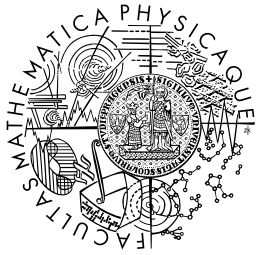

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

**MASTER THESIS**

Jiří Lžičař

# Solving Methods for Bilevel Optimization Problems

Department of Probability and Mathematical Statistics

| | |
|---|---|
| Supervisor of the master thesis: | doc. RNDr. Ing. Miloš Kopa, PhD. |
| Study programme: | Mathematics |
| Study branch: | Probability, Mathematical Statistics and Econometrics |

Prague 2019

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date .....  signature of the author

Title: Solving Methods for Bilevel Optimization Problems

Author: Jiří Lžičař

Department: Department of Probability and Mathematical Statistics

Supervisor: doc. RNDr. Ing. Miloš Kopa, PhD., Department of Probability and Mathematical Statistics

Abstract: The presented thesis discusses bilevel programming problems with the focus on solution algorithms. Bilevel programming problem is a hierarchical programming problem, where constraints contain another programming problem.

We formulate basic bilevel optimization theory and describe three types of solution algorithms for bilevel programming problems: Algorithms based on KKT reformulation where the lower level is replaced by its KKT conditions, algorithms based on optimal value function where the bilevel programming problem is reduced to a single level problem using the optimal value function of the lower level problem, and algorithms solving linear bilevel programming problems.

Using real data for portfolio optimization bilevel programming problems, we compare ability to solve the problems and computing time of some of the presented algorithms.

Keywords: Optimization, bilevel programming, algorithms, KKT reformulation, Optimal value function reformulation, mean-risk model.

First and foremost, I would like to express the deepest gratitude to my supervisor doc. Miloš Kopa for numerous excellent and effective consultations and for his enormous willingness to help me even during the most hectic time of the year.

I thank Zuza for moral support and her patience with me and I thank my parents for financial support during the most of my studies.

I would also like to thank my friend and colleague Jirka for guidance during the writing of this thesis and my friend Anna for the language advice.

# Contents

# 1. Introduction

Bilevel programming problem (BLPP) is a mathematical programming problem (MPP) with another MPP as a constraint (formal definition will be given in chapter 2). BLPP was introduced and mathematically formulated in Bracken and McGill [1], but the first problem that can be considered as a BLPP is the Stackelberg duopoly in von Stackelberg [26] known as well as the leader-follower problem. This is one (and probably most used and best known) of the applications of BLPP: Modeling a two-level system with two decision makers (one on the lower level and one on the upper level).

There is an increase in research of properties of BLPP, its applications and solving algorithms for BLPP. Because BLPP (even in Linear case) is in general strongly NP-hard (proven in Hansen et al. [17]), applications are limited by the computing power of accessible technology. This implies a demand for an invention of new solving methods and algorithms.

A short overview of bilevel optimization can be found in Colson et al. [3]. Two monographs Dempe [5] and Dempe et al. [11] are dedicated to bilevel optimization. The former focuses on theoretical foundations of BLPP and on optimality conditions, the latter contains an overview of some solving methods of BLPP with the main focus on mixed integer bilevel programming problems and applications of the BLPP. Solution algorithms for the BLPP can be found in the two mentioned monographs and in the papers Dempe [4], Dempe and Franke [7], Dempe and Franke [8], Ma [19] or Zheng et al. [31]. Optimality and stationary conditions of BLPP are discussed in Dempe and Zemkoho [10], Dempe and Zemkoho [10], Ye [28] or Ye and Zhu [29].

This work aims to list the most useful known solving methods and compare them, primarily in terms of the methods' prerequisites. In chapter 2, we acquaint the reader with the theory of BLPP and with examples showing difficulties of BLPP. When thinking of a way of BLPP solution, the first thing that comes to mind is replacing the lower level problem with its Karush-Kuhn-Tucker (or Fritz-John) optimality conditions. This so-called KKT-reformulation is the main topic of chapter 3. KKT reformulation is a mathematical program with equilibrium constraints (MPEC), however as shown in the said chapter, MPEC and BLPP are not fully equivalent. Since the KKT reformulation has issues with common complementarity constraints being violated in every feasible point, algorithms focus on overcoming nonregularity of KKT reformulation.

Chapter 4 focuses on another common approach - replacing the lower level problem by its optimal value function. While, in contrast with the KKT reformulation, the optimal value function reformulation is fully equivalent to the original BLPP, it generally results in a nonsmooth problem with the optimal value function being difficult to express. An algorithm based on the optimal value function reformulation is described at the end of the chapter.

Chapter 5 aims at a special case of BLPP - a linear BLPP. Even though linearity does in general case not imply a possibility to reduce the BLPP to a linear single level problem, it (especially on the lower level) leads to some significant simplifications. Two algorithms are described in the chapter: An algorithm based on duality and an algorithm based on optimal value function reformulation.

Chapter 6 compares the mentioned algorithms in terms of assumptions.

Chapter 7 reminds basic concepts of mean-risk models and states a bilevel programming problem using a mean-risk model at its lower level.

The rest of the thesis (chapter 8) is dedicated to solving practical the problem stated in chapter 7 on real-life data by implementing some of the aforementioned algorithms.

# 2. Preliminaries

This chapter focuses on the basic theory of bilevel programming. In the first section, we introduce to the reader the possible definitions of the bilevel programming problem, supplied with illustrative examples. The second section 2.2 describes possible types of bilevel programming - optimistic, pessimistic and selection function approaches. In the last section 2.3, we describe another problem which can occur when the upper level constraint in a bilevel programming problem depends on the lower level variable.

## 2.1 The Bilevel Programming problem

This section acquaints the reader with basic definitions and possible notations of the BLPP and illustrates the basic principles on two examples.

First, we state some possible definitions of BLPP. Consider a parametric programming problem

$$\begin{aligned} &\min_y f(x,\, y) \\ &s.t.\ g(x,\, y) \leq 0, \\ &\quad y \in T, \end{aligned} \tag{2.1}$$

where $T \subseteq \mathbb{R}^m$ is a closed set, $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ and $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ are functions. This problem shall be called the *lower level problem* and the function $f$ shall be called the *lower level objective function.* Consider a point-to-set mapping (from $\mathbb{R}^n$ to $\mathbb{R}^m$)

$$Y(x) = \{y \in \mathbb{R}^m : g(x,\, y) \leq 0,\, y \in T\},$$

a function

$$\varphi(x) = \min_y \{f(x,\, y) : y \in Y(x)\}$$

and finally a point-to-set mapping (from $\mathbb{R}^n$ to $\mathbb{R}^m$)

$$\Psi(x) = \{y \in Y(x) : f(x,y) \leq \varphi(x)\} = \arg\min_y \{f(x,\, y) : g(x,\, y) \leq 0,\, y \in T\}.$$

We shall call the mapping $Y$ *(lower level) feasible set mapping*, the function $\varphi$ is called *(lower level) optimal value function* and the mapping $\Psi$ is called *(lower level) solution set mapping.*

*Remark.* We shall denote a point-to-set mapping $X$ from a set $\mathbb{A}$ to a set $\mathbb{B}$ (i.e. a mapping $X : A \to 2^B$, where $2^B$ is the power set of $B$) by $X : A \rightrightarrows B$.

The bilevel programming problem is then given by

$$\begin{aligned} &\min_x F(x,\, y) \\ &s.t.\ G(x,\, y) \leq 0, \\ &\quad (x,\, y) \in \mathbf{gph}\, \Psi, \\ &\quad x \in X, \end{aligned} \tag{2.2}$$

where $X \subseteq \mathbb{R}^n$ is a closed set and $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ and $G : \mathbb{R}^n \to \mathbb{R}^q$ are functions and $\mathbf{gph}\,\Psi = \{(x,\, y) \in \mathbb{R}^n \times \mathbb{R}^m : y \in \Psi(x)\}$ denotes the graph of $\Psi$. The function $F$ is called the *upper level objective funtion*.

Note that the (in general point-to-set) mapping $x \to F(x, \Psi(x))$ does not need to be a function. An example of this issue will be given at the beginning of section 2.2. More detailed view on this topic is given in the first chapter of Dempe [5].

A formal definition of the BLPP, introduced in (2.2) follows:

**Definition 1.** *[3] Let $T \subseteq \mathbb{R}^m$, $X \subseteq \mathbb{R}^n$ be closed sets, let $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$, $G : \mathbb{R}^n \to \mathbb{R}^q$ be functions. The* bilevel programming problem *(BLPP) is given as*

$$\min_x\{F(x,\, y) :\, G(x,\, y) \,\leq\, 0,\, (x,\, y) \in \mathbf{gph}\,\Psi,\, x \in\, X\}, \tag{2.3}$$

*where*

$$\Psi(x) \,=\, \arg\min_y \{f(x,\, y) :\, g(x,\, y) \leq 0,\, y \in T\}. \tag{2.4}$$

A simple example of a bilevel programming problem together with its solution is given in the following (original) example.

*Example.* Consider a bilevel programming problem

$$\min_x\{(x-4)^2 \,+\, (y-3.5)^2 :\, (x,\, y) \in \mathbf{gph}\,\Psi\}.$$

where $\Psi(x)$ is the solution of the following parametric problem:

$$\min_y\{y^3 :\, y \,\geq\, x^2,\, y \,\leq\, 6.25\}.$$

The lower level feasible set is then equal to

$$Y(x) = \{y :\, y \,\geq\, x^2,\, y \,\leq\, 6.25\},$$

the lower level optimal value function is

$$\varphi(x) = x^6,\; x \in \langle -2.5,\, 2.5\rangle,$$

and the lower level solution set mapping is

$$\Psi(x) = x^2,\; x \in \langle -2.5,\, 2.5\rangle.$$

The problem together with the graphical solution is depicted in Figure (2.1). Since $\Psi(x)$ is a singleton for every feasible $x$, the BLPP is equal to the single level problem

$$\min_x\{(x-4)^2 \,+\, (y-3.5)^2 :\, y = x^2, x \in \langle -2.5,\, 2.5\rangle\}.$$

This problem has a global optimal solution $(x^*,\, y^*) = (2,\, 4)$ with $F(2,\, 4) = 4.25$, $Y(2) \,=\, \langle 4,\, 6.25\rangle$, $f(2,\, 4) \,=\, \varphi(2) \,=\, 64$ and $\Psi(2) \,=\, 4$, and one local optimal solution $(x,\, y) = (-1,\, 1)$ with $F(-1,\, 1) = 31.25$, $Y(-1) = \langle 1,\, 6.25\rangle$, $f(-1,\, 1) = \varphi(-1) = 1$ and $\Psi(-1) = 1$.

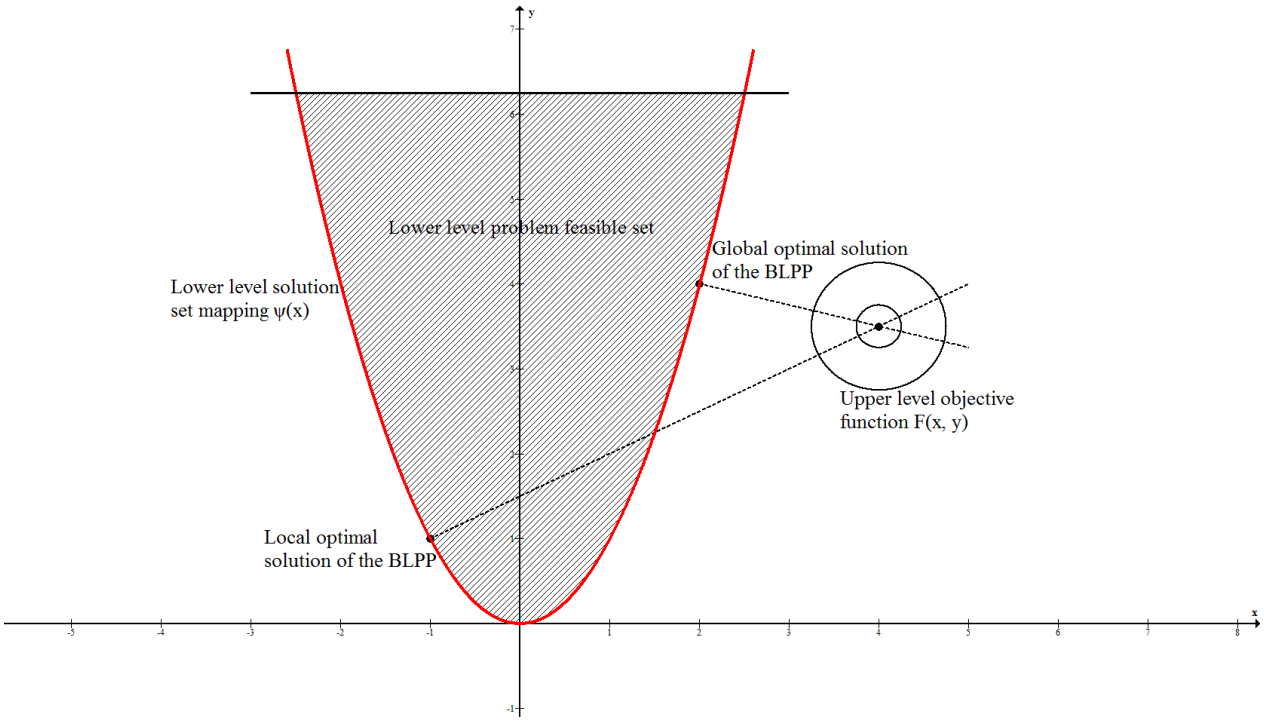


Figure 2.1: A graphical solution to a simple BLPP. The problem has one global optimal solution $(x, y) = (2, 4)$ and one local optimal solution $(x, y) = (-1, 1)$.

*Remark.* The problem (2.3) - (2.4) is equivalent to the problem (2.2) - (2.1). A shortened notation of the BLPP (2.3) - (2.4) is

$$\begin{aligned} &\min_x F(x, y) \\ &s.t.\ G(x, y) \leq 0, \\ &\quad \min_y f(x, y) \\ &\quad s.t.\ g(x, y) \leq 0, \\ &\quad x \in X,\ y \in T. \end{aligned} \tag{2.5}$$

We end this section with an (original) example illustrating one of the major issues of the BLPP: It is generally non-convex, even in spite of such strong conditions like linearity.

*Example.* Consider the following problem:

$$\min_x \{-x + y : y \in \Psi(x);\ x, y \geq 0\}, \tag{2.6}$$

where

$$\Psi(x) = \arg\min_y \{-y : 2x - y \leq 0,\ y \leq 2,\ 2x + y \leq 8,\ y \geq 0\}. \tag{2.7}$$

The whole problem is depicted in Figure 2.2. As we can see, all the functions $F$, $f$, $G$ and $g$ (with $G \equiv 0$) are linear, but the bilevel problem feasible set $\{(x, y) \in \mathbb{R}^2 : 2x - y = 0,\ x \in \langle 0, 1\rangle\} \cup \{(x, y) \in \mathbb{R}^2 : y = 2,\ x \in \langle 1, 3\rangle\} \cup \{(x, y) \in \mathbb{R}^2 : 2x + y = 8\, x \in \langle 3, 4\rangle\}$ is non-convex. That means that even a very simple linear bilevel programming problem is generally not convex.

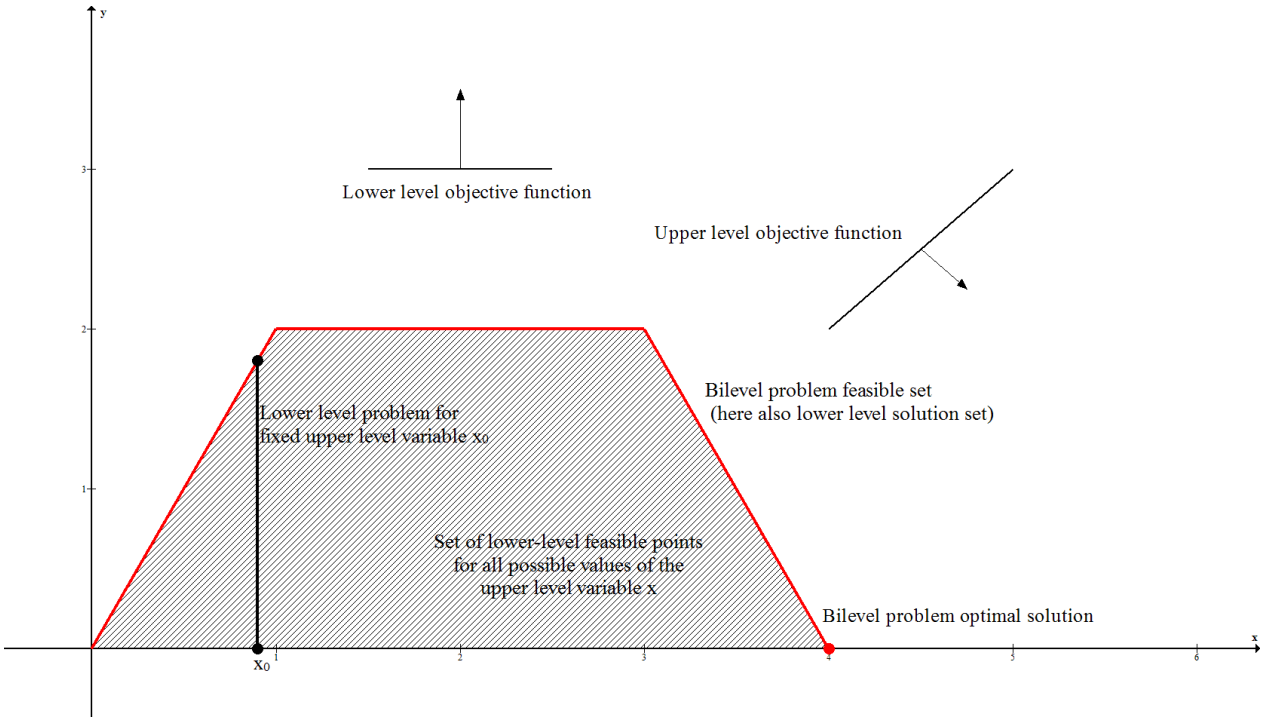


Figure 2.2: A Graphical solution to a simple linear BLPP (problem (2.6) - (2.7)). The bilevel problem feasible set is not convex even though both upper and lower level objective functions are linear as well as all the constraints.

## 2.2 The three approaches to BLPP

The motivation for this section is the second issue with general BLPP: Non-uniqueness of the solution of the lower level problem (the first issue is the earlier mentioned nonconvexity in general case). This implies a necessity for ways how to select only one of the solutions of the lower level problem for each value of the upper level variable.

This section presents three possible ways of doing so - choosing the "best" one, choosing the "worst" one or choosing the "predicted" one. The following simple example a shows problem related to the nonuniqueness of the lower level solution.

*Example.* (Dempe et al. [11], chapter 1) Consider the following problem:

$$\min_x \{x^2 + y : (x, y) \in \Psi(x)\} \tag{2.8}$$

where

$$\Psi(x) = \arg\min_y \{xy : 0 \leq y \leq 1\} \tag{2.9}$$

Then it holds

$$\Psi(x) = \begin{cases} 1, & x < 0, \\ 0, & x > 0, \\ \langle 0, 1 \rangle, & x = 0. \end{cases} \tag{2.10}$$

Thus the mapping $x \to F(x, \Psi(x))$ is, in fact, not a function, since it takes the

following form:

$$F(x, \Psi(x)) = \begin{cases} x^2, & x > 0, \\ x^2 + 1, & x < 0, \\ \{1\} \times \langle 0, 1 \rangle & x = 0. \end{cases} \tag{2.11}$$

The whole problem is depicted in Figure (2.3). As we can see, if this is a leader-follower problem, it has no solution since the leader does not know which value of (the set) $\Psi(0)$ the follower chooses. Denote $y(x_0) \in \Psi(x_0)$ a point from the lower level feasible set for a given $x_0$. If $y(0) = 0$ is chosen, the problem has a (unique) solution $(x,\ y) = (0,\ 0)$ and we speak of *optimistic BLPP*. If any nonzero value $y(0) = y_0$ is chosen, the problem has no solution, because we have $\inf_{x \in \mathbb{R}} F(x, y_0) = 0$, however $\min_{x \in \mathbb{R}} F(x, y_0)$ does not exist. The case when $y(0) = 1$ is taken, is called *pessimistic BLPP*.

