described

in Section 3.1 can be used to generate a box covering of the corresponding connected component of the Pareto set. To be more precise, we use a fixed subdivision depth d and define the BL-Recovering-PS algorithm by taking the classical Recovering algorithm and replacing the corrector step by a suitable method for solving $\tilde{F}(\tilde{z}) = 0$. For $z \in \mathbb{R}^N$ denote by $\Pi_{(x,y)}(z)$ the projection from the z-space $\mathbb{R}^N$ to the (x, y)-space $\mathbb{R}^{n+m}$. Moreover, denote by $B(z, d)$ the unique box $B \subset \mathbb{R}^{n+m}$ of subdivision depth d with $\Pi_{(x,y)}(z) \in B$. For every box B generated by the algorithm, let $z_B \in \mathbb{R}^N$ be the particular solution of $\tilde{F}(\tilde{z}) = 0$ which led to B, that is, $\tilde{F}(z_B) = 0$ and $\Pi_{(x,y)}(z_B) \in B$.

With these notations, for a given (incomplete) box collection $\mathcal{B}_j \subset \mathbb{R}^{n+m}$ an iteration of the BL-Recovering-PS algorithm reads as follows:

(i) for all $B \in \mathcal{B}_j$

$B.active := TRUE$

(ii) for $i = 1, \ldots, MaxStep$

$\hat{\mathcal{B}}_j := \mathcal{B}_j$
for all $\{B \in \mathcal{B}_j : B.active == TRUE\}$
choose a set of starting points $\mathcal{X} \in \mathbb{R}^N$ near z_B
$\mathcal{Y} := \emptyset$
for all $z \in \mathcal{X}$
starting from z solve $\tilde{F} = 0$ and get the solution Z
$\mathcal{Y} := \mathcal{Y} \cup Z$
$B.active := FALSE$
for all $z \in \mathcal{Y}$:
if $B(z, d) \notin \hat{\mathcal{B}}_j$
$B(z, d).active := TRUE$
$\hat{\mathcal{B}}_j := \hat{\mathcal{B}}_j \cup B(z, d)$
if $\hat{\mathcal{B}}_j == \mathcal{B}_j$ **STOP**
$\mathcal{B}_{j+1} := \hat{\mathcal{B}}_j$

In each loop of step (ii) the aim is to extend the current box collection $\mathcal{B}_j$ by adding further boxes. The algorithm terminates when no further boxes can be found or when the maximum number of iterations $MaxStep$ is exceeded.

Observe that in contrast to the BL-Subdivision algorithm, which uses boxes in $\mathbb{R}^N$, the BL-Recovering-PS algorithm works with boxes in the smaller space $\mathbb{R}^{n+m}$, which is an advantage concerning the computational effort. Certainly, one could

implement a variant of the BL-Recovering-PS algorithm using boxes in $\mathbb{R}^N$. But this would be inefficient, since for every solution $(x^\star, y^\star)$ of the BLMOP there are in general many solutions $z \in \mathbb{R}^N$ with $\tilde{F}(z) = 0$ and $\Pi_{(x,y)}(z) = (x^\star, y^\star)$, such that the algorithm would generate many boxes in $\mathbb{R}^N$ which all correspond to $(x^\star, y^\star)$, although one box would be sufficient. A main difference to the BL-Subdivision algorithm is, that according to the local nature of the BL-Recovering-PS algorithm all starting points z are chosen in the neighborhood of solutions z_B found so far. Consequently, we expect that with these starting points the zero finding method converges to further solutions of $\tilde{F}(\tilde{z}) = 0$. The situation is different for the BL-Subdivision algorithm, because here we do not know whether the starting points z are close enough to the solution of $\tilde{F}(\tilde{z}) = 0$. Therefore, the BL-Subdivision algorithm has to work with boxes in $\mathbb{R}^N$.

Let us now make a remark on a variant of the BL-Recovering-PS algorithm. In general, the applied iteration scheme might lead to substationary points which are far away from the region to be explored in order to find further Pareto points. But originally, the idea of our recovering methods was to find in every step further Pareto points (or boxes) in the neighborhood of a certain Pareto point (or box) found in the previous step. For some applications, particularly if a certain part of the Pareto set is required, it could be desired to maintain this local behavior of the algorithm even for starting points farther from the Pareto set. To this end, we propose the following variant of the BL-Recovering-PS algorithm. Instead of just solving the system $\tilde{F}(\tilde{z}) = 0$, we can search for a solution of this system with the additional property that the distance to the starting point with respect to (x, y) is minimized, that is, we have to solve

$$\begin{aligned} &\min_{\tilde{z}} ||\Pi_{(x,y)}(\tilde{z}) - \Pi_{(x,y)}(z_0)||, \\ &s.t.\ \ \tilde{F}(\tilde{z}) = 0, \end{aligned} \tag{6.27}$$

where z_0 denotes the respective starting point which now has the additional role of a reference point. This variant was applied to the following

Example 6.6

$$\min_{x \in \mathbb{R}^3,\, y \in \mathbb{R}} F(x, y) = \begin{pmatrix} (x_0 + 1)^2 + (x_1 - 1 - y)^4 + x_2^2 \\ (x_0 - 1)^2 + (x_1 + 1 - y)^2 + (x_2 - 0.5)^4 \end{pmatrix},$$

such that $H_1(x,y) = x_0^2 + x_2 - y^2 = 0,$
$G_1(x,y) = x_0 + x_1^3 + 0.5 \leq 0,$

and x solves:

$$\min_{x \in \mathbb{R}^3} f(x,y) = \begin{pmatrix} (x_0-1)^2 + 0.5(x_1+y)^2 + (x_2-1)^4 \\ (x_0+1)^2 + 0.5(x_1+y)^2 + (x_2+1)^2 \\ x_0^2 + x_1^2 + (x_2+1)^2 \end{pmatrix}$$

$$\text{such that } h_1 = x_0 - x_1 y = 0.$$

Figure 20 and Figure 21 show the projections to the $x-$space of the sets computed by the BL-Recovering-PS algorithm based on the solution of $\tilde{F}(\tilde{z}) = 0$ and $\bar{F}(\bar{z}) = 0$, respectively. To be more precise, Figure 20 shows a representation of the entire Pareto set according to the system $\tilde{F}(\tilde{z}) = 0$, and Figure 21 shows a representation of the 'inner' part of the Pareto set according to the system $\bar{F}(\bar{z}) = 0$. This 'inner' part is characterized by the fact that the upper level inequality constraint is inactive, that is, $G_1 < 0$.

Choice of Starting Points There are different possibilities for generating $\mathcal{X}$, the set of starting points. Of course, randomly chosen starting points can be used. Another approach is to take points $z_j = c + \sum_i \lambda_i e_i$, where c is the center of the current box, the e_i are basis vectors of the search space and, the λ_i are real numbers such that every $\Pi_{(x,y)}(z_j)$ lies within one of the neighboring boxes where new substationary points are supposed to be. If derivatives are available, the points z_j can be generated by computing a basis b_i of the tangent space to the set $\tilde{F}^{-1}(0)$ at a previously computed point $z^\star \in \tilde{F}^{-1}(0)$ and then setting $z_j = z^\star + \sum_i \lambda_i b_i$. In this latter approach, in particular when the dimension of the search space is very large, the number of points z_j can be chosen small compared to other approaches. Moreover, the points z_j are already located relatively close to the solution set $\tilde{F}^{-1}(0)$ and so the corrector step needs just a few iterations. For details on computing the basis vectors b_i see [25].

Convergence of Recovering-PS Algorithms

Since our algorithms generate substationary points of the given BLMOP by solving the system of equations $\tilde{F}(\tilde{z}) = 0$, we give a more general convergence proof concerning recovering techniques for the location of zeros of a vector valued function $F : \mathbb{R}^n \to \mathbb{R}^m$, $n > m$. The proof is achieved in two steps: first, Theorem 6.7 gives a guideline for the choice of a minimal number of starting points p_i within a given

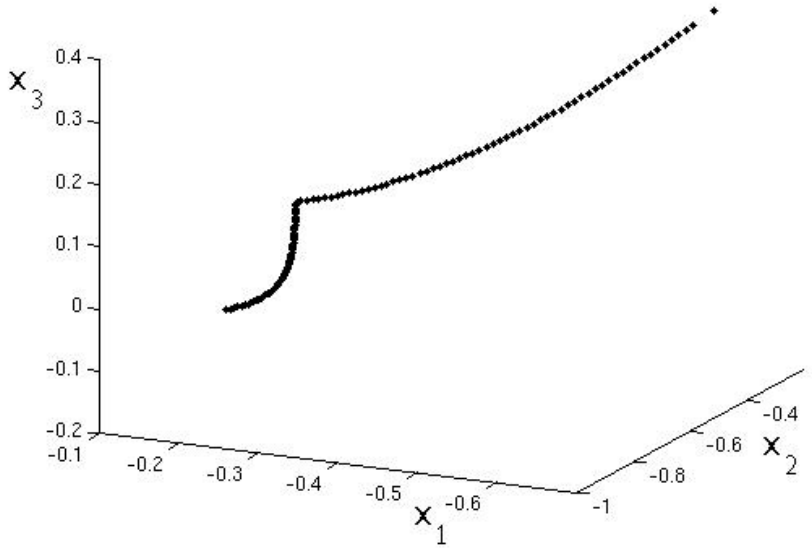

Figure 20: The entire solution of Example 6.6 obtained by solving $\tilde{F}(\tilde{z}) = 0$ (projection to x-space).

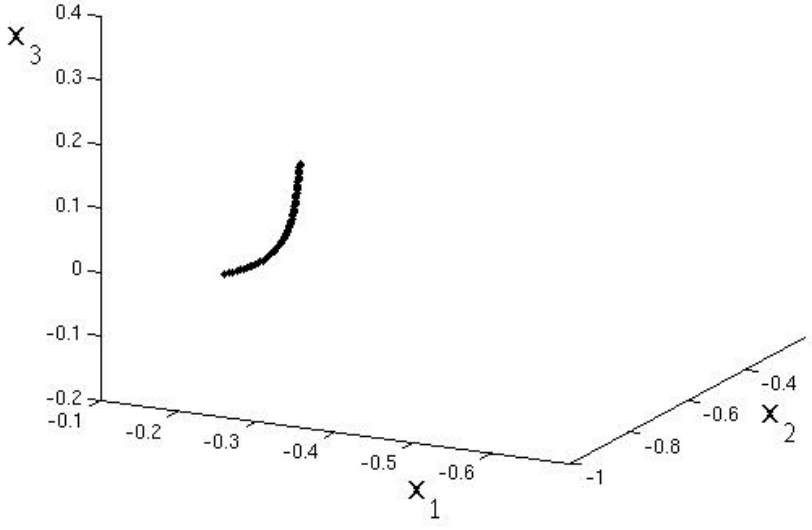

Figure 21: The 'inner' part of the solution of Example 6.6 obtained by solving $\bar{F}(\bar{z}) = 0$ (projection to x-space).

subset $B \subset \mathbb{R}^n$, e.g., a box, such that, under certain assumptions on the iteration scheme $\Phi : \mathbb{R}^n \to \mathbb{R}^n$, convergence towards a solution of $F(x) = 0$ within B is guaranteed in the sense that

$$\lim_{s\to\infty} \Phi^s(p_i) \in \{x : F(x) = 0\} \cap B$$

for at least one p_i. Thereafter, this result is used in Corollary 6.9 in order to show convergence for general recovering methods. For a norm $||\cdot||$ denote by $\operatorname{dist}(x, A) = \min_{y\in A} ||x-y||$ the distance between the point $x \in \mathbb{R}^n$ and the compact set $A \subset \mathbb{R}^n$.

THEOREM 6.7 *Let $F(x) : \mathbb{R}^n \to \mathbb{R}^m$, denote $P := \{x \in \mathbb{R}^n : F(x) = 0\}$ and let $B \subset \mathbb{R}^n$ be an open set such that $P \cap B \neq \emptyset$. Let $\Phi : \mathbb{R}^n \to \mathbb{R}^n$ be a mapping and let $\delta > 0, \lambda \geq 1$ such that for all $x_0 \in B$ with $\operatorname{dist}(x_0, P) < \delta$ we have*

(i) $\lim_{s\to\infty} \Phi^s(x_0) \in P$,

(ii) $\lim_{s\to\infty} ||\Phi^s(x_0) - x_0|| \leq \lambda \operatorname{dist}(x_0, P)$.

Then there is $d > 0$ such that for any set $\mathcal{X} \subset B$ of starting points with $\operatorname{dist}(x, \mathcal{X}) < d$ for all $x \in B \cap \{x : \operatorname{dist}(x, P) < \delta\}$ there is a point $p_0 \in \mathcal{X}$, such that

$$\lim_{s\to\infty} \Phi^s(p_0) \in P \cap B.$$

Proof: Since B is open and $P \cap B \neq \emptyset$, there is $\bar{x} \in P$, $\varepsilon > 0$, and an open neighborhood $U_\varepsilon(\bar{x})$ of $\bar{x}$, such that

$$P \cap U_\varepsilon(\bar{x}) \subset B.$$

For $0 < \theta < 1$ let $d := \theta \min\{\delta, \frac{\varepsilon}{2\lambda}\}$. Then, for any set $\mathcal{X} \subset B$ of starting points with $\operatorname{dist}(x, \mathcal{X}) < d$ for all $x \in B \cap \{x : \operatorname{dist}(x, P) < \delta\}$, there is $p_0 \in \mathcal{X}$, such that

$$\operatorname{dist}(p_0, P) \leq ||p_0 - \bar{x}|| < \min\{\delta, \frac{\varepsilon}{2\lambda}\}$$

and therefore

$$x^\star := \lim_{s\to\infty} \Phi^s(p_0) \in P.$$

Moreover,

$$||x^\star - p_0|| = \lim_{s\to\infty} ||\Phi^s(p_0) - p_0|| \leq \lambda \operatorname{dist}(p_0, P) \leq \lambda ||p_0 - \bar{x}|| < \frac{\varepsilon}{2}$$

and consequently

$$||x^\star - \bar{x}|| \leq ||x^\star - p_0|| + ||p_0 - \bar{x}|| < \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon,$$

that is, $x^\star \in U_\varepsilon(\bar{x})$. Since $U_\varepsilon, (\bar{x}) \subset B$, it follows that $x^\star$ is a solution of $F(x) = 0$ within B. □

REMARK 6.8 If the mapping Φ is given by an iteration step for the solution of (6.27), the requirement on λ in Theorem 6.7 is satisfied with the choice $\lambda = 1$.

To guarantee that a recovering method converges towards the union of those connected components of the solution set which correspond to the initial box collection $\mathcal{B}_0$, in every step of the algorithm the set of starting points p_i has to be chosen properly, such that all desired boxes are found, that is, boxes which are both neighbors of the boxes generated in the respective previous step and contain a part of the respective connected component of the solution. To this end, we denote by $\bar{B}$ the closure of a box B and we state the following

COROLLARY 6.9 *Using the notations of Theorem 6.7 and denoting by $\mathcal{B}_0$ a box collection of (fixed) subdivision depth d covering a part of P, assume that every step of the Recovering or BL-Recovering-PS algorithm, respectively, is realized in a way such that for every $B \in \mathcal{B}_j \backslash \mathcal{B}_{j-1}$ starting points are chosen according to Theorem 6.7 within all boxes $C \in \{C : \bar{C} \cap \bar{B} \neq \emptyset, C \notin \mathcal{B}_j\}$. Moreover, assume that P is bounded. Then, if the mapping defined by the iteration steps for solving $F(x) = 0$ fulfills the requirements on the mapping Φ stated in Theorem 6.7, the algorithm terminates after a finite number of steps such that the final box collection covers those connected components of P, which correspond to at least one $B \in \mathcal{B}_0$.*

BL-Recovering-IS Algorithm

Similarly as presented for classical multi-objective optimization problems in Section 4, image set-oriented variants of the recovering methods can also be defined for BLMOPs with a convex lower level problem. To this end the search space has to be changed and the distance minimization subproblems, which have to be solved within every step of the algorithm, have to be replaced by zero finding problems. To be more precise, the higher level objectives have to be replaced by the scalar valued objective $||F(x, y) - T||$, and we have to solve the system $\tilde{F}(\tilde{z}) = 0$ for this altered BLMOP, that is, according to Remark 6.2, $\tilde{F}(\tilde{z})$ simplifies to

$$\tilde{F}(\tilde{z};T)=\begin{pmatrix}\nabla||F(x,y)-T||+\sum_{i=1}^{n}\lambda_i\nabla\mathcal{L}_{x_i}(x,y,\alpha,\zeta)+\sum_{i=1}^{l}(\nu-\mu_i)\,e_{n+m+i}\\ +\sum_{i=1}^{p}\omega_i\nabla h_i(x,y)+\sum_{i=1}^{r}\rho_i\nabla H_i(x,y)+\sum_{i=1}^{s}\delta_i\nabla G_i(x,y)\\ h(x,y)\\ H(x,y)\\ G(x,y)+w\circ w\\ \bar{\nabla}\mathcal{L}(x,y,\alpha,\zeta)\\ \sum_{i=1}^{l}\alpha_i-1\\ \mu\circ\alpha\\ \delta\circ G(x,y)\\ \alpha-s\circ s\\ \mu-u\circ u\\ \delta-v\circ v\end{pmatrix}=\mathbf{0},$$

where T is the fixed target corresponding to the individual subproblem, and $\tilde{z}$ is understood to be reduced in the sense that there are no components β and t. Therefore, in the following we can change notation by using small letters t or t_i for the targets. Observe that the number of variables agrees with the number of equations, that is, a Newton method can be used for the solution of the system. For a given box collection $\mathcal{B}_j \subset \mathbb{R}^k$ (in image space) of subdivision depth d and denoting by z_B and F_B the previously generated solution (in parameter and image space, respectively) associated with a box $B \in \mathcal{B}_j$, a step of the BL-Recovering-IS algorithm can be written as follows:

(i) for all $B \in \mathcal{B}_j$

$B.active := TRUE$

(ii) for $j = 1, \ldots, MaxStep$

$\hat{\mathcal{B}}_j := \mathcal{B}_j$

for all $B \in \{B \in \mathcal{B}_j : B.active == TRUE\}$

choose target vectors $\{t_i\}_{i=1,\ldots,n_t}$ near B with $t_i <_p F_B$

find $\tilde{z}_i^\star$ with $\tilde{F}(\tilde{z}_i^\star; t_i) = 0, i = 1, \ldots, n_t$

$F_i^\star := F(\Pi_{(x,y)}(\tilde{z}_i^\star)), i = 1, \ldots, n_t$

$B.active := FALSE$

for all $i = 1, \ldots, n_t$:

if $B(F_i^\star, d) \notin \hat{\mathcal{B}}_j$

$\check{B} := B(F_i^\star, d),\ \tilde{z}_{\check{B}} := \tilde{z}_i^\star,\ F_{\check{B}} := F_i^\star$

$\check{B}.active := TRUE$

$\hat{\mathcal{B}}_j := \hat{\mathcal{B}}_j \cup \check{B}$

if $\hat{\mathcal{B}}_j == \mathcal{B}_j$ **STOP**

$\mathcal{B}_{j+1} := \hat{\mathcal{B}}_j$

Since the set of higher level objectives is replaced by a scalar distance minimization problem, another variant of this algorithm can be defined by using standard methods for the solution of classical optimization problems. To be more precise, in every step the expressions $||F(x,y) - t_i||$ have to be minimized under the restriction that only those components of $\tilde{F}$ vanish which guarantee that the optimality conditions for the lower level problem hold and that the higher level constraints of the BLMOP are fulfilled. With

$$\hat{F}(x,y,\alpha,\zeta,w,s) := \begin{pmatrix} h(x,y) \\ H(x,y) \\ G(x,y) + w \circ w \\ \bar{\nabla}_l \mathcal{L}(x,y,\alpha,\zeta) \\ \sum_{i=1} \alpha_i - 1 \\ \alpha - s \circ s \end{pmatrix} = 0$$

and denoting $\hat{z} := (x,y,\alpha,\zeta,w,s)$ and $S := \{\hat{z} : \hat{F}(\hat{z}) = 0\}$, the variant of the BL-Recovering-IS algorithm can be defined by replacing the parameter space by the $\hat{z}$-space and replacing the solution of $\tilde{F}(\tilde{z}; t_i) = 0$ by the distance minimization problem

$$\min_{\hat{z} \in S} ||F(\Pi_{(x,y)}(\hat{z})) - t_i||,$$

where $t_i, i = 1, \ldots, n_t$ denote the targets which have to be chosen individually in every cycle of the algorithm.

The efficiency of the BL-Recovering-IS algorithm will be demonstrated in a more general context by the examples in Section 6.5, where the algorithm is extended by a smoothing technique in order to solve BLMOPs which include lower level inequality constraints. Moreover, we refer to the example in Section 7.3, where the algorithm is combined with a sensitivity analysis based technique for the adaptive choice of targets.

Convergence of Recovering-IS Algorithms

Since the described BL-Recovering-IS algorithm is realized by minimizing a reformulation of the BLMOP, which can be understood as a constrained MOP, in the following we prove convergence for the more general class of image set-oriented recovering algorithms, which includes both the Recovering-IS algorithm and the BL-Recovering-IS algorithm. The proof is carried out in two steps: first, Theorem 6.10 states that for every subset $B \subset \mathbb{R}^k$ containing a part of the Pareto optimal solution in image space, there is a minimal set of targets, such that for at least one of these targets the corresponding distance minimization subproblem leads to a Pareto point $x^\star$ with $F(x^\star) \in B$. Then, this result is used in Corollary 6.11 to complete the proof.

