12-2021

An Algorithm for Biobjective Mixed Integer Quadratic Programs

Pubudu Jayasekara Merenchige
pwijesi@clemson.edu

12-2021

An Algorithm for Biobjective Mixed Integer Quadratic Programs

Pubudu Jayasekara Merenchige

An Algorithm for Biobjective Mixed Integer Quadratic Programs

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Mathematical Sciences

by
Pubudu L. W. Jayasekara
December 2021

Accepted by:
Dr. Margaret M. Wiecek, Committee Chair
Dr. Pietro Belotti
Dr. Yuyuan Ouyang
Dr. Matthew Saltzman
Dr. Boshi Yang

Abstract

Multiobjective quadratic programs (MOQPs) are appealing since convex quadratic programs have elegant mathematical properties and model important applications. Adding mixed-integer variables extends their applicability while the resulting programs become global optimization problems. Thus, in this work, we develop a branch and bound (BB) algorithm for solving biobjective mixed-integer quadratic programs (BOMIQPs). An algorithm of this type does not exist in the literature.

The algorithm relies on five fundamental components of the BB scheme: calculating an initial set of efficient solutions with associated Pareto points, solving node problems, fathoming, branching, and set dominance. Considering the properties of the Pareto set of BOMIQPs, two new fathoming rules are proposed. An extended branching module is suggested to cooperate with the node problem solver. A procedure to make the dominance decision between two Pareto sets with limited information is proposed. This set dominance procedure can eliminate the dominated points and eventually produce the Pareto set of the BOMIQP. Numerical examples are provided.

Solving multiobjective quadratic programs (MOQPs) is fundamental to our research. Therefore, we examine the algorithms for this class of problems with different perspectives. The scalarization techniques for (strictly) convex MOPs are reviewed and the available algorithms for computing efficient solutions for MOQPs are discussed. These algorithms are compared with respect to four properties of MOQPs. In addition, methods for solving parametric multiobjective quadratic programs are studied. Computational studies are provided with synthetic instances, and examples in statistics and portfolio optimization. The real-life context reveals the interplay between the scalarizations and provides an additional insight into the obtained parametric solution sets.

Dedication

To my beloved family.

Acknowledgments

First and foremost I would like to thank my Ph.D. advisor, Dr. Margaret M. Wiecek for her constant support, guidance, encouragement, and patience during my Ph.D. study. Her immense knowledge and experience have encouraged me in all the time of my academic and personal life. She has inspired me and has helped shape me into who I am today.

I would like to thank my committee members Dr. Pietro Belotti, Dr. Yuyuan Ouyang, Dr. Matthew Saltzman, and Dr. Boshi Yang for their great help during my Ph.D. research. Their suggestions made my research so much richer, and my dissertation something I can be proud of having written. I also thank Dr. Akshay Gupte and Dr. Georges Fadel for their time commitment for my comprehensive exam.

My sincere thanks go to Dr. Nathan Adelgren for his support with the coding and the implementation. Without his precious support, it would not be possible to conduct this research. I would also like to thank Andrew Pangia for his work and willingness to help as a co-author of my paper.

I would like to express my sincere gratitude to the Office of Naval Research for funding my research through grant number N00014-16-1-2725. Also, I thank the

School of Mathematical and Statistical Sciences for providing me financial support during my stay at Clemson.

Last but not the least, I would like to thank my family. To my parents Mala Senevirathne and Sunil Wijesiri, for believing in me and supporting me in this endeavor. To my beloved wife Shalika Marasinghe, for her tremendous support along this long path sacrificing her own career to encourage and support me. To my loving son Binula Jayasekara, for brightening my world. To my brother Malindu Wijesiri, for his continuous great patience and encouragement. Without all your support, this hard work would not have been possible.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Multiobjective programming (MOP) is a branch of mathematical optimization involving several conflicting objective functions. Unlike the single objective programs (SOPs) that typically have a unique optimal solution in objective space, multiobjective programs have a solution set from which decision makers select the best solution based on their preference and experience. MOPs occur frequently in many application areas including finance [106, 134, 115, 68], economics [83, 9], engineering [11, 112], statistics [1, 122], nanotechnology [129] and others. Many of those problems require all or a subset of decision variables to be described using discrete quantities, meaning there are two subclasses of problems with discrete variables: (i) multiobjective integer programs (MOIPs), in which all decision variables take on integer values, and (ii) multiobjective mixed integer programs (MOMIPs), in which some of the decision variables take on integer values.