**Definition 2.** *(Dempe et al. [11], chapter 1) Let $\Psi(x) = \{y \in Y(x) \cap T : f(x, y) \leq \varphi(x)\}$ be a lower level solution set mapping defined in section 2.1. We shall then define the* optimistic solution function *as*

$$\varphi_o(x) = \min_y \{F(x,\ y) :\ y \in \Psi(x)\} \tag{2.12}$$

*and the* pessimistic solution function *as*

$$\varphi_p(x) = \max_y \{F(x,\ y) :\ y \in \Psi(x)\}. \tag{2.13}$$

*Then the* optimistic bilevel programming problem *is defined as*

$$\min_x \{\varphi_o(x) :\ G(x,\ y) \leq 0,\ x \in X\} \tag{2.14}$$

*and the* pessimistic bilevel programming problem *is defined as*

$$\min_x \{\varphi_p(x) :\ G(x,\ y) \leq 0,\ x \in X\}. \tag{2.15}$$

The optimistic BLPP is the case of leader-follower problems where the follower is able and willing to cooperate (and the leader is aware of it). The pessimistic BLPP is a useful approach for leader-follower problems where the leader knows nothing about the follower's decision and wants to minimize the leader's risk by assuming the worst situation.
The optimistic BLPP is closely related (see below) to the problem (Dempe et al. [11], chapter 1)

$$\min_{x,\ y} \{F(x,\ y) :\ G(x,\ y) \leq 0,\ (x,\ y) \in \mathbf{gph}\, \Psi,\ x \in X\} \tag{2.16}$$

together with the lower level problem

$$\Psi(x) = \arg\min_y \{f(x,\ y) :\ g(x,\ y) \leq 0,\ y \in T\}. \tag{2.17}$$

The monography Dempe [5] focuses on relationships between optimistic and pessimistic BLPP and various reformulations. Generally, if we have a point $x^*$ which is a local minimum of the optimistic solution function on the set $\{G(x,\ y) \leq 0,\ x \in X\}$ and a point $y^* \in \Psi(x^*)$ then the point $(x^*,\ y^*)$ is a local minimum of the problem (2.16) - (2.17) (Dempe et al. [11], chapter 1). The following (original) example shows that in the general case the opposite implication does not hold:

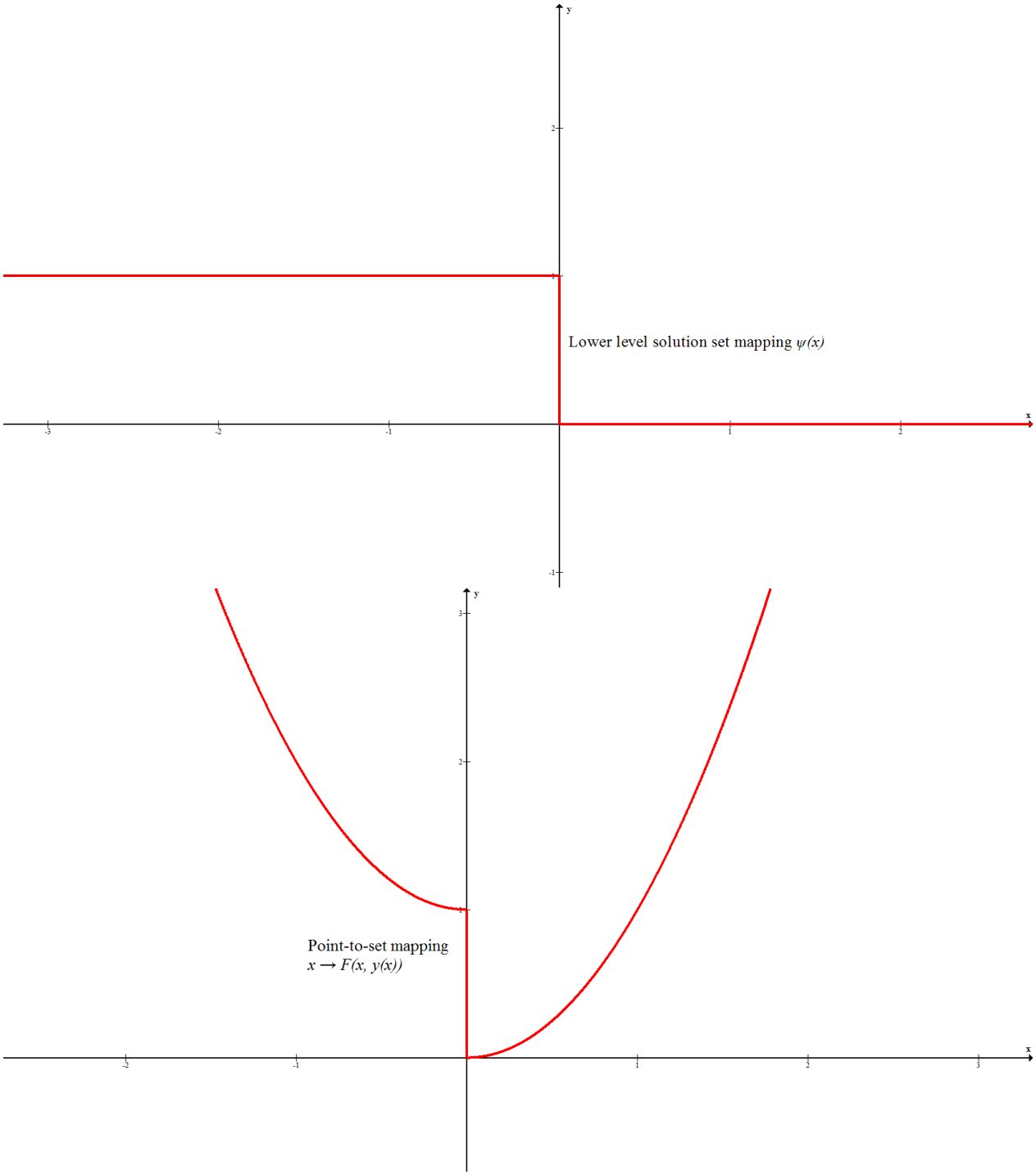


Figure 2.3: Graphical depiction of lower level solution set mapping (the upper graph) and the mapping $x \to F(x, \Psi(x))$ of the problem (2.8) - (2.9). The problem has no solution due to nonuniqueness of solution of the lower level problem for $x = 0$.

*Example.* Consider a problem

$$\min_{x,\,y}\{-x:\ x\in\langle 0,\,1\rangle,\ (x,\,y)\in \mathbf{gph}\,\Psi\}$$

with

$$\Psi(x)=\arg\min_{y}\{-x^2y:\ y\ \in\ \langle 0,\,1\rangle\}.$$

The graphical solution of the problem is depicted in figure 2.4. Then we have

$$\Psi(x)=\begin{cases}\langle 0,\,1\rangle, & x=0\\ 1, & x\in(0,\,1\rangle.\end{cases}$$

The point $(x,\,y)=(0,\,0)$ is then a local minimum of the bilevel programming problem since $F(0,\,0)\ \leq\ F(x,\,y)\ \forall\,(x,\,y)\in U^{0.5}(0,\,0)$. However the optimistic solution function is equal to

$$\varphi_o(x)\ =\ -x,$$

thus has only one local minimum on $\langle 0,\,1\rangle$, which is at $x=1$.

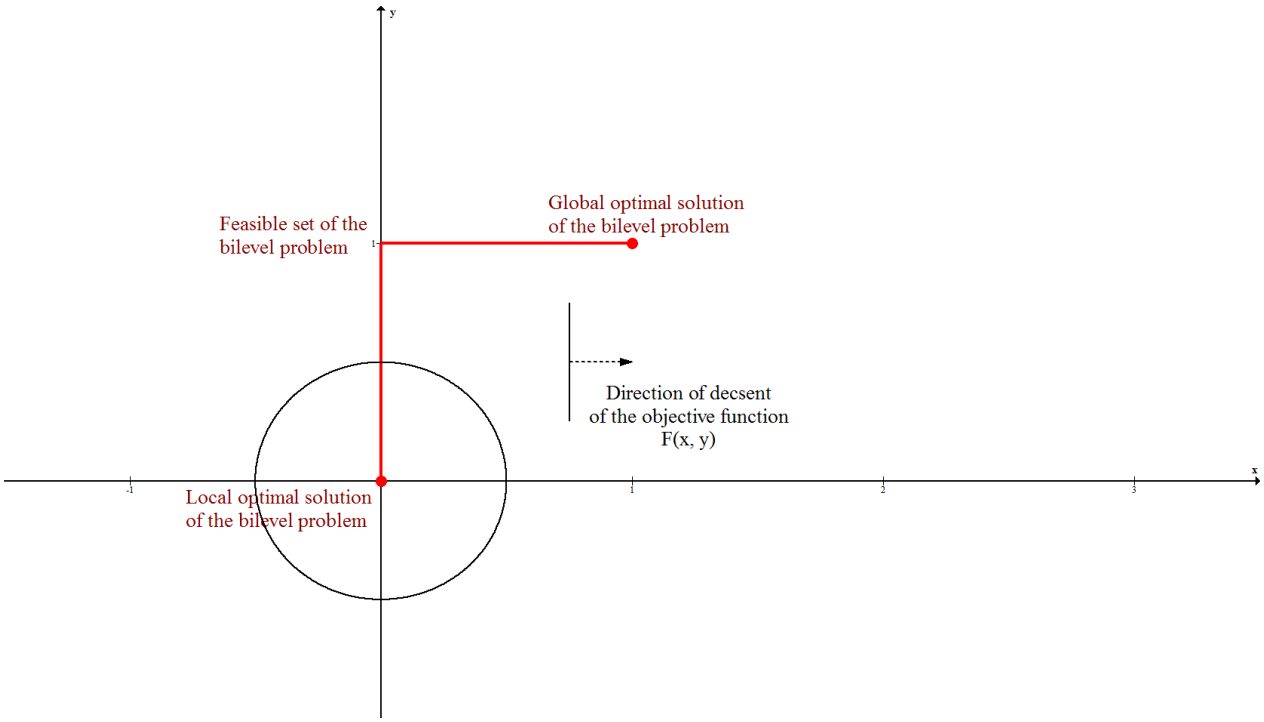


Figure 2.4: A graphical solution to an example illustrating that a local optimal solution of the problem (2.25) - (2.26) does not need to be a local optimal solution to an optimistic bilevel problem.

*Remark.* Even though the problem (2.16) - (2.17) is not equivalent to the optimistic bilevel problem (2.14) together with the lower level problem (2.4), we shall call the problem (2.16) - (2.17) optimistic BLPP as well.

There is nevertheless a third possible scenario: The leader is able to somehow predict the follower's decision and thus for every feasible $x$ select a unique lower level solution $y(x)\ \in\ \Psi(x)$. Then $y:\ \mathbb{R}^n\ \rightarrow\ \mathbb{R}^m$ is a function (compared to $\Psi$ being a point-to-set mapping) and we can give the following definition.

**Definition 3.** *The above mentioned function* $y : \mathbb{R}^n \to \mathbb{R}^m$ *is called* selection function of the point-to-set mapping $\Psi$ *and the problem*

$$\min_x \{F(x, y(x)) : G(x, y) \leq 0, x \in X\} \tag{2.18}$$

*is called* selection function approach bilevel programming problem.

In any case of leader-follower type of problem, we can decide, which of the three approaches should be applied, based on the nature of the problem at hand.

## 2.3 Connecting Constraints

In this short section we show that the BLPP as defined in (2.3) - (2.4) can have an issue with so-called connecting constraints and propose a solution (simplification) to solve this issue.

Consider a BLPP (2.3) - (2.4) with an unconnected feasible set. Then shifting the upper level constraint $G(x, y) \leq 0$ into the lower level problem can result in a (new) bilevel programming problem that has a connected feasible set. Such a problem is illustrated in the following (original) example.

*Example.* Consider the following bilevel programming problem

$$\min_{x,y} \{0.5x - y : y \leq x + 1, y \leq -x + 7, x \geq 0, y \in \Psi(x)\} \tag{2.19}$$

with the lower level problem

$$\Psi(x) = \arg\min_y \{-y : y \leq 2x, y \leq -2x + 12, y \geq 0\}. \tag{2.20}$$

The problem is depicted in figure 2.5. The global optimal solution is $(x, y) = (1, 2)$. The problem also has one local optimal solution $(x, y) = (5, 2)$. If we shift the constraints depending on the lower level variable from the upper level problem to the lower level problem, we get

$$\min_{x,y} \{0.5x - y : x \geq 0, y \in \Psi(x)\} \tag{2.21}$$

with the lower level problem

$$\Psi(x) = \arg\min_y \{-y : y \leq 2x, y \leq -2x + 12, y \leq x + 1, y \leq -x + 7, y \geq 0\}. \tag{2.22}$$

The problem is depicted in figure 2.6. The bilevel programming problem feasible set has been enlarged (the two connected components of the feasible set of the problem (2.19) - (2.20) have been connected into a single connected set). There exists a unique global optimal solution $(x, y) = (3, 4)$ and no local optimal solutions.

*Remark.* Note that both the problems (2.19) - (2.20) and (2.21) - (2.22) are in the form of the problem (2.16) - (2.17), however since the set $\Psi(x)$ is a singleton for every $x$, the problems (2.16) - (2.17) and (2.3) - (2.4) are clearly equivalent.

To overcome the issue of connecting constraints, we shall consider bilevel programming problems where the upper level constraint is not dependent on the lower level variable in the rest of this work. This means the problem (2.3) - (2.4) changes to

$$\min_x\{F(x,\, y) :\, G(x) \,\leq\, 0,\, (x,\, y) \in \mathbf{gph}\, \Psi,\, x \in\, X\}, \tag{2.23}$$

where

$$\Psi(x) \,=\, \arg\min_y \{f(x,\, y) :\, g(x,\, y) \leq 0,\, y \in T\}, \tag{2.24}$$

and the problem (2.16) - (2.17) changes to

$$\min_{x,\, y}\{F(x,\, y) :\, G(x) \,\leq\, 0,\, (x,\, y) \in \mathbf{gph}\, \Psi,\, x \in\, X\} \tag{2.25}$$

together with the lower level problem

$$\Psi(x) = \arg\min_y\{f(x,\, y) :\, g(x,\, y) \leq\, 0,\, y \,\in\, T\}. \tag{2.26}$$

Similarly to the remark at the end of section 2.2, we shall call the problem (2.25) - (2.26) optimistic BLPP even though it's not fully equivalent to the problem (2.14) together with the lower level problem (2.4).

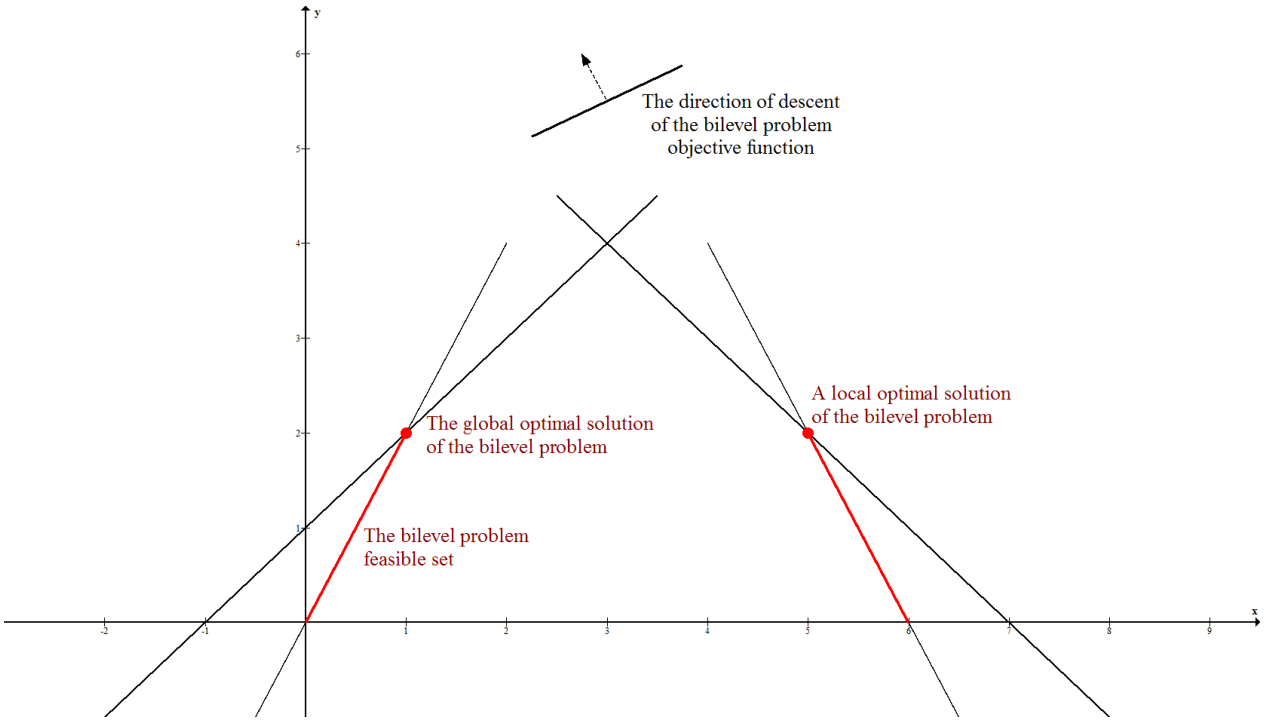


Figure 2.5: A graphical solution to the problem (2.19) - (2.20). Some of the constraints are in the upper level problem, thus not "contributing" to the bilevel programming problem feasible set, which is not connected (depicted in maroon). The problem thus has a unique global optimal solution and one local optimal solution.

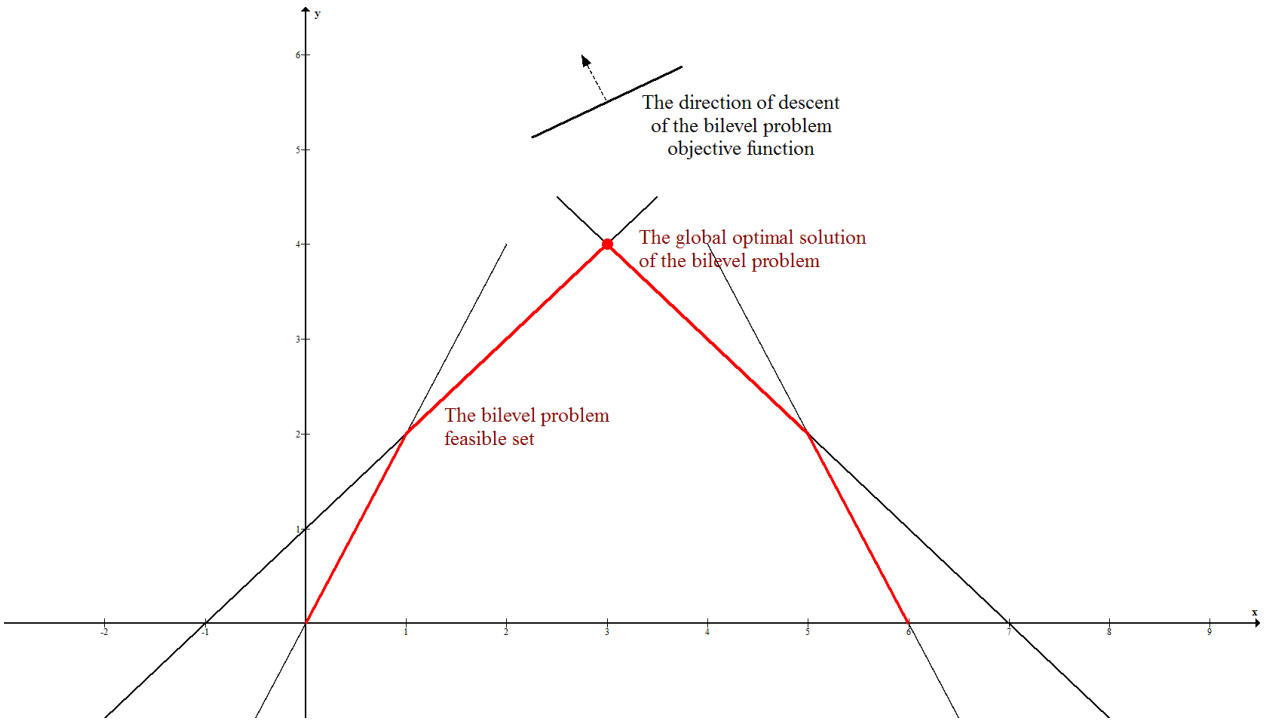


Figure 2.6: A graphical solution to the problem (2.21) - (2.22). All of the constraints depending on the lower level variable have been shifted to the lower level problem. The bilevel programming problem feasible set is connected (depicted in maroon) and the problem has a unique global optimal solution and no local optimal solutions.

# 3. KKT Reformulations of the Bilevel Programming Problem

This chapter focuses on a reduction of a bilevel programming problem to a single level programming problem by replacing the lower level problem with its Karush-Kuhn-Tucker (KKT) conditions.

We remind some necessary basic definitions of various cones, KKT conditions and constraint conditions in section 3.1. Section 3.2 focuses on the KKT reformulation itself, together with a discussion when is the KKT reformulated problem equivalent to the original BLPP. The last section of this chapter proposes three different algorithms based on KKT reformulation, which all focus on dealing with the violation of common complementarity constraints in the KKT reformulated problem. Derivation of stationary and optimality conditions based on KKT reformulation is beyond the extent of this work, an interested reader is referred to Zemkoho [30] and Dempe and Zemkoho [10].

## 3.1 Basic Definitions

This section reminds the reader needed basic definitions of various cones, KKT conditions and the Mangasarian-Fromovitz constraint qualification (MFCQ; the CQ that is mainly used in this work).

First, we need to state definitions of various cones:

**Definition 4.** *(Rockafellar and Wets [24]) Let $M \subseteq \mathbb{R}^n$ be non-empty, let $x \in clo\,(M)$. Then we define the* tangent cone of $M$ in $x$

$$T_M(x) = \{s \in \mathbb{R}^n : \exists\,\{x_k\} \subseteq M, \exists\,\{\lambda_k\}; \lim_{k\to\infty} x_k = x; \lim_{k\to\infty} \lambda_k = \infty, k \in \mathbb{N}; \lim_{k\to\infty} \lambda_k(x_k - x) = s\}. \quad (3.1)$$

**Definition 5.** *(Rockafellar and Wets [24]) Let $M \subseteq \mathbb{R}^n$ be non-empty, let $x \in M$. Then a vector $v$ is* regular normal to $M$ in $x$, *if*

$$\langle v,\, u\rangle \leq 0 \;\forall\, u \in T_M(x).$$

*The set of all regular normals to $M$ in $x$ is denoted by $\hat{N}_M(x)$ and called* regular normal cone to $M$ in $x$.

*We say a vector $w$ is a* normal (normal in the general sense) to $M$ in $x$, *if there are sequences $\{x_k\} \subseteq M$ and $\{w_k\}$ such that $w_k \in \hat{N}_M(x_k)\ \forall\, k \in \mathbb{N}$ and it holds $\lim_{k\to\infty} x_k = x$ and $\lim_{k\to\infty} w_k = w$.*
*The set of all normal vectors is denoted by $N_M(x)$ and called* normal cone to $M$ in $x$.

*Remark.* It is easy to see that both sets $\hat{N}_M(x)$ and $N_M(x)$ are closed cones, so the denomination regular normal cone and normal cone respectively is correct. Furthermore, $\hat{N}_M(x)$ is convex for any point $x \in M$, $M \in \mathbb{R}^n$. Proof of this property (as well as more properties of tangent cone and both normal and regular normal cones) can be found in chapter 6 of [24].

**Definition 6.** *(Rockafellar and Wets [24]) Let $M_k$ be a sequence of subsets in $\mathbb{R}^n$. We define the* Kuratowski lower limit of $M_k$ *as*

$$\begin{aligned}\liminf_{k\to\infty} M_k &= \{x \in \mathbb{R}^n : \limsup_{k\to\infty} dist(x,\, M_k) = 0\} \\ &= \{x \in \mathbb{R}^n : \exists\, n_0 \in \mathbb{N}; \forall\, U(x)\ \forall\, n \geq n_0 : U(x) \cap M_n \neq \emptyset\}\end{aligned}$$

*and the* Kuratowski upper limit of $M_k$ *as*

$$\begin{aligned}\limsup_{k\to\infty} M_k &= \{x \in \mathbb{R}^n : \liminf_{k\to\infty} dist(x,\, M_k) = 0\} \\ &= \{x \in \mathbb{R}^n : \forall\, U(x) :\ U(x) \cap M_n \neq \emptyset \text{ for infinitely many } n \in \mathbb{N}\},\end{aligned}$$

*where $dist(x,\, M_k) = \inf_{a\in M} \|x - a\|$ is the distance between the point $x \in \mathbb{R}^n$ and set $M$ and $U(x)$ denotes an open neighborhood of point $x$.*
*We say the* Kuratowski limit *of the sequence $M_k$ exists if the upper and lower limits are equal:*

$$\lim_{k\to\infty} M_k = \liminf_{k\to\infty} M_k = \limsup_{k\to\infty,\, x\neq 0} M_k.$$

The following theorem is from Rockafellar and Wets [24], the proof is a reformulation (to match the notations used in this work).