THEOREM 6.10 *Let $F : \mathbb{R}^n \to \mathbb{R}^k, S \subset \mathbb{R}^n$ and denote by P the Pareto set of the constrained MOP:*

$$\min_{x \in S} F(x).$$

Assume that the norm $||\cdot||$ is strictly monotonically increasing. Let $B \subset \mathbb{R}^k$ be an open subset such that $B \cap F(P) \neq \emptyset$. Then there is $d > 0$, such that for any set $\mathcal{X} \subset B$ of targets with $\operatorname{dist}(y, \mathcal{X}) < d$ for all $y \in B \cap F(P)$ there exists a target $t \in \mathcal{X}$ with $F(x^\star) \in F(P) \cap B$, where $x^\star := \arg\min_{x \in S} ||F(x) - t||$.

Proof: There are $\bar{y} \in F(P)$ and $\varepsilon > 0$, such that $U_\varepsilon(\bar{y}) \subset B$. Let $d := \frac{\varepsilon}{8\sqrt{k}}$ and $c := \bar{y} - 2\,d \sum_{i=1}^{k} e_i$, where e_i denotes the i-th standard basis vector in $\mathbb{R}^k$. Then, for every $y \in U_d(c)$, we have

$$||y - \bar{y}|| \leq ||y - c|| + ||c - \bar{y}|| \leq d + 2d\sqrt{k} = \frac{\varepsilon}{8\sqrt{k}} + \frac{\varepsilon}{4} < \frac{\varepsilon}{2},$$

that is, $U_d(c) \subset U_\varepsilon(\bar{y})$. Consequently, there is a target $t = c + v \in \mathcal{X}, ||v|| \leq d$, such that

$$\min_{x \in S_t} ||F(x) - t|| \leq ||t - \bar{y}|| < \frac{\varepsilon}{2}$$

and

$$t_i = c_i + v_i = \bar{y}_i - 2d + v_i < \bar{y}_i \quad for \quad i = 1, \ldots, k.$$

With $x^\star = \arg\min_{x \in S_t} ||F(x) - t||$, it follows that

$$||F(x^\star) - \bar{y}|| \leq ||F(x^\star) - t|| + ||t - \bar{y}|| < \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon$$

and therefore

$$F(x^\star) \in U_\varepsilon(\bar{y}) \subset B.$$

Now we have to show that $F(x^\star)$ is not dominated by any $\hat{y} \in F(P) \cap S_t$. For $F(x^\star) = \hat{y}$ this nondominance is obvious. For the case $F(x^\star) \neq \hat{y}$ we have to show that $F_i(x^\star) < \hat{y}_i$ for at least one $i = 1, \ldots, k$. To see this, assume that the opposite is true. Then $F_i(x^\star) \geq \hat{y}_i > t_i$ for all $i = 1, \ldots, k$, where, since $F(x^\star) \neq \hat{y}$, strict inequality holds for at least one $i \in \{1, \ldots, k\}$. Consequently, since $||\cdot||$ is strictly monotonically increasing,

$$||F(x^\star) - t|| > ||\hat{y} - t||,$$

which is a contradiction to $||F(x^\star) - t|| = \min_{x \in S_t} ||F(x) - t||$. Finally, since $F(x^\star)$ is not dominated by any $\hat{y} \in F(P) \cap S_t$, we have $F(x^\star) \in F(P)$, which completes the proof. □

To guarantee that an image set-oriented recovering method converges towards the union of those connected components of $F(P)$ which correspond to the initial box collection $\mathcal{B}_0$, in every step of the algorithm the set of targets t_i has to be chosen properly, such that all desired boxes are found, that is, boxes which are both neighbors of the boxes generated in the respective previous step and contain a part of the respective connected component of $F(P)$. To this end, we recall that $\bar{B}$ denotes the closure of a box B and we state the following

COROLLARY 6.11 *Using the notations of Theorem 6.10 and denoting by $\mathcal{B}_0$ a box collection of (fixed) subdivision depth d covering a part of $F(P)$, assume that every step of the Recovering-IS or BL-Recovering-IS algorithm, respectively, is realized in a way such that for every $B \in \mathcal{B}_j \backslash \mathcal{B}_{j-1}$ targets are chosen according to Theorem 6.10 within all boxes $C \in \{C : \bar{C} \cap \bar{B} \neq \emptyset, C \notin \mathcal{B}_j\}$. Moreover, assume that $F(P)$ is bounded. Then, the algorithm terminates after a finite number of steps such that the final box collection covers those connected components of $F(P)$, which correspond to at least one $B \in \mathcal{B}_0$.*

6.3 Pareto Set Constrained Multi-Objective Optimization Problems

In this section we introduce particular BLMOPs, which are termed *Pareto set constrained multi-objective optimization problems* PSCMOPs.

From Theorem 6.1 we will derive corollaries, which are concerned with necessary optimality conditions for PSCMOP. Then we will develop algorithms for the solution of this class of problems. In particular, we will present derivative-free algorithms, which turned out to work very satisfactorily in combination.

A PSCMOP can be understood as a variant of BLMOP, where the hierarchical structure is relaxed in the sense that – in contrast to classical bi-level structures – the lower level problem is not a parametrized one. Moreover, there are no explicitly given higher or lower level constraints. To be more precise, consider the vector valued functions $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^k$ and $f : \mathbb{R}^n \to \mathbb{R}^l$. Then a (PSCMOP) can be stated as follows:

$$\begin{aligned} &\min_{x\in\mathbb{R}^n, y\in\mathbb{R}^m} F(x,y), \\ \text{where } x \text{ solves:} \quad &\min_{x\in\mathbb{R}^n} f(x), \end{aligned} \tag{PSCMOP}$$

where minimization again has to be understood in the sense of the partial order $\leq_p$. Based on Theorem 6.1 the following necessary conditions for Pareto optimality for a PSCMOP can be stated. For this let $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}$, defined by

$$\mathcal{L}(x,\alpha) := \sum_{i=1}^{l} \alpha_i f_i(x)$$

denote the lower level Lagrangian. Let $I = \{i : \alpha_i = 0\} \subset \{1,\dots,l\}$ and denote by e_i the $i-$th vector of the standard basis in $\mathbb{R}^{n+m+l}$. Furthermore, for $i = 1,\dots,n$, denote by

$$\mathcal{L}_{x_i} := \frac{\partial}{\partial x_i}\mathcal{L}$$

the derivative of $\mathcal{L}$ with respect to x_i and let $\bar{\nabla} := \nabla_{(x)}$ and $\nabla := \nabla_{(x,y,\alpha)}$.

Corollary 6.12 *Let (x^*, y^*) be a Pareto point of PSCMOP.*
Then there exist $\alpha_1,\dots,\alpha_l \geq 0$ such that

$$(i) \qquad \bar{\nabla}\mathcal{L}(x^*,\alpha) = \sum_{i=1}^{l} \alpha_i \bar{\nabla} f_i(x^*) = 0, \tag{6.29}$$

$$(ii) \qquad \sum_{i=1}^{l} \alpha_i = 1. \tag{6.30}$$

If in addition all solutions of (i) and (ii) comply with Pareto points of the lower level problem and if the Hessian $\bar{\nabla}^2\mathcal{L}(x^*,\alpha)$ of $\mathcal{L}$ is regular at (x^*,α), then there exist

$\beta_1, \ldots, \beta_k, \mu_1, \ldots, \mu_l \geq 0$, $\nu, \lambda_1, \ldots, \lambda_n \in \mathbb{R}$ *such that*

$$(iii) \qquad \sum_{i=1}^{k} \beta_i \nabla F_i(x^*, y^*) + \sum_{i=1}^{n} \lambda_i \nabla \mathcal{L}_{x_i}(x^*, \alpha) + \sum_{i=1}^{l} (\nu - \mu_i)\, e_{n+m+i} = 0, \tag{6.31}$$

$$(iv) \qquad \sum_{i=1}^{k} \beta_i = 1, \tag{6.32}$$

$$(v) \qquad \mu_i = 0 \text{ for } i \notin I. \tag{6.33}$$

Proof: Let (x^*, y^*) be a solution of PSCMOP. Then x^* is a solution to the lower level problem (minimization of f) and the Kuhn-Tucker condition (2.2) holds: there are $\alpha_1, \ldots, \alpha_l \in \mathbb{R}$ such that

$$\bar{\nabla} \mathcal{L}(x^*, \alpha) = \sum_{i=1}^{l} \alpha_i \bar{\nabla} f_i(x^*) = 0, \tag{6.34}$$

$$\sum_{i=1}^{l} \alpha_i = 1 \tag{6.35}$$

and

$$\alpha_i \geq 0, \ i = 1, \ldots, l. \tag{6.36}$$

Since the set of solutions of (6.34), (6.35) and (6.36) is exactly the feasible region for the higher level problem, we can also apply the Kuhn-Tucker conditions (2.2) to the upper level problem while regarding (6.34), (6.35) and (6.36) as constraints. For this we have to ensure that the gradients with respect to (x, y, α) of the active constraints, that is

$$\nabla \mathcal{L}_{x_i}(x^*, \alpha), \qquad i = 1, \ldots, n, \tag{6.37}$$

$$\nabla \sum_{i=1}^{l} \alpha_i = \sum_{i=1}^{l} e_{n+m+i}, \tag{6.38}$$

$$\nabla(-\alpha_i) = -e_{n+m+i}, \qquad i \in I, \tag{6.39}$$

are linearly independent or equivalently, that the matrix

$$\begin{pmatrix} \bar{\nabla}^2 \mathcal{L}(x^*, \alpha) & \mathbf{0} & \bar{\nabla} f_1(x^*) \ldots \bar{\nabla} f_l(x^*) \\ \mathbf{0} & \mathbf{0} & -\tilde{I} \\ \mathbf{0} & \mathbf{0} & 1 \ldots 1 \end{pmatrix} \in \mathbb{R}^{n+|I|+1 \times n+m+l} \tag{6.40}$$

has maximal rank, where the rows of $\tilde{I}$ are given by the transposed of distinct standard basis vectors of $\mathbb{R}^l$ according to (6.39). Since (6.40) has a certain triangular block structure and $\bar{\nabla}^2\mathcal{L}(x^*,\alpha)$ is regular, it remains to show that the submatrix

$$\begin{pmatrix} -\tilde{I} \\ 1\ldots 1 \end{pmatrix} \quad \in \mathbb{R}^{|I|+1\times l} \tag{6.41}$$

has maximal rank. Indeed, with (6.35) and (6.36) we have $|I| < l$, that is, (6.41) has at most l rows. Obviously, these rows are linearly independent and thus both (6.41) and also (6.40) have maximal rank. Consequently, (6.37), (6.38) and (6.39) are linearly independent and we can write down the Kuhn-Tucker conditions (2.2) for the (constrained) upper level problem: there are $\beta_1,\ldots,\beta_k,\mu_1,\ldots,\mu_l \geq 0$, $\nu,\lambda_1,\ldots,\lambda_n \in \mathbb{R}$, such that

$$\sum_{i=1}^{k}\beta_i\nabla F_i(x^*,y^*) + \sum_{i=1}^{n}\lambda_i\nabla\mathcal{L}_{x_i}(x^*,\alpha) + \nu\sum_{i=1}^{l}\nabla\alpha_i + \sum_{i=1}^{l}\mu_i\nabla(-\alpha_i) = 0, \tag{6.42}$$

$$\sum_{i=1}^{k}\beta_i = 1 \tag{6.43}$$

and

$$\mu_i = 0 \ \text{ for } \ i \notin I. \tag{6.44}$$

Observing that $\nabla(-\alpha_i) = -e_{n+m+i}$ for $i = 1,\ldots,l$, we obtain from (6.42)

$$\sum_{i=1}^{k}\beta_i\nabla F_i(x^*,y^*) + \sum_{i=1}^{n}\lambda_i\nabla\mathcal{L}_{x_i}(x^*,\alpha) + \sum_{i=1}^{l}(\nu-\mu_i)\,e_{n+m+i} = 0 \tag{6.45}$$

□

Since for positive coefficients a linear combination of symmetric positive definite matrices is again symmetric positive definite, we can conclude the following corollary, which will be relevant for some of our numerical examples later on.

COROLLARY 6.13 *Let (x^*,y^*) be a Pareto point of PSCMOP.*
Then there exist $\alpha_1,\ldots,\alpha_l \geq 0$, such that

$$(i) \qquad \sum_{i=1}^{l}\alpha_i\bar{\nabla}f_i(x^*) = 0, \tag{6.46}$$

$$(ii) \qquad \sum_{i=1}^{l}\alpha_i = 1. \tag{6.47}$$

$$\tag{6.48}$$

If in addition all solutions of (i) and (ii) comply with Pareto points of the lower level problem and if the Hessian $\bar{\nabla}^2 f_i(x^*)$ of f_i is positive definite at x^* for $i = 1, \ldots, l$, then there exist $\beta_1, \ldots, \beta_k, \mu_1, \ldots, \mu_l \geq 0$, $\nu, \lambda_1, \ldots, \lambda_n \in \mathbb{R}$, such that

$$(iii) \qquad \sum_{i=1}^{k} \beta_i \nabla F_i(x^*, y^*) + \sum_{i=1}^{n} \lambda_i \nabla \mathcal{L}_{x_i}(x^*, \alpha) + \sum_{i=1}^{l} (\nu - \mu_i)\, e_{n+m+i} = 0, \tag{6.49}$$

$$(iv) \qquad \sum_{i=1}^{k} \beta_i = 1, \tag{6.50}$$

$$(v) \qquad \mu_i = 0 \text{ for } i \notin I. \tag{6.51}$$

Proof: The Hessians $\bar{\nabla}^2 f_i(x^*)$, $i = 1, \ldots, l$, are symmetric positive definite and consequently, with $\sum_{i=1}^{l} \alpha_i = 1$ and $\alpha_i \geq 0$ for $i = 1, \ldots, l$, also the Hessian $\bar{\nabla}^2 \mathcal{L}(x^*, \alpha) = \sum_{i=1}^{l} \alpha_i \bar{\nabla}^2 f_i(x^*)$ of $\mathcal{L}(x^*, \alpha)$ is symmetric positive definite at (x^*, α). Since a symmetric positive definite matrix is regular we can use Corollary 6.12 to complete the proof. □

REMARK 6.14 (i) If x^* is an inner point of the solution set of the lower level problem, that is, $I = \emptyset$, then (6.31) and (6.49) simplify to

$$\sum_{i=1}^{k} \beta_i \nabla f_i(x^*, y^*) + \sum_{i=1}^{n} \lambda_i \nabla \mathcal{L}_{x_i}(x^*, \alpha) + \sum_{i=1}^{l} \nu\, e_{n+m+i} = 0 \tag{6.52}$$

(ii) Since the Hessian $\bar{\nabla}^2 f$ of a convex function $f : \mathbb{R}^n \to \mathbb{R}$ is positive definite, Corollary 6.13 is in particular relevant for PSCMOPs with convex lower level problems.

(iii) If all lower and higher level objectives and the solution set of the lower level problem are convex, then all points satisfying the equations stated in Corollary 6.12 and Corollary 6.13 are Pareto points of the given PSCMOP.

6.4 Methods for Solving PSCMOPs

In this section we present algorithms for the solution of PSCMOP. The first two algorithms *PSC-Subdivision* and *PSC-Recovering*, which are based on the results stated

in Corollary 6.12 and Corollary 6.13, are special variants of the more general algorithms BL-Subdivision and BL-Recovering, respectively, for the solution of BLMOP. Since the algorithms BL-Subdivision and BL-Recovering are described in detail in the previous sections, we will restrict ourselves to the consideration of the essential item which distinguishes the PSC-variants from the BL-variants, that is, the simplification of the system of equations to be solved in order to find substationary points.

Thereafter we present the methods *PSC-Sampling* and *PSC-SamRec*, which are based solely on the computation and comparison of objective values and therefore give the opportunity to solve PSCMOPs with objectives that do not meet any smoothness or convexity assumptions. Moreover, we propose some guidelines on how to combine these algorithms in order to increase the performance of the respective numerical schemes.

PSC-Subdivision and PSC-Recovering Algorithm

As mentioned above, the PSC-variants of the Subdivision and Recovering algorithm, respectively, are very similar to the corresponding BL-variants. The essential difference is the system of equations to be solved in order to find Pareto candidates. With $\tilde{z} := (x, y, \alpha, \beta, \mu, \nu, \lambda, s, t, u)$, the system simplifies to

$$\tilde{F}(\tilde{z}) = \begin{pmatrix} \sum_{i=1}^{k} \beta_i \nabla F_i(x,y) + \sum_{i=1}^{n} \lambda_i \nabla \mathcal{L}_{x_i}(x,y,\alpha) + \sum_{i=1}^{l} (\nu - \mu_i)\, e_{n+m+i} \\ \bar{\nabla} \mathcal{L}(x,y,\alpha) \\ \sum_{i=1}^{k} \beta_i - 1 \\ \sum_{i=1}^{l} \alpha_i - 1 \\ \mu \circ \alpha \\ \alpha - s \circ s \\ \beta - t \circ t \\ \mu - u \circ u \end{pmatrix} = \mathbf{0}.$$

Apart from this simplification of the system of equations, the PSC-Subdivision and PSC-Recovering algorithms are identical to the BL-Subdivision and BL-Recovering algorithm, respectively. But there is one more important aspect to be mentioned. Since the lower level problem of a PSCMOP is not parametrized by the variable $y \in \mathbb{R}^m$, the PSC-algorithms can be combined with archiving strategies based on

the two-step nondominance test described below. In this way, it is possible to approximate the Pareto set of the PSCMOP even if substationary points of the lower level problem are not necessarily Pareto optimal for the lower level problem. The two-step nondominance test might theoretically be extended to handle the general BLMOP. But for this, a two-step nondominance test associated with at least every y of a representative discretization of the y-Space has to be performed, such that – in practice – the computational effort for such an extension would be unjustifiable.

PSC-SamRec Algorithm

In many applications the computation of derivatives of the given objectives is a considerable problem. In the majority of cases symbolic derivatives are not available at all and the numerical computation of derivatives is very expensive. Particularly, there is no hope to get derivatives of higher order in justifiable time. This motivates the construction of derivative-free algorithms. In the following we describe the PSC-SamRec algorithm which is similar to the preceding PSC-Recovering algorithm, but does not use derivatives. Moreover, at every considered point (x, y) the higher level function F must only be evaluated if x is nondominated with respect to the lower level function f by any other point accepted so far. The predictor-corrector concept for finding new nondominated points in the neighborhood of a given box is interchanged by just expanding the box to a larger test box, choosing test points in it and then performing a local nondominance test in two steps: first, all points within the test box are checked for nondominance with respect to the lower level function f. Only those points which are in fact f-nondominated are taken into account for the nondominance test concerning the upper level function F. All points "surviving" both steps of the nondominance test build up the basis for the decision whether a box is added to the current box collection or not. Now for a fixed subdivision depth d and a given box collection $\mathcal{B}_j \subset \mathbb{R}^{n+m}$ an iteration of the PSC-Sam-Rec algorithm can be stated as follows:

(i) for all $B \in \mathcal{B}_j$

$B.active := TRUE$

(ii) for $i = 1, \ldots, MaxStep$

$\hat{\mathcal{B}}_j := \mathcal{B}_j$
for all $B \in \{B \in \mathcal{B}_j : B.active == TRUE\}$

create a **test box** $\bar{B}$ such that

$$B \subset \bar{B}$$

and

$$\operatorname{diam}(\bar{B}) = \theta_j \operatorname{diam}(B),$$

where $\theta_j \geq \theta_{min} > 1$.
choose a set of **test points** $\mathcal{X} \subset \bar{B}$
$N_f := f$-nondominated points of $\mathcal{X}$
$N_F := F$-nondominated points of N_f
$B.active := FALSE$
for all $z \in N_F$:
  if $B(z,d) \notin \hat{\mathcal{B}}_j$
    $B(z,d).active := TRUE$
    $\hat{\mathcal{B}}_j := \hat{\mathcal{B}}_j \cup B(z,d)$
if $\hat{\mathcal{B}}_j == \mathcal{B}_j$ **STOP**
$\mathcal{B}_{j+1} := \hat{\mathcal{B}}_j$

Unlike the PSC-Recovering algorithm, in the PSC-SamRec algorithm there is no corrector step. As a result, the probability is small that a randomly chosen test point is a nondominated point. This fact has to be taken into account by considering a sufficiently large number of test points within every test box.
As well as the PSC-Recovering algorithm, the PSC-SamRec algorithm is of local nature, that is, nondominance tests are performed only with respect to a local neighborhood of the current box.

PSC-Sampling Algorithm

The following PSC-Sampling algorithm is an algorithm of subdivision type and therefore it is tailored to generate a box collection that covers the global Pareto set and not only a locally nondominated set. Like in all algorithms of subdivision type, at the beginning of an iteration each box of the current collection is subdivided to generate a finer collection. Thereafter a selection step is employed. For this, test points within each box are chosen, and similar as in the PSC-SamRec algorithm, a nondominance test is performed in two steps. The crucial difference is that test points are not only compared to test points within a local region, but to all test points within the whole box collection. This makes the PSC-Sampling algorithm a global approach. Given a box collection $\mathcal{B}_{j-1} \subset \mathbb{R}^{n+m}$ the collection $\mathcal{B}_j \subset \mathbb{R}^{n+m}$ is obtained by:

(i) Subdivision

Construct from $\mathcal{B}_{j-1}$ a new system $\hat{\mathcal{B}}_j$ of subsets such that

$$\bigcup_{B\in\hat{\mathcal{B}}_j} B = \bigcup_{B\in\mathcal{B}_{j-1}} B$$

and

$$\operatorname{diam}(\hat{\mathcal{B}}_j) = \theta_j \operatorname{diam}(\mathcal{B}_{j-1}),$$

where $0 < \theta_{min} \leq \theta_j \leq \theta_{max} < 1$.