In our work, we primarily focus on biobjective mixed integer quadratic problems (BOMIQPs) with linear constraints. There are several applications for this class of problems such as portfolio optimization [111] and electronics [121]. Our goal is to develop an algorithm for computing the set of Pareto efficient solutions to the BOMIQP. To achieve this goal, we propose an approach based on a branch and bound (BB) algorithm. According to our knowledge this is the first ever method to solve this class of problems. To aid in the description of BOMIQP we first introduce properties of multiobjective programs with linear constraints.

1.2 Basic Concepts and Notations

In this section, we introduce definitions and notations explaining the basic concepts of MOPs before we introduce BOMIQPs. However, these concepts are also valid for BOMIQPs. The reader may refer to Ehrgott [42] and Steuer [116] for a detailed description of MOPs.

Let $\mathbb{R}^n$, $\mathbb{R}^r$, $\mathbb{R}^p$ be Euclidean vector spaces such that $n, r, p \in \mathbb{N}$, $\mathbb{Z}$ be the set of integers and $\mathbb{Q}$ be the set of rational numbers. For any two vectors $\mathbf{y^1}, \mathbf{y^2} \in \mathbb{R}^r$, we use the notation $\mathbf{y^1} < \mathbf{y^2}$ if and only if $y_i^1 < y_i^2$ for all $i = 1, \cdots, r$, $\mathbf{y^1} \leqq \mathbf{y^2}$ if and only if $y_i^1 \leq y_i^2$ for all $i = 1, \cdots, r$, and $\mathbf{y^1} \leq \mathbf{y^2}$ if and only if $y_i^1 \leq y_i^2$ for all $i = 1, \cdots, r$ and $\mathbf{y^1} \neq \mathbf{y^2}$. Further, we define $\mathbb{R}^r_{\geqq} = \{\mathbf{y} \in \mathbb{R}^r : \mathbf{y} \geqq \mathbf{0}\}$ and the sets $\mathbb{R}^r_{\geq}, \mathbb{R}^r_{>}$ are defined accordingly. We also define $\mathbb{R}^2_{\gtreqqless} = \{\mathbf{y} \in \mathbb{R}^2 : y_1 \geq 0, y_2 \leq 0\}$, $\mathbb{R}^2_{\gtreqless} = \mathbb{R}^2_{\gtreqqless} \setminus \{\mathbf{0}\}$ and $\mathbb{R}^2_{\gtrless} = \{\mathbf{y} \in \mathbb{R}^2 : y_1 > 0, y_2 < 0\}$. For any, subset $S \subseteq \mathbb{R}^2$, we define $S_{\geqq} = S + \mathbb{R}^2_{\geqq}$. The sets $S_{\geq}$ and $S_{>}$ are defined accordingly. In addition, $conv(S), int(S)$, $bd(S)$ and $|S|$ denote the convex hull, interior, boundary and the cardinality of set S, respectively.