**Theorem 1.** *The regular normal cone is equivalent to the* Fréchet normal cone*, which is (for set $M \subseteq \mathbb{R}^n$ and point $x \in M$) defined as*

$$\hat{N}_M^F(x) = \{s \in \mathbb{R}^n :\ \langle s,\, u - x\rangle \leq o(\|u - x\|)\ \forall\, u \in M\}, \tag{3.2}$$

*where the symbol 'o' has the common meaning:*

$$\lim_{x\to 0,\, x\neq 0} \frac{o(f(x))}{f(x)} = 0.$$

*Proof.* First, consider a vector $s \in \mathbb{R}^n \setminus \hat{N}_M^F(x)$. We prove there exists a sequence $\{x_k\} \subseteq M$ with $\lim_{k\to\infty} x_k = x,\ x_k \neq x\ \forall\, k \in \mathbb{N}$, such that

$$\liminf_{k\to\infty} \frac{\langle s,\, x_k - x\rangle}{\|x_k - x\|} > 0. \tag{3.3}$$

Suppose the opposite, i.e. for each sequence $\{x_k\} \subseteq M$ with $\lim_{k\to\infty} x_k = x,\ x_k \neq x\ \forall\, k \in \mathbb{N}$ it holds

$$\limsup_{k\to\infty} \frac{\langle s,\, x_k - x\rangle}{\|x_k - x\|} \leq 0. \tag{3.4}$$

However, (3.4) is equivalent to $max\{0,\, \langle s,\, x_k - x\rangle\}$ being the term $o(\|x_k - x\|)$, which implies $\langle s,\, x_k - x\rangle \leq o(\|x_k - x\|)$. Because we have $\{x_k\} \subseteq M$, this means that $s \in \hat{N}_M^F(x)$. This is a contradiction and thus there exists a sequence $\{x_k\} \subseteq M$ with $\lim_{k\to\infty} x_k = x,\ x_k \neq x\ \forall\, k \in \mathbb{N}$ satisfying (3.3).

We set $u_k = \dfrac{x_k - x}{\|x_k - x\|}$. Then clearly $\|u_k\| = 1$ and $\liminf_{k\to\infty} \langle s,\, u_k\rangle > 0$. Let $u$ be a limit of (any) subsequence of $u_k$, thus $\langle s,\, u\rangle > 0$. Let us set $\dfrac{1}{\|x_k - x\|} = \lambda_k$

from (3.1). Then we have $\lim_{k\to\infty} \lambda_k(x_k - x) = u$, so $u \in T_M(x)$, but $\langle s,\, u\rangle > 0$, hence, from the definition of regular normal cone it follows $s \notin \hat{N}_M(x)$.

Let's now assume $s \in \hat{N}_M^F(x)$. Consider any vector $u \in T_M(x)$ and sequences $\{x_k\} \subseteq M$ and $\{\lambda_k\} \subseteq \mathbb{R}$ with $\lim_{k\to\infty} x_k = x$ and $\lim_{k\to\infty} \lambda_k = \infty$, such that $\lim_{k\to\infty} \lambda_k(x_k - k) = u$. Then (3.2) implies

$$\langle s,\, u_k\rangle \leq o(u_k) = o\Big(\frac{x_k - x}{\|x_k - x\|}\Big) \to 0 \text{ as } k \to \infty$$

and since $\langle s,\, u_k\rangle \to \langle s,\, u\rangle$, we get $\langle s,\, u\rangle \leq 0$, hence we have $s \in \hat{N}_M(x)$. □

*Remark.* The (general) normal cone is the Kuratowski upper limit of a sequence of regular normal cones: Let $M \in \mathbb{R}^n$. For any sequence $\{x_k\}$ of points in $M$ with $\lim_{k\to\infty} x_k = x \in M$ holds

$$N_M(x) = \limsup_{k\to\infty} \hat{N}_M(x_k),$$

i.e.

$$N_M(x) = \{s \in \mathbb{R}^n : \exists\, \{x_k\} \subseteq M,\ \exists\, \{s_k\} \subseteq \mathbb{R}^n \text{ with } \lim_{k\to\infty} x_k = x \in M,\ \lim_{k\to\infty} s_k = s,\ s_k \in \hat{N}_M(x_k)\ \forall\, k \in \mathbb{N}\}. \quad (3.5)$$

For proof see Rockafellar and Wets [24]. The normal cone defined in (3.5) is usually referred to as *Mordukhovich normal cone* in literature (e.g. in Dempe et al. [11]). This denomination shall be used in the rest of this work.

We shall now remind the KKT optimality conditions. These are defined for a single level programming problem, which can be viewed as the lower level problem in the context of bilevel programming.

**Definition 7.** *(Lachout [18]) Consider an optimization problem*

$$\begin{aligned} &\min_x f(x) \\ &s.t.\ g(x) \leq 0, \\ &\quad x \in X, \end{aligned} \quad (3.6)$$

*where $f : \mathbb{R}^n \to \mathbb{R}$ is a real function, $g : \mathbb{R}^n \to \mathbb{R}^p$ is a vector-valued function (i.e. functions $g_1, \ldots, g_p : \mathbb{R}^n \to \mathbb{R}$) and $X \subseteq \mathbb{R}^n$ is a set. Let us denote $I = \{1, \ldots, p\}$.*
*Let $x^* \in X$, $\lambda^* \in \mathbb{R}^p$ and let $\nabla f(x^*)$ and $\nabla g_i(x^*)$, $i \in I$, exist. We say the pair $(x^*,\, \lambda^*)$ fulfills the* Karush-Kuhn-Tucker optimality conditions *(abbreviated as* KKT conditions*), if*

$$\nabla_x L(x^*,\, \lambda^*) = \nabla f(x^*) + \sum_{i=1}^{p} \lambda_i^* \nabla g_i(x^*) = 0, \quad (3.7)$$

$$g_i(x) \leq 0,\ \forall\, i \in I, \quad (3.8)$$

$$\lambda_i^* \geq 0\ \forall\, i \in I,\ \sum_{i=1}^{p} \lambda_i^* g_i(x^*) = 0. \quad (3.9)$$

*(3.7) is called* Optimality Condition*, (3.9) is called* Complementarity Slackness condition *and (3.8) is called* Primal feasibility condition*. The pair fulfilling the KKT conditions is called the* Karush-Kuhn-Tucker point (KKT point)*.*

*Remark.* We shall refer to the problem (3.6) as to non-linear programming problem or non-linear program (NLP). We can also consider an NLP with equality constraints

$$\begin{aligned} &\min_x f(x) \\ &s.t.\ g(x) \leq 0, \\ &\quad h(x) = 0, \\ &\quad x \in X, \end{aligned} \tag{3.10}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a real function, $g : \mathbb{R}^n \to \mathbb{R}^p$ and $h : \mathbb{R}^n \to \mathbb{R}^r$ are vector-valued function (i.e. functions $g_1, \ldots, g_p : \mathbb{R}^n \to \mathbb{R}$ and $h_1, \ldots, h_r : \mathbb{R}^n \to \mathbb{R}$) and $X \subseteq \mathbb{R}^n$ is a set.

**Definition 8.** *(Fukushima and Pang [13]) Consider an NLP 3.10 and denote its feasible set by* $M = \{x \in \mathbb{R}^n : \ g(x) \leq 0,\ h(x) = 0,\ x \in X\}$*. We say that a point* $x^* \in M$ *is* Bouligand stationary (B-stationary) *provided that for every* $s \in T_M(x^*)$ *it holds*

$$\nabla f(x^*)^T s \ \geq \ 0.$$

*Remark.* (Fukushima and Pang [13]) Note that every local optimal solution of the problem 3.10 is a B-stationary point.

**Definition 9.** *(Dempe et al. [11]) Let* $x^* \in \mathbb{R}^n$ *satisfy KKT conditions for some optimization problem. Then we denote the set of all Lagrange multipliers satisfying the KKT conditions for* $x^*$ *by*

$$\Lambda(x^*) = \{\lambda \geq 0 : \ \nabla f(x^*) \ + \ \sum_{i=1}^{p} \lambda_i \nabla g_i(x^*) = 0,\ \sum_{i=1}^{p} \lambda_i g_i(x^*) = 0\}. \tag{3.11}$$

*Remark.* In the case when the functions $f$ and $g$ are not differentiable (but they are convex) we can use

$$\Lambda(x) = \{\lambda \geq 0 : \ 0 \in \partial f(x^*) \ + \ \lambda^T \partial g_i(x^*),\ \lambda^T g(x^*) = 0\}, \tag{3.12}$$

where $\partial f$ (or $\partial g$) denotes the subdifferential of the function $f$ (or $g$, respectively).

**Definition 10.** *(Lachout [18]) We define* set of active constraints *for an NLP at some feasible point* $\overline{x}$ *as* $I_0(\overline{x}) = \{i \in I : \ g_i(\overline{x}) = 0\}$

We need to introduce the following approximation of the tangent cone for the Abadie constraint qualification.

**Definition 11.** *(Geiger and Kanzow [14]) Consider an NLP (3.6). Let, let* $\overline{x}$ *be a feasible point. The* Linearized cone (for NLP) in $x$ *is defined as*

$$L(\overline{x}) = \{s \in \mathbb{R}^n : \ \nabla g_i(\overline{x})^T s \leq 0 \ \forall\, i \in I_0(\overline{x})\}. \tag{3.13}$$

**Definition 12.** *(Geiger and Kanzow [14]) Consider an NLP (3.6). Let $M$ be its set of feasible points and let $\overline{x} \in M$ be a feasible point. We say the* Abadie constraint qualification (ACQ) *holds in $\overline{x}$, if*

$$T_M(\overline{x}) = L(\overline{x}). \tag{3.14}$$

When the ACQ holds at some feasible point $\overline{x}$ of the NLP, the following implication is true

$$\begin{aligned} &\overline{x} \text{ is a local optimal solution of NLP} \\ &\Rightarrow \text{there exists some } \overline{u} \in \mathbb{R}^p \text{ such that } \langle \overline{u}, \overline{x} \rangle \text{ is a KKT-point of NLP.} \end{aligned} \tag{3.15}$$

If the ACQ holds at some feasible point $\overline{x}$ of an NLP, from the definition of linearized cone we have that the KKT conditions are equivalent to

$$0 = \nabla f(\overline{x}) + \hat{N}_M(\overline{x}), \tag{3.16}$$

where $M = X \cap \{x \in \mathbb{R}^n : \ g(x) \leq 0\}$ is the feasible set of the NLP.

**Definition 13.** *(Dempe et al. [11]) Consider an NLP (3.6). Let $\overline{x}$ be a feasible point. We say the* Mangasarian-Fromovitz constraint qualification (MFCQ) *holds at $\overline{x}$ if there exists a vector $s \in \mathbb{R}^n$ such that*

$$\nabla g_i(\overline{x})^T s \ < \ 0 \ \forall \, i \in I_0(\overline{x}). \tag{3.17}$$

*Remark.* (Geiger and Kanzow [14]) The MFCQ for an NLP with equality constraints 3.10 is satisfied at $\overline{x}$ if in addition to the inequality 3.17 gradients $\nabla h_j(x), \ j = 1, \ \ldots \ r$, are linearly independent and it holds

$$\nabla h_i(\overline{x})^T s \ = \ 0 \ \forall \, j \in \{1, \ \ldots \ r\}, \tag{3.18}$$

where $s \in \mathbb{R}^n$ is the vector from the definition of MFCQ (Definition 13).

Note that MFCQ implies ACQ (see Geiger and Kanzow [14]).

**Definition 14.** *(Lachout [18]) Consider an NLP (3.6). We say the* Slater's condition *is satisfied if there exists $\overline{x} \in X$ such that $g_i(\overline{x}) < 0$ for all $i \in I$.*

**Definition 15.** *(Dempe and Dutta [6]) Consider an NLP (3.6). Let $\overline{x}$ be a feasible point. We say the* Constant rank constraint qualification (CRCQ) *holds at $\overline{x}$ if there exists an open neighborhood $U(\overline{x})$ such that for every subset $J \subseteq I_0(\overline{x})$ the family of gradient vectors $\{\nabla g_i(x) : i \in J\}$ has the same rank for every $x \in U(\overline{x})$.*

The set of all Lagrange multipliers $\Lambda(x)$ can be also viewed as a point-to-set mapping $x \to \Lambda(x)$. Under mild conditions, this mapping has a property called upper semicontinuity that is important for solving bilevel programming problems (as we will see later in this chapter).

**Definition 16.** *(Dempe et al. [11]) A point-to-set mapping $\Gamma : \mathbb{R}^m \rightrightarrows \mathbb{R}^n$ (a mapping from $\mathbb{R}^m$ to $2^{\mathbb{R}^m}$) is called* upper semicontinuous *at a point $x_0 \in \mathbb{R}^m$, if for each open set $V \supseteq \Gamma(x^0)$ there is an open set $U$ with $x^0 \in U$ such that $\Gamma(x) \subseteq V$ for all $x \in U$.*

For the theorem stating the conditions under which is the mapping $x \to \Lambda(x)$ upper semicontinuous, we first need the following definition.

**Definition 17.** *Consider a problem 2.26 and a point $x \in \mathbb{R}^n$. The set*

$$SP(x) = \{y \in \mathbb{R}^m : \Lambda(x, y) \neq \emptyset\}$$

*is then called the* set of stationary solutions.

**Theorem 2.** *(Dempe et al. [11], Theorem 3.3) Consider a problem 2.26 at $x = x^*$. Let $\{y : g(x^*, y) \leq 0\}$ be nonempty and compact, the functions $f$, $g_i$ be at least continuously differentiable for all $i \in I$ and let (MFCQ) be satisfied at all points $y \in Y(x^*)$. Let $y^* \in SP(x^*)$. Then, the point-to-set mapping $(x, y) \to \Lambda(x, y)$ is upper semicontinuous at $(x^*, y^*)$.*

*Proof.* Can be found in Dempe et al. [11]. □

## 3.2 KKT Reformulation

The first idea that comes to mind if we want to reduce a bilevel programming problem to a single level problem, is replacing the lower level problem with its Karush-Kuhn-Tucker conditions. This is called KKT reformulation.

This section focuses on the KKT reformulation, and on a discussion on equality of the KKT reformulated problem with the original BLPP, with theorems stating when a solution of the KKT reformulated problem is also a solution of the original problem and vice-versa.

We shall now introduce the KKT reformulation (Dempe et al. [11]). If problem (2.25) - (2.26) is replaced by

$$\min\{F(x, y) : G(x) \leq 0, 0 \in \partial_y f(x, y) + N_{Y(x)\cap T}(y)\}, \tag{3.19}$$

where $\partial_y f(x, y)$ denotes the subdifferential of the function $y \to f(x, y)$, we talk about *primal KKT reformulation*.
If the problem (2.25) - (2.26) is replaced by

$$\begin{aligned} &\min_{x,y,\lambda} F(x, y) \\ &s.t.\ G(x) \leq 0, \\ &\quad \nabla f(x, y) + \sum_{i=1}^{p} \lambda_i \nabla g_i(x, y) = 0 \\ &\quad g_i(x, y) \leq 0 \ \forall\, i \in \{1, \ldots, p\}, \\ &\quad \sum_{i=1}^{p} \lambda_i g_i(x, y) = 0, \\ &\quad \lambda_i \geq 0 \ \forall\, i \in \{1, \ldots, p\}, \end{aligned} \tag{3.20}$$

we talk about *classical KKT reformulation*.

As we can see, both reformulations consider the optimistic bilevel programming problem (2.25) - (2.26). KKT reformulations for pessimistic BLPP are more complicated and can be found in (Zemkoho [30]).

We can immediately notice that stationary points and local optimal solutions of the lower level problem are added to $\Psi$ in the case of nonconvexity of the lower level problem in both (3.19) and (3.20).

Thus, for equivalence of the primal KKT reformulation (3.19) with the problem (2.25) - (2.26) we need to assume convexity of the function $y \to f(x, y)$ and convexity of the sets $Y(x)$ and $T$. For equivalence of the classical KKT reformulation with problem (2.25) - (2.26) we furthermore need to assume that $X = \mathbb{R}^n$ and $T = \mathbb{R}^m$, all the functions $y \to g_i(x, y)$ are convex and e.q the MFCQ (or Slater's condition) holds for the lower level problem. Moreover since the KKT conditions used in (3.20) clearly require differentiability of the functions $f$ and $g_i$ for all $i = 1, \ldots, p$, the classical KKT reformulation (3.20) requires differentiability of these function as well.

*Remark.* (Dempe et al. [11], chapter 3) In a general case (i.e. when functions $f$ and $g$ are not differentiable), the following classical KKT reformulation can be used:

$$\begin{aligned} &\min_{x,\,y,\,\lambda} F(x,\,y) \\ &s.\,t.\; G(x) \leq 0, \\ &\qquad 0 \in \partial_y f(x,\,y) + \lambda^T \partial_y g(x,\,y) \\ &\qquad g(x,\,y) \leq 0 \\ &\qquad \lambda^T g(x,\,y) = 0,\; \lambda \geq 0, \\ &\qquad x \in X. \end{aligned} \tag{3.21}$$

Clearly, this problem is reduced to 3.20 when the functions $f$ and $g$ are differentiable. Furthermore if the Slater's condition holds at x, from the definition of the set $Y(x)$ and from the definition of normal cone we have $N_{Y(x)} = \{s \in \mathbb{R}^m : \exists\lambda > 0 :\; \lambda^T g(x,\,y) = 0,\; s = \lambda^T \partial_y g(x,\,y)\}$ and primal KKT reformulation (3.19) reduces to classical KKT reformulated problem (3.20).

The following theorem, including an alternative original proof, shows the main issue of the KKT reformulation, which is violation of common constraint qualifications in every feasible point.

**Theorem 3.** *(Dempe et al. [11]), (Scheel and Scholtes [25]) The Mangasarian-Fromovitz constraint qualification is violated at every feasible point of the problem 3.21.*

*Proof.* Let $(x^*,\, y^*, \lambda^*) \in \mathbb{R}^{n+m+p}$ be a feasible point of the problem 3.21 and assume the MFCQ holds at $(x^*,\, y^*, \lambda^*)$. Let $s \in \mathbb{R}^n$ be the direction from the definition of MFCQ. Choose an index $i_0 \in \{1, \ldots, p\}$. Since the point $(x^*,\, y^*, \lambda^*) \in \mathbb{R}^{n+m+p}$ is feasible, it holds

$$\lambda^*_{i_0} g_{i_0}(x^*,\, y^*) = 0, \tag{3.22}$$

$$g_{i_0}(x^*,\, y^*) \leq 0, \tag{3.23}$$

$$\lambda^*_{i_0} \geq 0. \tag{3.24}$$

Hence, one of the following three possibilities occurs:

1. $g_{i_0}(x^*, y^*) < 0,\ \lambda^*_{i_0} = 0,$
2. $g_{i_0}(x^*, y^*) = 0,\ \lambda^*_{i_0} > 0,$
3. $g_{i_0}(x^*, y^*) = 0,\ \lambda^*_{i_0} = 0.$

In the first case it holds $\nabla\lambda^*_{i_0} s_{n+m+i_0} = s_{n+m+i_0} > 0$. This, together with $\lambda^*_{i_0} = 0$, implies

$$(\nabla\lambda^*_{i_0} g_{i_0}(x^*, y^*))^T s = \\ = \lambda^*_{i_0}(\nabla g_{i_0}(x^*, y^*))^T(s_1, \ldots, s_{m+n}) + g_{i_0}(x^*, y^*)s_{m+n+i_0} < 0, \quad (3.25)$$

which is a contradiction with MFCQ.

In the second case it holds

$$\nabla g_{i_0}(x^*, y^*))^T s = (\nabla g_{i_0}(x^*, y^*))^T(s_1, \ldots, s_{m+n}) < 0,$$

which, together with $g_{i_0}(x^*, y^*) = 0$ and $\lambda^*_{i_0} > 0$ again implies 3.25, which is a contradiction with MFCQ.

Finally, in the third case, it holds both $s_{n+m+i_0} > 0$ and $(\nabla g_{i_0}(x^*, y^*))^T(s_1, \ldots, s_{n+m}) < 0$, which again implies 3.25 and we have a contradiction with MFCQ. □

Note that the problem 3.20 (or 3.21, respectively) is not fully equivalent with (2.23) - (2.24) due to the introduction of new variables $\lambda$. The following theorem states conditions for an equivalence of local optimal solutions:

**Theorem 4.** *(Dempe et al. [11], Theorem 3.1). Let the lower level problem (2.26) be convex (i.e. let functions $f$ and $g_i, i = 1, \ldots, p$, be convex) and let Slater's condition hold for all $x \in X$ with $\varphi(x) \neq \emptyset$. A feasible point $(x^*, y^*)$ of problem (2.25) - (2.26) is a local optimal solution of the problem (2.25) - (2.26) if and only if $(x^*, y^*, \lambda^*)$ is a local optimal solution of the problem (3.21) for each $\lambda^* \in \Lambda(x^*, y^*)$.*

*Proof.* Can be found in (Dempe et al. [11]). □

*Remark.* If the point $(x^*, y^*)$ from Theorem 4 is a local optimal solution of the problem (2.25) and the Slater's condition is satisfied only at $x^*$, it still holds that $(x^*, y^*, \lambda^*)$ is a local optimal solution of the problem (3.21) for each $\lambda^* \in \Lambda(x^*, y^*)$.

In Theorem 4, the condition that $(x^*, y^*, \lambda^*)$ needs to be a local optimal solution of the problem (3.21) for each $\lambda^* \in \Lambda(x^*, y^*)$ is necessary. The following example shows that if $(x^*, y^*, \lambda^*)$ is local optimal solution of the problem (3.21) only for some $\lambda^* \in \Lambda(x^*, y^*)$, the point $(x^*, y^*)$ is not necessary a local optimal solution of the problem (2.25) - (2.26). The example is from Dempe et al. [11], however, the proof that there exists a neighborhood of $(0, 1, 1, 0)$ such that $\lambda_1 > 0$ on this neighborhood is original.

*Example.* Consider an optimistic bilevel programming problem

$$\min_{x,\,y}\{(x-1)^2 + (y-1)^2 :\ (x,\,y) \in \mathbf{gph}\Psi,\ x \in \langle -1,\,1\rangle\}$$

where

$$\Psi(x) = \arg\min_{y}\{-y :\ y \le x+1,\ y \le -x+1,\}.$$

The KKT conditions for the lower level problem are

$$\begin{aligned} 0 &= -1 + \lambda_1 + \lambda_2,\\ y &\le -x+1,\\ y &\le x+1,\\ 0 &= \lambda_1(-x+y-1) + \lambda_2(x+y-1),\\ \lambda_1,\,\lambda_2 &\ge 0. \end{aligned}$$

The lower level optimal value function equals to

$$\varphi(x) = \begin{cases} x+1, & x \le 0,\\ -x+1, & x \ge 0, \end{cases}$$

and the Lagrange multipliers are

$$\Lambda(x,\,y) = \begin{cases} (1,\,0), & x < 0,\\ (0,\,1), & x > 0,\\ conv\{(1,\,0),\,(0,\,1)\}, & x = 0. \end{cases}$$

The related MPEC (KKT-reformulated problem) is

$$\begin{aligned} \min_{x,\,y,\,\lambda}\ & (x-1)^2 + (y-1)^2\\ s.t.\ & 0 = -1 + \lambda_1 + \lambda_2,\\ & y \le -x+1,\\ & y \le x+1,\\ & 0 = \lambda_1(-x+y-1) + \lambda_2(x+y-1),\\ & \lambda_1,\,\lambda_2 \ge 0,\ -1 \le x \le 1. \end{aligned}$$

The BLPP has clearly one global optimal solution $(x^*,\,y^*) = (0.5,\,0.5)$ and no other local optimal solution. We will now show that the point $(0,\,1,\,1,\,0)$ is a local optimal solution to the related MPEC:
The set $\{y :\ g(0,\,y) \le 0\} = \{0\} \times \langle 0,\,1\rangle$ is clearly nonempty and compact. The functions $f$, $g_1$, $g_2$ are linear, thus continuously differentiable. For the set of active constraints at the point $(0,\,y)$ it holds

$$I_0(0,\,y) = \begin{cases} \{1,\,2\}, & y = 1,\\ \emptyset, & y \in \langle 0,\,1\rangle, \end{cases}$$

Thus for the MFCQ to be satisfied for every $y \in Y(0) = \langle 0,\,1\rangle$, we need to find a vector $s \in \mathbb{R}^2$ such that $\nabla g_1(0,\,1)^T s = (1,\,1)^T s < 0$ and $\nabla g_2(0,\,1)^T s = (-1,\,1)^T s < 0$. We can take for example $s = (0,\,-1)$. We have just verified

the assumptions of Theorem 2, thus the mapping $\Lambda$ is upper semicontinuous at $(x,\, y) = (0,\, 1)$, hence for each open set $V \supseteq conv\{(1,\, 0),\, (0,\, 1)\}$ there exists an open neighborhood $U$ of the point $(0,\, 1)$ such that $\lambda_1 > 0\ \forall\, (x,\, y) \in U$. This implies that the first constraint $y \leq x + 1$ is active on some open neighborhood of the point $(0,\, 1,\, 1,\, 0)$ and thus $x \leq 0$. If we substitute this into the upper level objective function, we get

$$F(x,\, x+1) = (x-1)^2 + x^2 \geq 1.$$

Since $F(0,\, 1) = 1$, we have just shown that $(0,\, 1,\, 1,\, 0)$ is a local minimum of the MPEC.