(ii) Selection

for all $B \in \hat{\mathcal{B}}_j$

choose a set of **test points** $\mathcal{X}_B \subset B$

$N_f := f$-nondominated points of $\bigcup_{B\in\hat{\mathcal{B}}_j} \mathcal{X}_B$

$N_F := F$-nondominated points of N_f

$\mathcal{B}_j := \left\{ B \in \hat{\mathcal{B}}_j : \mathcal{X}_B \cap N_F \neq \emptyset \right\}$

Combination of the PSC-Sampling and PSC-SamRec Algorithms

All the proposed approaches are self-contained algorithms, but in many applications one may obtain even better results when combining them. Moreover, a suitable combination can reduce the computational effort.

The PSC-Sampling algorithm finds global Pareto points due to the fact that it works with comparisons within the entire image spaces of both the upper and lower level of the PSCMOP. However, there always remains some uncertainty, in particular when the boxes are big and/or the dimension of the PSCMOP is large. To be more precise, boxes containing global Pareto points may get lost because the selection is performed by considering finitely many test points within each box. Nevertheless in practice it turned out that this algorithm works satisfactorily. The PSC-SamRec algorithm, which is local in nature, successfully extends the computed box covering of the set of substationary points. To obtain an even better performance – i.e. to compute a robust approximation of the global Pareto set *and* to use a moderate amount of function calls – we propose the following combination of the PSC-Recovering and the PSC-Sampling algorithm.

step 0 Apply n_0 iterations of the PSC-Sampling algorithm using p_0 test points per box.

step 1 Apply n_1 iterations of the PSC-SamRec algorithm to the current box collection with p_1 test points per box. This fills the gaps which have possibly been generated in previous steps.

step 2 Use n_2 iterations of the PSC-Sampling algorithm using p_2 test points per box for further refinement of the box collection. In this way boxes which only contain local Pareto points that are not Pareto-optimal from a global point of view can be removed from the covering.

step 3 Carry out the PSC-SamRec algorithm using p_3 test points per box until no more missing boxes are added.

Step 1 and step 2 are repeated until the desired box size is obtained.

With a proper choice of the parameters n_i and p_i, this combination works much better and faster than using just one of the algorithms on its own. Of course, the choices for the n_i and p_i depend on the concrete problem, but experience gives reason to some general rules as follows: p_1 and p_2 can be chosen small compared to p_0 and p_3. The n_i should be chosen based on how many "good" boxes get lost during the application of the PSC-Sampling algorithm. The number of boxes which are added to the collection during step 1 is a measure for the number of test points p_2 needed in step 2.

EXAMPLE 6.15 Given a higher level function $F = (F_1, F_2)^t : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^2$ and a lower level function $f = (f_1, f_2)^t : \mathbb{R}^2 \to \mathbb{R}^2$, we consider the following example:

$$\min_{x \in \mathbb{R}^2,\, y \in \mathbb{R}} F(x,y) = \begin{pmatrix} (x_1-1)^4 + (x_2-1)^2 + (y-1)^2 \\ (x_1+1)^2 + (x_2+1)^4 + (y+1)^2 \end{pmatrix}, \tag{6.53}$$

such that x solves:

$$\min_{x \in \mathbb{R}^2} f(x) = \begin{pmatrix} (x_1-2)^2 + (x_2-2)^4 \\ (x_1+0.5)^2 + (x_2+1)^2 \end{pmatrix}. \tag{6.54}$$

Observe that the Hessians $\bar{\nabla}^2 f_1(x)$ and $\bar{\nabla}^2 f_2(x)$ of the lower level objectives f_1 and f_2 with respect to x are given by

$$\bar{\nabla}^2 f_1(x) = \begin{pmatrix} 2 & 0 \\ 0 & 12\,(x_2-2)^2 \end{pmatrix} \text{ and } \bar{\nabla}^2 f_2(x) = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \tag{6.55}$$

for all $x \in \mathbb{R}^2$. $\bar{\nabla}^2 f_2(x)$ is obviously constant positive definite. $\bar{\nabla}^2 f_1(x)$ is positive definite unless $x_2 = 2$, but in the latter case one can easily show that for all $x_1, y \in \mathbb{R}$

the point (x_1, x_2, y) cannot be a solution of the given PSCMOP. In other words, the assumptions of Corollary 6.13 are satisfied. Moreover, all objectives of both the lower and the higher level problem are convex. Thus it is a good choice to use the PSC-Subdivision or the PSC-Recovering algorithm for the solution of this problem. To state the system of equations which has to be solved in every corrector step of the PSC-Recovering algorithm observe that the optimality conditions for the lower level system (6.54) are as follows:
There exist non-negative multipliers $\alpha_1, \alpha_2 \in \mathbb{R}$ such that

$$\sum_{i=1}^{2} \alpha_i = 1 \tag{6.56}$$

$$\text{and} \quad \sum_{i=1}^{2} \alpha_i \bar{\nabla} f_i(x) = 0. \tag{6.57}$$

From this we formulate the following constraints for the higher level problem (6.53):

$$H(x, \alpha) = \begin{pmatrix} 2\alpha_1(x_1 - 2) + 2\alpha_2(x_1 + 0.5) \\ 4\alpha_1(x_2 - 2)^3 + 2\alpha_2(x_2 + 1) \end{pmatrix} = 0, \tag{6.58}$$

$$\alpha_1 + \alpha_2 - 1 = 0, \tag{6.59}$$

$$\alpha_1 > \varepsilon, \tag{6.60}$$

$$\alpha_2 > \varepsilon. \tag{6.61}$$

As motivated in Section 6.2, ε avoids that x is on the boundary of the solution set of the lower level problem. Thus, according to Remark 6.14, the optimality conditions for the higher level problem under the constraints defined above are:
There exist multipliers $\beta_1, \beta_2 \in \mathbb{R}^+$ and $\lambda_0, \lambda_1, \lambda_2 \in \mathbb{R}$ such that

$$\sum_{i=1}^{2} \beta_i = 1 \tag{6.62}$$

$$\text{and} \quad \sum_{i=1}^{2} \beta_i \nabla F_i(x, y) + \sum_{i=1}^{2} \lambda_i \nabla H_i(x, \alpha) + \lambda_0 \sum_{i=1}^{2} \nabla \alpha_i = 0, \tag{6.63}$$

where ∇ acts on (x, y, α). Calculating the gradients of F and H and substituting them into (6.63), together with the equations above, we get the system which has to be solved in every corrector step:

$$\tilde{F} = \begin{pmatrix} 4\beta_1(x_1-1)^3 + 2\beta_2(x_1+1) + 2\lambda_1(\alpha_1+\alpha_2) \\ 2\beta_1(x_2-1) + 4\beta_2(x_2+1)^3 + \lambda_2(12\alpha_1(x_2-2)^2 + 2\alpha_2) \\ 2\beta_1(y-1) + 2\beta_2(y+1) \\ \lambda_1(2x_1-4) + 4\lambda_2(x_2-2)^3 + \lambda_0 \\ \lambda_1(2x_1+1) + 2\lambda_2(x_2+1) + \lambda_0 \\ 2\alpha_1(x_1-2) + 2\alpha_2(x_1+0.5) \\ 4\alpha_1(x_2-2)^3 + 2\alpha_2(x_2+1) \\ \alpha_1 - s_1^2 - \varepsilon \\ \alpha_2 - s_2^2 - \varepsilon \\ \alpha_1 + \alpha_2 - 1 \\ \beta_1 - t_1^2 \\ \beta_2 - t_2^2 \\ \beta_1 + \beta_2 - 1 \end{pmatrix} = 0. \quad (6.64)$$

The result of the PSC-Recovering algorithm was used to create Figures 22 and 23 as follows. First the algorithm was applied to obtain a tight box covering of the solution. Then test points in parameter space were generated within each box and – similar as in the PSC-Sampling and PSC-SamRec algorithms – a nondominance test was performed in two steps to select representatives of the solution set. In doing so both the nondominated points in parameter space and the corresponding objective values were archived. Thereafter the Recovering algorithm for solving classical MOPs as described in Section 3.1 was applied both to the lower level problem and to the unconstrained higher level problem. Again, nondominance tests were performed on sets of test points, such that we obtained representatives of the corresponding solutions both in parameter and objective space. The representatives of the lower level problem in parameter space were expanded, in a certain sense, to the three-dimensional higher level parameter space. They were used in Figure 22 to depict the *constraint surface* which defines the feasible set for the higher level problem of the PSCMOP. The representatives of the solution to the PSCMOP in parameter space were plotted in the same figure and consequently one can see that this solution is embedded in the constraint surface, as expected. In Figure 23 the representatives in higher level image space of both the unconstrained higher level problem and the PSCMOP were depicted for the purpose of comparison. Obviously, the solution to PSCMOP (which is a constrained problem) can not be better than the solution to the unconstrained higher level problem and is thus located above the latter one.

For alternatively solving the problem with the PSC-Sampling and the PSC-SamRec algorithms we used a combination as follows:

for i **from** 1 **to** 3 **do**

6 **iterations PSC-Sampling with** 300 **test points/box**

1 **iteration PSC-SamRec with** 500 **test points/box**

end

3 **iterations PSC-Sampling with** 500 **test points/box**

The result of this combined algorithm is shown in Figure 24. It should be mentioned that, to obtain a result like this by using the pure PSC-Sampling algorithm, 5000 test points/box were needed. Consequently the computational time for the combined algorithm was about 10 times faster than that for the pure PSC-Sampling algorithm.

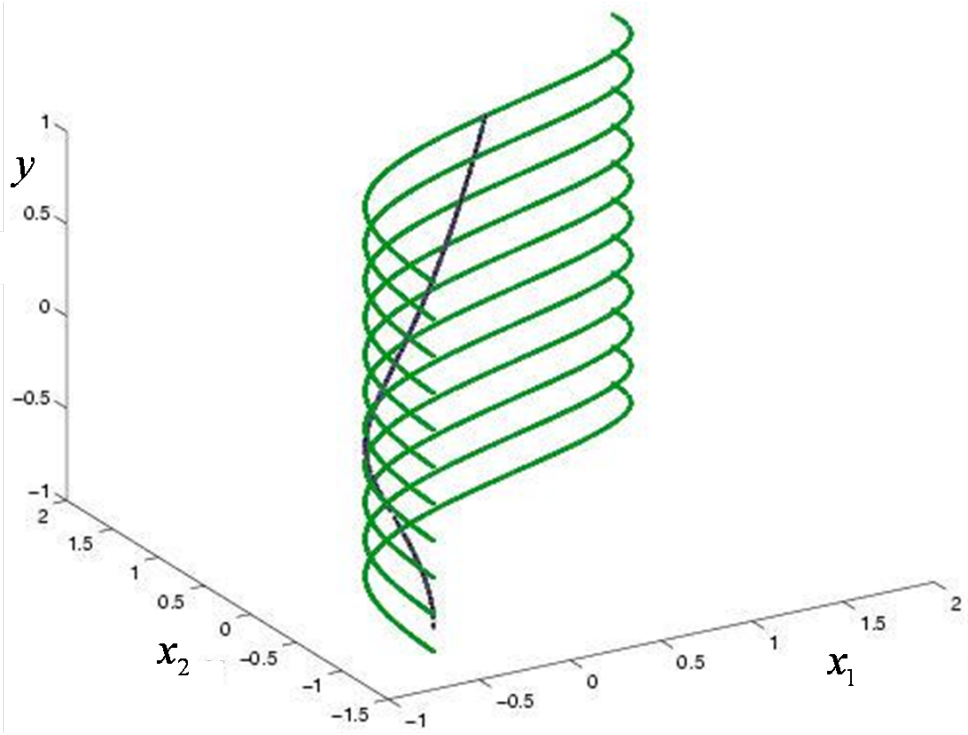

Figure 22: The solution lies within the constraint surface defined by the lower level problem. (the constraint surface was computed separately just for the purpose of visualization, but need not be explicitly computed by our algorithms).

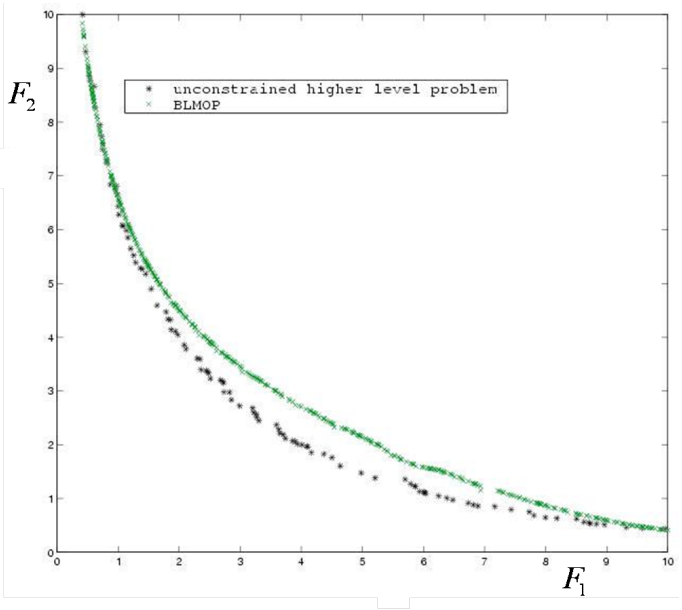

Figure 23: Comparison of the solution of the PSCMOP to the solution of the (unconstrained) higher level problem (higher level objective space).

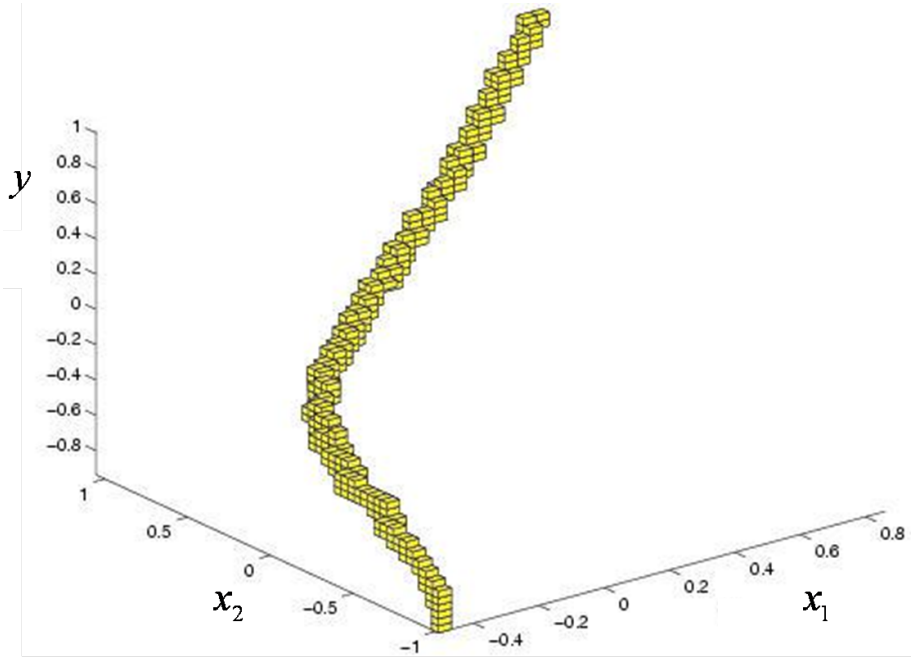

Figure 24: Box covering computed by a combination of the PSC-Sampling and PSC-SamRec algorithms.

EXAMPLE 6.16 Let us now consider the following example with a higher level function $F = (F_1, F_2, F_3)^t : \mathbb{R}^3 \to \mathbb{R}^3$ and a lower level function $f = (f_1, f_2, f_3)^t : \mathbb{R}^3 \to \mathbb{R}^3$:

$$\min_{x \in \mathbb{R}^3} F(x) = \begin{pmatrix} (x_1 - 2)^4 + (x_2 - 2)^2 + (x_3 - 2)^2 \\ (x_1 + 1)^2 + (x_2 + 1)^2 + (x_3 + 1)^4 \\ (x_1 - 1)^2 + (x_2 - 1)^2 + (x_3 + 1)^4 \end{pmatrix}, \tag{6.65}$$

such that x solves:

$$\min_{x \in \mathbb{R}^3} f(x) = \begin{pmatrix} (x_1 - 2)^2 + (x_2 - 2)^4 + (x_3 - 2)^2 \\ (x_1 + 0.5)^2 + (x_2 + 1)^4 + (x_3 - 2)^2 \\ (x_1 + 0.5)^2 + (x_2 - 1)^2 + (x_3 + 2)^4 \end{pmatrix}. \tag{6.66}$$

Here, F does not depend on y, that is, $F(x, y) = F(x)$. Again, all objectives are convex. The solution to this problem was computed using the PSC-Recovering algorithm. The results represented in different spaces are shown in Figures 25 to 28. For a demonstrative representation also the solutions in lower and higher level image space, respectively, are depicted as box collections (Figures 27 and 28). Figure 26 does not only show the solution to the PSCMOP but also – in a different color – the solution to the lower level problem. The latter was computed using the Recovering algorithm designed for solving classical MOPs as described in Section 3.1. One can observe that – in parameter space – the solution to the PSCMOP complies with a part of the solution to the lower level Problem.

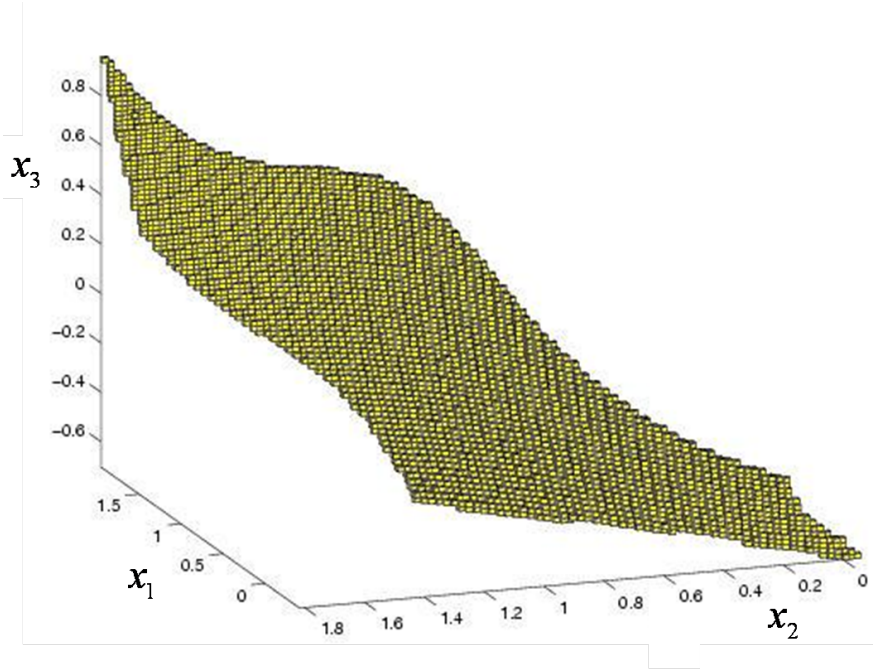

Figure 25: Box covering in parameter space computed by the PSC-Recovering algorithm.

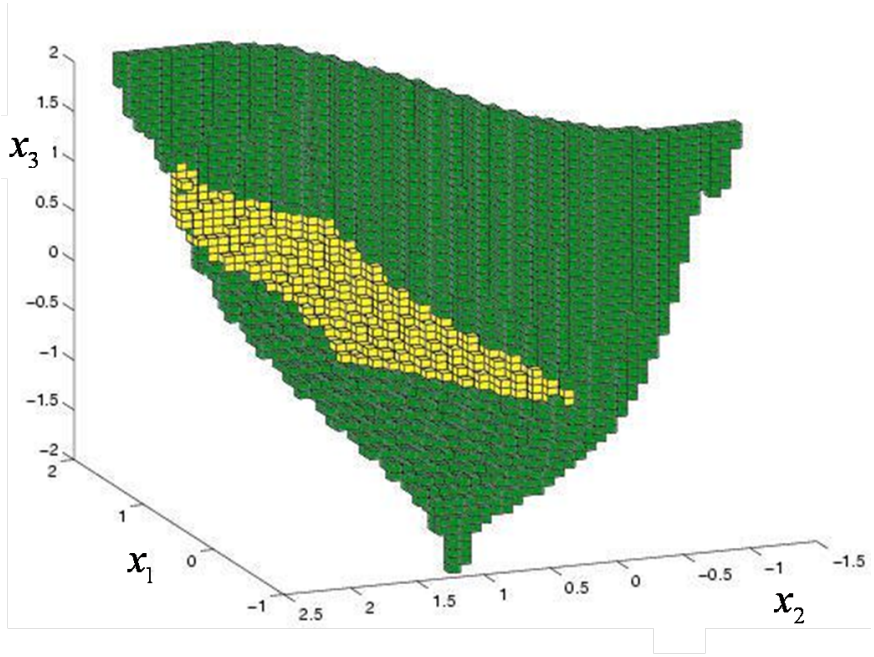

Figure 26: The solution to the PSCMOP – computed by the PSC-Recovering algorithm – is also a part of the solution to the lower level problem – computed by the Recovering algorithm.