Consider the following multiobjective program (MOP):

$$\begin{aligned} \min \quad & \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_r(\mathbf{x})] \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{X}, \end{aligned} \tag{1.1}$$

where $f_i : \mathbb{R}^n \longrightarrow \mathbb{R}$, $i = 1, \cdots, r$ and $\mathcal{X} \subseteq \mathbb{R}^n$ is a feasible set. Throughout this dissertation we assume $f_i(\mathbf{x})$ is twice continuously differentiable and the MOP is (strictly) convex, that is, all objective functions $f_i(\mathbf{x})$, $i = 1, \cdots, r$ are (strictly) convex and unless specified otherwise the feasible set $\mathcal{X}$ is nonempty, convex, and compact. The spaces $\mathbb{R}^n$ and $\mathbb{R}^r$ serve as the solution (decision) space and the objective (performance) space respectively. The attainable set in the objective space for the MOP is defined as the image of the feasible set $\mathcal{X}$ under the vector-valued objective function mapping $\mathbf{f}$.

$$\mathcal{Y} := \mathbf{f}(\mathcal{X}) = \{\mathbf{y} \in \mathbb{R}^r : \mathbf{y} = \mathbf{f}(\mathbf{x}), \mathbf{x} \in \mathcal{X}\}.$$

Definition 1.2.1. *Let $\boldsymbol{y}^1, \boldsymbol{y}^2 \in \mathcal{Y}$. If $\boldsymbol{y^1}(\leqq) \leq \boldsymbol{y^2}$, then $\boldsymbol{y^1}$ is said to (weakly) dominate $\boldsymbol{y}^2$.*

Definition 1.2.2. *Let $\boldsymbol{y^1} \in \mathcal{Y}$. We say that $\boldsymbol{y^1}$ is Pareto to MOP (1.1) if there does not exist $\boldsymbol{y^2} \in \mathcal{Y}$ such that $\boldsymbol{y^2} \leq \boldsymbol{y^1}$. The set of all Pareto points is denoted by $\mathcal{Y}_P$ and is called the Pareto set.*

Solving MOP (1.1) is understood as finding its efficient solutions.

Definition 1.2.3. *Let $\boldsymbol{x}^i \in \mathcal{X}$ and $\boldsymbol{y}^i = \boldsymbol{f}(\boldsymbol{x}^i)$ for $i = 1, 2$. Then $\boldsymbol{x}^1$ is called a (weakly) efficient solution to MOP* (1.1) *if there does not exist $\boldsymbol{x}^2$ such that $\boldsymbol{y}^2(<) \leq \boldsymbol{y}^1$. The set of (weakly) efficient solutions is denoted by $(\mathcal{X}_{wE})\mathcal{X}_E$.*

The solution to MOP (1.1) determines $(\mathcal{X}_{wE})$ $\mathcal{X}_E$ along with $\mathcal{Y}_P$.

The following proposition is one of the main results of our work. It specifies the properties of the (weakly) efficient set and the attainable set when the MOP is strictly convex.

Proposition 1.2.4. *Let MOP (1.1) be strictly convex. Then the following holds:*

1. $\mathcal{X}_E = \mathcal{X}_{wE}$;

2. *The set* $\mathcal{Y} + \mathbb{R}^r_>$ *is strictly convex.*

Proof. 1. Refer to [24].

2. We show that $\mathcal{Y} + \mathbb{R}^r_>$ is strictly convex, that is, for $\mathbf{z}_1, \mathbf{z}_2 \in \mathcal{Y} + \mathbb{R}^r_>$ such that $\mathbf{z}_1 \neq \mathbf{z}_2, \mathbf{z} = \lambda \mathbf{z}_1 + (1-\lambda)\mathbf{z}_2 \in \operatorname{int}(\mathcal{Y} + \mathbb{R}^r_>)$ for $\lambda \in (0,1)$.

   Let $\mathbf{z}_1, \mathbf{z}_2 \in \mathcal{Y} + \mathbb{R}^r_>$, such that $\mathbf{z}_1 \neq \mathbf{z}_2$ where $\mathbf{z}_k = \mathbf{y}_k + \mathbf{d}_k$, $\mathbf{y}_k \in \mathcal{Y}$ and $\mathbf{d}_k \in \mathbb{R}^r_>$, $\mathbf{y}_k = \mathbf{f}(\mathbf{x}_k)$ for some $\mathbf{x}_k \in \mathcal{X}$ and for $k = 1, 2$. For $\lambda \in (0,1)$ calculate