The following theorem states when is the KKT reformulated problem equivalent to the original BLPP in respect of the global optimal solution.

**Theorem 5.** *(Dempe and Dutta [6], Theorem 2.3) Let $(x^*,\, y^*, \lambda^*)$ be a global optimal solution of problem 3.21. Let the lower level problem (2.26) in (2.25) be convex (i.e. let functions $f$ and $g_i, i = 1,\, \ldots,\, p$, be convex) and assume that Slater's condition is satisfied for the lower level problem for each $x \in X$. Then, $(x^*,\, y^*)$ is a global optimal solution of the bilevel programming problem (2.25) - (2.26).*

*Proof.* Can be found in Dempe and Dutta [6]. □

*Remark.* An opposite implication of Theorem 5 with similar mild assumptions is proven in Dempe and Dutta [6], Theorem 2.1.

The following example from Dempe et al. [11] shows that the assumption of Slater's condition being satisfied at every $x \in X$ is necessary.

*Example.* Consider the bilevel programming problem

$$\min_{x,\, y_1,\, y_2} \{x :\ x \geq 0,\, (y_1,\, y_2) \in \Psi(x)\}$$

with the lower level problem

$$\min_{y_1,\, y_2} \{y_1 :\ y_1^2 - y_2 \leq x,\, y_1^2 + y_2 \leq 0\}.$$

Then, for $x = 0$ is $y = (0,\, 0)$ the only feasible point for the lower level problem, and the Slater's condition is violated for $x = 0$. The KKT conditions for the lower level problem are

$$\begin{aligned}
0 &= 1 + 2\lambda_1 y_1 + 2\lambda_2 y_1,\\
0 &= -\lambda_1 y_2 + \lambda_2 y_2,\\
y_1^2 - y_2 &\leq x,\\
y_1^2 + y_2 &\leq 0,\\
0 &= \lambda_1(y_1^2 - y_2 - x) + \lambda_2(y_1^2 + y_2),\\
\lambda_1,\, \lambda_2 &\geq 0.
\end{aligned}$$

The solution of the lower level problem is

$$\Psi(x) = \begin{cases} (0,\, 0), & x = 0\\ (-\sqrt{x/2},\, -x/2), & x > 0, \end{cases}$$

and the Lagrange multipliers are $\lambda_1 = \lambda_2 = \dfrac{1}{4\sqrt{x/2}}$ for $x > 0$. For $x = 0$, the lower level problem is not regular and there exists no KKT-point.

The bilevel problem has unique global optimal solution $(x,\, y_1,\, y_2) = (0,\, 0,\, 0)$ and no other local optimal solutions. The related MPEC is

$$\begin{aligned} \min_{x,\, y_1,\, y_2,\, \lambda} \; & x \\ s.t.\; 0 &= 1 + 2\lambda_1 y_1 + 2\lambda_2 y_1, \\ 0 &= -\lambda_1 y_2 + \lambda_2 y_2, \\ y_1^2 - y_2 &\le x, \\ y_1^2 + y_2 &\le 0, \\ 0 &= \lambda_1(y_1^2 - y_2 - x) + \lambda_2(y_1^2 + y_2), \\ \lambda_1,\, \lambda_2 &\ge 0. \end{aligned} \tag{3.26}$$

The feasible set of this problem is the set

$$\left\{\left(t,\, -\sqrt{t/2},\, -t/2,\, \frac{1}{4\sqrt{t/2}},\, \frac{1}{4\sqrt{t/2}}\right) :\; t > 0\right\}.$$

From here we can see that the MPEC (3.26) has no optimal solution since we have

$$F(x,\, y_1,\, y_2) = x \searrow 0, \text{ for } x \searrow 0,$$

however, there exists no point feasible to the MPEC with $x = 0$.

It might seem that the BLPP is a special case of MPEC due to the possibility of KKT-reformulation, which is indeed an MPEC. This was a common belief until the publication of Dempe and Dutta [6]. Theorem 5 together with the following remark imply that under mild assumptions (convexity of the lower level problem and Slater's condition) a global optimal solution of the KKT-reformulation (3.21) is associated with a global optimal of an optimistic BLPP (2.25) - (2.26).

On the other hand, the assumption that $(x^*,\, y^*,\, \lambda^*)$ needs to be a local optimal solution of the problem (3.21) for each $\lambda^* \in \Lambda(x^*,\, y^*)$ in Theorem 4 means we need to check local optimality of every feasible solution $(x^*,\, y^*,\, \lambda^*)$, $\lambda^* \in \Lambda(x^*,\, y^*)$, of the problem (3.21). This can be an infinite number of points with only one of them not being a local optimal solution. In other words, not every local optimal solution of the KKT reformulation needs to be associated with a local optimal solution of the original bilevel programming problem. (Dempe and Dutta [6])

Thus, whereas we can say that the BLPP is a special case of MPEC in terms of the global optimal solution, in general case we can not say the same thing in terms of a local optimal solution.

The possibly infinite number of points needed to be checked for local optimality can be reduced to a finite number by adding constant rank constraint qualification to the assumptions of Theorem 4, which has been proven in Dempe and Dutta [6] or Dempe et al. [11]. An original modification of the proof for the following theorem

**Theorem 6.** *(Dempe and Dutta [6], Corollary 3.3) Let the lower level problem (2.26) be convex (i.e. let functions $f$ and $g_i, i = 1, \ldots, p$, be convex), let $(x^*, y^*, \lambda^*)$ be a feasible point of the problem (3.21). Let Slater's condition and (CRCQ) be satisfied for the lower level problem (2.26) at the point $x^*$. Then, if the point $(x^*, y^*, \lambda^*)$ is a local optimal solution of the problem (3.21) for all $\lambda^*$ that are vertices of the set $\Lambda(x^*, y^*)$, the point $(x^*, y^*)$ is a local optimal solution of the problem (2.25) - (2.26).*

*Proof.* Let $(x^*, y^*)$ not be a local optimal solution of the BLPP. Then there exists a sequence $\{(x^k, y^k)\} \subset \mathbf{gph}\,\Psi$ with a limit $(x^*, y^*)$ and with $x^k \in X$ satisfying $F(x^k, y^k) < F(x^*, y^*)\ \forall k \in \mathbb{N}$. Since the Slater's condition is satisfied at $x^*$, and thus, due to persistence in some open neighborhood of $x^*$, there exists a point $\lambda^k$ that is a vertex of the set $\Lambda(x^k, y^k)\ \forall k \in \mathbb{N}$. The sequence $\{\lambda^k\}$ has an accumulation point $\hat{\lambda} \in \Lambda(x^*, y^*)$ by upper semicontinuity of the Lagrange multiplier set mapping. The CRCQ implies that $\hat{\lambda}$ is a vertex of the set $\Lambda(x^*, y^*)$. We have found a sequence $\{(x^k, y^k, \lambda_k)\}$ of feasible solutions of the problem 3.21 converging to a feasible point $(x^*, y^*, \hat{\lambda})$ such that $F(x^k, y^k) < F(x^*, y^*)\ \forall k \in \mathbb{N}$. Thus $(x^*, y^*, \lambda^*)$ is not a local optimal solution of the problem 3.21 for some $\lambda^*$ that is a vertex of the set $\Lambda(x^*, y^*)$.
Hence, $(x^*, y^*)$ is a local optimal solution of the bilevel programming problem (2.25) - (2.26). □

Further reading on KKT reformulated problem (3.21) can be found for example in Zemkoho [30], Dempe and Zemkoho [10] or chapter 3 in Dempe et al. [11].

## 3.3 Algorithms based on KKT reformulation

Since common constraint qualifications are violated in the KKT-reformulated problem (3.21) (see Theorem 3), it usually cannot be implemented into a solver software. Three possible methods that circumvent this problem are described in this section.

### 3.3.1 Reduction into single level mixed integer problem

This algorithm is proposed in (Dempe et al. [11]). The problem (3.21) can be replaced by

$$\begin{aligned} \min_{x,\,y,\,z,\,\lambda}\ & F(x,\,y) \\ s.\,t.\ & G(x) \le 0, \\ & 0 \in \partial_y f(x,\,y) + \lambda^T \partial_y g(x,\,y), \\ & 0 \le \lambda_i \le Q z_i,\ i = 1,\,\ldots\,p, \\ & 0 \ge g_i(x,\,y) \ge -Q(1 - z_i),\ i = 1,\,\ldots\,p, \\ & x \in X,\ z_i \in \{0,\,1\},\ i = 1,\,\ldots\,p, \end{aligned} \tag{3.27}$$

where $Q$ is a sufficiently large real number. In the case when MFCQ is satisfied for the lower level problem and the set $\{(x,\,y) :\ g(x,\,y) \le 0,\ G(x) \le 0,\ x \in X\}$

is compact, the number $Q$ is bounded. Since the problem (3.27) is a mixed integer problem (MIP), it can be solved by using MIP solving methods (which are usually implemented in commercial solving software). These methods typically find the global optimal solution, thus implementing this reformulation into a solver software most likely results in the global optimal solution of the BLPP.

### 3.3.2 Regularization of the KKT transformation

This algorithm is proposed in (Dempe et al. [11]). As mentioned, the classical KKT reformulation (3.21) has a problem with the failure of constraint qualifications, it cannot be solved by algorithms requiring regularity. The problem can, however, be regularized:
Let $T = \mathbb{R}^m$, $X = \mathbb{R}^n$ and assume that the functions $f$ and $g_i$, $i \in I$, are differentiable. We replace the problem (3.21) with

$$\begin{aligned} &\min_{x,\,y,\,\lambda} F(x,\,y) \\ &s.t.\; G(x) \leq 0, \\ &\quad 0 \in \partial_y f(x,\,y) + \lambda^T \partial_y g(x,\,y) \\ &\quad g(x,\,y) \leq 0,\, \lambda \geq 0, \\ &\quad -\lambda^T g(x,\,y) \leq \varepsilon \end{aligned} \tag{3.28}$$

and solve a sequence of such problems with $\varepsilon = \varepsilon_k$ satisfying $\varepsilon_k \searrow 0$. Assume that the set $\mathcal{M} = \{(x,\,y) :\; g(x,\,y) \leq 0,\; G(x) \leq 0\}$ is nonempty and bounded. Further assume that the lower level problem (2.26) is a convex parametric optimization problem and that Slater's condition is satisfied for each x with $(x,\,y) \in \mathcal{M}$ for some $y$. Then problem 3.21 has a global optimal solution $(x^*,\,y^*,\,\lambda^*)$ (see Dempe et al. [11]). It clearly holds

$$F(x_k,\,y_k,\,\lambda_k) \leq F(x^*,\,y^*,\,\lambda^*),$$

where $F(x_k,\,y_k,\,\lambda_k)$ is a global optimal solution of the problem (3.28) with $\varepsilon = \varepsilon_k$. Then the sequence $\{(x_k,\,y_k,\,\lambda_k)\}_{k=1}^{\infty}$ of global optimal solutions of a sequence of problems (3.28) with $\varepsilon = \varepsilon_k$, $k = 1,\,2,\,\ldots$, has an accumulation point and by Theorem 5 each accumulation point $(x^*,\,y^*)$ of a corresponding sequence $\{(x_k,\,y_k)\}_{k=1}^{\infty}$ is a global optimal solution of the problem (2.25) - (2.26).

### 3.3.3 Smoothing Continuation Method

The previous subsection presented one way how to regularize the KKT reformulated problem (3.21). An alternative approach of regularization is proposed in this subsection.

The smoothing continuation algorithm is described for a general MPEC in chapter 3 of Fukushima and Pang [13], whereas a version for BLPP is from chapter 3 of Dempe et al. [11].

The main idea of smoothing continuation method is replacing the complementarity constraint

$$\lambda^T g(x,\,y) \geq 0$$

with a nonlinear complementarity problem function:

**Definition 18.** *(Yamashita and Fukushima [27]) A function* $\phi : \mathbb{R}^2 \to \mathbb{R}$ *is called a* nonlinear complementarity problem function *(NCP-function) if it satisfies the following equivalence:* $\phi(a, b) = 0$ *if and only if* $ab = 0,\ a \geq 0,\ b \geq 0$.

**Definition 19.** *(Fukushima and Pang [13]) The* perturbed Fischer-Burmeister function $\kappa : \mathbb{R}^2 \times (0, \infty) \to \mathbb{R}$ *is defined by*

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

*The function* $\kappa^0 : \mathbb{R}^2 \to \mathbb{R}$ *defined by* $\kappa^0(a, b) = \kappa(a, b, 0)$ *is called* Fischer-Burmeister function.

*Remark.* (Yamashita and Fukushima [27]) The Fischer-Burmeister function is an NCP-function. Since the definition of the perturbed Fischer-Burmeister functions assumes $\varepsilon > 0$, it holds

$$\kappa(a, b, \varepsilon) = 0 \Leftrightarrow a > 0,\ b > 0,\ ab = \frac{\varepsilon}{2}.$$

Assume that the functions $F$, $f$, $G$, $g$ are all differentiable and that $T = \mathbb{R}^m$ and $X = \mathbb{R}^n$. Then we can replace the problem (3.21) with

$$\begin{aligned} &\min_{x,\,y,\,\lambda} F(x, y) \\ &s.t.\ G(x) \leq 0, \\ &\qquad \nabla_y f(x, y) + \lambda^T \nabla_y \partial_y g(x, y) = 0, \\ &\qquad \Phi(x, y, \lambda, \varepsilon) = 0, \end{aligned} \tag{3.29}$$

where

$$\Phi(x, y, \lambda, \varepsilon) = \begin{pmatrix} \kappa(\lambda_1, -g_i(x, y), \varepsilon) \\ \vdots \\ \kappa(\lambda_p, -g_p(x, y), \varepsilon) \end{pmatrix}. \tag{3.30}$$

A sequence of problems (3.29) with $\varepsilon = \varepsilon_k$ satisfying $\varepsilon_k \searrow 0$ is solved. A sequence of optimal solutions $\{(x_k, y_k, \lambda_k)\}_{k=1}^{\infty}$ is obtained this way. An accumulation point $(x^*, y^*, \lambda^*)$ of this sequence, however, does not necessarily exist and if it does, it does not necessarily need to be a global optimal solution of the problem (3.21).

**Theorem 7.** *(Fukushima and Pang [13]), (Dempe et al. [11]). Let* $(x_k, y_k, \lambda_k)$ *be a feasible solution of the problem (3.29) for* $\varepsilon = \varepsilon_k > 0$. *If* $(x_k, y_k, \lambda_k) \to (x^*, y^*, \lambda^*)$ *for* $\varepsilon = \varepsilon_k \searrow 0$ *as* $k \to \infty$, *then* $(x^*, y^*, \lambda^*)$ *is a feasible solution of the problem (3.21).*

*Proof.* Can be found in (Fukushima and Pang [13], Lemma 3.2). □

A theorem describing conditions under which the limit $(x^*, y^*, \lambda^*)$ of a sequence $\{(x_k, y_k, \lambda_k)\}$ obtained by the described algorithm is a B-stationary point of the problem (3.21) can be found in Fukushima and Pang [13] or Dempe et al. [11].

# 4. Optimal Value Function Reformulation

The main idea of this chapter, described in section 4.1, is, similarly to chapter 3, to reduce the bilevel programming problem to a single level problem. Section 4.2 proposes algorithms (one local and one global) based on the optimal value function reformulation and approximation of the optimal value function by a piecewise linear function.

## 4.1 Theory

This section focuses on presenting the optimal value function reformulation to the reader, followed by introducing partial calmness condition, a constraint qualification specific to this reformulation.

Consider a BLPP (2.25) - (2.26). We use the optimal value function

$$\varphi(x) = \{\min_{y} f(x, y) : y \in Y(x) \cap T\}$$

defined in chapter 2 to replace this problem by the optimal value function reformulated problem

$$\begin{aligned} &\min_{x,y} F(x, y) \\ &s.t.\ G(x) \leq 0, \\ &\quad f(x, y) - \varphi(x) \leq 0, \\ &\quad g(x, y) \leq 0, \\ &\quad x \in X,\ y \in T. \end{aligned} \tag{4.1}$$

Stationary conditions based on optimal value function reformulation can be found in Ye and Zhu [29], various constraint qualifications related to optimal value function reformulation can be found in Ye [28]. A global solution algorithm (which will be included later in this chapter) is described in Dempe et al. [11] and extended to a local version in Dempe and Franke [8].

Note that, in contrast to the KKT reformulation, the problem 4.1 is clearly fully equivalent to the BLPP (2.25) - (2.26), however, it is generally nonsmooth, even if all the functions $F$, $f$, $G$, $g$ are continuously differentiable (Ye and Zhu [29]). Furthermore, Mangasarian-Fromovitz constraint qualification is in general too strong to be satisfied for the problem (4.1).

**Theorem 8.** *(Dempe et al. [11]) Let the optimal value function $\varphi(x)$ be locally Lipschitz continuous, let the functions $F$, $f$, $G$, $g$ be at least continuously differentiable and let $T \subseteq \mathbb{R}^n$ be convex. Then the Mangasarian-Fromovitz constraint qualification is violated at every feasible point of the problem (4.1).*

*Proof.* Can be found in Dempe et al. [11], Theorem 3.30. □

*Remark.* The previous Theorem 8 is formulated for the non-smooth version of MFCQ in Dempe et al. [11] or Ye and Zhu [29].

The stated Theorem 8 requires local Lipschitz continuity of the optimal value function. The following theorem shows that this requirement is however not very strong.

**Theorem 9.** *(Dempe et al. [11]) Consider the lower level problem (2.26) at $x = x^*$, with $T = \mathbb{R}^n$. Let MFCQ be satisfied at every $y \in Y(x^*)$, let the functions $F$, $f$, $G$, $g$ be at least continuously differentiable and let the set $\{(x, y) : g(x, y) \leq 0\}$ be nonempty and compact. Then the optimal value function $\varphi(x)$ is locally Lipschitz continuous.*

*Proof.* Can be found in Dempe et al. [11], Theorem 3.5. □

Further discussion on why is the MFCQ (and other usual constraint qualifications) too strong to hold for the problem 4.1 can be found in chapter 3 of Ye and Zhu [29]. For this reason, we define another constraint qualification.

*Remark.* We shall denote an open ball of radius $\delta > 0$ and center $x \in \mathbb{R}^n$ by $B^\delta(x)$.

**Definition 20.** *(Ye and Zhu [29]) Let $(x^*, y^*)$ be a local optimal solution of the problem (4.1). We say that (4.1) is* partially calm *at $(x^*, y^*)$ if there exist $\varkappa > 0$ and $\delta > 0$ such that for all $u \in B^\delta(0) \subseteq \mathbb{R}$ and for all $(x', y') \in B^\delta(x^*, y^*) \subseteq \mathbb{R}^n \times \mathbb{R}^m$ which are feasible for the problem*

$$\begin{aligned} &\min_{x,\,y} F(x, y) \\ &s.t.\,,\ G(x) \leq 0, \\ &\qquad f(x, y) - \varphi(x) + u \leq 0, \\ &\qquad g(x, y) \leq 0, \\ &\qquad x \in X,\ y \in T, \end{aligned} \tag{4.2}$$

*holds the following inequality*

$$F(x', y') - F(x^*, y^*) + \varkappa |u| \leq 0.$$

*Remark.* (Ye and Zhu [29]) Let $(x^*, y^*)$ be a local optimal solution of the problem (4.1). Then the problem (4.1) is partially calm at $(x^*, y^*)$ if and only if there exists $\gamma > 0$ such that the point $(x^*, y^*)$ is a local optimal solution to the *exact penalization problem*:

$$\begin{aligned} &\min_{x,\,y} F(x, y) + \gamma(f(x, y) - \varphi(x)) \\ &s.t.\ G(x) \leq 0, \\ &\qquad g(x, y) \leq 0, \\ &\qquad x \in X,\ y \in T. \end{aligned} \tag{4.3}$$

A discussion on the partial calmness conditions can be found in Ye and Zhu [29]. The exact penalization problem 4.3 is used to derive stationarity conditions for the problem (4.1) in Ye and Zhu [29], a short overview of these results is in Dempe et al. [11]. Stationary conditions for the BLPP based on optimal value function reformulation (both with and without partial calmness) are derived in Dempe and Zemkoho [10].

## 4.2 Solution Algorithms

In this section, we describe a global solution algorithm for the problem (4.1), which was proposed in Dempe et al. [11] and its modification into a local solution algorithm, proposed in Dempe and Franke [8]. Both algorithms assume convexity of the optimal value function $\varphi(\cdot)$, which is i.e. implied by joint convexity of the functions $f$ and $g$. Further, assume $T = \mathbb{R}^m$ and assume that $X$ is a polytope (i.e. a bounded polyhedron).

Denote $Z = \{x_1, \ldots, x_\tau\}$ set of its vertices. The paper Dempe and Franke [8] assumes $X = \mathbb{R}^n$ and $Z = \{x_1, \ldots, x_\tau\}$ such that the $\mathcal{M}_G = \{x \in X : G(x) \leq 0\} \supseteq \mathbf{conv}(Z)$, with the desired optimal solution being element of the set $\mathbf{conv}(Z)$. Both local and global algorithms however work even for $Z = \{x_1, \ldots, x_\tau\}$ such that the $\mathcal{M}_G = \{x \in X : G(x) \leq 0\} \subseteq \mathbf{conv}(Z)$.