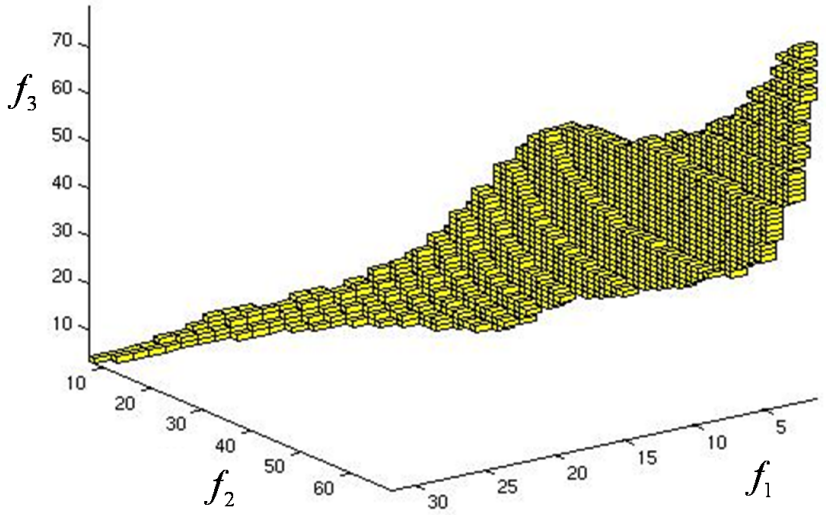

Figure 27: The solution in lower level image space as computed by the PSC-Recovering algorithm.

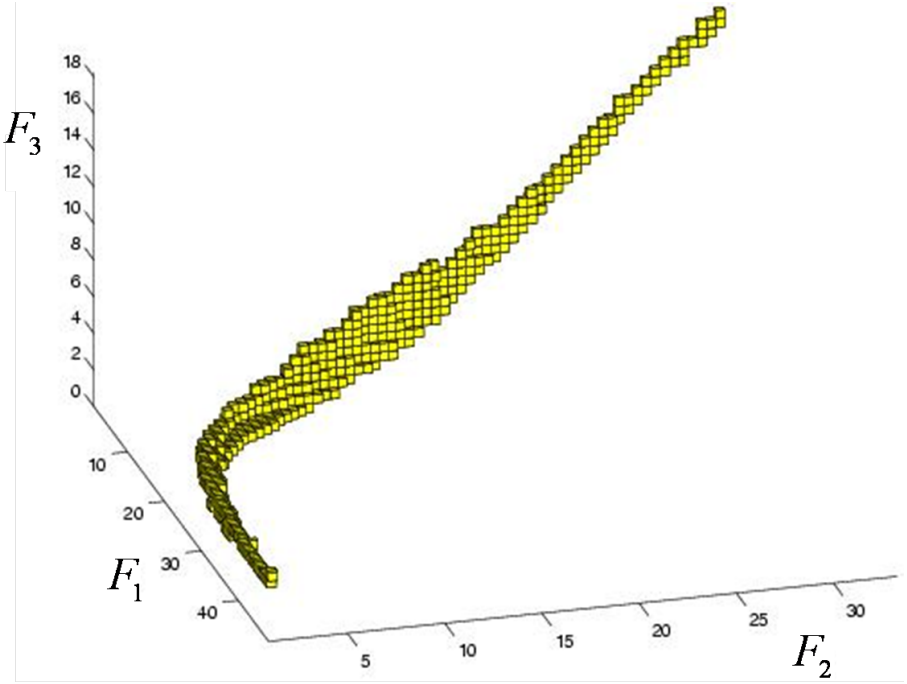

Figure 28: The solution in higher level image space as computed by the PSC-Recovering algorithm.

EXAMPLE 6.17 Application to Electromagnetic Shielding

In the following we deal with the optimization of a new multi-layer shielding material with $N = 3$ layers, which is designed to protect an electronic device against electromagnetic radiation. A similar problem was considered in [40], from where we have taken the mathematical model which describes the relevant physical properties of the material. For a deeper insight to this interesting field of applications the reader is also referred to [40]. We consider a scenario where radiation of frequency $f_1 = 50 \cdot 10^6\, Hz$ could cause light temporal disturbance, whereas radiation of frequency $f_2 = 3.35 \cdot 10^7\, Hz$ would end up in heavy destruction of the electronic device. Consequently, protection against f_2 is much more important than protection against f_1. In addition, due to production tolerances, the obtained degrees of shielding against f_1 and f_2, respectively, are required to be robust against small variations of the layers' thickness. Thus, the shielding material has to be designed such that both protection against f_2 and the corresponding robustness are optimized in a first instance, and the same objectives related to f_1 have to be optimized in a second instance.

For our example we have chosen a 3-layered material, where the middle layer

consists of a particular polymer and the two outer layers consist of polyaniline polyurethane (PAni/PU). We denote by $Z_i, \sigma_i, \epsilon_i$, and d_i $(i = 1, \ldots, 3)$ the impedance, the conductivity, the permittivity, and the thickness of the i-th layer, respectively. Due to the manufacturers possibilities both thickness and conductivity of the two outer layers serve as design variables that is, $x = (d_1, d_3, \sigma_1, \sigma_3)^t$.

For every frequency f the characteristic matrix $M_i^f \in \mathbb{C}^{2\times 2}$ of the i-th layer is given by

$$M_i^f = \begin{pmatrix} \cos(A_i^f) & -jZ_i^f \sin(A_i^f) \\ -\frac{j}{Z_i^f}\sin(A_i^f) & \cos(A_i^f) \end{pmatrix},$$

where

$$A_i^f = \omega d_i \sqrt{\mu_0\epsilon_0\left(\epsilon_i - j\frac{\sigma_i}{\omega\epsilon_0}\right)} \quad \text{and} \quad Z_i^f = \sqrt{\frac{\mu_0}{\epsilon_0\left(\epsilon_i - j\frac{\sigma_i}{\omega\epsilon_0}\right)}},$$

where $\omega = 2\pi f$ and j denotes the imaginary unit. $\mu_0 = 4\pi \cdot 10^{-7}$ and $\epsilon_0 = 8.854 \cdot 10^{-12}$ are the common physical constants. Due to their contact to air media, the impedances of the outer layers are set to $Z_0 = Z_{N+1} = 377$. The characteristic matrix for the entire 3-layered compound is given by

$$M^f = M_1^f M_2^f M_3^f = \begin{pmatrix} M_{11}^f & M_{12}^f \\ M_{21}^f & M_{22}^f \end{pmatrix}.$$

With this notation the transmission coefficient is given by

$$T^f = \frac{2\left(M_{22}^f(M_{11}^f Z_0 - M_{12}^f) + M_{12}^f(M_{22}^f - M_{21}^f Z_0)\right)}{(M_{11}^f Z_0 - M_{12}^f) + Z_0(M_{22}^f - M_{21}^f Z_0)}$$

and the electromagnetic shielding against radiation of frequency f can be expressed by

$$s_f = 20\log(|T^f|).$$

Furthermore, we express the shielding robustness against variations of the layers' thickness by

$$r_f = (\Delta_{d_1} s_f)^2 + (\Delta_{d_2} s_f)^2,$$

where $\Delta_{d_1} s_f$ and $\Delta_{d_2} s_f$ denote finite differences of s_f with respect to d_1 and d_2,

respectively. Finally we are in the position to state the PSCMOP we have to solve:

$$\min_{d_1,d_3,\sigma_1,\sigma_3} \begin{pmatrix} s_{f_1}(d_1,d_3,\sigma_1,\sigma_3) \\ r_{f_1}(d_1,d_3,\sigma_1,\sigma_3) \end{pmatrix} \tag{6.67}$$

such that $(d_1, d_3, \sigma_1, \sigma_3)$ solves:

$$\min_{d_1,d_3,\sigma_1,\sigma_3} \begin{pmatrix} s_{f_2}(d_1,d_3,\sigma_1,\sigma_3) \\ r_{f_2}(d_1,d_3,\sigma_1,\sigma_3) \end{pmatrix} \tag{6.68}$$

Observe that, caused by the trigonometric functions involved in both the higher and lower level objectives, this problem is not convex at all and therefore there might be local solutions which are not necessarily globally Pareto optimal. Moreover, for both the higher level and lower level problem, respectively, points satisfying the Kuhn-Tucker conditions are not necessarily Pareto optimal even from a local point of view. Thus, it was advantageous to use the PSC-Sampling algorithm to solve this problem. After generating the box collections with the PSC-Sampling algorithm, we have generated test points within each generated box. Thereafter, the nondominated points have been selected by a two-step nondominance test. The solution in higher and lower level objective space, respectively, are depicted in Figure 29. Two different projections of the Pareto set in parameter space to representative 3-dimensional subspaces are shown in Figure 30.

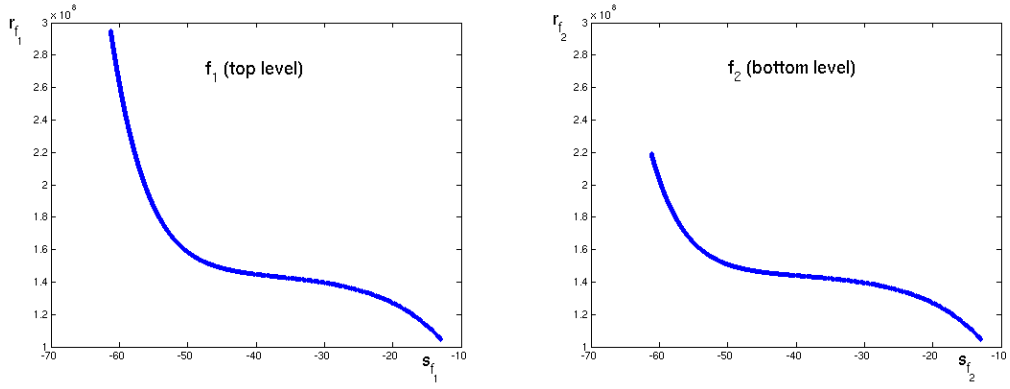

Figure 29: The solution of the shielding problem in higher level (left) and lower level (right) objective space.

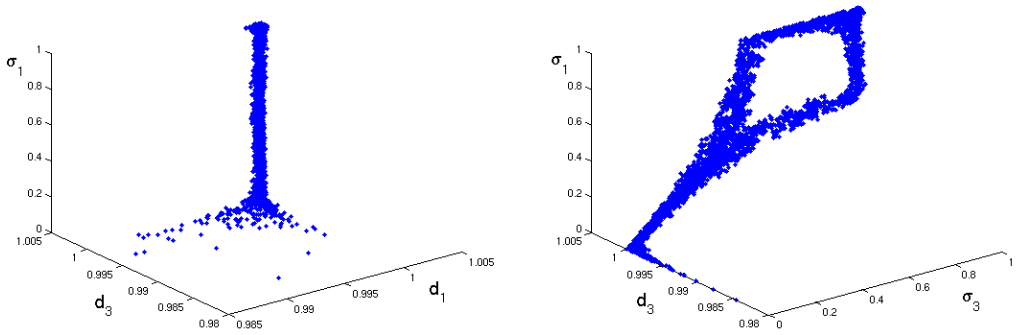

Figure 30: The solution of the shielding problem in parameter space (projections to the (d_1, d_3, σ_1)-subspace (left) and $(\sigma_3, d_3, \sigma_1)$-subspace (right), respectively).

6.5 Methods for Solving BLMOPs with Lower Level Inequality Constraints

As stated in Section 6.1, Theorem 6.1 and consequently also the algorithms based on it are restricted to problems without lower level inequality constraints, because the constraint qualifications which allow to apply the Kuhn-Tucker conditions to the higher level problem can not be satisfied if lower level inequality constraints are involved, see Theorem 6.3. Nevertheless, if there are lower level inequality constraints, we can benefit from Theorem 6.1 when the BLMOP is adequately approximated. To be more precise, for the reasons stated in Section 5.1, instead of solving the given BLMOP directly, we solve successively a sequence of related auxiliary problems. Here, the lower level inequality constraints $g_i(x, y)$, the requirement on the related multipliers τ_i to be non-negative, and the corresponding complementarity relations $\tau_i\, g_i(x, y) = 0$, are replaced by equality constraints constructed by the use of the perturbed Fischer-Burmeister function Φ described in Section 5.1. The resulting equations have the form

$$\Phi_i(\tau_i, -g_i(x, y), \varepsilon) = \tau_i - g_i(x, y) - \sqrt{\tau_i^2 + g_i(x, y)^2 + \varepsilon}, \quad i = 1, \ldots, q. \tag{6.69}$$

Now the series of auxiliary problems has to be defined such that the smoothing parameter ε is positive at the beginning, decreases with every auxiliary problem and finally converges towards 0. Recall that $\Phi_i(\tau_i, -g_i(x, y), \varepsilon)$ is smooth for every $\varepsilon > 0$ and that $\Phi_i(\tau_i, -g_i(x, y), 0) = 0$ if and only if $g_i(x, y) \leq 0$, $\tau_i \geq 0$ and $\tau_i\, g_i(x, y) = 0$. Consequently, the auxiliary problem associated with $\varepsilon = 0$ complies with the original problem, that is, the given BLMOP is approximated by a sequence

of smooth problems without lower level inequality constraints which can be solved based on Theorem 6.1. This strategy can be used to modify the algorithms of Section 6.2 such that they are capable of solving BLMOP with lower level inequality constraints.

BL2-Recovering-IS algorithm

As an example, in the following we describe the *BL2-Recovering-IS algorithm*, which is a modified version of the BL-Recovering-IS algorithm. For this, we adopt the notations of Section 6.2, expand $\tilde{z}$ by $(\xi, \tau) \in \mathbb{R}^{2q}$, and denote by $\tilde{F}(\tilde{z}; t, \varepsilon)$ the variant of $\tilde{F}(\tilde{z}; t)$, which includes the expressions (6.69) associated with the lower level inequality constraints $g_i, i = 1, \ldots, q$, that is, with

$$\mathbf{\Phi}(x, y, \tau, \varepsilon) := (\Phi_1(\tau_1, -g_1(x, y), \varepsilon), \ldots, \Phi_q(\tau_q, -g_q(x, y), \varepsilon))^t$$

and

$$\mathcal{L}(x, y, \alpha, \zeta, \tau) := \sum_{i=1}^{l} \alpha_i \, f_i(x, y) + \sum_{i=1}^{p} \zeta_i \, h_i(x, y) + \sum_{i=1}^{q} \tau_i \, g_i(x, y),$$

we obtain

$$\tilde{F}(\tilde{z}; t, \varepsilon) = \begin{pmatrix} \nabla ||F(x, y) - t|| + \sum_{i=1}^{n} \lambda_i \nabla \mathcal{L}_{x_i}(x, y, \alpha, \zeta, \tau) + \sum_{i=1}^{l} (\nu - \mu_i) \, e_{n+m+i} \\ + \sum_{i=1}^{p} \omega_i \nabla h_i(x, y) + \sum_{i=1}^{q} \xi_i \mathbf{\Phi}_i(x, y, \tau, \varepsilon) + \sum_{i=1}^{r} \rho_i \nabla H_i(x, y) \\ + \sum_{i=1}^{s} \delta_i \nabla G_i(x, y) \\ h(x, y) \\ \mathbf{\Phi}(x, y, \tau, \varepsilon) \\ H(x, y) \\ G(x, y) + w \circ w \\ \bar{\nabla} \mathcal{L}(x, y, \alpha, \zeta, \tau) \\ \sum_{i=1}^{l} \alpha_i - 1 \\ \mu \circ \alpha \\ \delta \circ G(x, y) \\ \alpha - s \circ s \\ \mu - u \circ u \\ \delta - v \circ v \end{pmatrix} = \mathbf{0}.$$

Now, for a fixed subdivision depth d, with $\varepsilon_0 > 0$ and the notations above, an iteration of the BL2-Recovering-IS algorithm, which can also handle lower level inequality constraints, can be written as follows:

(i) for all $B \in \mathcal{B}_j$

$B.active := TRUE$

(ii) for $j = 1, \ldots, MaxStep$

$\hat{\mathcal{B}}_j := \mathcal{B}_j$
for all $B \in \{B \in \mathcal{B}_j : B.active == TRUE\}$
choose target vectors $\{t_i\}_{i=1,\ldots,n_t}$ near B with $t_i <_p F_B$
for all $i = 1, \ldots, n_t$
$\varepsilon := \varepsilon_0$
$\tilde{z} := \tilde{z}_B$
while termination criterion is not satisfied do
$\varepsilon := c\,\varepsilon$ for some $c \in (0, 1)$
$\tilde{z}_0 := \tilde{z}$
starting from $\tilde{z}_0$ **find** $\tilde{z}$ **with** $\tilde{F}(\tilde{z}; t_i, \varepsilon) = 0$
$\tilde{z}_i^\star := \tilde{z}$
$F_i^\star := F(\Pi_{(x,y)}(\tilde{z}_i^\star)), i = 1, \ldots, n_t$
$B.active := FALSE$
for all $i = 1, \ldots, n_t$:
if $B(F_i^\star, d) \notin \hat{\mathcal{B}}_j$
$\check{B} := B(F_i^\star, d),\ \tilde{z}_{\check{B}} := \tilde{z}_i^\star,\ F_{\check{B}} := F_i^\star$
$\check{B}.active := TRUE$
$\hat{\mathcal{B}}_j := \hat{\mathcal{B}}_j \cup \check{B}$
if $\hat{\mathcal{B}}_j == \mathcal{B}_j$ **STOP**
$\mathcal{B}_{j+1} := \hat{\mathcal{B}}_j$

There are several possibilities for the definition of a suitable termination criterion. For example, one can use the simple criterion, that the variation of two successive vectors of higher and lower level objectives has to be sufficiently small. For this, let $\tilde{z}_\varepsilon$ be the solution of $\tilde{F}(\tilde{z}; t_i, \varepsilon) = 0$ and $(x_\varepsilon, y_\varepsilon) := \Pi_{(x,y)}(\tilde{z}_\varepsilon)$ and denote

$$\mathcal{F}_\varepsilon := (F(x_\varepsilon, y_\varepsilon), f(x_\varepsilon, y_\varepsilon))^t \in \mathbb{R}^{k+l}.$$

Then the termination criterion is given by the requirement

$$||\mathcal{F}_\varepsilon - \mathcal{F}_{c\varepsilon}||_2 < \varepsilon_{stop}$$

for some small $\varepsilon_{stop} > 0$. In some applications, such a termination criterion might be satisfactory. But since $\Phi_i(\tau_i, -g_i(x,y), \varepsilon) = 0$ is equivalent to the original constraints $g_i(x,y) \leq 0, \tau_i \geq 0$ and $\tau_i\, g_i(x,y) = 0$ only for the case $\varepsilon = 0$, it is worth to include these original constraints into the termination criterion as follows:

$$\begin{aligned} &||\mathcal{F}_\varepsilon - \mathcal{F}_{c\varepsilon}||_2 \;+\; ||\tau_\varepsilon \circ g(x_\varepsilon, y_\varepsilon)||_2 \\ &+ ||\max\{0, g(x_\varepsilon, y_\varepsilon)\}||_2 \;+\; ||\max\{0, -\tau_\varepsilon\}||_2 \;\;<\;\; \varepsilon_{stop}. \end{aligned}$$

Here, max is understood to act component-wise and $\tau_\varepsilon := \Pi_\tau(\tilde{z}_\varepsilon)$.

Example 6.18 We consider now an example which was taken from [16]. In this example there are explicitly defined inequality constraints $G_i : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}, i = 1, 2$ and $g_i : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}, i = 1, \ldots, 4$ for the higher and lower level problem, respectively. With these constraints, a higher level function $F = (F_1, F_2)^t : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^2$ and, a lower level function $f = (f_1, f_2)^t : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^2$, the BLMOP is given as follows:

$$\min_{x \in \mathbb{R}^2, y \in \mathbb{R}} F(x,y) = \begin{pmatrix} x_1 + x_2^2 + y + \sin^2(x_1 + y) \\ \cos(x_2) \cdot (0.1 + y) \cdot \exp(-\frac{x_1}{0.1 + x_2}) \end{pmatrix}, \tag{6.70}$$

such that

$$\begin{aligned} G_1(x,y) &= -y \leq 0, \\ G_2(x,y) &= y \leq 10, \end{aligned}$$

and x solves:

$$\min_{x \in \mathbb{R}^2} f(x,y) = \begin{pmatrix} \frac{(x_1-2)^2 + (x_2-1)^2}{4} + \frac{x_2 y + (5-y)^2}{16} + \sin(\frac{x_2}{10}) \\ \frac{x_1^2 + (x_2-6)^4 - 2x_1 y - (5-y)^2}{80} \end{pmatrix}, \tag{6.71}$$

such that

$$\begin{aligned} g_1(x,y) &= x_1^2 - x_2 \leq 0, \\ g_2(x,y) &= 5x_1^2 + x_2 - 10 \leq 0, \\ g_3(x,y) &= x_2 + \tfrac{y}{6} - 5 \leq 0, \\ g_4(x,y) &= -x_1 \leq 0. \end{aligned}$$

Of course, in (6.71) there is the non-convex term $s(x_2) := \sin(\frac{x_2}{10})$. But the explicitly given constraints on both levels guarantee that $x_2 \in [0, 5]$ and obviously the restriction of s to the interval $[0, 5]$ is convex. Consequently, since all other lower level terms are also convex for fixed y, the entire lower level problem (6.71) is convex. The

problem was solved with the BL2-Recovering-IS algorithm using a few initial guesses and initial targets. As a result, several local Pareto sets were computed, which, as shown in Figure 31, intersect each other in image space. Therefore, not all computed solutions are global solutions of the BLMOP. Nevertheless, since all lower level problems are convex, every local solution $(x^\star, y^\star)$ corresponds to a global solution $x^\star$ of the lower level problem associated with the parameter $y^\star$. Consequently, the global solution of the BLMOP can be selected from the computed points by a nondominance test with respect to the higher level. In Figure 31, also the result of this nondominance test is marked.