$$
\begin{aligned}
\mathbf{z} &= \lambda(\mathbf{y}_1 + \mathbf{d}_1) + (1-\lambda)(\mathbf{y}_2 + \mathbf{d}_2) \\
&= \lambda \mathbf{y}_1 + (1-\lambda)\mathbf{y}_2 + \lambda \mathbf{d}_1 + (1-\lambda)\mathbf{d}_2 \\
&= \lambda \mathbf{f}(\mathbf{x}_1) + (1-\lambda)\mathbf{f}(\mathbf{x}_2) + \lambda \mathbf{d}_1 + (1-\lambda)\mathbf{d}_2 \\
&> \mathbf{f}(\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) + \lambda \mathbf{d}_1 + (1-\lambda)\mathbf{d}_2
\end{aligned}
$$

   where the last inequality results from the fact that $\mathbf{f}$ is composed of strictly convex real-valued functions. Since $\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2 \in \mathcal{X}$, $\mathbf{f}(\lambda \mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) = \bar{\mathbf{y}} \in \mathcal{Y}$, and $\bar{\mathbf{d}} = \lambda \mathbf{d}_1 + (1-\lambda)\mathbf{d}_2 \in \mathbb{R}^r_>$, we obtain

$$
\mathbf{z} > \bar{\mathbf{y}} + \bar{\mathbf{d}} \qquad \text{where } \bar{\mathbf{y}} \in \mathcal{Y}, \;\; \bar{\mathbf{d}} \in \mathbb{R}^r_>,
$$

which can be converted to an equality

$$\mathbf{z} = \bar{\mathbf{y}} + \bar{\mathbf{d}} + \Delta\mathbf{z} \qquad \text{where } \Delta\mathbf{z} \in \mathbb{R}^r_>.$$

We obtain

$$\mathbf{z} = \bar{\mathbf{y}} + \bar{\bar{\mathbf{d}}}, \text{ where } \bar{\bar{\mathbf{d}}} = \bar{\mathbf{d}} + \Delta\mathbf{z} \in \mathbb{R}^r_>, \text{ or equaivalently, } \mathbf{z} \in \mathcal{Y} + \mathbb{R}^r_>.$$

Since $\mathcal{Y} + \mathbb{R}^r_>$ is an open set, $\mathbf{z} \in \text{int } (\mathcal{Y} + \mathbb{R}^r_>)$. Hence, $\mathcal{Y} + \mathbb{R}^r_>$ is strictly convex.

□

Our main focus in this research is to solve BOMIQPs. Therefore, first we consider a broad area, the multiobjective discrete optimization problems. As mentioned earlier, there are two different types of discrete optimization problems: either all decision variables are discrete ($\mathbf{x} \in \mathbb{Z}^n$), or only a subset of decision variables is discrete while the other subset is continuous ($\mathbf{x} \in \mathbb{R}^p \times \mathbb{Z}^{n-p}$).

1.3 A Review of Branch and Bound Algorithms for Solving Multiobjective Discrete Optimization Problems

In this section, we summarize earlier work on multiobjective pure or mixed integer programs using a variation of the BB technique. Some of these algorithms were developed to solve a specific optimization problem, for example, the biobjective spanning tree problem, while some of this research addressed a certain class of optimization problems such as mixed binary linear programs and mixed integer linear programs.

1.3.1 Multiobjective Mixed Binary Linear Programs (MOMBLPs)

Early studies on multiobjective discrete linear programming began 30 years ago when Bitran and Rivera [18] and Kızıltan and Yucaoglu [71] developed BB algorithms specifically for the multiobjective binary case. A decade later, Mavrotas and Diakoulaki [86] introduced a BB algorithm for the generation of the efficient set for a MOMBLP using a node fathoming procedure. However, their method was not capable of obtaining the complete set of Pareto points. They obtained only a subset of Pareto points throughout the solution process. Vincent et al. [125] improved this method by proposing a new algorithm to obtain the complete Pareto set. In addition, this new algorithm introduced refinements such as an improved fathoming rule, a branching strategy, and bound sets. Stidsen et al. [118] proposed a new BB method for solving a biobjective mixed binary problem with one of the two objectives having binary variables only. This proposed method is able to find the complete set of Pareto points.