Define a function

$$\Xi_Z(x) = \\ = \min_{\mu} \left\{ \sum_{j=1}^{\tau} \mu_j \varphi(x^j) : \mu_j \geq 0,\, j = 1, \ldots, \tau,\, \sum_{j=1}^{\tau} \mu_j = 1,\, \sum_{j=1}^{\tau} \mu_j x^j = x \right\}. \quad (4.4)$$

The function $\Xi_Z(\cdot)$ is an objective function of a linear optimization problem with right hand side perturbations, thus it is convex (and piece-wise affine). Due to convexity of $\varphi(\cdot)$ we have

$$\varphi(x) \leq \sum_{j=1}^{\tau} \mu_j \varphi(x^j),$$

where $\mu = \{\mu_1, \ldots, \mu_\tau\}$ is from the definition of the function $\Xi_Z(x)$, thus for all $x \in X$ it holds

$$\varphi(x) \leq \Xi_Z(x). \quad (4.5)$$

Thereby, if we denote optimal function value of the problem (4.1) by $\upsilon^*$ and $\upsilon(Z)$ optimal function value of the problem

$$\begin{aligned} &\min_{x,\,y} F(x,\,y) \\ &s.\,t.\; G(x) \leq 0, \\ &\quad f(x,\,y) - \Xi_Z(x) \leq 0, \\ &\quad g(x,\,y) \leq 0, \\ &\quad x \in X,\, y \in T, \end{aligned} \quad (4.6)$$

we have $\upsilon(Z) \leq \upsilon^*$, since the feasible set of the problem (4.1) is a subset of the feasible set of the problem (4.6). Both global and local algorithms are based on solving a sequence of problems (4.6) for a sequence of sets $\{Z_k\}$ with $\{Z_k\} \subset \{Z_{k+1}\}$.

### 4.2.1 Global Algorithm

Consider the Problem (4.1) with the functions $f$, $g$ being jointly convex and $X = \mathbb{R}^n$. Choose a set $Z = \{x_1, \ldots, x_\tau\}$ described in the previous section. Then, $\mathcal{M}_G$ can be partitioned into regions of stability of the function

$\Xi_Z(\cdot)$, i.e. (without loss of generality bounded) sets $X_1, \ldots, X_\nu$ such that $\mathbf{rint}\, X_i \cap \mathbf{rint}\, X_j = \emptyset \,\forall\, i, j = 1, \ldots, \nu,\ i \neq j,\ X \subseteq \bigcup_{i=1}^{\nu} X_i$ and $\Xi_Z(\cdot)$ is continuous affine linear on each of these sets. Then $X_i = \mathbf{conv}\,\{x_{ij_1}, \ldots, x_{ij_\iota}\}$, where $\{x_{ij_1}, \ldots, x_{ij_\iota}\} \subseteq Z$.

The global algorithm follows:

1. Compute the function $\Xi_Z(\cdot)$. Set $t := 1,\ Z_1 = Z$.
2. Solve the problem (4.6) with $Z = Z_t$ and denote the global optimal solution by $(x^t, y^t)$.
3. If $y^t \in \Psi(x^t)$, stop. $(x^t, y^t)$ is the global optimal solution. Else, set $Z_{t+1} := Z_t \cup \{x^t\}$, compute $\varphi(x^t)$ and $\Xi_{Z_{t+1}}(\cdot)$. Set $t := t + 1$. Go to step 2.

*Remark.* Computing the function $\Xi_{Z_{t+1}}(\cdot)$ in the step 3 can be done by updating the function $\Xi_{Z_t}(\cdot)$. Only the region of stability containing $x^t$ needs to be recalculated. In a case when $x^t$ lies on a boundary between two (or more) regions of stability $\{X_{t_k}\}_{k\in K}$, i.e. there exists some $\varepsilon > 0$ such that $B^\varepsilon(x^t) \cap X_{t_k} \neq \emptyset \forall\, k \in K$, all regions $\{X_{t_k}\}_{k\in K}$ need to be recalculated.

Correctness of the algorithm is shown by the following theorem:

**Theorem 10.** *(Dempe et al. [11]) Suppose that the mentioned algorithm leads to an infinite sequence $\{(x^t, y^t)\}_{t=1}^{\infty}$. Let the following assumptions be satisfied:*

**Assumption A1.** *The set $\{(x, y) : G(x) \leq 0, g(x, y) \leq 0, x \in X\}$ is nonempty and compact,*

**Assumption A2.** *For every $\widetilde{x} \in X$ there exists a point $(\widetilde{x}, \widetilde{y})$ such that $g(\widetilde{x}, \widetilde{y}) < 0$.*

*Then the sequence $\{(x^t, y^t)\}_{t=1}^{\infty}$ has an accumulation point. Furthermore, each accumulation point of this sequence is a global optimal solution of the problem (4.1).*

*Proof.* Can be found in Dempe et al. [11], Theorem 3.32. □

*Remark.* The convexity of the functions $f$, $g$ that is required in the description of the algorithm, is needed for the Theorem 10 as well.

### 4.2.2 Local Algorithm

Consider problem (4.1) with the functions $f$, $g$ being jointly convex and the set $X = \mathbb{R}^n$. Choose a set $Z = \{x_1, \ldots, x_\tau\}$ described above.

The algorithm is very similar to the global algorithm:

1. Select the set $Z = \{x_1, \ldots, x_\tau\} \subset \mathcal{M}_G$ and compute the function $\Xi_Z(\cdot)$. Set $t := 1,\ Z_1 = Z$.
2. Solve the problem (4.6) with $Z = Z_t$ and denote a local optimal solution by $(x^t, y^t)$.

3. If $y^t \in \Psi(x^t)$, stop. $(x^t, y^t)$ is the global optimal solution. Else, set $Z_{t+1} := Z_t \cup \{x^t\}$, compute $\varphi(x^t)$ and $\Xi_{Z_{t+1}}(\cdot)$. Set $t := t + 1$. Go to step 2.

*Remark.* Note that the only difference between the local algorithm and the global algorithm is in step 2: In global algorithm, we solve the problem (4.6) globally, whereas in local algorithm, we solve the problem (4.6) locally.

*Remark.* Similarly to the global algorithm, only the regions of stability $\{X_{t_k}\}_{k \in K}$ containing $x^t$ (we assume boundedness of the sets $X_i$) in the third step of the local algorithm need to be updated when calculating the function $\Xi_{Z_{t+1}}(\cdot)$.

*Remark.* In the second step of the first iteration of the local algorithm, we can solve the problems

$$\begin{aligned} &\min_{x,\,y} F(x,\,y) \\ &s.\,t.\; G(x) \,\leq\, 0, \\ &\qquad f(x,\,y) \,-\, \Xi_Z(x) \,\leq\, 0, \\ &\qquad g(x,\,y) \,\leq\, 0, \\ &\qquad x \,\in\, X_k,\; y \,\in\, T, \end{aligned}$$

for all regions of stability $X_k$, $k = 1, \ldots, \nu$, of the function $\Xi_{Z_1}$ and take the best solution. The same procedure can be used in the global algorithm.

We shall now include a series of theorems from Dempe and Franke [8], stating assumptions under which the local algorithm yield a correct solution to the BLPP.

**Theorem 11.** *(Dempe and Franke [8]) Suppose that the mentioned local algorithm leads to an infinite sequence $\{(x^t,\, y^t)\}_{t=1}^{\infty}$. Let the assumptions A1 and A2 be satisfied, together with*

> **Assumption A3.** *There exists an open set $W$, $W \in \mathcal{M}_G$ such that $|\varphi(x)| < \infty \,\forall\, x \in W$.*

*Then the sequence $\{(x^t,\, y^t)\}_{t=1}^{\infty}$ has an accumulation point. Furthermore, each accumulation point of this sequence is feasible to the problem (4.1).*

*Proof.* Can be found in Dempe and Franke [8]), theorem 4.3. □

*Remark.* Similarly to Theorem 10, the convexity of the functions $f$, $g$ that is required in the description of the local algorithm, is needed for Theorem 11 as well.

Note that Theorem 11 is much weaker than Theorem 10, since it does not guarantee optimality of accumulation points of the sequence computed with the local algorithm. We need to add one more assumption to achieve local optimality:

**Theorem 12.** *(Dempe and Franke [8]) Suppose that the mentioned local algorithm leads to an infinite sequence $\{(x^t,\, y^t)\}_{t=1}^{\infty}$. Let the assumptions A1, A2 and A3 be satisfied. Denote by $\mathcal{M}_t = \{(x,\, y) :\; g(x,\, y) \leq 0,\; G(x) \leq 0,\; f(x,\, y) \leq \Xi_{Z_t}(x)\}$ the feasible set of t-th iteration of the problem (4.6). Assume that the following assumption is satisfied:*

> **Assumption A4.** *There exists an $\varepsilon > 0$ and an iteration index $t_0 \in \mathbb{N}$ such that:*
>
> $$\forall\, t \,\geq\, t_0,\; \forall\, (x,\, y) \in \mathcal{M}_t \text{ with } \|(x,\, y) \,-\, (x^t,\, y^t)\| \,\leq\, \varepsilon \quad \text{we have } F(x,\, y) \,\geq\, F(x^t,\, y^t), \quad (4.7)$$

*Then, each accumulation point of the local algorithm is a local optimal solution of the problem (4.1).*

*Proof.* Can be found in Dempe and Franke [8]), Theorem 4.4. □

The final theorem in this chapter proposes an assumption alternative to A4.

**Theorem 13.** *(Dempe and Franke [8]) Suppose that the mentioned local algorithm leads to an infinite sequence* $\{(x^t, y^t)\}_{t=1}^{\infty}$. *Suppose that the function* $G$ *is convex. Suppose the following assumption is satisfied:*

> **Assumption A5.** *There exists a* $\delta > 0$ *and an iteration index* $t_0 \in \mathbb{N}$ *such that:*
>
> $$\forall t \geq t_0, \forall d \in T_{\mathcal{M}_t}(x^t, y^t), \|d\| = 1, \text{ we have } \nabla F(x^t, y^t)^T d \geq \delta. \quad (4.8)$$

*Then, the assumption A4 is satisfied.*

*Proof.* Can be found in Dempe and Franke [8]), Theorem 4.5. □

# 5. Linear Bilevel Programming Problem

Linear programming problems are an important class of optimization problems. This section is dedicated to linear bilevel programming problems.

In the first section 5.1 of this chapter we shall define the linear BLPP and derive out its basic properties with the focus on the properties needed for solution algorithms. The algorithms are given in section 5.2. An interesting solution algorithm for a very special case of the linear BLPP is included at the end of the chapter. Reader interested in linear BLPP is also referred to Dempe et al. [11], chapter 2.

## 5.1 Linear BLPP Theory

This section formulates the linear bilevel programming problem and states its basic properties.

**Definition 21.** *Let $A$ be an $(p \times n)$-matrix and let $C$ be an $(n \times m)$-matrix. Let $a, x \in \mathbb{R}^n$; $b, d, y \in \mathbb{R}^m$ and $c \in \mathbb{R}^p$. We then define the* Linear Bilevel Programming Problem (Linear BLPP) *as*

$$\min_{x,\,y}\{a^T x + b^T y : Ax \leq c, (x, y) \in \mathbf{gph}\,\Psi\} \tag{5.1}$$

*where*

$$\Psi(x) = \arg\min_{y}\{d^T y : Cy \leq x\}. \tag{5.2}$$

Note that this is a special case of the optimistic bilevel programming problem (2.25) - (2.26). A reader can notice the (upper level) constraints in 5.1 only depend on the upper level variable $x$. A $(p \times n)$-matrix $B$ could be added to the constraint $Ax + By \leq c$, however this could result in connecting constraints (described in section 2.3). Thus all the constraints involving the lower level variable are in the lower level problem in Definition 21 ($Cy \leq x$).

It can be proven (see e.g. Grygarová [16]) that $\mathbf{gph}\,\Psi$ is a connected union of faces of the set $\mathcal{M} = \{(x, y) \in \mathbb{R}^{n+m} : Cy \leq x\}$. The set $\mathcal{M}$ is connected and its convex haul is a convex polytope.

Since a linear programming problem has a solution in a one of the vertices of its feasible set, the problem (5.1) - (5.2), if it has an optimal solution, has at least one global optimal solution in a vertex of the set $\{(x, y) \in \mathbb{R}^{n+m} : Ax \leq c, Cy \leq x\}$ (the proof can be found in Candler and Townsley [2]).

The following example from Dempe et al. [11] shows an interesting property of the bilevel programming: The global optimal solution can change, if an inequality constraint, which is inactive at the global optimal solution, is added to the lower level problem.

*Example.* Consider the bilevel programming problem

$$\min_{x,\,y}\{(x-1)^2 \,+\, (y-1)^2 :\; y \in \Psi(x)\} \tag{5.3}$$

with the lower level problem

$$\Psi(x) = \arg\min_{y}\{0.5y^2 \,+\, 500y \,-\, 50xy\}. \tag{5.4}$$

The problem has a unique global optimal solution $= (50102/5002,\, 4100/5002)$ with the optimal objective function value of $F(x^*,\, y^*) = 81.33$. If we add the constraint $y \geq x$ to the lower level problem, we get the BLPP

$$\min_{x,\,y}\{(x-1)^2 \,+\, (y-1)^2 :\; y \in \Psi(x)\} \tag{5.5}$$

with the lower level problem

$$\Psi(x) = \arg\min_{y}\{0.5y^2 \,+\, 500y \,-\, 50xy :\; y\, geq\, 0\}. \tag{5.6}$$

The problem (5.5) - (5.6) has a unique global optimal solution $(\overline{x},\, \overline{y}) = (1,\, 0)$ with the optimal objective function value of $F(\overline{x},\, \overline{y}) = 1 < 81.33 = F(x^*,\, y^*)$, however the point $(\overline{x},\, \overline{y})$ is not feasible for the problem (5.3) - (5.4). The point $(x^*,\, y^*)$ is a local optimal solution of the problem (5.5) - (5.6).

Stationarity or optimality conditions for a Linear BLPP can be derived using the KKT reformulation (see references in chapter 3), which takes the following form:

$$\begin{aligned}
&\min_{x,\,y,\,\lambda} (a^T x \,+\, b^T y) \\
&s.\,t.\; Ax \,\leq\, c, \\
&\qquad d^T \,+\, \lambda^T C \,=\, 0, \\
&\qquad Cy \,\leq\, x, \\
&\qquad \lambda^T(Cy \,-\, x) = 0, \\
&\qquad \lambda_i \geq 0 \;\forall\, i \in \{1, \ldots, p\},
\end{aligned} \tag{5.7}$$

As mentioned in chapter 3, this is an MPEC, and thus NP-hard to solve. However, in a special case of Linear BLPP

$$\min_{x,\,y,\,z}\{F(y) :\; x \in \mathcal{B},\, z \in \mathcal{C},\, (y) \in \Psi(x,\, z)\},$$

where

$$\mathcal{B} \,=\, \{x :\; Bx = b\},\; \mathcal{C} \,=\, \{z :\; Cz = \widetilde{c}\}$$

and

$$\Psi(x,\, z) = \arg\min_{y}\{c^T y :\; Ay \,=\, x,\; y \,\geq\, 0\},$$

where $x \in \mathbb{R}^n$, $y \in \mathbb{R}^m$, $z \in \mathbb{R}^m$, $b \in \mathbb{R}^p$, $c \in \mathbb{R}^q$, $A$ is an $(n \times m)$-matrix, $B$ is a $(p \times n)$-matrix and $C$ is a $(q \times m)$-matrix. Note that the upper level objective function depends only on the lower level variable. For this problem, optimality conditions, which are verifiable in a polynomial time, can be formulated, see Dempe and Lohse [9].

## 5.2 Algorithms

Two algorithms solving the LBLPP are described in this section. A basic algorithm, proposed in (Dempe [4]), is described in the first part of this section. The second part contains a global (and local) algorithm for a special case of LBLPP.

### 5.2.1 Local Algorithm

Consider the LBLPP (5.1) - (5.2). A point $y^*$ is an optimal solution of the lower level problem (5.2) if and only if there exists $u \in \mathbb{R}^n$ such that

$$u \leq 0,\ u^T(Cy - x) = 0,\ C^T u = d.$$

Note that $u = -\lambda$ from (5.7). Let $rank(C) = n$. The point $y^*$ lies in a vertex of the lower level problem feasible set $Y(x) = \{y :\ Cy \leq x\}$, i.e. there exists a set of indexes $J = \{j_1, \ldots, j_n\}$ such that

$$C_j y^* = x_j,\ j \in J,$$

and

$$C_j y^* \leq x_j,\ j \notin J,$$

where $C_j$ denotes the $j$-th row of the matrix $C$. Let, without loss of generality, be $J = \{1, \ldots, n\}$. Let $D = (C_1^T, \ldots, C_n^T)^T$ be the basic matrix related to $y^*$ and $N = (C_{n+1}^T, \ldots, C_m^T)^T$, i.e. $C = (D^T, N^T)^T$. Let $x = (x_D^T, x_N^T)^T$ be a corresponding decomposition of the vector $x$. Then we have $y^* = D^{T\,-1} x_D$. For a solution of the dual problem $u^*$ we have $u^* = (u_D^{*T}, u_N^{*T}) = ((D^{T\,-1}d)^T, 0^T)^T$ and it holds

$$u_D \leq 0,\ D_y = x_D,\ Ny \leq X_N,\ u_N = 0,\ Ny \leq x.$$

A set of all $x$ for which the basic matrix $D$ (with $D^{T\,-1}d \leq 0$) remains the same in the primal problem is

$$\mathcal{R}_D = \{x :\ Dy = x_D,\ Ny \leq x_N \text{ for some } y \in \mathbb{R}^m\}.$$

For $x \notin \mathcal{R}_D$, the basic matrix consists of other rows of $C$. We can now proceed to the exact description of the algorithm:

1. Choose an optimal basic solution $(x^1, y)$ of the problem

$$\min_{x,\,y}\{a^T x + b^T y :\ Ax \leq c,\ Cy \leq x\}.$$

2. Compute an optimal basic solution $y^1$ of the problem (5.2) with $x = x^1$, i.e. choose an $y^1 \in \Psi(x^1) = \arg\min_y\{d^T y :\ Cy \leq x^1\}$. Set $k := 1$.

3. Select a basic matrix $D$ for $y^k$ compute $\mathcal{R}_D$ and solve the problem

$$\min_x\{a^T x + b^T y :\ x = (x_D^T, x_N^T)^T,\ y = D^{T\,-1} x_D,\ x \in \mathcal{R}_D\}.$$

   Denote the optimal solution $(x^*, D^{T\,-1}x_D^*)$.

4. Set $x^{k+1} = x^*$. Compute an optimal basic solution $y^{k+1}$ of the problem

$$\min_y \{b^T y : \ y \in \Psi(x^{k+1})\}.$$

5. If $(x^{k+1}, y^{k+1}) = (x^k, y^k)$, stop, $(x^k, y^k)$ is the solution. Else, go to step 3.

The proof that the algorithm computes a local solution after a finite number of steps can be found in (Dempe [4]).

### 5.2.2 Alternative Global Algorithm

In this subsection we shall describe a global algorithm, proposed in (Dempe and Franke [7]), which is (under mild assumptions) easily modifiable into a local algorithm. The algorithm solves the problem

$$\min_{x,\,y} \{a^T x + b^T y : \ Ax \leq c, \, (x, \, y) \in \mathbf{gph}\, \Psi\}, \tag{5.8}$$

where

$$\Psi(x) = \arg\min_y \{x^T y : \ By \leq d\}. \tag{5.9}$$

Note that this is a special case of the optimistic bilevel programming problem (2.25) - (2.26), and whilst being linear in the sense that both objective functions as well as all the constraints are linear (the variable $x$ is passed as a parameter to the lower level problem), it differs from the problem (5.1) - (5.2). The problem (5.1) - (5.2) has the upper level decision variable in the constraint of the lower level problem $Cy \leq x$, whereas the problem (5.8) - (5.9) has the upper level decision variable in the objective function of the lower level problem $x^T y$.

Consider an optimal value function reformulation of the problem (5.8) - (5.9):

$$\min_{x,\,y} \{a^T x + b^T y : \ Ax \leq c, \, By \leq d, \, x^T y \leq \varphi(x)\}, \tag{5.10}$$

where

$$\varphi(x) = \min_y \{x^T y : \ By \leq d\}. \tag{5.11}$$

Note that the lower level constraints are added to the upper level problem to avoid connecting constraints. We remind that the optimal value function reformulation is fully equivalent to the original bilevel programming problem. The main idea of the algorithm is somewhat similar to the idea of the algorithm described in chapter 4. The optimal value function of the problem (5.10) is replaced by an approximation, resulting in

$$\min_{x,\,y} \{a^T x + b^T y : \ Ax \leq c, \, By \leq d, \, x^T y \leq \min_{z \in Z} x^T z\}, \tag{5.12}$$

with $Z \subseteq \{y : \ By \leq d\}$. Note that this is not a linear programming problem, since the constraint $x^T y \leq \min_{z \in Z} x^T z$ is not linear. Let us denote the global optimal solution of the problem (5.10) by $(\overline{x}_B, \, \overline{y}_B)$ and the global optimal solution of the

problem (5.12) by $(\overline{x}_R, \overline{y}_R)$. Then, the following series of implications is obviously satisfied:

$$Z \subseteq \{y : By \leq d\} \Rightarrow \min_{z \in Z} x^T z \geq \varphi(x) \Rightarrow F((\overline{x}_R, \overline{y}_R)) \leq F((\overline{x}_B, \overline{y}_B)).$$

Moreover, every global optimal solution of the problem (5.12) which is feasible for the problem (5.10), is also a global optimal solution for this problem, see (Dempe and Franke [7]).

The algorithm is as follows:

1. Choose an initialization point $z_1$ (see the remark below). Set $Z_1 := \{z^1\}$, set $k := 1$.
2. Solve the problem (5.12) globally with $Z = Z_k$. Denote the optimal solution $(x^k, y^k)$.
3. If $(x^k, y^k)$ is feasible for the problem (5.10), stop. $(x^k, y^k)$ is the global optimal solution. Else, compute $z^{k+1} = \varphi(x^k)$ from (5.11).
4. Set $Z_{k+1} := Z_k \cup \{z^{k+1}\}$. Set $k := k + 1$. Go to 2.

*Remark.* The initialization point $z_1$ can be obtained as $z_1 = \overline{y}$, where $(\overline{x}, \overline{y})$ is the global optimal solution of the problem

$$\min_{x,\, y} \{a^T x + b^T y : Ax \leq c, By \leq d\}. \tag{5.13}$$

For the following theorems, we set

$$\mathcal{M} = \{(x, y) : Ax \leq c, By \leq d\},$$

$$\mathcal{M}_{R_k} = \{(x, y) : Ax \leq c, By \leq d, x^T y \leq \min_{z \in Z_k} x^T z\}$$

and the upper level objective function $F(x, y) = a^T x + b^T y$

**Theorem 14.** *(Dempe and Franke [7]). Suppose that the mentioned algorithm yields a sequence $(x^k, y^k)$. Assume the set $\mathcal{M}$ is bounded. Then, the sequence $(x^k, y^k)$ has an accumulation point and every accumulation point of this sequence is a global optimal solution of the problem (5.10).*

*Proof.* Can be found in (Dempe and Franke [7], Theorem 3.4). □

*Remark.* (Dempe and Franke [7]) If the polytope $\{y : By \leq d\}$ is bounded, only its vertices need to be considered for the computation of optimal solutions in the mentioned algorithm. Thus, the algorithm stops after a finite number of steps.