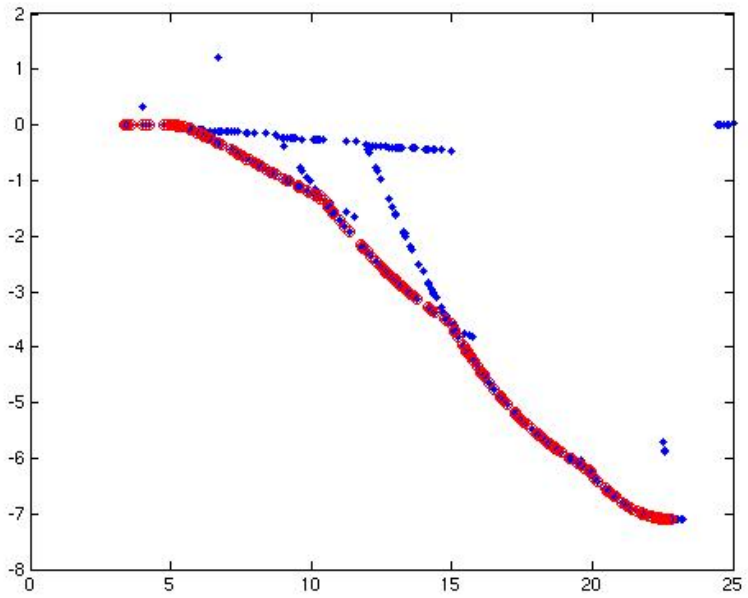

Figure 31: The solutions of Problem (6.70,6.71) as computed by the BL-Recovering-IS algorithm are locally but not necessarily globally Pareto optimal.

To show the behavior of the smoothing strategy, we have once more computed a small part of the solution while achieving all intermediate solutions $\tilde{z}_\varepsilon$ and the corresponding higher and lower level objective values. The resulting sequences for the considered part of the Pareto set in parameter space and lower and higher level image space are shown in Figures 32, 33 and 34, respectively. Here, the bigger points mark the final points of the sequences, that is, they are substationary points of the given BLMOP.

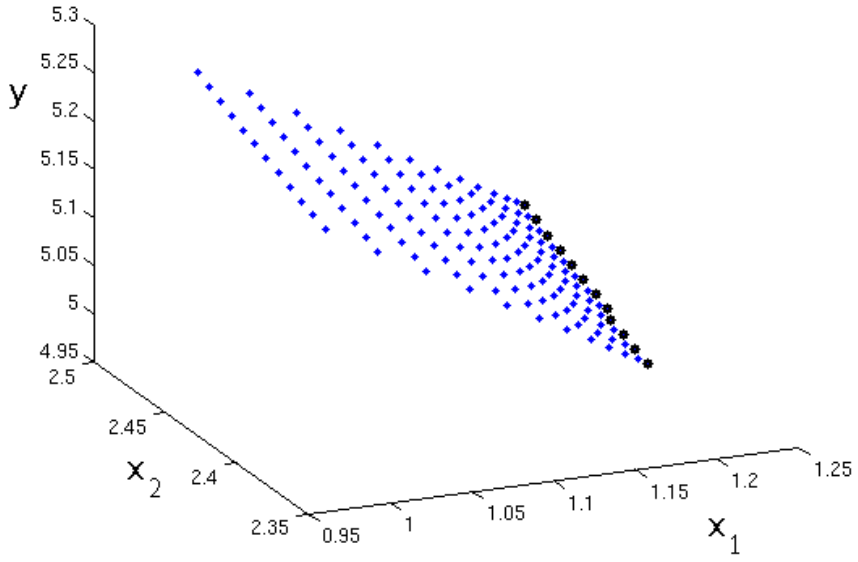

Figure 32: Convergence of the smoothing method (parameter space)

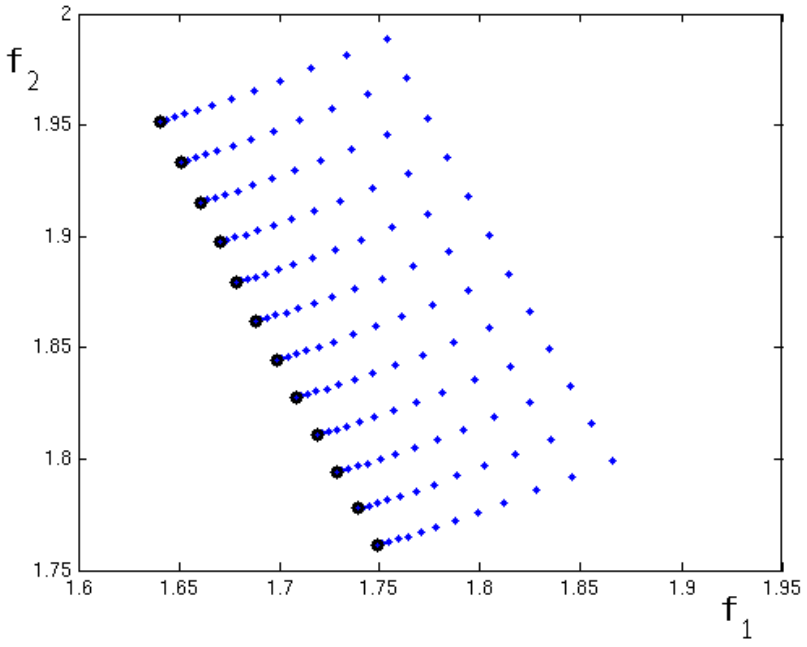

Figure 33: Convergence of the smoothing method (lower level objective space)

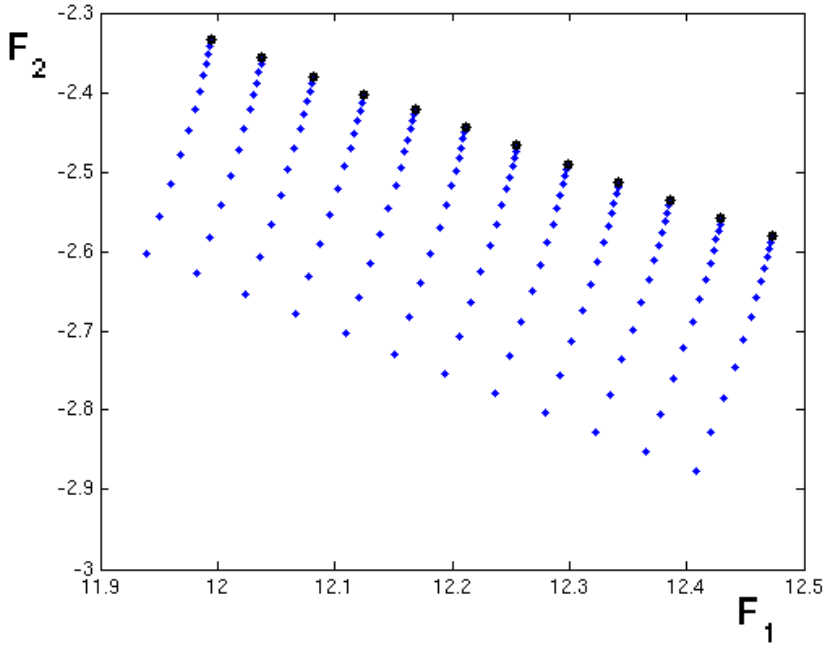

Figure 34: Convergence of the smoothing method (higher level objective space)

Example 6.19 Application to Medical Engineering
This example is also taken from [16] and deals with a real world application appearing in medical engineering, where the task is to find an optimal configuration of coils. For a deeper insight to this applications we refer to [16] and the references therein. The resulting BLMOP is given as follows:

$$\min_{x\in\mathbb{R}^{14},y\in\mathbb{R}} \begin{pmatrix} ||x||_2^2 \\ ||x - x_{old}||_2^2 \end{pmatrix}, \tag{6.72}$$

$$\text{such that } y \in [0,\pi],$$

$$\text{and } x \text{ solves:}$$

$$\min_{x\in\mathbb{R}^{14}} \begin{pmatrix} ||A(y)\cdot Vx - b(y)||_2^2 \\ ||x||_2^2 \end{pmatrix}, \tag{6.73}$$

$$\text{such that } ||A(y)\cdot Vx - b(y)||_2^2 \leq \Delta_{max}$$

with

$$A(y) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cos y & \sin y & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sin y & \cos y \\ 0 & 0 & 0 & -\sin y & 0 & 0 & \cos y & -\sin y \end{pmatrix},$$

$b(y) = (0, \cos y, \sin y, 1, 0, 0)^t$ and $\Delta_{max} = 0.3$.
For $x_{old} \in \mathbb{R}^{14}$ we choose

$$x_{old} = \begin{matrix} (0.1247, & 0.1335, & -0.0762, & -0.1690, & 0.2118, & -0.0534, & -0.1473, \\ 0.3170, & -0.0185, & -0.1800, & 0.1700, & -0.0718, & 0.0058, & 0.0985)^t. \end{matrix}$$

The matrix $V \in \mathbb{R}^{8\times 14}$ is a non-sparse matrix with $\operatorname{rank}(V) = 8$ and depends on the special medical therapy to be administrated. For our calculation we have taken the same (randomly chosen) matrix $V = (V_1|V_2)$ as in [16] with

$$V_1 = \begin{pmatrix} 0.9501 & 0.8214 & 0.9355 & 0.1389 & 0.4451 & 0.8381 & 0.3046 \\ 0.2311 & 0.4447 & 0.9169 & 0.2028 & 0.9318 & 0.0196 & 0.1897 \\ 0.6068 & 0.6154 & 0.4103 & 0.1987 & 0.4660 & 0.6813 & 0.1934 \\ 0.4860 & 0.7919 & 0.8936 & 0.6038 & 0.4186 & 0.3795 & 0.6822 \\ 0.8913 & 0.9218 & 0.0579 & 0.2722 & 0.8462 & 0.8318 & 0.3028 \\ 0.7621 & 0.7382 & 0.3529 & 0.1988 & 0.5252 & 0.5028 & 0.5417 \\ 0.4565 & 0.1763 & 0.8132 & 0.0153 & 0.2026 & 0.7095 & 0.1509 \\ 0.0185 & 0.4057 & 0.0099 & 0.7468 & 0.6721 & 0.4289 & 0.6979 \end{pmatrix}$$

and

$$V_2 = \begin{pmatrix} 0.3784 & 0.8180 & 0.8385 & 0.7948 & 0.8757 & 0.2844 & 0.4329 \\ 0.8600 & 0.6602 & 0.5681 & 0.9568 & 0.7373 & 0.4692 & 0.2259 \\ 0.8537 & 0.3420 & 0.3704 & 0.5226 & 0.1365 & 0.0648 & 0.5798 \\ 0.5936 & 0.2897 & 0.7027 & 0.8801 & 0.0118 & 0.9883 & 0.7604 \\ 0.4966 & 0.3412 & 0.5466 & 0.1730 & 0.8939 & 0.5828 & 0.5298 \\ 0.8998 & 0.5341 & 0.4449 & 0.9797 & 0.1991 & 0.4235 & 0.6405 \\ 0.8216 & 0.7271 & 0.6946 & 0.2714 & 0.2987 & 0.5155 & 0.2091 \\ 0.6449 & 0.3093 & 0.6213 & 0.2523 & 0.6614 & 0.3340 & 0.3798 \end{pmatrix}.$$

Observe that for every fixed $y \in [0, \pi]$, $A(y) \cdot Vx - b(y)$ is linear with respect to x. Consequently, the lower level problem (6.73) is a convex one for every fixed $y \in [0, \pi]$.

The solution of this problem was computed with a variant of the BL2-Recovering-IS algorithm, which is based on the minimization of the distance between the vector of higher level objectives and the targets under the restriction that only those components of $\tilde{F}$ vanish which guarantee that the optimality conditions for the lower

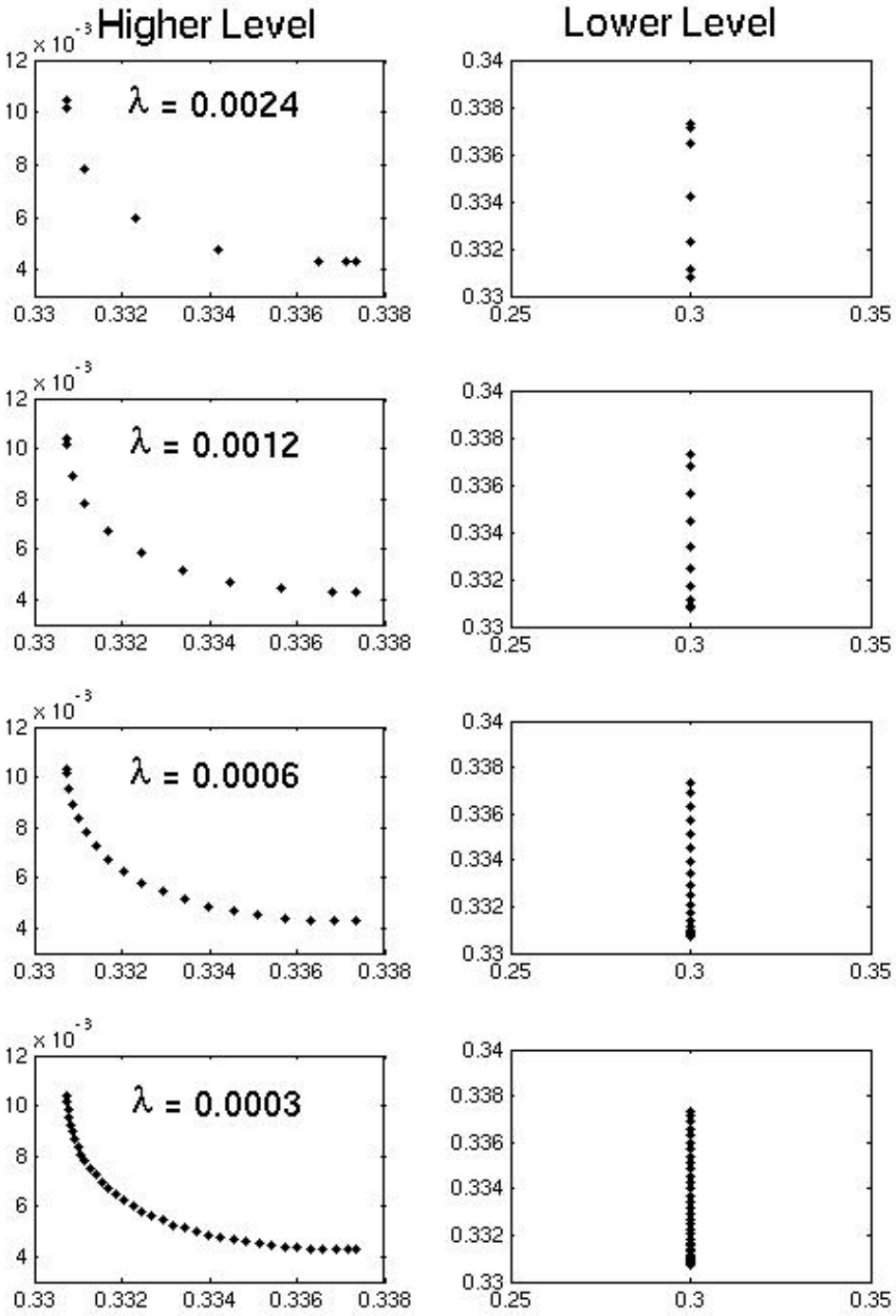

Figure 35: The Pareto set of Problem (6.72, 6.73) in higher level image space (left) and lower level image space (right) for different parameters λ.

level problem hold and that the higher level constraints of the BLMOP are fulfilled. To be more precise, with $\hat{z} := (x, y, \alpha, \zeta, \tau, \delta, w, s)$ and

$$\hat{F}(\hat{z}; \varepsilon) := \begin{pmatrix} h(x,y) \\ H(x,y) \\ G(x,y) + w \circ w \\ \Phi(x,y,\tau,\varepsilon) \\ \bar{\nabla}\mathcal{L}(x,y,\alpha,\zeta,\tau) \\ \sum_{i=1}^{l} \alpha_i - 1 \\ \alpha - s \circ s \end{pmatrix},$$

in every step we have to solve

$$\min_{\hat{z} \in S} ||F(\Pi_{(x,y)}(\hat{z})) - t_i||, \tag{6.74}$$

where $S := \{\hat{z} : \hat{F}(\hat{z}; \varepsilon) = 0\}$ and $t_i, i = 1, \ldots, n_t$ denote the targets which have to be chosen individually in every cycle of the algorithm. This variant works without boxes $B \subset \mathbb{R}^k$. Instead, the individual targets and consequently the Pareto points are successively computed using the targets $t^\star$ and the solutions $\hat{z}^\star$ from the respective previous step, that is, we solve (6.74) for a target

$$t = F^\star + \lambda b + \lambda_0 (t^\star - F^\star),$$

where b is a basis vector of the one-dimensional tangent space $T_{F^\star} F(P)$ of the Pareto set $F(P)$ in higher level image space at $F^\star := F(\Pi_{(x,y)}(\hat{z}^\star))$. Here, λ and λ_0 make up a parameterization for the new target t. To demonstrate the advantage of these methods concerning the user's influence on the diversity and the granularity, we have computed different solutions corresponding to different fixed values for the parameter λ, see Figure 35. While computing these solutions it turned out that in this particular application the simple choice $\lambda_0 = 0$ leads always to suitable targets, that is, we have always $t <_p F(\Pi_{(x,y)}(\bar{z}^\star))$, where $\bar{z}^\star$ is the respective solution of (6.74). As can be observed on the right hand side of Figure 35, the lower level inequality constraint $||A(y) \cdot Vx - b(y)||_2^2 \leq \Delta_{max} = 0.3$, which is related to the first lower level objective, see (6.73), is active for the entire solution of this problem.

6.6 Non-Convex and Non-Smooth BLMOPs

Due to the conditions given by many real world applications, in this section we assume that for the objectives and constraints of both the higher and lower level

either no derivatives are available at all or the computation of derivatives is too time consuming. Moreover, we abandon the lower level convexity assumption which was – as well as the availability of derivatives – essential for the application of the algorithms based on Theorem 6.1.

BL2-Subdivision Algorithm

Motivated by the facts mentioned above, we want to describe an algorithm which belongs to the family of subdivision algorithms and is tailored to solve non-convex and non-smooth BLMOPs. A naive way to handle such problems would be to solve for 'sufficiently many' parameters $y \in \mathbb{R}^m$ the lower level problem, collecting the respective solutions in a common archive, and then performing at first a feasibility test and thereafter a nondominance test, both with respect to the higher level problem, among all collected solutions. But there is the question how to choose 'sufficiently many' parameters $y \in \mathbb{R}^m$ in order to obtain a representative approximation of the y-space while keeping the computational effort justifiable. The following set-oriented algorithm finds a way out by the use of a subdivision technique. To be more precise, the algorithm works generally with boxes in $\mathbb{R}^{n+m}$, but for the representation of the y-space and the x-space, respectively, also boxes of respective lower dimension are used, which can be regarded as projections of boxes in $\mathbb{R}^{n+m}$ to the space of boxes in $\mathbb{R}^n$ or $\mathbb{R}^m$, respectively. The subdivision depth is increased adaptively, that is, the size of the boxes shrinks, such that both the number of points $y_i \in \mathbb{R}^m$ representing the y-space and the number of generated Pareto points of the associated lower level problems is increased adaptively as the algorithm proceeds.

For a detailed description of the *BL2-Subdivision algorithm*, denote for every box collection $\mathcal{B} \subset \mathbb{R}^{n+m}$ by $\Pi_y(\mathcal{B}) \subset \mathbb{R}^m$ the 'box collection-valued projection' of $\mathcal{B}$ to the space of boxes in $\mathbb{R}^m$. Moreover, for every fixed $y \in \mathbb{R}^m$ and $\delta \in \mathbb{N}$, denote by $BLS(y, \delta)$ a box collection in $\mathbb{R}^n$ with $\operatorname{diam}(BLS(y, \delta)) = \delta$ covering the solution of the lower level problem associated with y, e.g., a box collection generated by a combination of the classical SamRec and Sampling algorithms.

With these notations, starting with an initial box $Q = \mathcal{B}_0 \subset \mathbb{R}^{n+m}$, an iteration of the BL2-Subdivision algorithm reads as follows:

(i) Subdivision

Construct from $\mathcal{C}_{j-1} := \Pi_y(\mathcal{B}_{j-1})$ a new system $\hat{\mathcal{C}}_j$ of subsets such that

$$\bigcup_{C \in \hat{\mathcal{C}}_j} C = \bigcup_{C \in \mathcal{C}_{j-1}} C$$

and

$$\delta_j := \operatorname{diam}(\hat{\mathcal{C}}_j) = \theta_j \operatorname{diam}(\mathcal{C}_{j-1}),$$

where $0 < \theta_{min} \leq \theta_j \leq \theta_{max} < 1$.

(ii) Selection

$\hat{\mathcal{B}}_j := \emptyset$
for all $C \in \hat{\mathcal{C}}_j$

choose a set of **test points** $\mathcal{X}_C \subset C$
$\hat{\mathcal{B}}_j := \hat{\mathcal{B}}_j \bigcup\limits_{y \in \mathcal{X}_C} (BLS(y, \delta_j) \times C)$

for all $B \in \hat{\mathcal{B}}_j$

choose a set of **test points** $\mathcal{X}_B \subset B$

$\mathcal{X} := \bigcup\limits_{B \in \hat{\mathcal{B}}_j} \mathcal{X}_B$
$N_F := F_{(G,H)}$-nondominated points of $\mathcal{X}$
$\mathcal{B}_j := \left\{ B \in \hat{\mathcal{B}}_j : \mathcal{X}_B \cap N_F \neq \emptyset \right\}$

As in all our algorithms of subdivision type, a termination criterion for the BL2-Subdivision algorithm can be defined by the condition $\operatorname{diam}(\mathcal{B}_j) < \varepsilon$ for some small $\varepsilon > 0$, which is equivalent with the termination after a number of iterations.