Based on these general purpose mixed binary integer BB algorithms, later researchers focused on solving a specific type of problem. For example, Sourd and Spanjaard [114] proposed a set of fathoming rules and developed a BB procedure for solving a biobjective spanning tree problem. A key aspect of this work is that the bounding is performed using a set of points rather than a single ideal point. In addition, Delort and Spanjaard [37] proposed a BB technique for solving the biobjective mixed binary knapsack problem.

1.3.2 Biobjective Mixed Integer Linear Programs (BOMILPs)

Two techniques for solving general BOMILPs have been developed. The Triangle Splitting method suggested by Boland et al. [19] is an iterative method in which the objective space is partitioned into smaller rectangular or triangular search regions, and then various scalarization techniques for solving biobjective linear programs are employed to obtain the Pareto points.

The second is a variation of the BB technique. Adelgren [5] presented a generic BB method for finding all efficient solutions for BOMILPs along with new algorithms for obtaining dual bounds at a node, checking node fathoming, presolving the problem and obtaining duality gap measurements. In that work, the reader can find a concise review of methods to solve BOMILPs. In addition, Adelgren [4] used a new data structure to store and dynamically update the primal bound set at each node throughout the BB process and to compare each element of the dual bound set with the primal bound set. The author suggests that this procedure is more efficient than comparing each element in the primal bound set with the dual bound set since, in practice, for a particular node of the BB tree, the primal bound set typically contains far more points and segments than the dual bound set.

Belotti et al. [14] introduced two new practical fathoming rules that can be used to solve BOMILPs. These fathoming rules are implemented by solving an auxiliary LP to determine whether the upper bound and the lower bound can be separated by a hyperplane.

Przybylski and Gandibleux [99] reviewed BB algorithms for solving multiobjective linear optimization problems with binary variables and mixed binary variables. Parragh and Tricoirea [93] proposed a new branch-and-bound algorithm to solve BOMIQPs with new branching rule. The developed algorithm is applied to biobjective facility location problems, biobjective set covering problem, and the biobjective team orienteering problem. Perini et al. [96] proposed a new criterion space search method called the boxed line method for solving the BOMILPs. The provided computational results demonstrate the relative strengths and weaknesses of this method.

The reader may refer to [5] for a concise review of these methods. Also, Dächer and Klamroth [33], Alves and Costa [6], Boland et al. [20], Rasmi and Türkay [103], Forget et al. [54], and Rasmiet al. [104] studied the mixed/pure integer programs with more than two objectives.

1.3.3 Multiobjective Mixed Integer Nonlinear Programs (MOMINLPs)

Martin et al. [85] discussed the use of constraint propagation in interval BB for biobjective mixed integer nonlinear optimization problems and proposed a generic BB algorithm. In addition to this algorithm, they introduced a mechanism that exploits an upper bound set using dominance relations.

Cacchiani and D'Ambrosio [26] proposed a heuristic BB algorithm enhanced with a refinement procedure to derive an approximated set of Pareto points for convex MOMINLPs. The proposed algorithm obtains an initial set of feasible points and its image set in the objective space by solving ϵ-constraint problems. Lower bounds at each node in the BB tree are considered as the ideal point of the relaxed problem. A simple fathoming rule is implemented where a node can be fathomed if its lower bound vector is dominated by the points currently identified as nondominated. To obtain an improved nondominated set, a weighted sum problem is solved at each leaf node and the current set of nondominated points is updated by applying a pairwise comparison. In addition, a refinement procedure is suggested for discarding the dominated points. The idea behind this method can be specialized for solving BOMIQPs. Since the authors solve an SOP each time, a general purpose quadratic program (QP) solver can be used for solving this QP. In addition, the proposed fathoming rule is easy to implement and no assumptions on the number of objective functions or on the type of the variables are made. However, this work suggests that for solving a large number of SOPs, the time complexity could be high and the exact solution may not be obtained.