The mentioned algorithm calculates the global optimal solution of the problem (5.10), however, if the problem (5.12) is solved locally instead of globally in step 2, the algorithm is turned into a local algorithm.

**Theorem 15.** *(Dempe and Franke [7]). Suppose that the local version of the global algorithm described in this subsection yields a sequence* $(x^k, y^k)$*. Assume the set* $\mathcal{M}$ *is bounded and that*

$$\nabla F(x^k, y^k)^T s > 0 \text{ for all } s \in T_{\mathcal{M}_{R_k}}(x^k, y^k) \; \forall \, k \in \mathbb{N}.$$

*Then, the sequence* $(x^k, y^k)$ *has an accumulation point and every accumulation point of this sequence is a global optimal solution of the problem (5.10).*

*Proof.* Can be found in (Dempe and Franke [7], Theorem 3.7). □

# 6. Comparison of the Described Algorithms

In this chapter, we shall focus on a comparison of the algorithms described in chapters 3, 4 and 5.

A total of six algorithms have been described in this work. The three algorithms based on the KKT reformulations are in chapter 3, namely in subsections 3.3.1 (reduction to a mixed integer single level problem), 3.3.2 (regularization of the KKT reformulation) and 3.3.3 (smoothing continuation method). An algorithm (both global and local versions) based on optimal value function reformulation is described in section 4.2. Finally, two algorithms for linear bilevel programming (each for a different type of problem) are described in chapter 5, namely in section 5.2.

All of the included algorithms consider a bilevel programming problem in the form (2.25) - (2.26), i.e. a problem close to an optimistic problem. For all of the algorithms based on the KKT reformulation, we need to assume that $X = \mathbb{R}^n$ and $T = \mathbb{R}^m$, all the functions $y \to g_i(x, y)$, $i = 1, \ldots, p$, and $y \to f_(x, y)$ are convex and differentiable and that e.q the MFCQ (or Slater's condition) holds for the lower level problem.

The algorithm based on optimal value function reformulation does not require MFCQ to be satisfied for the lower level problem, however, the functions $f$ and $g_i$, $i = 1, \ldots, p$, have to be jointly convex. Furthermore, should the algorithm compute the global optimal solution of the problem (2.25) - (2.26), we need the assumptions A1 and A2 from the chapter 4 to be satisfied. The local version of this algorithm needs, in addition to the mentioned, the assumptions A3 and A4 to be satisfied (the latter can be replaced by the assumption A5 together with convexity of the function $G$).

The algorithms for LBLPP both require specific types of linear bilevel problem, both being a special case of the problem (2.25) - (2.26). The local algorithm based on duality (subsection 5.2.1) considers the problem (5.1) - (5.2) (the upper level variable is in the constraint of the lower level problem), whereas the algorithm based on optimal value function reformulation (subsection 5.2.2) considers the problem (5.8) - (5.9) (the upper level variable is in the objective function of the lower level problem).

# 7. Portfolio Optimization Problems

This section is dedicated to one of the possible uses of bilevel programming in portfolio management: Finding a mean-risk efficient portfolio (mean-risk efficiency being the lower level problem) that is in some way closest to some other portfolio (minimum distance being the upper level problem). Solving this problem may be desirable e.g. to minimize the transaction costs when finding a new portfolio that is efficient.

Basic theory about risk measures is defined in the section 7.1. Section 7.2 states the use of bilevel optimization in the problem of finding a mean-risk efficient portfolio.

## 7.1 Theory

This section reminds the reader definitions regarding portfolio optimization.

By *loss* of an asset (additive inverse of return) we mean any random variable $Z$. We shall denote the set of all random variables by $\mathcal{L}$ and the set of all random variables $Z$ such that $\mathsf{E}\,|Z|^p$ by $\mathcal{L}_p$ for some $p \in \mathbb{N}$.

**Definition 22.** *(Rocciolietti [21])* Risk measure *is a functional* $\rho : \mathcal{L}_p \to \mathbb{R}$ *for some* $p \in \mathbb{N}$.

From now on, assume a set of $n$ assets. Denote $Z = (Z_1, \ldots, Z_n)^T \in \mathbb{L}_p^n$ vector of losses of these assets. Consider a portfolio $X$ of these assets with weights $x = (x_1, \ldots, x_n)^T$ such that $\sum_{i=1}^{n} x_i = 1$ and $x_i \geq 0$, $i = 1, \ldots, n$, and a portfolio $Y$ with weights $y = (y_1, \ldots, y_n)^T$ such that $\sum_{i=1}^{n} y_i = 1$ and $y_i \geq 0, i = 1, \ldots, n$.

Denote $Z_X = x^T Z$ loss of the portfolio $X$ and $Z_Y = y^T Z$ the loss of the portfolio $Y$. The following definitions are inspired by Dupačová et al. [12], where the efficiency of a portfolio using variance as a risk measure is defined, however any risk measure can be used.

**Definition 23.** *Let* $\rho$ *be a risk measure on* $\mathbb{L}_p$ *and consider two portfolios* $X$ *and* $Y$. *We say that the portfolio* $X$ dominates the portfolio $Y$ with respect to $\rho$ and mean loss*, if it holds either*

$$\mathsf{E}(Z_Y) < \mathsf{E}(Z_X) \;\&\; \rho(Z_Y) \leq \rho(Z_X)$$

*or*

$$\mathsf{E}(Z_Y) \leq \mathsf{E}(Z_X) \;\&\; \rho(Z_Y) < \rho(Z_X).$$

**Definition 24.** *Let* $\rho$ *be a risk measure on* $\mathbb{L}_p$. *We say that a portfolio* $X$ *is* efficient with respect to $\rho$ and mean loss*, if there exists no other portfolio* $Y$ *such that* $Y$ *dominates* $X$ *with respect to* $\rho$ *and mean loss.*

Finally we need to define a commonly used risk measure:

**Definition 25.** *(Rockafellar and Uryasev [23]) Let* $\alpha \in (0, 1)$ *The* conditional value at risk at level $\alpha$ *is a functional* $CVaR_\alpha : \mathcal{L}_1 \to \mathbb{R}$ *defined by*

$$CVaR_\alpha(Z) = \min_a \left\{ a + \frac{1}{1-\alpha} E[Z - a]^+ \right\}.$$

*Remark.* The level $\alpha$ is usually chosen close to 1, e.g. 0.95. If we define *value at risk at level* $\alpha$ as the $\alpha$-quantile of the probability distribution of loss, i.e.

$$VaR_\alpha(Z) = F_X^{-1}(\alpha) = \inf_x \{x : P(Z \leq x) \geq \alpha\},$$

it holds

$$VaR_\alpha(Z) \in \arg\min_a \left\{ a + \frac{1}{1-\alpha} E[Z - a]^+ \right\}. \tag{7.1}$$

A proof of (7.1) can be found in Rockafellar and Uryasev [22].

## 7.2 Bilevel Programming Problem Based on a Mean-Risk Model

First, part of this section states one of the possible formulations of the problem of minimizing CVaR while minimizing loss (maximizing return). The rest of the subsection describes bilevel programming problem based on the aforementioned portfolio optimization problem - finding a portfolio, that is efficient (with respect to CVaR and mean loss) and in some way closest to another portfolio.

As in the previous section, consider a set of $n$ assets with vector of losses $Z = (Z_1, \ldots, Z_n)^T \in \mathbb{L}_p^n$. Suppose we can observe a set of $S$ scenarios (realizations of $Z$) $\{\xi^1, \ldots, \xi^S\}$ with $\xi^s = (\xi_1^s, \ldots, \xi_n^s)$, $s = 1, \ldots, S$, each scenario with probability $p^s$, $s = 1, \ldots, S$.

Let the desired maximal loss $l_{max} \in \mathbb{R}$ and a level $\alpha \in (0, 1)$ be fixed parameters. The problem of finding a portfolio $X$ with weights $x = (x_1, \ldots, x_n)^T$ minimizing $CVaR_\alpha$ subject to loss not exceeding $l_{max}$ is

$$\begin{aligned} \min_{a,\, z,\, x} & \left( a + \frac{1}{1-\alpha} \sum_{s=1}^{S} p^s z^s \right) \\ s.t.\ z^s &\geq \sum_{i=1}^{n} x_i \xi_i^s - a,\ s = 1, \ldots, S, \\ l_{max} &\geq \sum_{i=1}^{n} \sum_{s=1}^{S} x_i \xi_i^s p^s, \\ 1 &= \sum_{i=1}^{n} x_i, \\ x &\geq 0,\ z \geq 0,\ a \in \mathbb{R}. \end{aligned} \tag{7.2}$$

The problem 7.2 is a linear programming problem.

We shall now describe the motivation for a bilevel programming problem. Suppose that $\alpha$ is fixed. Denote $\Psi(l_{max})$ the set of all (global) optimal solutions of the problem (7.2) for some value of the parameter $l_{max}$. Suppose we have a portfolio $X^0$ with vector of weights $x^0 = (x_1^0, \ldots, x_n^0)$ such that $\sum_{i=1}^{n} x_i^0 = 1$ and $x_i^0 \geq 0,\ i = 1, \ldots, n$. We shall call $X^0$ the *initial portfolio*. Consider a problem of finding a portfolio $X$ which is efficient and (in some way) closest to $X^0$. This leads to an optimistic bilevel programming problem:

$$\min_{l_{max},\, a,\, z,\, x} \{\|x - x^0\| :\ (a,\, z,\, x) \in \Psi(l_{max})\}. \tag{7.3}$$

In the computation part, two different norms $\|\cdot\|$ are considered. The *absolute value norm*

$$\|x\|_1 = \sum_{i=1}^{n} |x_i| \tag{7.4}$$

and the *Euclidean norm*

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}. \tag{7.5}$$

Let $b = (b_1, \ldots, b_n) \in \mathbb{R}^n$. The problem (7.3) with the norm (7.4), which has been considered in Goduľová [15], then takes the form

$$\begin{aligned} \min_{l_{max},\, b,\, a,\, z,\, x} & \sum_{i=1}^{n} b_i \\ s.t.\ b_i &\geq x_i - x_i^0,\ i = 1, \ldots, n, \\ b_i &\geq x_i^0 - x_i,\ i = 1, \ldots, n, \\ (a,\, z,\, x) &\in \Psi(l_{max}),\ l_{max} \in \mathbb{R}, \end{aligned} \tag{7.6}$$

and the problem (7.3) with the norm (7.5) is equivalent to

$$\begin{aligned} \min_{l_{max},\, b,\, a,\, z,\, x} & \sum_{i=1}^{n} b_i^2 \\ s.t.\ b_i &\geq x_i - x_i^0,\ i = 1, \ldots, n, \\ b_i &\geq x_i^0 - x_i,\ i = 1, \ldots, n, \\ (a,\, z,\, x) &\in \Psi(l_{max}),\ l_{max} \in \mathbb{R}. \end{aligned} \tag{7.7}$$

Note that the (upper level) objective function of the problem 7.7 is the square of the Euclidean norm rather than the Euclidean norm itself.

# 8. Results on Real-Life Data

This chapter compares empirical results obtained by using four of the algorithms described in this work. The first subsection describes the data used. We obtain the initial portfolio $X^0$ described in chapter 7 in the second subsection. Last two subsections 8.2.1 and 8.2.2 present the empirical results from solving the problem (7.3) - (7.2).

## 8.1 Data and Initial portfolio

This section describes the data we use both in calculations of the initial portfolio $X^0$ (subsection 8.1.1) and for empirical comparison of some of the algorithms described in this work on solving the bilevel programming problem (7.3) - (7.2) (subsection 8.1.2).

### 8.1.1 Data

We observe prices of stocks of 10 companies: Apple, Advanced Micro Devices, Bank of America Corporation, Ford Motor Company, Foot Locker, General Electrics, Chesapeake Energy Corporation, Microsoft Corporation, QIAGEN N. V. and AT&T. We observe closing price $P_i$ on the last day of each month for the ten stocks from October 1998 to October 2018 (241 observations). The loss of i-th stock on q-th month is $\xi_i^q = (P_i^{q-1} - P_i^q),\ i = 1, \ldots, 10,\ q = 1, \ldots, 240$.

The observed losses have been split into two halves of 120 observations by 10 stocks. This means that for each half we have $S = 120$ scenarios, each with probability $p^s = 1/120,\ s = 1, \ldots, 120$, and $n = 10$ assets.

The first half has been used to obtain the initial portfolio $X^0$. The second half of the observed losses has then been used to find a portfolio that is efficient with respect to $CVaR_\alpha$ (we use $\alpha = 0.95$) and mean loss and that is closest to $X^0$ using both of the stated norms $\|\cdot\|_1$ and $\|\cdot\|_2$ by solving the problems (7.6) - (7.2) and (7.7) - (7.2), respectively.

The initial solution of the solved problem is described in the subsection 8.1.2. The solution to the problem (7.6) - (7.2) using all three of the KKT-reformulation algorithms stated in subsections 3.3.1, 3.3.2 and 3.3.3 is described in the subsection 8.2.1 and the solution to the problem (7.7) - (7.2) using the KKT reformulation algorithms is described in subsection 8.2.2. The solution to the problem (7.6) - (7.2) using the algorithm described in section 4.2 is in section 8.3.

### 8.1.2 Initial Solution

In this subsection the initial portfolio $X^0$ is found, using the first half of the available data, i.e. the observed losses from November 1998 to October 2008. Ten levels of $l_{max}$ have been considered in the calculation of the initial solution, i.e. $l_{max} \in \{l_{max}^1, \ldots, l_{max}^{10}\}$, where

$$l_{max}^1 = \min_i \left\{ \sum_{s=1}^{S} \xi_i^s p^s :\ i \in \{1, \ldots, 10\} \right\}$$

is the lowest obtainable mean loss (mean loss of the stock with the highest mean return) and

$$l_{max}^{10} = \sum_{i=1}^{n} \sum_{s=1}^{S} x_i^* \xi_i^s p^s,$$

where $x_i^*$ is the (global) optimal solution of the problem

$$\begin{aligned} \min_{a,\,z,\,x} & \left( a + \frac{1}{1-\alpha} \sum_{s=1}^{S} p^s z^s \right) \\ s.t.\ z^s & \geq \sum_{i=1}^{n} x_i \xi_i^s \ - \ a, \ s = 1, \ldots, S, \\ 1 & = \sum_{i=1}^{n} x_i, \\ x & \geq 0, \ z \geq 0, \ a \in \mathbb{R}. \end{aligned}$$

In other words, $l_{max}^{10}$ is the mean loss of the portfolio with the lowest possible $CVaR_\alpha$. $l_{max}^2, \ldots, l_{max}^9$ are equidistant values between $l_{max}^1$ and $l_{max}^{10}$.

Ten portfolios have been obtained by solving the problem 7.2 with $l_{max} = l_{max}^i$, $i = 1, \ldots, 10$. The problem has been solved using R with package *lpSolve*. The considered values of $l_{max}$ with achieved values of $CVaR$ are in Table 8.1. The computed portfolios can be found in attachments in Table 8.10 and in Table 8.11.

Table 8.1: $l_{max}$ and achieved $CVaR$ of the initial portfolios.

| $l_{max}$ | $CVaR$ |
|---|---|
| -0.1241 | 9.259 |
| -0.1020 | 4.185 |
| -0.0799 | 3.283 |
| -0.0578 | 3.013 |
| -0.0357 | 2.827 |
| -0.0136 | 2.651 |
| 0.0085 | 2.529 |
| 0.0306 | 2.418 |
| 0.0527 | 2.308 |
| 0.0747 | 2.226 |

## 8.2 Algorithms based on KKT Reformulation

This section describes results from solving problems (7.6) - (7.2) and (7.7) - (7.2) using all three algorithms described in chapter 3:

- The algorithm reducing the KKT reformulated problem 3.21 to a single level mix integer problem (henceforth will be referred to as KKT MIP algorithm, subsection 3.3.1),

- the regularization algorithm (subsection 3.3.2),
- the algorithm using NCP functions to regularize the KKT reformulated problem (henceforth will be referred to as KKT NCP function algorithm 3.3.3).

All of the algorithms were implemented using GAMS software. The first subsection focuses on solving the linear bilevel programming problem (7.6) - (7.2), while the second subsection focuses on solving the (nonlinear) bilevel programming problem (7.7) - (7.2).

### 8.2.1 Absolute-value norm

With having the initial portfolio $X^0$ from the subsection 8.1.2, we use the observed losses from November 2008 to October 2018 to find a portfolio $X$ that is efficient with respect to $CVaR_\alpha$ and mean loss such that the distance $\|x - x^0\|_1$ is minimal. This means we solve the problem (7.6) - (7.2) for each of the initial portfolios obtained in section 8.1.2.

The efficiency of each of the obtained portfolios was checked using R with package *lpSolve*. Note that if a portfolio is not efficient, the solution of the bilevel problem (7.6) - (7.2) related to this portfolio is not feasible for (7.6) - (7.2).

The algorithm using reduction to a single lev1el mixed integer problem (subsection 3.3.1) uses a control variable $Q$ which has been set to 1000. The tolerance for feasible solution was set to $10^{-8}$ and the time limit has been set to 1800 seconds. An overview of the results is in Table 8.2. All of the obtained portfolios are efficient. The computed portfolios can be found in attachments in Table 8.12 and in Table 8.13.

Table 8.2: Results obtained by the KKT MIP algorithm. Achieved Loss, $CVaR$, objective function value (equal to $\|x - x^0\|_1$) of obtained solutions of the problem (7.6) - (7.2) with initial solutions $X^0$ from subsection 8.1.2 and total computing time. All of the solutions are feasible to the BLPP. Note that the computing time varies considerably, however never exceeds 190 seconds.

| $l_{max}$ Initial Solution | Loss | $CVaR_\alpha$ | Objective function value | Time (s) |
|---|---|---|---|---|
| -0.1241 | -0.063 | 1.314 | 1.896 | 76.563 |
| -0.102 | -1.365 | 16.953 | 0.198 | 2.18 |
| -0.0799 | -1.126 | 11.814 | 0.763 | 7.248 |
| -0.0578 | -1.029 | 9.998 | 1.05 | 40.532 |
| -0.0357 | -1.008 | 9.624 | 1.343 | 87.682 |
| -0.0136 | -0.609 | 4.743 | 1.441 | 186.275 |
| 0.0085 | -0.609 | 4.743 | 1.223 | 58.305 |
| 0.0306 | -0.553 | 4.311 | 1.021 | 71.474 |
| 0.0527 | -0.459 | 3.572 | 0.87 | 39.956 |
| 0.0747 | -0.453 | 3.527 | 0.861 | 34.781 |

The algorithm using regularization of the KKT reformulated problem (subsection 3.3.2) and the algorithm using NCP function to circumvent irregularity

of the KKT reformulation (subsection 3.3.3) use a control sequence $\varepsilon_k$. For each initial portfolio, several (usually about ten) sequences were used and one of the obtained solutions was chosen (usually the solution closest to being feasible). The tried sequences were $\frac{c}{k^\beta}$, $c\,e^{-k}$ or $\frac{c}{log(1+k)}$ with $c \in \langle 10^{-8}, 10^6 \rangle$ and $\beta \in \langle 0.25, 8 \rangle$.

200 iterations ($k = 1, \ldots, 200$) were used for the KKT regularization algorithm, except the first problem, where $k = 1, \ldots, 100$ was used. The results were highly dependent on the chosen control sequence, especially on the rate of convergence of the sequence. Only the solution for the first problem (first initial portfolio) is feasible and optimal, all the other solutions are infeasible and not optimal, no matter which control sequence was used. An overview of the results is in Table 8.3. The computed portfolios can be found in attachments in Table 8.14 and in Table 8.15.

Table 8.3: Results obtained by the KKT regularization algorithm. Achieved loss, $CVaR$, objective function value (equal to $\|x - x^0\|_1$) of obtained solutions of the problem (7.6) - (7.2) with initial solutions $X^0$ from subsection 8.1.2 and total computing time. All except the first solutions are infeasible to the BLPP and not optimal. The computing time depends mainly on the number of iterations used.

| $l_{max}$ Initial Solution | Loss | $CVaR_\alpha$ | Objective function value | Time (s) | Remark |
|---|---|---|---|---|---|
| -0.1241 | -0.062 | 1.315 | 1.896 | 12.228 | $\varepsilon_k = 4/log(1+k)$ |
| -0.1020 | -0.072 | 1.421 | 1.91 | 24.816 | Infeasible |
| -0.0799 | -0.257 | 6.957 | 1.001 | 26.219 | Infeasible |
| -0.0578 | -0.116 | 5.056 | 1.369 | 26.613 | Infeasible |
| -0.0357 | -0.114 | 4.889 | 1.339 | 26.453 | Infeasible |
| -0.0136 | -0.072 | 1.421 | 1.585 | 24.93 | Infeasible |
| 0.0085 | -0.09 | 4.137 | 1.202 | 26.069 | Infeasible |
| 0.0306 | -0.072 | 1.421 | 1.207 | 25.327 | Infeasible |
| 0.0527 | -0.072 | 1.421 | 1.031 | 26.411 | Infeasible |
| 0.0747 | -0.072 | 1.421 | 0.991 | 25.737 | Infeasible |

500 iterations ($k = 1, \ldots, 500$) were used for the KKT NCP function algorithm. The solution is highly dependent on the chosen control sequence: A sequence with high convergence rate usually yielded a feasible solution with the same optimal objective function value as the MIP KKT algorithm. The solution was also highly dependent on the value of $\varepsilon_1$, i.e. the first element of the control sequence. An overview of the results is in Table 8.4. The computed portfolios can be found in attachments in Table 8.16 and in Table 8.17.

If we compare the three algorithms, the KKT MIP algorithm is the most reliable one. Even though it can use more computing time than the other two algorithms, the absence of the requirement of choosing the correct control sequence makes up for this. The KKT regularization algorithm yielded a feasible (and probably optimal as well) solution only for one of the ten problems and the solution obtained by this algorithm depended more on the chosen control sequence than on the initial portfolio.

Table 8.4: Results obtained by the KKT NCP function algorithm. Achieved loss, $CVaR$, objective function value (equal to $\|x - x^0\|_1$) of obtained solutions of the problem (7.6) - (7.2) with initial solutions $X^0$ from subsection 8.1.2 and total computing time. Three of the obtained solutions have lower optimal objective function value than the according solutions obtained by the KKT MIP algorithm. Since each problem used the same number of iterations, the computing time does not show a great variability.

| $l_{max}$ Initial Solution | Loss | $CVaR_\alpha$ | Objective function value | Time (s) | Remark |
|---|---|---|---|---|---|
| -0.1241 | -0.063 | 1.314 | 1.896 | 81.552 | $\varepsilon_k = 1/k^2$ |
| -0.1020 | -1.297 | 15.475 | 0.198 | 64.836 | $\varepsilon_k = 10/k^3$ |
| -0.0799 | -1.103 | 11.376 | 0.763 | 76.95 | $\varepsilon_k = 10/k^3$ |
| -0.0578 | -1.005 | 9.555 | 1.05 | 69.447 | $\varepsilon_k = 10/k^3$ |
| -0.0357 | -1.365 | 16.953 | 1.343 | 70.059 | $\varepsilon_k = 1/k^4$ |
| -0.0136 | -0.063 | 1.314 | 1.69 | 66.632 | Not optimal |
| 0.0085 | -0.138 | 1.442 | 1.439 | 65.373 | Not optimal |
| 0.0306 | -0.553 | 4.311 | 1.021 | 66.323 | $\varepsilon_k = 100/k^{3.5}$ |
| 0.0527 | -0.417 | 3.246 | 0.905 | 69.432 | Not optimal |
| 0.0747 | -0.453 | 3.527 | 0.861 | 67.112 | $\varepsilon_k = 0.1/k^3$ |

### 8.2.2 Euclidean norm

In this subsection we solve the problem (7.7) - (7.2) using the initial portfolio $X^0$ obtained in the subsection 8.1.2. The solution is analogical to the previous subsection. The solver parameters for the KKT MIP algorithm remain the same, as well as the procedure for choosing the best control sequence $\varepsilon_k$ (including the number of iterations) for the KKT regularization algorithm and the KKT NCP function algorithm.