Observe that at the end of each iteration the number of boxes $B \in \mathcal{B}_j$ is minimized by the use of a feasibility and nondominance test. Thus, the refinement of the y-space in the next iteration is restricted to those regions where solutions of the BLMOP are expected to be. Both, the adaptive refinement itself and the restriction of the respective refinements to the relevant regions help to keep the computational effort low.

Let us make a remark concerning the choice of test points for the higher level nondominance test. If the method for the solution of the lower level problems produces for every y a representative number of $f_{(g,h)}$-nondominated points, these points can be collected in an archive, which serves as a basis for the choice of the set $\mathcal{X}$ of testpoints, the higher level nondominance test has to be performed on. In this way, the computational effort for the construction of $\mathcal{X}$, which is particularly time consuming in the high-dimensional case, can be reduced to a minimum.

EXAMPLE 6.20 Given a higher level function $F = (F_1, F_2)^t : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^2$, a lower level function $f = (f_1, f_2)^t : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^2$, and a lower level inequality constraint

function $g = (g_1, g_2)^t : \mathbb{R}^2 \to \mathbb{R}^2$, we consider the following BLMOP:

$$\min_{x \in \mathbb{R}^2,\, y \in \mathbb{R}} F(x,y) = \begin{pmatrix} x_1^4 + x_2^2 + (y - 6\pi)^2 \\ x_1^2 + (x_2 - 6\pi)^2 + y^4 \end{pmatrix}, \tag{6.75}$$

$$\begin{aligned} &\text{such that } x \text{ solves:} \\ &\min_{x \in \mathbb{R}^2} f(x,y) = \begin{pmatrix} \sin x_1 + \sin(x_2 - y) \\ \cos(x_1 - y) + \cos x_2 \end{pmatrix}, \\ &\text{such that } g(x) = \begin{pmatrix} 0.2 - x_1^2 + x_2^2 \\ x_1^2 + x_2^2 - 5 \end{pmatrix} \leq 0, \end{aligned} \tag{6.76}$$

where $\leq$ has to be understood component-wise.

The solution of this problem in parameter space, as computed by the BL2-Subdivision algorithm, is shown in Figure 36. For comparison reasons, we have also computed the union of the lower level solutions associated with an extensive set of parameters $y \in \mathbb{R}^m$ using the BL-Subdivision algorithm. This union of lower level solutions makes up the feasible set for the higher level problem and is also depicted in Figure 36. Observe, that the solution of the BLMOP complies with just a small part of the union. In Figure 37 the solution of the above BLMOP in higher level objective space is shown.

7 Sensitivity Analysis

In this section we present results concerning the sensitivity of a BLMOP, that is, we are interested in the behavior of the solution when the BLMOP is disturbed due to additional parameters. To be more precise, we derive estimations for the distance of a particular Pareto point of the original problem to the Pareto sets of the problems arising when the perturbation parameters vary within a certain range. Such information can help the decision maker to choose a particular solution out of the computed Pareto set for the adjustment of the system under consideration, particularly, if this system is affected by uncertainties. Moreover, as we will see in Section 7.3, the derived estimations can be used to influence the representation of the Pareto set computed by the algorithms described in the previous sections.

7.1 Sensitivity Analysis for Classical Optimization Problems

Before we can derive the sensitivity results for MOP and BLMOP we recall the basic sensitivity theorem stated in [17], which is concerned with (scalar valued) parametric

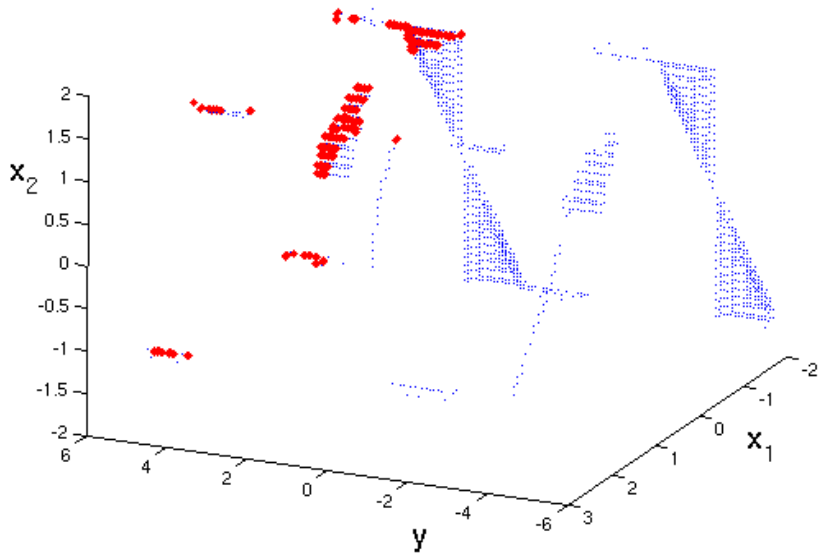

Figure 36: The solution of the non-convex BLMOP in parameter space.

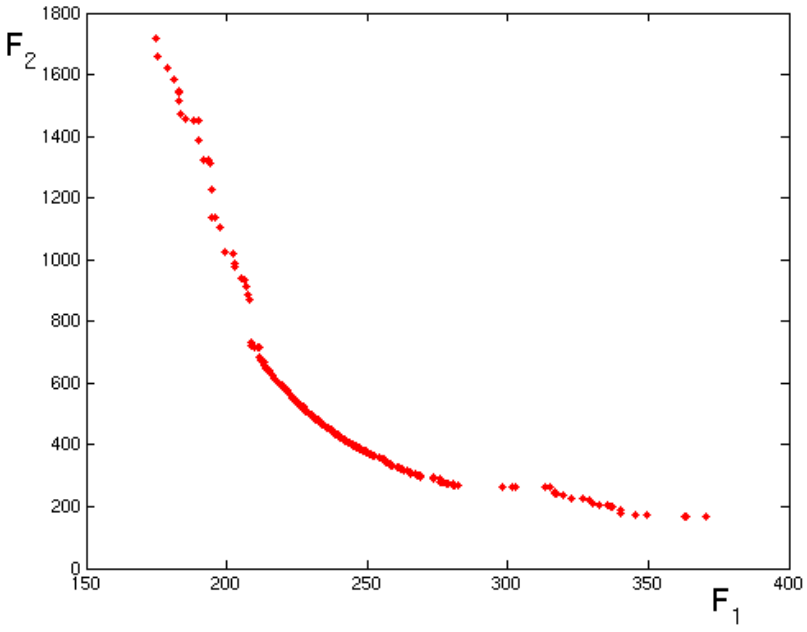

Figure 37: The solution of the non-convex BLMOP in higher level objective space.

optimization problems of the form

$$\begin{aligned} \min_x \; & F(x,\epsilon) \\ \text{s.t.} \quad & G_i(x,\epsilon) \le 0, \quad i = 1,\dots,q, \\ & H_j(x,\epsilon) = 0, \quad j = 1,\dots,p, \end{aligned} \tag{P(ϵ)}$$

where $\epsilon \in \mathbb{R}^m$ is a (fixed) perturbation parameter vector and all functions $F, G_i, H_j : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ are twice continuously differentiable with respect to x. For the remainder of this section let $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^q \times \mathbb{R}^p \times \mathbb{R}^m \to \mathbb{R}$, defined by

$$\mathcal{L}(x,\mu,\lambda,\epsilon) := F(x,\epsilon) + \sum_{i=1}^{q} \mu_i G_i(x,\epsilon) + \sum_{j=1}^{p} \lambda_j H_j(x,\epsilon),$$

be the Lagrangian of P(ϵ). The gradient $\nabla F(x,\epsilon)$ of a function $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is understood to be a row vector and the Hessian of F is denoted by $\nabla^2 F(x,\epsilon)$. Accordingly, the $i-$th row of the Jacobian $\nabla G(x,\epsilon)$ of a vector valued function $G : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^k$ is given by the gradient $\nabla G_i(x,\epsilon)$ of the $i-$th component of G. Moreover, denote $\bar{\nabla} := \nabla_{(x)}$, $\nabla := \nabla_{(\epsilon)}$, and let $I_G(x,\epsilon) := \{i : G_i(x,\epsilon) = 0\}$.

THEOREM 7.1 *Let $x^\star \in \mathbb{R}^n$ be a solution of* P(0)*, such that the following conditions hold at $x^\star$:*

(i) The gradients $\bar{\nabla} F(x,\epsilon)$, $\bar{\nabla} G_i(x,\epsilon)$, $\bar{\nabla} H_j(x,\epsilon)$ and the constraints $G_i(x,\epsilon)$, $H_j(x,\epsilon)$, $i = 1,\dots,q$, $j = 1,\dots,p$, are continuously differentiable with respect to ϵ in a neighborhood of $(x^\star, 0)$.

(ii) The gradients $\bar{\nabla} G_i(x^\star,0), i \in I_G(x^\star,0), \bar{\nabla} H_j(x^\star,0), j = 1,\dots,p$, are linearly independent.

(iii) There are $\mu_i^\star \ge 0, i = 1,\dots,q$ and $\lambda_j^\star \in \mathbb{R}, j = 1,\dots,p$, such that

$$\bar{\nabla}\mathcal{L}(x^\star,\mu^\star,\lambda^\star,0) = 0, \tag{7.1}$$

and

$$\mu_i \; G_i(x^\star,0) = 0, \quad i = 1,\dots,q. \tag{7.2}$$

(iv)

$$z \; \bar{\nabla}^2\mathcal{L}(x^\star,\mu^\star,\lambda^\star,0) \; z > 0 \tag{7.3}$$

for all $z \neq 0$ *with*

$$\begin{aligned} \bar{\nabla} F(x^\star, 0) z &= 0, \\ \bar{\nabla} G_i(x^\star, 0) z &\leq 0, \quad i \in I_G(x^\star, 0), \\ \bar{\nabla} H_j(x^\star, 0) z &= 0, \quad j = 1, \ldots, p. \end{aligned}$$

(v)

$$\mu_i > 0, \quad i \in I_G(x^\star, 0). \tag{7.4}$$

Then

(a) $x^\star$ *is an isolated local solution of* P(0) *and the multipliers* $\mu_i^\star, \lambda_j^\star$ *are unique,*

(b) for ϵ *in a neighborhood of* 0*, there exists a unique, continuously differentiable vector function* $v(\epsilon) = (x(\epsilon)^t, \mu(\epsilon)^t, \lambda(\epsilon)^t)^t$ *satisfying the second order sufficient conditions for a local minimum of* P(ϵ) *such that* $v(0) = (x^{\star t}, \mu^{\star t}, \lambda^{\star t})^t$ *and hence* $x(\epsilon)$ *is a unique local minimum of* P(ϵ) *with associated multipliers* $\mu(\epsilon), \lambda(\epsilon)$*,*

(c) for ϵ *near 0, the set* $I_G(x(\epsilon), \epsilon)$ *of active inequality constraints is unchanged, strict complementarity slackness* $\mu_i G_i(x(\epsilon), \epsilon) = 0$ *holds for* $i = 1, \ldots, q$*, and the gradients* $\bar{\nabla} G_i(x(\epsilon), \epsilon), \bar{\nabla} H_j(x, \epsilon), i \in I_G(x, \epsilon), j = 1, \ldots, p$ *of the active constraints with respect to* x *are linearly independent at* $x(\epsilon)$*.*

Proof: See [17]. □

Differentiating the equations

$$\begin{aligned} \bar{\nabla} \mathcal{L}(x(\epsilon), \mu(\epsilon), \lambda(\epsilon), \epsilon) &= 0, & (7.5) \\ \mu_i(\epsilon)\, G_i(x(\epsilon), \epsilon) &= 0, \quad i = 1, \ldots, q, & (7.6) \\ H(x(\epsilon), \epsilon) &= 0, & (7.7) \end{aligned}$$

with respect to ϵ, it follows that

$$M(\epsilon) \nabla v(\epsilon) = N(\epsilon), \tag{7.8}$$

where

$$M(\epsilon) := \begin{pmatrix} \bar{\nabla}^2 \mathcal{L}(x(\epsilon), \mu(\epsilon), \lambda(\epsilon), \epsilon) & (\bar{\nabla} G(x(\epsilon)), \epsilon)^t & (\bar{\nabla} H(x(\epsilon), \epsilon))^t \\ I_q \mu(\epsilon) \bar{\nabla} G(x(\epsilon), \epsilon) & I_q\, G(x(\epsilon), \epsilon) & 0 \\ \bar{\nabla} H(x(\epsilon), \epsilon) & 0 & 0 \end{pmatrix}$$

and

$$N(\epsilon) := -\begin{pmatrix} \nabla(\bar{\nabla}\mathcal{L}(x(\epsilon),\mu(\epsilon),\lambda(\epsilon),\epsilon)^t) \\ I_q\mu(\epsilon)\nabla G(x(\epsilon),\epsilon) \\ \nabla H(x(\epsilon),\epsilon) \end{pmatrix}$$

and I_q denotes the unity matrix in $\mathbb{R}^{q\times q}$. As stated in [17], $M(\epsilon)$ is regular under the assumptions of Theorem 7.1 for ϵ near 0. Thus, it follows that

$$\nabla v(\epsilon) = M(\epsilon)^{-1}N(\epsilon)$$

for ϵ near 0, where the quantities are evaluated as in (7.8). This expression can be substituted into the Taylor expansion of $v(\epsilon)$ in order to obtain a first-order approximation, as is stated in the following

COROLLARY 7.2 *Under the assumptions of Theorem 7.1, a first-order approximation of $v(\epsilon)$ in a neighborhood of $\epsilon = 0$ is given by*

$$v(\epsilon) = \begin{pmatrix} x(\epsilon) \\ \mu(\epsilon) \\ \lambda(\epsilon) \end{pmatrix} = \begin{pmatrix} x^\star \\ \mu^\star \\ \lambda^\star \end{pmatrix} + M(0)^{-1}N(0)\epsilon + o(||\epsilon||), \tag{7.9}$$

where $\Phi(\epsilon) := o(||\epsilon||)$ means that $\Phi(\epsilon)/||\epsilon|| \to 0$ as $\epsilon \to 0$.

7.2 Sensitivity Analysis for MOP and BLMOP

In this section we develop methods for the sensitivity analysis of both parametrized MOPs and parametrized BLMOPs. Since later on in this section the analysis for BLMOPs is realized by applying the analysis for MOPs on a suitable reformulation of BLMOP, we begin by extending the previously presented concept, which was adapted from [17], in order to analyze parametrized MOPs of the following form:

$$\begin{aligned} \min_x\ & F(x,\epsilon) && (\mathrm{PM}(\epsilon)) \\ \text{s.t.}\quad & G_i(x,\epsilon) \le 0, \quad i = 1,\dots,q, \\ & H_j(x,\epsilon) = 0, \quad j = 1,\dots,p. \end{aligned}$$

Here, $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^l$ is a vector-valued function and minimization has to be understood in the sense of Definition 2.1, that is, in the sense of multi-objective optimization.

To be more precise, we are interested in finding the maximal distance of a solution $x^\star = x(0)$ of PM(0) to a (compact) Pareto set of PM(ϵ), when ϵ varies within a

neighborhood of 0, that is, we want to estimate

$$\Delta(x^\star, \delta) := \max_{\epsilon} \min_{x} \{||x^\star - x|| : x \text{ solves PM}(\epsilon),\ ||\epsilon|| \leq \delta\}$$

for some $\delta > 0$ and some norm $||\cdot||$. For this we will also have to consider the following weighted sums problems corresponding to fixed weighting vectors $\alpha \in S_\alpha := \{\alpha : \sum_{i=1}^{l} \alpha_i = 1, \alpha_i \geq 0, i = 1, \ldots, l\}$:

$$\begin{aligned} \min_{x} \sum_{i=1}^{l} \alpha_i F_i(x, \epsilon) & \qquad (\text{PWS}(\epsilon, \alpha)) \\ \text{s.t.} \quad & G_i(x, \epsilon) \leq 0, \quad i = 1, \ldots, q, \\ & H_j(x, \epsilon) = 0, \quad j = 1, \ldots, p. \end{aligned}$$

In the following we make the assumption

(A1) for every local solution $x^\star$ of PM(ϵ) there is a corresponding weighting vector $\alpha \in S_\alpha$ such that $x^\star$ is also a unique local solution for problem PWS(ϵ, α).

W.l.o.g., let $\epsilon = 0$ and let $x^\star = x(0)$ be both a local Pareto point of PM(0) and a local solution of PWS(0, $\alpha^\star$) for a weighting vector $\alpha^\star \in S_\alpha$. Since the unique solution of PWS($\epsilon, \alpha^\star$) is contained in the Pareto set of PM(ϵ), we can conclude that

$$\Delta(x^\star, \delta) \leq \max_{\epsilon} \{||x^\star - x|| : x \text{ solves PWS}(\epsilon, \alpha^\star),\ ||\epsilon|| \leq \delta\},$$

and thus an upper bound of first order for $\Delta(x^\star, \delta)$ can be obtained as follows. We substitute $F(x, \epsilon)$ by the weighted sums scalarization $\sum_{i=1}^{l} \alpha_i^\star F_i(x, \epsilon)$ and observe that from Corollary 7.2 we have

$$||x^\star - x(\epsilon)|| = \left|\left| \tilde{M}(0)^{-1} N(0)\epsilon + o(||\epsilon||) \right|\right|,$$

where $M(0)$ and $N(0)$ are the above defined matrices associated with the mentioned substitution for F and the matrix $\tilde{M}(0)^{-1}$ is given by the first n rows of $M(0)^{-1}$. Consequently,

$$\Delta(x^\star, \delta) \quad \dot{\leq} \quad \max_{\epsilon} \{||\tilde{M}(0)^{-1} N(0)\epsilon|| : ||\epsilon|| \leq \delta\}, \tag{7.10}$$

where $a \dot{\leq} b$ means $a \leq b + o(||\epsilon||)$ for all $a, b > 0$.

As we will present in the following, (7.10) can be improved in the sense that $x^\star$ is compared to the entire Pareto sets of the perturbed problems PM(ϵ) and not only to those particular solutions out of these sets, which are given by the solutions

of the problems PWS$(\epsilon, \alpha^\star)$. For this we consider the problems PWS(ϵ, α), while regarding not only ϵ but also α as perturbation parameters. In this case, Corollary 7.2 yields

$$x(\epsilon, \alpha) = x^\star + \tilde{M}(0, \alpha^\star)^{-1} N(0, \alpha^\star) \begin{pmatrix} \epsilon \\ \alpha - \alpha^\star \end{pmatrix} + o\left(\left|\left| \begin{pmatrix} \epsilon \\ \alpha - \alpha^\star \end{pmatrix} \right|\right| \right). \quad (7.11)$$

According to the assumption (A1), for all ϵ near 0, the distance of $x^\star$ to the Pareto set of PM(ϵ) is given by the distance of $x^\star$ to the unity of the solutions of the problems PWS(ϵ, α), $\alpha \in S_\alpha$:

$$\Delta(x^\star, \delta) = \max_{\epsilon} \min_{\alpha} \{||x^\star - x(\epsilon, \alpha)|| : \alpha \in S_\alpha, ||\epsilon|| \leq \delta\}. \quad (7.12)$$

Consequently, with $A^\star := \tilde{M}(0, \alpha^\star)^{-1} N(0, \alpha^\star)$ and $\xi(\epsilon, \alpha) := (\epsilon, \alpha - \alpha^\star)^t$ the following equation holds up to first order:

$$\Delta(x^\star, \delta) \doteq \max_{\epsilon} \min_{\alpha} \{||A^\star \xi(\epsilon, \alpha)|| : \alpha \in S_\alpha, ||\epsilon|| \leq \delta\}. \quad (7.13)$$

Whereas the above considerations are based on comparisons of Pareto points in parameter space, it might also be important to know the behavior of the objective values caused by the variation of ϵ. Assuming that F_i is Lipschitz continuous with Lipschitz constant L_i, $i = 1, \ldots, l$, we have

$$|F_i(x, \epsilon) - F_i(x^\star, 0)| \dot{\leq} L_i \, \Delta(x^\star, \delta) \quad (7.14)$$

for all ϵ with $||\epsilon|| \leq \delta$ and $x \in \{x : x \text{ solves PM}(\epsilon), ||x - x^\star|| \leq \Delta(x^\star, \delta)\}$. The expression (7.14) considers the variation of F_i for the case that the distance of $x^\star$ to the Pareto set of PM(ϵ) in pre-image space is maximized subject to $||\epsilon|| \leq \delta$. In practice it can also be interesting to know the maximal distance of $F_i(x^\star, 0)$ to the i-th component of the Pareto set of PM(ϵ) in image space subject to $||\epsilon|| \leq \delta$, which for all $i = 1, \ldots, l$ can be expressed by

$$\Delta_{F_i}(x^\star, \delta) := \max_{\epsilon} \min_{x} \{|F_i(x^\star, 0) - F_i(x, \epsilon)| : x \text{ solves PM}(\epsilon),\ ||\epsilon|| \leq \delta\}.$$

Accordingly, if we are interested in the maximal distance of $F(x^\star, 0)$ to the Pareto set of PM(ϵ) in image space subject to $||\epsilon|| \leq \delta$, we have to consider

$$\Delta_F(x^\star, \delta) := \max_{\epsilon} \min_{x} \{||F(x^\star, 0) - F(x, \epsilon)|| : x \text{ solves PM}(\epsilon),\ ||\epsilon|| \leq \delta\}.$$

From (7.11), we have

$$F_i(x(\epsilon, \alpha), \epsilon) \quad \approx \quad F_i(x^\star + A^\star \xi(\epsilon, \alpha), \epsilon)$$

and therefore we can state

$$\Delta_{F_i}(x^\star,\delta) \;\approx\; \max_{\epsilon}\min_{\alpha}\{|F_i(x^\star,0)-F_i(x^\star+A^\star\xi(\epsilon,\alpha),\epsilon)| : x \text{ solves PM}(\epsilon),\ ||\epsilon||\le\delta\}$$

and

$$\Delta_{F}(x^\star,\delta) \;\approx\; \max_{\epsilon}\min_{\alpha}\{||F(x^\star,0)-F(x^\star+A^\star\xi(\epsilon,\alpha),\epsilon)|| : x \text{ solves PM}(\epsilon),\ ||\epsilon||\le\delta\}.$$

EXAMPLE 7.3 Consider $F=(F_1,F_2)^t:\mathbb{R}^2\times\mathbb{R}\to\mathbb{R}^2$ defined by

$$F_1(x,\epsilon) = (x_1-\epsilon)^2+x_2^2, \tag{7.15}$$
$$F_2(x,\epsilon) = (x_1-\epsilon-1)^2+(x_2-1)^2 \tag{7.16}$$

and the corresponding unconstrained parametrized MOP

$$\min_x \begin{pmatrix} F_1(x,\epsilon)\\ F_2(x,\epsilon)\end{pmatrix}. \tag{7.17}$$

The Pareto set of this MOPis given by $\{(x+\epsilon,x) : 0\le x\le 1\}$ for every $\epsilon\in\mathbb{R}$. Moreover, for every $\epsilon\in\mathbb{R}$ and every $\alpha\in S_\alpha$ we consider the weighted sums scalarization $F_\alpha:\mathbb{R}^2\times\mathbb{R}\to\mathbb{R}$ defined by

$$F_\alpha(x,\epsilon)=\alpha_1F_1(x,\epsilon)+\alpha_2F_2(x,\epsilon)$$

and the associated weighted sums problem

$$\min_x F_\alpha(x,\epsilon). \tag{7.18}$$

Observe that, since F_1 and F_2 are convex for every fixed $\epsilon\in\mathbb{R}$, F_α is convex for all fixed $\epsilon\in\mathbb{R}$, $\alpha\in S_\alpha$, and therefore the corresponding solutions of (7.18) are unique.