De Santis et al. [35] presented a branch and bound algorithm to solve MOMINLPs. They use both convex relaxation and the linear approximations of the objective space to compute the efficient solutions and the Pareto set. This method is the first non scalarization based deterministic algorithm devised to handle this class of problems.

Diessel [39] proposed a new algorithm for approximating the Pareto set of biobjective mixed integer problems with convex constraints. Computational results are provided with convex biobjective quadratic programs and convex biobjective linear/ nonlinear programs.

Burachik et al. [25] developed algorithms to solve MOMINLPs based on methods which were originally designed to solve continuous problems. They illustrated how to use different scalarization techniques to find or approximate the Pareto sets of this class problems. The proposed algorithms are tested with instances up to four objective functions.

We also note that multiobjective mixed integer quadratic programs with one quadratic function were studied by Dua et al. [41], Romanko [106] and Sawik [111], while Buchheim [23] studied a biobjective mixed integer portfolio optimization problem. De Santis and Eichfelder [34] presented a branch and bound algorithm for MOQPs with strictly convex quadratic objective functions over integer variables. However, this algorithm is not capable of finding the complete Pareto set but only finds a finite set of Pareto points.

1.4 Research Needs

Currently, multiobjective mixed integer quadratic programming models have seen widespread use, particularly fields such as finance, engineering, economics, business, and management. Developing a tool to solve such problems is in demand. Although some studies have been conducted on MOMILPs, research on the nonlinear programs with integer variables is limited. Based on this limited research, we propose to develop a BB algorithm for solving BOMIQPs. To the best of our knowledge, this study is the first attempt at proposing a general purpose BB algorithm for this class of problems.

Consider the BOMIQP in the form:

$$
\begin{aligned}
\mathcal{P}: \quad \min \quad & \mathbf{f}(\mathbf{x}) = \left[f_1(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T Q_1 \mathbf{x} + \mathbf{p}_1^T \mathbf{x}, \;\; f_2(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T Q_2 \mathbf{x} + \mathbf{p}_2^T \mathbf{x}\right] \\
\text{s.t.} \quad & \mathbf{x} \in \mathcal{X} = \{\mathbf{x} \in \mathbb{R}^p \times \mathbb{Z}^{n-p} : A\mathbf{x} \leqq \mathbf{b}, \mathbf{x} \geqq \mathbf{0}\},
\end{aligned}
\tag{1.2}
$$

where $n > 1, 0 \leq p < n, A \in \mathbb{Q}^{m \times n}, \mathbf{b} \in \mathbb{Q}^m, Q_1, Q_2 \in \mathbb{Q}^{n \times n}$ and $\mathbf{p}_1, \mathbf{p}_2 \in \mathbb{Q}^n$. Each quadratic function contains the quadratic term $\mathbf{x}^T Q_i \mathbf{x}$ with a symmetric positive definite matrix Q_i, $i = 1, 2$ and the linear term $\mathbf{p}_i^T \mathbf{x}$, $i = 1, 2$.

In addition, the problem (1.2) contains discrete and continuous variables and has a convex feasible set when the integrality of the variables is relaxed, and two objectives are strictly convex quadratic functions. These features make the determination of the Pareto points to BOMIQP (1.2) very challenging. The main contribution of this work is the design and development of a BB algorithm, which allows deriving the set of Pareto points for BOMIQP (1.2). In general, this Pareto set is the union of single points and (open/close) convex curves in the objective space. In addition, this set may not be connected, which brings even more challenges to the task of developing a technique for solving BOMIQPs. Refer to the examples in Section B in Chapter 2 and Appendix B to get an idea of the structure of the Pareto set. A BB algorithm for solving BOMIQPs should have the following five components:

1. computing an initial set of efficient solutions and Pareto outcomes
2. solving the node problem
3. bounding and fathoming rules
4. branching
5. set dominance

We discuss these components in the context of BOMIQPs and the work we propose. The BB algorithm starts with an initial set of Pareto points that evolves toward the Pareto set of the BOMIQP. These initial points are computed by solving the weighted sum problem for fixed weights. This weighted sum problem is a mixed integer quadratic program (MIQP) and a commercial package, GUROBI is used to solve the program.