The KKT MIP algorithm used more time in the case of the problem (7.7) - (7.2), the main reason for this is nonlinearity of this problem. An overview of the results is in Table 8.5. The computed portfolios can be found in attachments in Table 8.18 and in Table 8.19.

Similarly to the previous section, all of the solutions yielded by the KKT regularization algorithm (except the solution to the first of the ten problems) are identical and infeasible. The solutions were again highly dependent on the rate of convergence of the control sequence and in this case on the first element of the control sequence $\varepsilon_1$ as well. An overview of the results is in Table 8.6. The computed portfolios can be found in attachments in Table 8.20 and in Table 8.21.

The KKT NCP function works similarly as in the previous subsection, even though the dependency of the solution on the control sequence seems to be lower (several control sequences lead to the same solution). All of the obtained solutions are feasible. Five of them have higher optimal objective function value than the solution obtained by the KKT MIP algorithm, and one of the solutions (the third one) has lower objective function value than the solution obtained by the KKT MIP algorithm. Five of the solutions are identical. An overview of the solutions is in Table 8.7. The computed portfolios can be found in attachments in Table 8.22 and in Table 8.23.

Table 8.5: Results obtained by the KKT MIP algorithm. Achieved loss, $CVaR$, objective function value (equal to the square of $\|x - x^0\|_2$) of obtained solutions of the problem (7.7) - (7.2) with initial solutions $X^0$ from subsection 8.1.2 and total computing time. Note that the computing time shows great variability, even exceeding the limit of 1800 second in three cases.

| $l_{max}$ Initial Solution | Loss | $CVaR_\alpha$ | Objective function value | Time (s) | Remark |
|---|---|---|---|---|---|
| -0.1241 | -0.063 | 1.314 | 1.31 | 415.616 | |
| -0.1020 | -1.331 | 16.214 | 0.011 | 27.36 | |
| -0.0799 | -0.862 | 7.185 | 0.733 | 45.878 | Not optimal |
| -0.0578 | -1.139 | 12.047 | 0.281 | 467.409 | |
| -0.0357 | -1.184 | 13.032 | 0.463 | 1800 | |
| -0.0136 | -0.16 | 1.506 | 0.588 | 211.448 | |
| 0.0085 | -0.463 | 3.606 | 0.401 | 683.006 | |
| 0.0306 | -0.36 | 2.822 | 0.27 | 1800 | |
| 0.0527 | -0.063 | 1.314 | 0.224 | 1800 | |
| 0.0747 | -0.303 | 2.41 | 0.162 | 310.579 | |

Table 8.6: Results obtained by the KKT regularization algorithm. Achieved loss, $CVaR$, objective function value (equal to the square of $\|x - x^0\|_2$) of obtained solutions of the problem (7.7) - (7.2) with initial solutions $X^0$ from subsection 8.1.2 and total computing time. There are two facts to be mentioned: All solutions except the first one are identical and infeasible; and this algorithm works significantly faster than the KKT MIP algorithm.

| $l_{max}$ Initial Solution | Loss | $CVaR_\alpha$ | Objective function value | Time (s) | Remark |
|---|---|---|---|---|---|
| -0.1241 | -0.062 | 1.315 | 1.307 | 13.078 | $\varepsilon_k = 4/log(1+k)$ |
| -0.1020 | -0.072 | 1.421 | 1.125 | 25.772 | Infeasible |
| -0.0799 | -0.072 | 1.421 | 0.753 | 28.547 | Infeasible |
| -0.0578 | -0.072 | 1.421 | 0.625 | 25.193 | Infeasible |
| -0.0357 | -0.072 | 1.421 | 0.582 | 16.279 | Infeasible |
| -0.0136 | -0.072 | 1.421 | 0.583 | 29.343 | Infeasible |
| 0.0085 | -0.072 | 1.421 | 0.426 | 30.677 | Infeasible |
| 0.0306 | -0.072 | 1.421 | 0.299 | 25.88 | Infeasible |
| 0.0527 | -0.072 | 1.421 | 0.211 | 25.864 | Infeasible |
| 0.0747 | -0.072 | 1.421 | 0.191 | 25.87 | Infeasible |

If we compare the three algorithms, the KKT MIP algorithm is, similarly to the subsection 8.2.1 the most reliable one. All of the solutions it yielded are feasible and all but one seem to be optimal. The disadvantage of this algorithm is large computing time, in several cases even exceeding the limit of 1800 seconds. The KKT regularization algorithm computed only one feasible solution, while the other nine solutions are infeasible and their objective function value is in most higher than the objective function value of the solutions provided by the KKT

Table 8.7: Results obtained by the KKT NCP function algorithm. Achieved loss, $CVaR$, objective function value (equal to the square of $\|x - x^0\|_2$) of obtained solutions of the problem (7.7) - (7.2) with initial solutions $X^0$ from subsection 8.1.2 and total computing time. Five of the solutions are identical and five of the solutions are not optimal.

| $l_{max}$ Initial Solution | Loss | $CVaR_\alpha$ | Objective function value | Time (s) | Remark |
|---|---|---|---|---|---|
| -0.1241 | -0.063 | 1.314 | 1.31 | 71.036 | $\varepsilon_k = e^{-k}$ |
| -0.1020 | -1.331 | 16.214 | 0.011 | 64.968 | $\varepsilon_k = 10/k^3$ |
| -0.0799 | -1.234 | 14.111 | 0.146 | 75.942 | $\varepsilon_k = 10/k^3$ |
| -0.0578 | -0.063 | 1.314 | 0.726 | 68.071 | Not optimal |
| -0.0357 | -1.365 | 16.953 | 0.677 | 65.048 | Not optimal |
| -0.0136 | -0.063 | 1.314 | 0.681 | 69.543 | Not optimal |
| 0.0085 | -0.138 | 1.442 | 0.437 | 68.333 | Not optimal |
| 0.0306 | -0.063 | 1.314 | 0.331 | 65.636 | Not optimal |
| 0.0527 | -0.063 | 1.314 | 0.224 | 65.091 | $\varepsilon_k = 1/k^3$ |
| 0.0747 | -0.303 | 2.41 | 0.162 | 68.538 | $\varepsilon_k = 0.1/k^3$ |

MIP algorithm. The advantage of this algorithm is its speed. The KKT NCP function algorithm is between the two other algorithms. All of the solutions it computed are feasible and five of them seem to be optimal, while the computing time never exceeds 80 seconds.

## 8.3 Algorithm Based on Optimal Value Function Reformulation

This section describes results from solving the problem (7.6) - (7.2) using the global algorithm based on an approximation of the optimal value function in the optimal value function reformulated problem 4.1. The algorithm described in the section 4.2 and in the subsection 4.2.1. We shall abbreviate this algorithm as OVFRA algorithm.

We solve the same problem (and using the same data) as in the subsection 8.2.1.

Two implementations of the OVFRA algorithm were used. The section 8.3.1 describes results provided by the algorithm, which is implemented as described in the subsection 4.2.1 (and in the monograph Dempe et al. [11] and the paper Dempe and Franke [8]). The section 8.3.2 describes an original modification of this algorithm which significantly improves the computing time, and results provided by the modified algorithm.

Both the original and the modified algorithms were implemented using R with package *lpSolve*.

### 8.3.1 Original Implementation of the OVFRA Algorithm

The problem (7.6) - (7.2) has been solved using the OVFRA algorithm.

The remark in the subsection 4.2.1 stating that in step 3 of the algorithm only the region of stability containing $x^t$ needs to be updated during the computation of the function $\Xi_{Z_{k+1}}(\cdot)$ was implemented. The function $\Xi_{Z_{k+1}}(\cdot)$ is defined in 4.4. The stopping criterion $y^t \in \Psi(x^t)$ described in step 3 of the algorithm was given a tolerance of $10^{-6}$. In our case, it means that the algorithm stops, if $|l_{max}^{k+1} - l_{max}^{k}| < 10^{-6}$, where $l_{max}^{k}$ denotes the maximal loss of the portfolio computed in k-th step of the algorithm.

The low tolerance of $10^{-6}$ leads to reasonable computing times, however some inaccuracies occur on the solutions. With respect to these inaccuracies, all of the provided solutions are feasible and optimal (objective function value is the same as with the KKT MIP algorithm, see Table 8.2). An overview of the results is in Table 8.8. The computed portfolios can be found in attachments in Table 8.24 and in Table 8.25.

Table 8.8: Results obtained by the OVFRA algorithm. Achieved loss, $CVaR$, objective function value (equal to $\|x - x^0\|_1$) of obtained solutions of the problem (7.6) - (7.2) with initial solutions $X^0$ from subsection 8.1.2 and total computing time. All of the solutions are feasible and optimal (compare with Table 8.2). Note the high computing time of the first problem.

| $l_{max}$ Initial Solution | Loss | $CVaR_\alpha$ | Objective function value | Time (s) |
|---|---|---|---|---|
| -0.124 | -0.063 | 1.314 | 1.896 | 259.172 |
| -0.102 | -1.365 | 16.953 | 0.198 | 0.073 |
| -0.080 | -1.139 | 12.047 | 0.763 | 1.537 |
| -0.058 | -1.139 | 12.047 | 1.050 | 4.779 |
| -0.036 | -0.993 | 9.344 | 1.343 | 9.893 |
| -0.014 | -0.609 | 4.743 | 1.441 | 14.572 |
| 0.008 | -0.609 | 4.743 | 1.223 | 11.574 |
| 0.031 | -0.553 | 4.311 | 1.021 | 6.534 |
| 0.053 | -0.459 | 3.572 | 0.870 | 4.681 |
| 0.075 | -0.453 | 3.527 | 0.861 | 0.933 |

### 8.3.2 Modified Implementation OVFRA Algorithm

The OVFRA algorithm is much faster than the MIP algorithm (except for the problem with the first initial portfolio), however improving the tolerance in the stopping criterion from $|l_{max}^{k+1} - l_{max}^{k}| < 10^{-6}$ to $|l_{max}^{k+1} - l_{max}^{k}| < 10^{-7}$ could lead to very high computing times. An original modification of this algorithm improving its computing time (especially for the cases when the computing time is very high) is described in this section.

Since the lower level objective function in the problem (7.6) - (7.2) depends only on $l_{max}$, the optimal value function reformulation of the problem (7.6) - (7.2) with the optimal value function $\varphi(\cdot)$ replaced by the function $\Xi_Z(\cdot)$, defined in

4.4, takes the form

$$\begin{aligned}
\min_{l_{max},b,a,z,x} & \sum_{i=1}^{n} b_i \\
s.t.\ b_i &\geq x_i - x_i^0,\ i = 1, \ldots, n, \\
b_i &\geq x_i^0 - x_i,\ i = 1, \ldots, n, \\
0 &\geq \left(a + \frac{1}{1-\alpha}\sum_{s=1}^{S} p^s z^s\right) - \Xi_Z(l_{max}), \\
z^s &\geq \sum_{i=1}^{n} x_i \xi_i^s - a,\ s = 1, \ldots, S, \\
l_{max} &\geq \sum_{i=1}^{n}\sum_{s=1}^{S} x_i \xi_i^s p^s, \\
1 &= \sum_{i=1}^{n} x_i, \\
x &\geq 0,\ z \geq 0,\ a \in \mathbb{R},\ l_{max} \in \mathbb{R},
\end{aligned} \tag{8.1}$$

for some finite set $Z \subset \mathbb{R}$. In the implementation of the algorithm, in $k-th$ iteration we have $Z_k = \{l_1^{Z_k}, \ldots, l_{k+1}^{Z_k}\}$. If we (without loss of generality) reorder the set $Z_k$ so that $l_1^{Z_k} \leq \ldots \leq l_{k+1}^{Z_k}$, then $Z_k$ is a partition of the interval $\langle l_1^{Z_k}, l_{k+1}^{Z_k}\rangle$ where

$$l_1^{Z_k} = \min_i \left\{ \sum_{s=1}^{S} \xi_i^s p^s :\ i \in \{1, \ldots, 10\} \right\}$$

is the mean loss of the stock with the smallest mean loss and

$$l_{k+1}^{Z_k} = \max_i \left\{ \sum_{s=1}^{S} \xi_i^s p^s :\ i \in \{1, \ldots, 10\} \right\}$$

is the mean loss of the stock with the greatest mean loss. In the first step of the algorithm we use the set $Z_k$ to compute the function $\Xi_{Z_k}(l_{max})$, and in the second step we solve the problem 8.1 which means that for each $j = 1, \ldots, k$ a linear programming problem (which will be described shortly) and take the best solution. The problem solved is the problem (8.1) where we replace the third constraint

$$0 \geq \left(a + \frac{1}{1-\alpha}\sum_{s=1}^{S} p^s z^s\right) - \Xi_Z(l_{max}) \tag{8.2}$$

with

$$\begin{aligned}
0 &\geq \left(a + \frac{1}{1-\alpha}\sum_{s=1}^{S} p^s z^s\right) - w\, l_{max} - v, \\
l_j^{Z_k} &\leq l_{max} \leq l_{j+1}^{Z_k},
\end{aligned} \tag{8.3}$$

where

$$w = \frac{\varphi(l_{j+1}^{Z_k}) - \varphi(l_j^{Z_k})}{l_{j+1}^{Z_k} - l_j^{Z_k}}$$

is the gradient of the line which equals to $\Xi_Z(l_{max})$ on the region of stability $\langle l_j^{Z_k}, l_{j+1}^{Z_k} \rangle$ and

$$v = \varphi(l_j^{Z_k}) \; - \; w l_j^{Z_k}$$

is the intercept of this line. We remind that in our case $\varphi(l)$ is the (global) optimal function value of the problem (7.2) with $l_{max} = l$.

As we can see, in the second step of $k-th$ iteration of the OVFRAA algorithm, we have to solve $k$ linear programming problems, which can lead to very high computing time. The modification of the algorithm is adding a cache which stores the optimal values of the (upper level) objective function for each of the regions of stability $\langle l_j^{Z_k}, l_{j+1}^{Z_k} \rangle$. The second step of the OVFRAA algorithm then reduces to solving the problem (8.1) with the constraint (8.2) replaced by (8.3) only for the two regions of stability which contain the point added to $Z$ in the last iteration of the algorithm.

A similar modification would be possible even if the lower level objective function depended on more than one upper level variables.

The modification allowed us to set the tolerance in stopping criterion to $10^{-8}$ instead of $10^{-6}$, which was used for the original implementation. The modified algorithm obviously leads to the same solutions as the original algorithm, however, the solution of the fifth problem is different. This is caused by the lowering of the tolerance in stopping criterion, tolerance of $10^{-6}$ leads to the same solution as the original OVFRA algorithm. The computing time for most of the solutions is smaller (even with the smaller tolerance of $10^{-8}$), especially for the solutions with the largest computing time. An overview of the results is in Table 8.9. The computed portfolios can be found in attachments in Table 8.26 and in Table 8.27.

In section 8.3 we have used the OVFRA algorithm to solve the problem (7.6) - (7.2). We have described and used a modification of this algorithm (to solve the same problem). We used the KKT MIP algorithm, the KKT regularization algorithm and the KKT NCP function algorithm to solve the same problem (7.6) - (7.2) in subsection 8.2.1. The best of these three algorithms is the KKT MIP algorithm (for comparison see section 8.2.1). If we compare it to the modified version of the OVFRA algorithm, we see that the modified OVFRA is superior in terms of the computing time and does not yield worse solutions, however, the implementation of the modified OVFRA algorithm is more complicated than the implementation of the KKT MIP algorithm.

Table 8.9: Results obtained by the modified OVFRA algorithm. Achieved loss, $CVaR$, objective function value (equal to $\|x - x^0\|_1$) of obtained solutions of the problem (7.6) - (7.2) with initial solutions $X^0$ from subsection 8.1.2 and total computing time. All of the solutions are feasible and optimal (compare with Table 8.2). Note that the computing times are in most cases lower than with the original OVFRA algorithm with the greatest improvement being in the first problem (compare with Table 8.8). Also, note that lowering the tolerance in the stopping criterion caused that the fifth solution is different from the solution obtained by the original OVFRA algorithm.

| $l_{max}$ Initial Solution | Loss | $CVaR_\alpha$ | Objective function value | Time (s) |
|---|---|---|---|---|
| 0.015 | -0.063 | 1.314 | 1.896 | 24.142 |
| 1.000 | -1.365 | 16.953 | 0.198 | 0.111 |
| 0.671 | -1.139 | 12.047 | 0.763 | 1.182 |
| 0.671 | -1.139 | 12.047 | 1.050 | 2.231 |
| 0.671 | -1.139 | 12.047 | 1.343 | 3.121 |
| 0.131 | -0.609 | 4.743 | 1.441 | 3.612 |
| 0.131 | -0.609 | 4.743 | 1.223 | 3.318 |
| 0.110 | -0.553 | 4.311 | 1.021 | 2.198 |
| 0.075 | -0.459 | 3.572 | 0.870 | 1.471 |
| 0.072 | -0.453 | 3.527 | 0.861 | 0.619 |

# Conclusion

In this work, we have described the bilevel programming problem with its basic properties. We have presented solution algorithms which can be divided into three groups.

The first group is the approach of KKT transformation, where the lower level problem of an optimistic bilevel programming problem is replaced by its Karush-Kuhn-Tucker conditions. We have discussed assumptions, under which is the KKT reformulated problem equivalent to the original bilevel programming problem. The KKT reformulated problem itself is difficult to solve, due to violation of common constraint qualifications. Thus, three algorithms based on the KKT reformulation have been proposed: An algorithm which reduces the BLPP to a single level mixed-integer problem, an algorithm which regularizes the KKT reformulated problem and an algorithm which replaces the complementarity condition by a nonlinear complementarity function.

The second group is the approach of optimal value function reformulation, where an optimistic bilevel problem is reduced to a single level problem using the lower level optimal value function. We have presented a constraint qualification called partial calmness, specific to this. An algorithm based on the optimal value function reformulation for an optimistic bilevel programming problem with jointly convex lower level problem was described in both global and local versions.

In the last group there are algorithms for linear bilevel programming problems. Linearity in both upper and lower level problems results into simplifications based on parametric programming. We have described two solution algorithms for linear bilevel programming problems. The first one, based on duality, solves a linear bilevel programming problem where the upper level variable is present on the right-hand side of the lower level constraint. The second one, based on optimal value function, solves a linear bilevel programming problem where the upper level variable is present in the lower level objective function.

A comparison of the assumptions of the presented algorithms has been discussed in chapter 6. Chapter 7 has reminded the reader basic concepts of mean-CVaR portfolio optimization and proposed a bilevel programming problem which, for a given initial portfolio, finds a portfolio which is efficient with respect to mean loss and conditional value at risk and which is closest to the initial portfolio.

We have then solved the bilevel programming problem described in chapter 7 (with real-life data) using four of the presented algorithms: All three of the algorithms based on KKT reformulation and the algorithm based on optimal value function reformulation (abbreviated as the OVFRA algorithm). We have then proposed a new modification of the last algorithm, resulting in significant shortening of the computing time. Results, as well as computing times of all the algorithms, have then been compared.

Both the modified version and the non-modified version of the OVFRA algorithm and the algorithm based on KKT reformulation, which reduces the bilevel programming problem to a single level mixed-integer problem (KKT MIP algorithm), proved to be the most reliable since all the solutions obtained by these algorithms were feasible (i.e. the computed portfolios were efficient). Of the three,

the modified OVFRA algorithm was the fastest, followed by its non-modified version, whereas the KKT MIP was the slowest. This was offset by higher difficulty of the implementation of the OVFRA algorithm (especially the modified version).

The other two algorithms based on KKT reformulation proved to be less reliable: The algorithm using regularization computed optimal solutions for only two of the twenty solved problems (the other were infeasible for the bilevel problem), notwithstanding substantial efforts to find a suitable regularization sequence. The algorithm using NCP functions computed optimal solutions for twelve of the twenty solved problems (however, the other solutions were feasible). Both algorithms were satisfactorily fast, but in most cases slower than the modified OVFRA algorithm.

The original contribution of this thesis is the compilation of bilevel optimization theory, including several original examples, proofs and discussions, and solution algorithms for bilevel programming problems, together with the implementation of some of the presented algorithms on a real-life problem and empirical comparison of these algorithms. The most important contribution is, however, an original modification of one of the described algorithms.

We have described the most common and popular solving algorithms for bilevel programming problems, however, other possible algorithms not mentioned in this work exist, too.

More algorithms based on the KKT reformulation (e.g. different approaches to regularization) are described in Dempe et al. [11] and in Mersha [20].

Some algorithms solve only special cases of bilevel programming problems. An algorithm for a bilevel programming problem with a unique solution of the lower level problem is described in section 3.4 of Dempe et al. [11], the same algorithm was proposed earlier in Dempe [5]. Another special case with upper and lower level variables being the same and both upper objective functions as well as upper level feasible set being convex, resulting in a convex programming problem, is described in chapter 4 of Dempe et al. [11].

An evolutionary (genetic) algorithm, was described and applied to a product portfolio management problem in Ma [19].

A solution method for pessimistic linear mixed-integer bilevel programming problem was described in Zheng et al. [31]. More on mixed-integer bilevel programming problems can be found e.g. in chapter 5 of Dempe et al. [11].

# Bibliography


[1] J. Bracken and J. T. McGill. Defense applications of mathematical programs with optimization problems in the constraints. *Operations Research*, 22(5): 1086–1096, 1974. ISSN 0030364X, 15265463.

[2] W. Candler and R.J. Townsley. *A Linear Two-level Programming Problem*. Discussion Papers, Development Research Center, International Bank for Reconstruction and Development. World Bank, 1978.

[3] B. Colson, P. Marcotte, and G. Savard. An overview of bilevel optimization. *Annals of Operations Research*, 153(1):235–256, Sep 2007. ISSN 1572-9338.

[4] S. Dempe. A simple algorithm for the-linear bilevel programming problem. *Optimization*, 18(3):373–385, 1987. doi: 10.1080/02331938708843247.

[5] S. Dempe. *Foundations of Bilevel Programming*. Kluwer Academic Publishers, 2002. ISBN 1-4020-0631-4.

[6] S. Dempe and J. Dutta. Is bilevel programming a special case of a mathematical program with complementarity constraints? *Math. Program.*, 131 (1-2):37–48, 2012.

[7] S. Dempe and S. Franke. Solution algorithm for an optimistic linear stackelberg problem. *Comput. Oper. Res.*, 41:277–281, January 2014. ISSN 0305-0548.