For the following demonstration we fix $\alpha_1=\alpha_1^\star=\frac{3}{4}$, $\alpha_2=\alpha_2^\star=\frac{1}{4}$ and $\epsilon=\epsilon^\star=0$. Thus, the necessary optimality condition for a solution of (7.18) is

$$\bar\nabla F_{\alpha^\star}(x,0)=\frac{3}{4}\bar\nabla F_1(x,0)+\frac{1}{4}\bar\nabla F_2(x,0)=\frac{3}{4}\begin{pmatrix}2x_1\\2x_2\end{pmatrix}^t+\frac{1}{4}\begin{pmatrix}2x_1-2\\2x_2-2\end{pmatrix}^t=0,$$

from which we obtain the unique solution

$$x^\star=\begin{pmatrix}x_1^\star\\x_2^\star\end{pmatrix}=\frac{1}{4}\begin{pmatrix}1\\1\end{pmatrix}.$$

As mentioned before, for our analysis we have to regard both ϵ and α as perturbation parameters and therefore we have to calculate $M(\epsilon,\alpha)$ and $N(\epsilon,\alpha)$. The gradient $\bar\nabla F_\alpha(x^\star,\epsilon)$ turns out to be

$$\bar\nabla F_\alpha(x^\star,\epsilon)=2(\alpha_1(x_1-\epsilon)+\alpha_2(x_1-\epsilon-1),\ \alpha_1x_2+\alpha_2(x_2-1))$$

Since there are no active constraints in the neighborhood of the considered point, the matrix $M(\epsilon, \alpha)$ is simply

$$M(\epsilon,\alpha) = \bar{\nabla}^2 F_\alpha(x^\star,\epsilon) = \begin{pmatrix} 2(\alpha_1+\alpha_2) & 0 \\ 0 & 2(\alpha_1+\alpha_2) \end{pmatrix} = 2\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

and consequently

$$M(\epsilon,\alpha)^{-1} = \frac{1}{2}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Denoting by ∇ the gradient with respect to (ϵ, α), $N(\epsilon, \alpha)$ turns out to be

$$\begin{aligned} N(\epsilon,\alpha) &= -\nabla\bar{\nabla}F_\alpha(x^\star,\epsilon)^t \\ &= -2\begin{pmatrix} -(\alpha_1+\alpha_2) & x_1-\epsilon & x_1-\epsilon-1 \\ 0 & x_2 & x_2-1 \end{pmatrix} \\ &= -2\begin{pmatrix} -1 & x_1-\epsilon & x_1-\epsilon-1 \\ 0 & x_2 & x_2-1 \end{pmatrix}. \end{aligned}$$

It follows that

$$\nabla x(\epsilon^\star,\alpha^\star) = M^{-1}(0,\alpha^\star)N(0,\alpha^\star) = -\frac{1}{4}\begin{pmatrix} -4 & 1 & -3 \\ 0 & 1 & -3 \end{pmatrix}$$

and from (7.11) we have

$$x(\epsilon,\alpha) \doteq \frac{1}{4}\begin{pmatrix} 1 \\ 1 \end{pmatrix} - \frac{1}{4}\begin{pmatrix} -4 & 1 & -3 \\ 0 & 1 & -3 \end{pmatrix}\begin{pmatrix} \epsilon \\ \alpha_1-\alpha_1^\star \\ \alpha_2-\alpha_2^\star \end{pmatrix}.$$

Thus, for fixed $\alpha = \alpha^\star$ we have

$$x(\epsilon,\alpha^\star) \doteq \frac{1}{4}\begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} \epsilon \\ 0 \end{pmatrix},$$

that is, $x(\epsilon, \alpha^\star)$ moves in parallel to the x_1-axis when ϵ varies, see Figure 38. With these results, according to (7.10), we can write

$$\begin{aligned} \Delta(x^\star,\delta) &\dot{\leq} \max_\epsilon\{||x(\epsilon,\alpha^\star)-x^\star|| : ||\epsilon|| \leq \delta\} && (7.19) \\ &= \max_\epsilon\{||\epsilon|| : ||\epsilon|| \leq \delta\} = \delta. && (7.20) \end{aligned}$$

As one can see in Figure 38, for the case $\delta = 1$, both $x(-1, \alpha^\star)$ and $x(1, \alpha^\star)$ are located on the line $x_2 = \frac{1}{4}$ such that $||x(-1,\alpha^\star) - x^\star|| = ||x(1,\alpha^\star) - x^\star|| = 1$ and therefore we have

$$\Delta(x^\star, 1) \dot{\leq} 1. \qquad (7.21)$$

Now we go a step ahead and calculate a more accurate estimation according to (7.13). For this, we have to find for every ϵ near $\epsilon^\star = 0$ the minimum of

$$\left\| M^{-1}(0,\alpha^\star)N(0,\alpha^\star) \begin{pmatrix} \epsilon \\ \alpha_1 - \alpha_1^\star \\ \alpha_2 - \alpha_2^\star \end{pmatrix} \right\|$$

with respect to α or, equivalently, we have to solve

$$\min_{\alpha_2} \quad (-4\epsilon + 1 - 4\alpha_2)^2 + (1 - 4\alpha_2)^2 \tag{7.22}$$

$$s.t. \quad 0 \leq \alpha_2 \leq 1, \tag{7.23}$$

where α_1 has been eliminated by the substitution $\alpha_1 = 1 - \alpha_2$. Assuming for the moment that ϵ varies only in a small neighborhood such that the solution $\hat{\alpha}(\epsilon)$ of (7.22) can be expected to vary within the open interval $(0, 1)$, we can omit the constraints on α_2 and therefore the necessary optimality condition turns out to be

$$\bar{\nabla}_{\alpha_2}((-4\epsilon + 1 - 4\alpha_2)^2 + (1 - 4\alpha_2)^2) = -8(-4\epsilon + 2 - 8\alpha_2) = 0$$

or

$$2\epsilon - 1 + 4\alpha_2 = 0,$$

from which we obtain the solution

$$\hat{\alpha}_2 = \frac{1}{4} - \frac{\epsilon}{2} \text{ and } \hat{\alpha}_1 = 1 - \left(\frac{1}{4} - \frac{\epsilon}{2}\right) = \frac{\epsilon}{2} + \frac{3}{4}.$$

This result is valid for all $\epsilon \in (-\frac{3}{2}, \frac{1}{2})$, because in this case $\hat{\alpha}_2(\epsilon)$ remains within the open interval $(0, 1)$, that is, there are no active constraints at $\hat{\alpha}_2(\epsilon)$. For $\epsilon \in \mathbb{R}\backslash(-\frac{3}{2}, \frac{1}{2})$ the necessary optimality condition alters to be

$$2\epsilon - 1 + 4\alpha_2 - \mu = 0,$$

where the sign of $\mu \geq 0$ for $\alpha_2 = 1$ and $\mu \leq 0$ for $\alpha_2 = 0$. More precisely, we have

$$\mu = \begin{cases} 3 + 2\epsilon, & \epsilon \leq -\frac{3}{2} \\ 0, & \epsilon \in (-\frac{3}{2}, \frac{1}{2}) \\ 2\epsilon - 1, & \epsilon \geq \frac{1}{2} \end{cases} \tag{7.24}$$

and finally

$$\hat{\alpha}(\epsilon) = \begin{cases} (0,1)^t, & \epsilon \leq -\frac{3}{2} \\ (\frac{\epsilon}{2} + \frac{3}{4}, \frac{1}{4} - \frac{\epsilon}{2})^t, & \epsilon \in (-\frac{3}{2}, \frac{1}{2}) \\ (1,0)^t, & \epsilon \geq \frac{1}{2} \end{cases} . \tag{7.25}$$

As can be seen from this result, both $\hat{\alpha}_1(\epsilon)$ and $\hat{\alpha}_2(\epsilon)$ depend continuously and monotonously on ϵ. Moreover, it can be shown that the function

$$\epsilon \to ||M^{-1}(0, \alpha^\star) N(0, \alpha^\star)\, \xi(\epsilon, \hat{\alpha}(\epsilon))||$$

is also continuous and monotone. Thus, for $\delta = 1$, the calculation of (7.13) reduces to

$$\begin{aligned} \Delta(x^\star, 1) &\doteq \max_{\epsilon}\{||M^{-1}(0, \alpha^\star) N(0, \alpha^\star)\, \xi(\epsilon, \hat{\alpha}(\epsilon))|| : \epsilon \in \{-1, 1\}\} \\ &= \max\left\{\frac{1}{\sqrt{2}}, \frac{\sqrt{10}}{4}\right\} = \frac{\sqrt{10}}{4}. \end{aligned} \tag{7.26}$$

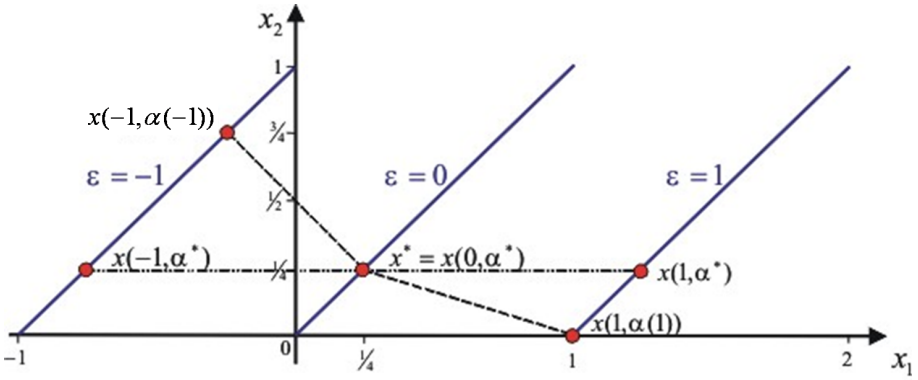

Figure 38: Comparison of the different estimations for $\Delta(x^\star, 1)$.

The sensitivity analysis for a BLMOP without lower level inequality constraints can be realized by the use of the sensitivity analysis for MOP as described above. To be more precise, the desired estimations are based on the application of (7.10) and (7.13), respectively, to the following reformulation obtained by replacing the

lower level problem by its Kuhn-Tucker conditions:

$$\min_{x\in\mathbb{R}^n, y\in\mathbb{R}^m, \alpha\in\mathbb{R}^l, \zeta\in\mathbb{R}^p} F(x,y), \tag{7.27}$$

$$\begin{aligned}
\text{s.t} \qquad G(x,y) &\leq_p 0,\\
H(x,y) &= 0,\\
\sum_{i=1}^{l} \alpha_i \nabla_x f_i(x,y) + \sum_{i=1}^{p} \zeta_i \nabla_x h_i(x,y) &= 0, \qquad (7.28)\\
h(x,y) &= 0,\\
\sum_{i=1}^{l} \alpha_i - 1 &= 0,\\
-\alpha_i &\leq 0,\ i = 1,\ldots,l,
\end{aligned}$$

where minimization has to be understood in the sense of multi-objective optimization. For a more detailed description we consider the following

EXAMPLE 7.4 Let $F = (F_1, F_2)^t, f = (f_1, f_2)^t : \mathbb{R}^2 \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}^2$ be defined by

$$\begin{aligned}
F_1(x,y,\epsilon) &= \frac{1}{2}\left((x_1 - y)^2 + (1+\epsilon)\, x_2^2\right),\\
F_2(x,y,\epsilon) &= \frac{1}{2}\left((x_1 - y - 1)^2 + (x_2 - 1)^2\right),\\
f_1(x,y,\epsilon) &= \frac{1}{2}\left((x_1 - 2)^2 + (x_2 - y - \epsilon)^2\right),\\
f_2(x,y,\epsilon) &= \frac{1}{2}\left((x_1 + 1)^2 + (x_2 - 1)^2\right),
\end{aligned}$$

and consider the following parametrized BLMOP:

$$\begin{aligned}
&\min_{x,y} F(x,y,\epsilon), \qquad (7.29)\\
\text{s.t. } x \text{ solves: } &\min_{x} f(x,y,\epsilon).
\end{aligned}$$

Observe that there are no explicitly given constraints for the upper level. Therefore, in the following reformulation used for the calculation of the desired estimations, we will use H and G to denote the constraints which arise from replacing the lower level

by its Kuhn-Tucker conditions. Thus, we consider the following auxiliary problem:

$$\min_{x,y,\alpha} \begin{pmatrix} F_1(x,\epsilon) \\ F_2(x,\epsilon) \end{pmatrix}, \tag{7.30}$$

$$\text{s.t.}$$

$$H_1(x,\alpha,\epsilon) := \alpha_1(x_1-2)+\alpha_2(x_1+1)=0, \tag{7.31}$$

$$H_2(x,\alpha,\epsilon) := \alpha_1(x_2-y-\epsilon)+\alpha_2(x_2-1)=0, \tag{7.32}$$

$$H_3(x,\alpha,\epsilon) := \alpha_1+\alpha_2-1=0, \tag{7.33}$$

$$G_1(x,\alpha,\epsilon) := -\alpha_1 \leq 0, \tag{7.34}$$

$$G_2(x,\alpha,\epsilon) := -\alpha_2 \leq 0. \tag{7.35}$$

Here, the equations H_1 and H_2 correspond to the rows of (7.28). As can be verified by the optimality conditions stated in Theorem 6.1, the point $(x_1^\star, x_2^\star, y^\star, \alpha_1^\star, \alpha_2^\star) = (\frac{1}{2}, \frac{1}{2}, 0, \frac{1}{2}, \frac{1}{2})$ along with corresponding multipliers $\lambda_1 = \lambda_2 = \lambda_3 = \mu_1 = \mu_2 = 0$ is both a solution of the given BLMOP for $\epsilon^\star = 0$ and a solution of the related weighted sums problem corresponding to the weights $(\beta_1^\star, \beta_2^\star) = (\frac{1}{2}, \frac{1}{2})$. Now, let

$$\mathcal{L}(x,y,\alpha,\beta,\mu,\lambda,\epsilon) = \sum_{i=1}^{k} \beta_i F_i(x,y,\epsilon) + \sum_{i=1}^{2} \mu_i G_i(x,y,\epsilon) + \sum_{i=1}^{3} \lambda_i H_i(x,y,\epsilon)$$

be the Lagrangian associated with the reformulation (7.30) – (7.35) and denote $\bar{\nabla} = \nabla_{(x,y,\alpha)}$ and $\nabla = \nabla_{(\epsilon,\beta)}$. Then we obtain

$$\bar{\nabla}^2\mathcal{L}(x^\star,y^\star,\alpha^\star,\beta^\star,\mu^\star,\lambda^\star,\epsilon^\star) = \begin{pmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\nabla\bar{\nabla}\mathcal{L}(x^\star,y^\star,\alpha^\star,\beta^\star,\mu^\star,\lambda^\star,\epsilon^\star) = \begin{pmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$\nabla H(x^\star,y^\star,\alpha^\star,\epsilon^\star) = \begin{pmatrix} 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$\bar{\nabla} G(x^\star,y^\star,\alpha^\star,\epsilon^\star) = \begin{pmatrix} 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 \end{pmatrix},$$

and

$$\bar{\nabla} H(x^\star, y^\star, \alpha^\star, \epsilon^\star) = \begin{pmatrix} 1 & 0 & 0 & -\frac{3}{2} & \frac{3}{2} \\ 0 & 1 & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix},$$

yielding

$$M(\epsilon^\star, \beta^\star) = \begin{pmatrix} 1 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & -\frac{3}{2} & \frac{1}{2} & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & \frac{3}{2} & -\frac{1}{2} & 1 \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 \\ 1 & 0 & 0 & -\frac{3}{2} & \frac{3}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -\frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

and

$$N(\epsilon^\star, \beta^\star) = \begin{pmatrix} 0 & \frac{1}{2} & -\frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Since we are particularly interested in $(x(\epsilon), y(\epsilon))$, let $\tilde{M}^{-1}(\epsilon^\star, \beta^\star)$ denote the first $n + m = 3$ rows of $M^{-1}(\epsilon^\star, \beta^\star)$. Then we have

$$A^\star = \tilde{M}^{-1}(\epsilon^\star, \beta^\star) N(\epsilon^\star, \beta^\star) = \frac{1}{316} \begin{pmatrix} -90 & 78 & -78 \\ -137 & -34 & 34 \\ -18 & -16 & 16 \end{pmatrix}.$$

We assume that ϵ may vary within $(0.5, 0.5)$. Thus we are interested in estimating $\Delta((x^\star, y^\star), 0.5)$. From (7.10), by fixing $\beta = \beta^\star$, we can calculate

$$\Delta((x^\star, y^\star), 0.5) \dot{\leq} \left\| A^\star \begin{pmatrix} 0.1 \\ 0 \\ 0 \end{pmatrix} \right\| \approx 0.2609. \tag{7.36}$$

To compute an estimation according to (7.13), observe that for $\beta \in S_\beta$ the gradient of

$$\left\| A^\star \begin{pmatrix} \epsilon \\ \beta_1 - \beta_1^\star \\ \beta_2 - \beta_2^\star \end{pmatrix} \right\|^2$$

with respect to (β_1, β_2) is given by

$$\frac{937}{6241} \begin{pmatrix} \beta_1 - \beta_2 - \frac{1037}{3748}\epsilon \\ -\beta_1 + \beta_2 + \frac{1037}{3748}\epsilon \end{pmatrix} = \frac{937}{6241} \begin{pmatrix} 1 - 2\beta_2 - \frac{1037}{3748}\epsilon \\ -1 + 2\beta_2 + \frac{1037}{3748}\epsilon \end{pmatrix},$$

and vanishes if and only if $\beta_1 = \frac{1}{2}(1 + \frac{1037}{3748}\epsilon)$ and $\beta_2 = \frac{1}{2}(1 - \frac{1037}{3748}\epsilon)$. Moreover, for $|\epsilon| \leq 0.5$, we have $\beta_1, \beta_2 \in (0, 1)$. Therefore,

$$\min_\beta \left\{ \left\| A^\star \begin{pmatrix} \epsilon \\ \beta_1 - \beta_1^\star \\ \beta_2 - \beta_2^\star \end{pmatrix} \right\| : \beta \in S_\beta \right\} = \left\{ \left\| A^\star \begin{pmatrix} \epsilon \\ \frac{1037}{3748}\epsilon \\ \frac{1037}{3748}\epsilon \end{pmatrix} \right\| \right\} = \sqrt{\frac{7993}{29984}}\epsilon.$$

Consequently, we can rewrite (7.13) as

$$\Delta((x^\star, y^\star), 0.5) \doteq \min_\beta \left\{ \left\| A^\star \begin{pmatrix} 0.5 \\ \beta_1 - \beta_1^\star \\ \beta_2 - \beta_2^\star \end{pmatrix} \right\| : \beta \in S_\beta \right\} = 0.5\sqrt{\frac{7993}{29984}} \approx 0.2582,$$

which, as expected, is a better estimation than (7.36).

7.3 A Concept for the Adaptive Choice of Targets

When computing a discrete representation of a Pareto set for MOP or BLMOP, respectively, by solving parametrized subproblems, sensitivity analysis can be used to generate suitable parameters in order to control the distance (in parameter space as well as in image space) between neighboring points of the computed representation. Here, we will present a new recovering algorithm of reference point type which can be understood as a variant of our image set-oriented recovering methods presented in Section 4.1 and Section 6.2. The new algorithm uses a concept for the adaptive choice of targets (which are considered as perturbation parameters), such that the generated Pareto points are well-distributed in parameter space rather than in image space. This is motivated by applications with high-dimensional parameter space, where on one hand a well-distributed representation of the Pareto set in parameter space is required (as can be computed by our classical set-oriented methods), but on the other hand one wants to benefit from the advantages of the new image set-oriented methods mentioned in Section 4.