Within the BB algorithm, the initial Pareto set (being a subset of the true Pareto set of the BOMIQP) evolves towards the true Pareto set. This evolving set is referred to as the incumbent set (working set). The evolution of the incumbent set involves adding new nondominated points and discarding dominated points. The new points added to the incumbent set are obtained from solving the biobjective quadratic programs (BOQPs) associated with the nodes of BB tree.

At each node of the BB tree, we solve a BOQP, a relaxed BOMIQP, to obtain the Pareto set. This set is used to construct the lower bound set for the algorithm. Either the mpLCP method [3] or the spLCP method [124, 69] can be used to solve the BOQP. These methods compute the complete Pareto set of the node problem under some assumptions. However, the spLCP method is not used in our work since the assumptions are hard to satisfy and it is computationally expensive relative to the mpLCP method [69]. Note that the node problem solver plays a key role in our BB algorithm. There are several other methods available to solve BOQPs. However, the mpLCP method emerges as the best approach to solve BOQPs in our work. The efficient solution and the Pareto set are available as a function of the weight parameter λ.

In the single objective case, the objective function value associated with the fractional solution to the node problem provides a lower bound while the objective function value of any integer feasible solution is an upper bound for every node problem of the BB tree. In the biobjective case, these bounds are no longer singletons in $\mathbb{R}$. Instead, they are subsets of $\mathbb{R}^2$. At each node, the lower bound set is obtained by adding the set $\mathbb{R}^2_{\geqq}$ to the Pareto set of the node problem while the upper bound set is obtained by adding the set $\mathbb{R}^2_{\geqq}$ to the set of points and curves implied by the Pareto points to the problem $\mathcal{P}$ in the current nondominated set.

The next important component of the BB algorithm is fathoming rules. A good set of fathoming rules make the BB algorithm effective. If the node problem is infeasible or the solution to the node problem is feasible to the problem $\mathcal{P}$, then the node can be fathomed immediately. If a node cannot be fathomed due to the above reasons, then the fathoming rules based on the previously defined bound sets become relevant. Two fathoming rules are implemented in a new way to align with the properties of the Pareto set of the BOMIQP. If a node cannot be fathomed or the node is fathomed by integer feasibility, either the complete Pareto set or a subset of the Pareto set of the node problem should be stored in the evolving nondominated set of the BOMIQP (1.2). A detailed description of the fathoming rules and bound sets is given in Chapter 2.

The idea of the branching operation in our BB algorithm is similar to the idea of the branching in the single objective case. If a node of the BB tree cannot be fathomed, we need to branch on an integer variable that has a fractional value. If a variable x_l takes a fractional value ϕ_l in an optimal solution to the node problem, a branching process is performed. Within branching operation, two new node problems are created by adding the constraints $x_l \geq \lceil \phi_l \rceil$ and $x_l \leq \lfloor \phi_l \rfloor$, respectively, to the current node problem. This regular branching idea is extended to be applicable to BOMIQPs and is integrated with the mpLCP method. Because the solution with mpLCP is available as a function of the weight parameter λ. Therefore, we first need to find the range of the values that the fractional integer variable can take. Then we create child node problems adding the additional constraint. Reader may refer to Chapter 2 for more details.

A method to discard the dominated points from the incumbent set is required in our BB algorithm. We introduce a set dominance procedure to fulfill this task. The Pareto sets or subsets of the Pareto sets to relaxed node problems are tested in the set dominance procedure with respect to the elements in the current incumbent set. In every iteration of the algorithm, the incumbent set is updated using this procedure. Our procedure can be easily extended to the MOPs producing strictly convex Pareto sets. According to our knowledge, there is very little research in this area.