[8] S. Dempe and S. Franke. On the solution of convex bilevel optimization problems. *Computational Optimization and Applications*, 63(3):685–703, 2016.

[9] S. Dempe and S. Lohse. Inverse linear programming. In A. Seeger, editor, *Recent Advances in Optimization*, pages 19–28. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006. ISBN 978-3-540-28258-7.

[10] S. Dempe and A. B. Zemkoho. The bilevel programming problem: reformulations, constraint qualifications and optimality conditions. *Mathematical Programming*, 138(1):447–473, Apr 2013. ISSN 1436-4646.

[11] S. Dempe, V. Kalashnikov, G. A. Perez-Valdes, and N. Kalashnykova. *Bilevel Programming Problems: Theory, Algorithms and Applications to Energy Networks*. Springer Publishing Company, Incorporated, 2015. ISBN 3662458268, 9783662458266.

[12] J. Dupačová, J. Hurt, and J. Štěpán. *Stochastic Modeling in Economics and Finance*. Springer US, 2002. ISBN 978-0-306-48167-3.

[13] M. Fukushima and J.-S. Pang. Convergence of a smoothing continuation method for mathematical progams with complementarity constraints. In A. Antipin, F. Vasil'ev, M. Théra, and R. Tichatschke, editors, *Ill-posed Variational Problems and Regularization Techniques*, Lecture Notes in Economics and Mathematical Systems 477. Springer-Verlag Berlin Heidelberg, 1 edition, 1999. ISBN 978-3-540-66323-2,978-3-642-45780-7.

[14] C. Geiger and C. Kanzow. *Theorie und Numerik restringierter Optimierungsaufgaben*. Springer-Lehrbuch Masterclass. Springer Berlin Heidelberg, 2002. ISBN 9783540427902.

[15] L. Goduľová. Dvojúrovňové optimalizačné modely a ich využitie v úlohách optimalizácie portfólia. Master's thesis, Charles University in Prague, 2018.

[16] L. Grygarová. *Úvod do parametrického programování*. Karolinum, 1994. ISBN 80-7066-876-8.

[17] P. Hansen, B. Jaumard, and G. Savard. New branch-and-bound rules for linear bilevel programming. *SIAM Journal on Sicentific and Statistical Computing*, 13(5):1194–1217, 1992. ISSN 0196-5204, 2168-3417.

[18] P. Lachout. Optimization theory - direct approach, October 2018. URL `http://www.karlin.mff.cuni.cz/~lachout/Vyuka/T-Optima/181014-TO-OptI-text.pdf`.

[19] S. Ma. A nonlinear bi-level programming approach for product portfolio management. *SpringerPlus*, 5(1):727, Jun 2016. ISSN 2193-1801.

[20] A. G. Mersha. *Solution Methods for Bilevel Programming Problems*. Verlag Dr. Hut, 2009. ISBN 9783899639315.

[21] S. Rocciolietti. *Backtesting Value at Risk and Expected Shortfall*. Gabler Verlag, 2016. ISBN 978-3-658-11908-9.

[22] R. T. Rockafellar and S. Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–41, 2000.

[23] R. T. Rockafellar and S. Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking and Finance*, 26:1443–1471, 2002.

[24] R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis*. Springer Verlag, Heidelberg, Berlin, New York, 1998.

[25] H. Scheel and S. Scholtes. Mathematical programs with complementarity constraints: Stationarity, optimality, and sensitivity. *Math. Oper. Res.*, 25: 1–22, 02 2000.

[26] H. von Stackelberg. *Marktform und Gleichgewicht*. Die Handelsblatt-Bibliothek "Klassiker der Nationalökonomie". J. Springer, 1934.

[27] N. Yamashita and M. Fukushima. Properties of restricted ncp functions for nonlinear complementarity problems. *Journal of Optimization Theory and Applications*, 98(3):701–717, 1998.

[28] J. J. Ye. Constraint qualifications and kkt conditions for bilevel programming problems. *Math. Oper. Res.*, 31(4):811–824, nov 2006. ISSN 0364-765X.

[29] J. J. Ye and D.i L. Zhu. Optimality conditions for bilevel programming problems. *Optimization*, 33(1):9–27, 1995. doi: 10.1080/02331939508844060.

[30] A. Zemkoho. *Bilevel programming: reformulations, regularity, and stationarity*. PhD thesis, TU Bergakademie Freiberg, 2012.

[31] Y. Zheng, G. Zhang, J. Han, and J. Lu. Pessimistic bilevel optimization model for risk-averse production-distribution planning. *Inf. Sci.*, 372(C): 677–689, December 2016. ISSN 0020-0255.

# List of Figures

# List of Tables

# Attachments

Table 8.10: The first five of the initial portfolios computed in subsection 8.1.2.

| $l_{max}$ | -0.124 | -0.102 | -0.08 | -0.058 | -0.036 |
|---|---|---|---|---|---|
| AAPL | 0.015 | 1 | 0.671 | 0.671 | 0.671 |
| AMD | 0.027 | 0 | 0 | 0 | 0 |
| BAC | 0 | 0 | 0 | 0 | 0 |
| F | 0.615 | 0 | 0 | 0 | 0 |
| FL | 0.047 | 0 | 0 | 0 | 0 |
| GE | 0.101 | 0 | 0 | 0 | 0 |
| CHK | 0.052 | 0 | 0 | 0 | 0 |
| MSFT | 0.003 | 0 | 0.329 | 0.329 | 0.329 |
| QGEN | 0 | 0 | 0 | 0 | 0 |
| ATT | 0.14 | 0 | 0 | 0 | 0 |

Table 8.11: The last five of the initial portfolios computed in subsection 8.1.2.

| $l_{max}$ | -0.014 | 0.008 | 0.031 | 0.053 | 0.075 |
|---|---|---|---|---|---|
| AAPL | 0.205 | 0.218 | 0.256 | 0.286 | 0.222 |
| AMD | 0 | 0 | 0 | 0.008 | 0.075 |
| BAC | 0 | 0 | 0 | 0 | 0 |
| F | 0.014 | 0.123 | 0.247 | 0.362 | 0.369 |
| FL | 0.535 | 0.463 | 0.361 | 0.268 | 0.291 |
| GE | 0.002 | 0 | 0 | 0 | 0 |
| CHK | 0.005 | 0.003 | 0.001 | 0 | 0.016 |
| MSFT | 0 | 0 | 0 | 0 | 0 |
| QGEN | 0.168 | 0.140 | 0.109 | 0.073 | 0 |
| ATT | 0.070 | 0.052 | 0.026 | 0.003 | 0.027 |

Table 8.12: The first five of the portfolios solving the problem (7.6) - (7.2) computed by the KKT MIP algorithm in subsection 8.2.1.

| $l_{max}$ Initial Solution | -0.124 | -0.102 | -0.08 | -0.058 | -0.036 |
|---|---|---|---|---|---|
| AAPL | 0.015 | 1 | 0.653 | 0.511 | 0.48 |
| AMD | 0.027 | 0 | 0 | 0 | 0 |
| BAC | 0 | 0 | 0 | 0 | 0 |
| F | 0.615 | 0 | 0 | 0 | 0 |
| FL | 0.047 | 0 | 0 | 0 | 0 |
| GE | 0.101 | 0 | 0 | 0 | 0 |
| CHK | 0.052 | 0 | 0 | 0 | 0 |
| MSFT | 0.003 | 0 | 0.347 | 0.489 | 0.520 |
| QGEN | 0 | 0 | 0 | 0 | 0 |
| ATT | 0.140 | 0 | 0 | 0 | 0 |

Table 8.13: The last five of the portfolios solving the problem (7.6) - (7.2) computed by the KKT MIP algorithm in subsection 8.2.1.

| $l_{max}$ Initial Solution | -0.014 | 0.008 | 0.031 | 0.053 | 0.075 |
|---|---|---|---|---|---|
| AAPL | 0.131 | 0.131 | 0.110 | 0.075 | 0.072 |
| AMD | 0 | 0 | 0 | 0 | 0 |
| BAC | 0 | 0 | 0 | 0 | 0 |
| F | 0.180 | 0.180 | 0.247 | 0.362 | 0.369 |
| FL | 0.135 | 0.135 | 0.132 | 0.129 | 0.128 |
| GE | 0 | 0 | 0 | 0 | 0 |
| CHK | 0 | 0 | 0 | 0 | 0 |
| MSFT | 0.555 | 0.555 | 0.511 | 0.435 | 0.431 |
| QGEN | 0 | 0 | 0 | 0 | 0 |
| ATT | 0 | 0 | 0 | 0 | 0 |

Table 8.14: The first five of the portfolios solving the problem (7.6) - (7.2) computed by the KKT regularization algorithm in subsection 8.2.1.

| $l_{max}$ Initial Solution | -0.124 | -0.102 | -0.08 | -0.058 | -0.036 |
|---|---|---|---|---|---|
| AAPL | 0.015 | 0.023 | 0.118 | 0.030 | 0.016 |
| AMD | 0.028 | 0.018 | 0.015 | 0.054 | 0.052 |
| BAC | 0 | 0 | 0 | 0 | 0.036 |
| F | 0.612 | 0.510 | 0.279 | 0.282 | 0.225 |
| FL | 0.047 | 0.077 | 0.134 | 0.070 | 0.080 |
| GE | 0.097 | 0.282 | 0.074 | 0.239 | 0.245 |
| CHK | 0.052 | 0.0006 | 0.013 | 0.039 | 0.037 |
| MSFT | 0.003 | 0 | 0.008 | 0.034 | 0.050 |
| QGEN | 0 | 0.043 | 0.243 | 0.195 | 0.213 |
| ATT | 0.144 | 0.048 | 0.114 | 0.057 | 0.046 |

Table 8.15: The last five of the portfolios solving the problem (7.6) - (7.2) computed by the KKT regularization algorithm in subsection 8.2.1.

| $l_{max}$ Initial Solution | -0.014 | 0.008 | 0.031 | 0.053 | 0.075 |
|---|---|---|---|---|---|
| AAPL | 0.023 | 0.013 | 0.023 | 0.023 | 0.023 |
| AMD | 0.018 | 0.043 | 0.018 | 0.018 | 0.018 |
| BAC | 0 | 0.025 | 0 | 0 | 0 |
| F | 0.510 | 0.148 | 0.510 | 0.510 | 0.510 |
| FL | 0.077 | 0.070 | 0.077 | 0.077 | 0.077 |
| GE | 0.282 | 0.228 | 0.282 | 0.282 | 0.282 |
| CHK | 0.0006 | 0.072 | 0.0006 | 0.0006 | 0.0006 |
| MSFT | 0 | 0.044 | 0 | 0 | 0 |
| QGEN | 0.043 | 0.242 | 0.043 | 0.043 | 0.043 |
| ATT | 0.048 | 0.058 | 0.048 | 0.048 | 0.048 |

Table 8.16: The first five of the portfolios solving the problem (7.6) - (7.2) computed by the KKT NCP function algorithm in subsection 8.2.1.

| $l_{max}$ Initial Solution | -0.124 | -0.102 | -0.08 | -0.058 | -0.036 |
|---|---|---|---|---|---|
| AAPL | 0.015 | 0.901 | 0.619 | 0.475 | 1 |
| AMD | 0.027 | 0 | 0 | 0 | 0 |
| BAC | 0 | 0 | 0 | 0 | 0 |
| F | 0.615 | 0 | 0 | 0 | 0 |
| FL | 0.047 | 0 | 0 | 0 | 0 |
| GE | 0.101 | 0 | 0 | 0 | 0 |
| CHK | 0.052 | 0 | 0 | 0 | 0 |
| MSFT | 0.003 | 0.099 | 0.381 | 0.525 | 0 |
| QGEN | 0 | 0 | 0 | 0 | 0 |
| ATT | 0.140 | 0 | 0 | 0 | 0 |

Table 8.17: The last five of the portfolios solving the problem (7.6) - (7.2) computed by the KKT NCP function algorithm in subsection 8.2.1.

| $l_{max}$ Initial Solution | -0.014 | 0.008 | 0.031 | 0.053 | 0.075 |
|---|---|---|---|---|---|
| AAPL | 0.015 | 0.003 | 0.11 | 0.059 | 0.072 |
| AMD | 0.027 | 0 | 0 | 0 | 0 |
| BAC | 0 | 0 | 0 | 0 | 0 |
| F | 0.615 | 0.557 | 0.247 | 0.412 | 0.369 |
| FL | 0.047 | 0.073 | 0.132 | 0.127 | 0.128 |
| GE | 0.101 | 0.16 | 0 | 0 | 0 |
| CHK | 0.052 | 0 | 0 | 0 | 0 |
| MSFT | 0.003 | 0.126 | 0.511 | 0.402 | 0.431 |
| QGEN | 0 | 0.047 | 0 | 0 | 0 |
| ATT | 0.140 | 0.036 | 0 | 0 | 0 |

Table 8.18: The first five of the portfolios solving the problem (7.7) - (7.2) computed by the KKT MIP algorithm in subsection 8.2.2.

| $l_{max}$ Initial Solution | -0.124 | -0.102 | -0.08 | -0.058 | -0.036 |
|---|---|---|---|---|---|
| AAPL | 0.015 | 0.95 | 0.267 | 0.671 | 0.737 |
| AMD | 0.027 | 0 | 0 | 0 | 0 |
| BAC | 0 | 0 | 0 | 0 | 0 |
| F | 0.615 | 0 | 0 | 0 | 0 |
| FL | 0.047 | 0 | 0 | 0 | 0 |
| GE | 0.101 | 0 | 0 | 0 | 0 |
| CHK | 0.052 | 0 | 0 | 0 | 0 |
| MSFT | 0.003 | 0.05 | 0.733 | 0.329 | 0.263 |
| QGEN | 0 | 0 | 0 | 0 | 0 |
| ATT | 0.140 | 0 | 0 | 0 | 0 |

Table 8.19: The last five of the portfolios solving the problem (7.7) - (7.2) computed by the KKT MIP algorithm in subsection 8.2.2.

| $l_{max}$ Initial Solution | -0.014 | 0.008 | 0.031 | 0.053 | 0.075 |
|---|---|---|---|---|---|
| AAPL | 0 | 0.076 | 0.07 | 0.015 | 0.049 |
| AMD | 0 | 0 | 0 | 0.027 | 0 |
| BAC | 0 | 0 | 0 | 0 | 0 |
| F | 0.522 | 0.356 | 0.524 | 0.615 | 0.579 |
| FL | 0.079 | 0.129 | 0.114 | 0.047 | 0.134 |
| GE | 0.183 | 0 | 0 | 0.101 | 0 |
| CHK | 0 | 0 | 0 | 0.052 | 0 |
| MSFT | 0.165 | 0.439 | 0.293 | 0.003 | 0.238 |
| QGEN | 0.050 | 0 | 0 | 0 | 0 |
| ATT | 0 | 0 | 0 | 0.140 | 0 |

Table 8.20: The first five of the portfolios solving the problem (7.7) - (7.2) computed by the KKT regularization algorithm in subsection 8.2.2.

| $l_{max}$ Initial Solution | -0.124 | -0.102 | -0.08 | -0.058 | -0.036 |
|---|---|---|---|---|---|
| AAPL | 0.015 | 0.023 | 0.023 | 0.023 | 0.023 |
| AMD | 0.028 | 0.018 | 0.018 | 0.018 | 0.018 |
| BAC | 0 | 0 | 0 | 0 | 0 |
| F | 0.612 | 0.510 | 0.510 | 0.510 | 0.510 |
| FL | 0.047 | 0.077 | 0.077 | 0.077 | 0.077 |
| GE | 0.097 | 0.282 | 0.282 | 0.282 | 0.282 |
| CHK | 0.052 | 0.0006 | 0.0006 | 0.0006 | 0.0006 |
| MSFT | 0.003 | 0 | 0 | 0 | 0 |
| QGEN | 0 | 0.043 | 0.043 | 0.043 | 0.043 |
| ATT | 0.144 | 0.048 | 0.048 | 0.048 | 0.048 |

Table 8.21: The last five of the portfolios solving the problem (7.7) - (7.2) computed by the KKT regularization algorithm in subsection 8.2.2.

| $l_{max}$ Initial Solution | -0.014 | 0.008 | 0.031 | 0.053 | 0.075 |
|---|---|---|---|---|---|
| AAPL | 0.023 | 0.023 | 0.023 | 0.023 | 0.023 |
| AMD | 0.018 | 0.018 | 0.018 | 0.018 | 0.018 |
| BAC | 0 | 0 | 0 | 0 | 0 |
| F | 0.510 | 0.510 | 0.510 | 0.510 | 0.510 |
| FL | 0.077 | 0.077 | 0.077 | 0.077 | 0.077 |
| GE | 0.282 | 0.282 | 0.282 | 0.282 | 0.282 |
| CHK | 0.0006 | 0.0006 | 0.0006 | 0.0006 | 0.0006 |
| MSFT | 0 | 0 | 0 | 0 | 0 |
| QGEN | 0.043 | 0.043 | 0.043 | 0.043 | 0.043 |
| ATT | 0.048 | 0.048 | 0.048 | 0.048 | 0.048 |

Table 8.22: The first five of the portfolios solving the problem (7.7) - (7.2) computed by the KKT NCP function algorithm in subsection 8.2.2.

| $l_{max}$ Initial Solution | -0.124 | -0.102 | -0.08 | -0.058 | -0.036 |
|---|---|---|---|---|---|
| AAPL | 0.015 | 0.950 | 0.809 | 0.015 | 1 |
| AMD | 0.027 | 0 | 0 | 0.027 | 0 |
| BAC | 0 | 0 | 0 | 0 | 0 |
| F | 0.615 | 0 | 0 | 0.615 | 0 |
| FL | 0.047 | 0 | 0 | 0.047 | 0 |
| GE | 0.101 | 0 | 0 | 0.101 | 0 |
| CHK | 0.052 | 0 | 0 | 0.052 | 0 |
| MSFT | 0.003 | 0.050 | 0.191 | 0.003 | 0 |
| QGEN | 0 | 0 | 0 | 0 | 0 |
| ATT | 0.140 | 0 | 0 | 0.140 | 0 |

Table 8.23: The last five of the portfolios solving the problem (7.7) - (7.2) computed by the KKT NCP function algorithm in subsection 8.2.2.

| $l_{max}$ Initial Solution | -0.014 | 0.008 | 0.031 | 0.053 | 0.075 |
|---|---|---|---|---|---|
| AAPL | 0.015 | 0.003 | 0.015 | 0.015 | 0.049 |
| AMD | 0.027 | 0 | 0.027 | 0.027 | 0 |
| BAC | 0 | 0 | 0 | 0 | 0 |
| F | 0.615 | 0.557 | 0.615 | 0.615 | 0.579 |
| FL | 0.047 | 0.073 | 0.047 | 0.047 | 0.134 |
| GE | 0.101 | 0.16 | 0.101 | 0.101 | 0 |
| CHK | 0.052 | 0 | 0.052 | 0.052 | 0 |
| MSFT | 0.003 | 0.126 | 0.003 | 0.003 | 0.238 |
| QGEN | 0 | 0.047 | 0 | 0 | 0 |
| ATT | 0.140 | 0.036 | 0.140 | 0.140 | 0 |

Table 8.24: The first five of the portfolios solving the problem (7.6) - (7.2) computed by the OVFRA algorithm in subsection 8.3.1.

| $l_{max}$ Initial Solution | -0.124 | -0.102 | -0.08 | -0.058 | -0.036 |
|---|---|---|---|---|---|
| AAPL | 0.015 | 1.000 | 0.671 | 0.671 | 0.458 |
| AMD | 0.027 | 0 | 0 | 0 | 0 |
| BAC | 0 | 0 | 0 | 0 | 0 |
| F | 0.615 | 0 | 0 | 0 | 0 |
| FL | 0.047 | 0 | 0 | 0 | 0 |
| GE | 0.101 | 0 | 0 | 0 | 0 |
| CHK | 0.052 | 0 | 0 | 0 | 0 |
| MSFT | 0.003 | 0 | 0.329 | 0.329 | 0.542 |
| QGEN | 0 | 0 | 0 | 0 | 0 |
| ATT | 0.140 | 0 | 0 | 0 | 0 |

Table 8.25: The last five of the portfolios solving the problem (7.6) - (7.2) computed by the OVFRA algorithm in subsection 8.3.1.

| $l_{max}$ Initial Solution | -0.014 | 0.008 | 0.031 | 0.053 | 0.075 |
|---|---|---|---|---|---|
| AAPL | 0.131 | 0.131 | 0.110 | 0.075 | 0.072 |
| AMD | 0 | 0 | 0 | 0 | 0 |
| BAC | 0 | 0 | 0 | 0 | 0 |
| F | 0.180 | 0.180 | 0.247 | 0.362 | 0.369 |
| FL | 0.135 | 0.135 | 0.132 | 0.129 | 0.128 |
| GE | 0 | 0 | 0 | 0 | 0 |
| CHK | 0 | 0 | 0 | 0 | 0 |
| MSFT | 0.555 | 0.555 | 0.511 | 0.435 | 0.431 |
| QGEN | 0 | 0 | 0 | 0 | 0 |
| ATT | 0 | 0 | 0 | 0 | 0 |

Table 8.26: The first five of the portfolios solving the problem (7.6) - (7.2) computed by the modified OVFRA algorithm in subsection 8.3.2.

| $l_{max}$ Initial Solution | -0.124 | -0.102 | -0.08 | -0.058 | -0.036 |
|---|---|---|---|---|---|
| AAPL | 0.015 | 1.000 | 0.671 | 0.671 | 0.671 |
| AMD | 0.027 | 0 | 0 | 0 | 0 |
| BAC | 0 | 0 | 0 | 0 | 0 |
| F | 0.615 | 0 | 0 | 0 | 0 |
| FL | 0.047 | 0 | 0 | 0 | 0 |
| GE | 0.101 | 0 | 0 | 0 | 0 |
| CHK | 0.052 | 0 | 0 | 0 | 0 |
| MSFT | 0.003 | 0 | 0.329 | 0.329 | 0.329 |
| QGEN | 0 | 0 | 0 | 0 | 0 |
| ATT | 0.140 | 0 | 0 | 0 | 0 |

Table 8.27: The last five of the portfolios solving the problem (7.6) - (7.2) computed by the modified OVFRA algorithm in subsection 8.3.2.

| $l_{max}$ Initial Solution | -0.014 | 0.008 | 0.031 | 0.053 | 0.075 |
|---|---|---|---|---|---|
| AAPL | 0.131 | 0.131 | 0.110 | 0.075 | 0.072 |
| AMD | 0 | 0 | 0 | 0 | 0 |
| BAC | 0 | 0 | 0 | 0 | 0 |
| F | 0.180 | 0.180 | 0.247 | 0.362 | 0.369 |
| FL | 0.135 | 0.135 | 0.132 | 0.129 | 0.128 |
| GE | 0 | 0 | 0 | 0 | 0 |
| CHK | 0 | 0 | 0 | 0 | 0 |
| MSFT | 0.555 | 0.555 | 0.511 | 0.435 | 0.431 |
| QGEN | 0 | 0 | 0 | 0 | 0 |
| ATT | 0 | 0 | 0 | 0 | 0 |