The variant of the new algorithm for the solution of BLMOP is realized by replacing the lower level problem by its Kuhn-Tucker conditions and computing a solution of the resulting reformulation, which has the form of a constrained MOP. Therefore, the following description is related to a general variant for the solution of constrained MOPs. Thereafter, Example 7.5 shows that this algorithm also works satisfactorily for BLMOPs. For the description of the new algorithm recall, that in image set-oriented recovering methods, every new Pareto point $\hat{x}$ is generated in the neighborhood of a known Pareto point $x^\star$ by minimizing the distance $||F(x) - t||$ between $F(x)$ and a target

$$t \in \{F(x^\star) + \sum_{j=1}^{k-1} \epsilon_j b_j + \tilde{\epsilon}(t^\star - F(x^\star))\},$$

where $\{b_j, j = 1, \ldots, k-1\}$ is a basis of the tangent space $T_{F(x^\star)}F(P)$ of the Pareto set in image space at $F(x^\star)$, and $t^\star$ is the target used previously for generating $x^\star$. The vector $\epsilon = (\epsilon_1, \ldots, \epsilon_{k-1})$ and the value $\tilde{\epsilon}$ make up a parameterization for the target t. Let us consider the case where the Pareto set is convex (in image space). In this case, we can always choose $\tilde{\epsilon} = 0$, and therefore we write $t = t(\epsilon)$. Observe that $t(0) = F(x^\star)$ and thus minimization of $||F(x) - t(0)||$ ends up in $x^\star$, that is, we have $x(0) = x^\star$ for the optimum function $x(\epsilon)$. Now suppose that, starting from a previously found Pareto point $x^\star$, we want to find a new Pareto point $\hat{x}$ such that approximately

$$||\hat{x} - x^\star|| \;=\; \delta \tag{7.37}$$

holds. According to Corollary 7.2, we have to find a parameter vector $\epsilon \in \mathbb{R}^{k-1}$ such that

$$||\tilde{M}^{-1}(0)N(0)\epsilon|| = \delta,$$

where $\tilde{M}$ and N are evaluated as in (7.8) while the objective $F(x)$ is substituted by the (differentiable) auxiliary function $||F(x)-t(\epsilon)||^2$. For every fixed $i = 1, \ldots, k-1$, since ϵ determines the target $\sum_{j=1}^{k-1} \epsilon_j b_j \in T_{F(x^\star)}F(P)$, we can choose $\epsilon_j = 0$ for $j \neq i$ in order to force that $F(\hat{x})$ is close to $F(x^\star) + \epsilon_i b_i$. Then, if $\tilde{M}^{-1}(0)N(0)e_i \neq 0$, where e_i denotes the i-th standard basis vector in $\mathbb{R}^{k-1}$, we can conclude, that we have to choose

$$\hat{\epsilon}_i := |\epsilon_i| = \frac{\delta}{||\tilde{M}^{-1}(0)N(0)e_i||}.$$

Observe that, since we are dealing with a linear approximation, we can use the targets

$$F(x^\star) + \hat{\epsilon}_i b_i \text{ and } F(x^\star) - \hat{\epsilon}_i b_i$$

for computing two different Pareto points with the required property expressed by (7.37). It should be mentioned that particularly for the case $k \geq 3$, the points computed with this technique certainly have approximately the desired distance to $x^\star$, but in general there is no information on the distance to the other Pareto points which have been computed so far. Here, additional strategies are needed in order to obtain diversity among *all* generated Pareto points. For the particular case $k = 2$ it is certainly a good choice to compute at first the individual minimum $x^\star$ of one of the objectives F_i and then – since $F(x^\star)$ makes up a vertex of the 1-dimensional Pareto set – using successively the technique described above and choosing the respective targets t in a way such that the i-th component t_i increases (and the other one decreases) successively, see Figure 39 (right). Another strategy is to start with an arbitrary target t, and after calculating the solution $x^\star$ associated with that target t, to extend the Pareto set into both directions with the mentioned technique by choosing two different targets $\tilde{t}$ and $\hat{t}$ in a way such that for a fixed $\tilde{t}_1$ and $\hat{t}_2$ increase and $\tilde{t}_2$ and $\hat{t}_1$ decreases (or vice versa) successively, see Figure 39 (left). In contrast to the image space oriented recovering algorithms described in Section 4.1 and Section 6.2, the concept of using boxes in image space can be omitted here, since for $k = 2$ the desired diversity among *all* generated Pareto points is obtained with the help of the adaptively chosen targets.

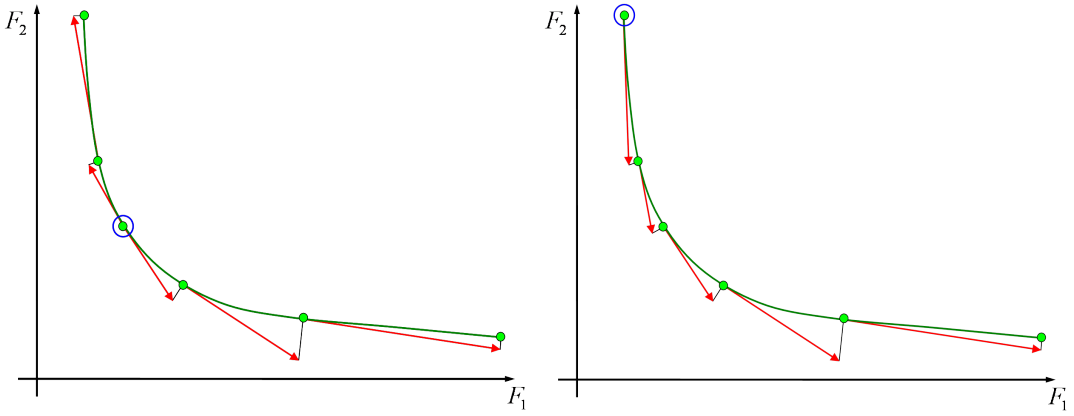

Figure 39: Adaptive target construction starting at an arbitrary Pareto point (left) and at the Pareto point corresponding to the individual minimum of F_1 (right).

EXAMPLE 7.5 Given a higher level function $F = (F_1, F_2)^t : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^2$ and a lower level function $f = (f_1, f_2)^t : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^2$, we consider the following BLMOP:

$$\min_{x,y \in \mathbb{R}} F(x,y) = \begin{pmatrix} 10(x_1-1)^4 + (x_2-1)^2 + (y-1)^2 \\ (x_1+1)^2 + (x_2+1)^2 + (y+1)^4 \end{pmatrix}, \tag{7.38}$$

such that x solves:

$$\min_{x \in \mathbb{R}} f(x,y) = \begin{pmatrix} (x_1-2)^2 + (x_2-2)^4 + 3y^2 \\ 2y(x_1+0.5)^2 + (x_2+1)^2 \end{pmatrix}. \tag{7.39}$$

We have calculated a solution for this problem using the BL-Recovering-IS method (which uses the Kuhn-Tucker reformulation for the lower level problem) in two different ways:

On one hand, the target parameter $\epsilon = 1$ was fixed and consequently the distance between neighboring computed Pareto points $F(x_i^\star)$ and $F(x_{i+1}^\star)$, $i = 1, 2, \ldots$, in image space corresponds approximately with this value $\epsilon = 1$, while the distance between the associated points $x_i^\star$ and $x_{i+1}^\star$, $i = 1, 2, \ldots$, in parameter space varies, see Figure 40.

On the other hand, the target parameter $\epsilon = \hat{\epsilon}$ was estimated in every step by the above sensitivity analysis based technique. Here, we have chosen $\delta = 0.15$ and as depicted in Figure 41, the distance between neighboring computed Pareto points in parameter space $x_i^\star$ and $x_{i+1}^\star$, $i = 1, 2, \ldots$, corresponds – as desired – approximately with this value $\delta = 0.15$, while the distance between the associated points $F(x_i^\star)$ and $F(x_{i+1}^\star)$, $i = 1, 2, \ldots$, in image space varies.

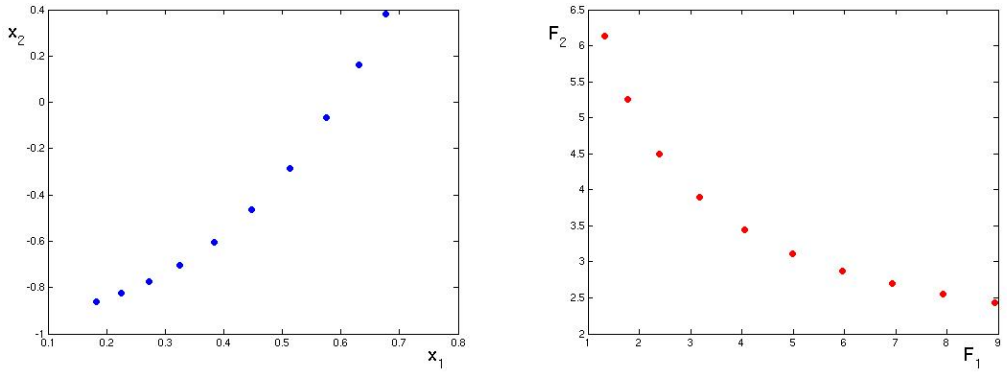

Figure 40: The solution in parameter space (left) and image space (right) for fixed target parameter $\epsilon = 1$.

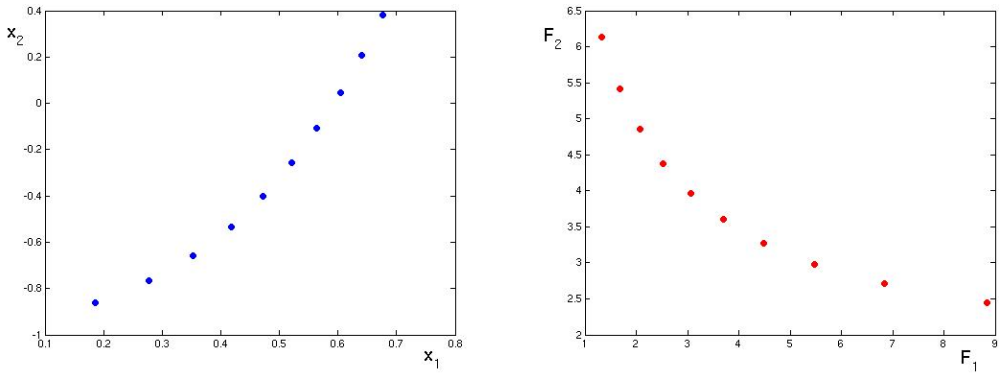

Figure 41: The solution in parameter space (left) and image space (right) for adaptive target parameter $\hat{\epsilon}$ related to $\delta = 0.15$.

8 Conclusion and Outlook

In this work we considered the class of bi-level multi-objective optimization problems (BLMOP), which can be understood as a generalization of classical bi-level optimization problems to the case where both the upper and lower level are given by multi-objective optimization problems (MOP) instead of scalar-valued optimization problems.

Based on the well-known Kuhn-Tucker optimality conditions for classical optimization problems, we have developed optimality conditions for BLMOP for the case that the lower level problem is a convex one without inequality constraints. These optimality conditions have been used for the definition of numerical algorithms for the solution of this particular subclass of BLMOP.

Since the solution of bi-level multi-objective optimization problems, the Pareto set, is typically an extensive set or, in mathematical terms, a manifold, most of our algorithms are embedded in a set-oriented framework. In particular, we have concentrated on two main directions, that is, algorithms of subdivision type and algorithms of recovering type. Both have their individual advantages and work satisfactorily on their own, but it turned out that performance can often be improved when they are used in combination.

Moreover, in this work we have considered the sensitivity analysis for MOP and BLMOP, that is, we have investigated the behavior of the Pareto set under the variation of additional parameters. The results have particularly been applied to one of our set-oriented algorithms in order to control the spreading of the discrete representation of the computed solution.

The restriction to the above mentioned subclass of BLMOP is motivated by the following mathematical circumstances: the lower level problem is assumed to be convex, because otherwise it can not be guaranteed that the points satisfying the Kuhn-Tucker conditions for the lower level comply exactly with the feasible region for the upper level problem. Moreover, the common constraint qualifications, which have to be fulfilled for applying the Kuhn-Tucker conditions to the higher level problem, are necessarily violated in the presence of lower level inequality constraints.

The development of advanced optimality conditions for the general BLMOP, which also includes non-convex lower level problems with inequality constraints, is an interesting field which shall be investigated in the future. Well-known second order sufficient optimality conditions for classical optimization may form a promising basis for further development. Moreover, the above mentioned sensitivity analysis may be an additional tool not only for the development of the desired optimality conditions, but also for designing algorithms for the solution of the general BLMOP.

Up to now, we have used the developed algorithms to solve several academic examples and technical applications. Our interest for the future is to apply both the existing and upcoming algorithms to BLMOPs arising from real world problems of any discipline.

References

[1] J. F. Bard, Practical Bilevel Optimization: Algorithms and Applications, Kluwer Academic Publishers, 1998.

[2] H. Bonnel and J. Morgan, Semivectorial Bilevel Optimization Problem: Penalty Approach, Journal of Optimization Theory and Application, vol. 131(3), pp. 365–382, 2006.

[3] M. Brown, R. E. Smith, Directed Multi-Objective Optimisation, International Journal of Computers, Systems and Signals, vol. 6(1), 2005.

[4] I. Chauduri, S. Sertl, H. Zoltán, M. Dellnitz and T. Fraunheim, Global Optimization of Silicon Nanoclusters, Applied Surface Science, vol. 226(1-3), pp. 108-113, 2004.

[5] B. Colson, P. Marcotte and G. Savard, Bilevel programming: A survey, A Quarterly Journal of Operations Research, vol. 3, pp. 87–107, 2005.

[6] I. Das, J. E. Dennis, A Closer Look at Drawbacks of Minimizing Weighted Sums of Objectives for Pareto Set Generation in Multicriteria Optimization Problems, Structural Optimization, vol. 14, pp. 63–69, 1997

[7] I. Das, J. E. Dennis, Normal-Boundary Intersection: A new Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems, SIAM Journal of Optimization, vol. 8, no.3, pp. 631–657, 1998

[8] J. P. Dauer, T. A. Fosnaugh, Optimization over the Efficient Set, Journal of Global Optimization, vol. 7, pp. 261–277, 1995.

[9] A. Dell'Aere, Numerische nichtlineare Optimierung in MuPAD, Diplomarbeit, Paderborn, 2003

[10] A. Dell'Aere, Multi-Objective Optimization in Self-Optimizing Systems, Proceedings of the 32nd Annual Conference of the IEEE Industrial Electronics Society, Paris, 2006

[11] M. Dellnitz, O. Schütze and T. Hestermeyer, Covering Pareto Sets by Multilevel Subdivision Techniques, Journal of Optimization Theory and Application, vol. 124(1), pp. 113–13, 2005.

[12] S. Dempe, Foundations of Bilevel Programming, Kluwer Academic Publishers, 2002.

[13] S. Dempe, Annotated Bibliography on Bilevel Programming and Mathematical Programs with Equilibrium Constraints, Optimization, vol. 52, pp. 333–359, 2003.

[14] P. Deuflhard, A. Hohmann, Numerische Mathematik I – Eine algorithmisch orientierte Einfhrung, de Gruyter & Co., 1993.

[15] M. Ehrgott, Multicriteria Optimization, Lecture Notes in Economics and Mathematical Systems, vol. 491, Springer, Berlin, 2000.

[16] G. Eichfelder, Parametergesteuerte Lösung nichtlinearer multikriterieller Optimierungsprobleme, PhD thesis, University of Erlangen-Nürnberg, Germany, 2006.

[17] A. V. Fiacco, Introduction to Sensitivity and Stability Analysis in Nonlinear Programming, Academic Press, London, 1983.

[18] J. Figueira, S. Greco, M. Ehrgott, Multiple Criteria Decision Analysis, Lecture Notes in Economics and Mathematical Systems, Springer, New York, 2005.

[19] M. Fukushima, Merit Functions for Variational Inequality and Complementarity Problems, Nonlinear Optimization and Applications, G. Di Pillo and F. Giannessi (eds.), Plenum Press, pp. 155-170, 1996.

[20] C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb and L. Thiele, Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003, Springer, Berlin, 2003.

[21] U. Frank, H. Giese, F. Klein, O. Oberschelp, A. Schmitt, B. Schulz, H. Vcking, K. Witting, J. Gausemeier (Hrsg.), Selbstoptimierende Systeme des Maschinenbaus - Definitionen und Konzepte, HNI-Verlagsschriften, vol. 155, Paderborn, 2004.

[22] M. Fukushima and J.-S. Pang, Convergence of a Smoothing Continuation Method for Mathematical Programs with Complementarity Constraints, Ill-Posed Variational Problems and Regularization Techniques, Lecture Notes in Economics and Mathematical Systems, vol. 477, pp. 99-110, 1999.

[23] K. Harada, J. Sakuma, S. Kobayashi, Local Search for Multiobjective Function Optimization: Pareto Descent Method, Proceedings of the 8th annual conference on genetic and evolutionary computation, pp. 659–666, New York, 2006.

[24] M. E. Henderson, Multiple Parameter Continuation: Computing Implictly Defined k-Manifilds, International Journal of Bifurcation and Chaos, vol. 12, pp. 451–476, 2002

[25] C. Hillermeier, Nonlinear Multiobjective Optimization - A Generalized Homotopy Approach, Birkhäuser, Berlin, 2001.

[26] R. Horst and H. Tuy, Global Optimization: Deterministic Approaches, Springer, Berlin, 1993.

[27] J. Jahn, Vector Optimization – Theory, Applications, and Extensions, Springer, Berlin, 2004.

[28] H. Jiang and D. Ralph, Smooth SQP Methods for Mathematical Programs with Nonlinear Complementarity Constraints, Journal on Optimization, vol. 10(3), pp. 779–808, 2000.

[29] H. Jiang and D. Ralph, Extension of Quasi-Newton Methods to Mathematical Programs with Complementarity Constraints, Computational Optimization and Applications, vol. 25(1-3), pp. 123–150, 2003

[30] T. Knoke, C. Romaus, J. Böcker, A. Dell'Aere, K. Witting, Energy management for an Onboard Storage System based on Multiobjective Optimization, Proceedings of the 32nd Annual Conference of the IEEE Industrial Electronics Society, Paris, 2006

[31] H. Kuhn and A. Tucker, Nonlinear programming, Proc. Berkeley Symp. Math. Statist. Probability, vol. 2, pp. 481–492, 1951.

[32] Y. C. Liou, X. Yang, and J. C. Yao, Mathematical Programs with Vector Optimization Constraints, Journal of Optimization Theory and Application, vol. 126(2), pp. 345–355, 2005.

[33] Z. Q. Luo, J.-S. Pang, and D. Ralph, Mathematical Programs with Equilibrium Constraints, Cambridge University Press, Cambridge, 1996.

[34] K. Miettinen, Nonlinear Multiobjective Optimization, Kluwer Academic Publishers, 1999.

[35] W. Oevel, Einführung in die Numerische Mathematik, Spektrum Akademischer Verlag, Heidelberg, 1996.

[36] P. M. Pardalos and M. G. C. Resende, Handbook of Applied Optimization, Oxford University Press, New York, 2002.

[37] S. Schäffler and R. Schultz and K. Weinzierl, A Stochastic Method for the Solution of Unconstrained Vector Optimization Problems, Journal of Optimization Theory and Application, vol. 114(1), pp. 209–222, 2002.

[38] O. Schütze, Set-oriented Methods for Global Optimization, PhD Thesis, University of Paderborn, Germany, 2004.

[39] O. Schütze, A. Dell'Aere and M. Dellnitz, On Continuation Methods for the Numerical Treatment of Multi-Objective Optimization Problems, in Dagstuhl Seminar Proceedings 04461, Practical Approaches to Multi-Objective Optimization, IBFI, Schloss Dagstuhl, Germany, 2005 http://drops.dagstuhl.de/opus/volltexte/2005/349.

[40] O. Schütze, L. Jourdan, T. Legrand, E.-G. Talbi and J. L. Wojkiewicz, A Multi-objective Approach to the Design of Conducting Polymer Composites for Electromagnetic Shielding, To appear in Evolutionary Multi-Criterion Optimization. Fourth International Conference, EMO 2007, Springer, Berlin, 2007.

[41] S. Sertl and M. Dellnitz, Global Optimization using a Dynamical Systems Approach, Submitted to Journal of Global Optimization, 2005

[42] X. Shi and H. S. Xia, Model and Interactive Algorithm of Bi-Level Multi-Objective Decision Making with Multiple Interconnected Decision Makers, Journal of Multi-Criteria Analysis vol. 10, pp. 27–34, 2001.

[43] H. v. Stackelberg, Marktform und Gleichgewicht, Springer, Berlin, 1934.

[44] P. T. Thach, H. Konno and D. Yokota, Dual Approach to Minimization on the Set of Pareto-Optimal Solutions, Journal of Optimization Theory and Application, vol. 88(3), pp. 689–707, 1996.

[45] L. N. Vicente and P. H. Calamai, Bilevel and Multilevel Programming: A Bibliography Review, Journal of Global Optimization, pp. 291–306, 1994.

[46] K. Witting, B. Schulz, A. Pottharst, M. Dellnitz, J. Böcker, N. Fröhleke, A new Approach for Online Multiobjective Optimization of Mechatronical Systems, Submitted to the International Journal on Software Tools for Technology Transfer STTT (Special Issue on Self-Optimizing Mechatronic Systems), 2004.

[47] N. Yamashita, K. Taji, M. Fukushima, Unconstrained Optimization Reformulations of Variational Inequality Problems, Journal of Optimization Theory and Applications, vol. 92, pp. 439–456, 1997.

[48] J. J. Ye, Constraint Qualifications and KKT Conditions for Bilevel Programming Problems, Mathematics of Operations Research, vol. 31(4), pp. 811–824, 2006.

List of Figures

Index