1.5 Completed Research Objectives

Our primary research objective is to develop a BB algorithm for BOMIQPs by combining the five necessary components presented in Section 1.4.

All the theoretical tasks addressing the components 1,3,4, and 5 have been accomplished and are presented in Chapter 2. The node problem solver, component 2 has been introduced and implemented by Adelgren [3, 2]. This solver is integrated with the proposed BB algorithm. Also, we have proved that our algorithm finds the complete Pareto set and efficient solutions. The algorithm is implemented in *MATLAB* and tested for randomly generated BOMIQP instances.

Finding a good node problem solver to solve BOQPs in the BB algorithm is a major challenge. In [68], we have reviewed algorithms for computing efficient solutions to multiobjective quadratic programs (MOQPs) and concluded that the mpLCP method [3] is currently the best method for computing exact solutions to convex MOQPs. In addition, we reviewed some well-established scalarizations in MOPs from the perspective of parametric optimization. A new scalarization for MOPs, modified hybrid scalarization is proposed. This is useful for a class of specially structured convex MOPs when the objective functions of the MOP come in two groups with different real-life scenarios. Then we used the mpLCP method and the newly introduced modified hybrid scalarization to solve multiobjective portfolio optimization problems with three or more quadratic objective functions, a class of problems that has not been solved before. Computational examples are provided to illustrate the new scalarization. The work is included in Chapter 3.

The spLCP method [124] is proposed to solve BOQPs as an alternative to the mpLCP method due to its implementation limitations. This method was proposed in 1993 however, to the best of our knowledge, was never implemented. Note that the spLCP method can solve linear complementarity problems with one parameter. Therefore we can only use this method to solve BOQPs. We compared mpLCP and spLCP methods in the paper [69]. We anticipated that the spLCP method would be more efficient than the mpLCP method, due to its special assumptions but discovered otherwise. Also, the mpLCP method turned out to be superior to the spLCP method in the current MATLAB environment. Consequently, we conclude that the mpLCP method determines the state-of-the-art in solving BOQPs with linear constraints and hence is used as the node problem solver in the proposed BB algorithm.

A Pareto set dominance module is proposed in Chapter 2 as a major component of the BB algorithm. However, this module can be generalized to make the set dominance decision with two strictly convex Pareto sets. In this module, we analyze the mutual location of two Pareto sets in $\mathbb{R}^2$ to conclude about the (partial) dominance between them, assuming each Pareto set is a strictly convex curve. We use the parametric description of the efficient set and the end points of the Pareto sets in this work. The set dominance module is implemented in *MATLAB*.

A preliminary numerical experiment is performed for the proposed BB algorithm. In this rudimentary implementation ten instances of BOMIQPs are tested and the results are discussed in Chapter 2.

The spLCP and mpLCp methods are broadly studied in our work. These methods are based on the linear complementarity problem (LCP). Therefore, we discuss the relationship between QPs and LCPs briefly in the following section.

1.6 Linear Complementarity Problem Formulation of the Quadratic Program

The linear complementarity problem (LCP) approach was conceived as a unifying formulation for quadratic programs (QPs). Several effective algorithms for solving QPs are based on the LCP formulation. For more details about these algorithms, the reader may refer to [31]. In this section, we discuss how to obtain the associated LCP by applying the Karush-Kuhn-Tucker (KKT) optimality conditions to the QP. We also give some preliminary information on the LCP.

Consider a single objective QP in the form:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{p}^T\mathbf{x} \\ \text{s.t.} \quad & A\mathbf{x} \leqq \mathbf{b} \\ & \mathbf{x} \geqq \mathbf{0} \end{aligned} \tag{1.3}$$

where $\mathbf{x} \in \mathbb{R}^n$, $Q \in \mathbb{R}^{n\times n}$, $\mathbf{p} \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$.

Assumption 1.6.1. *Matrix Q is a symmetric positive definite (PD) matrix. Therefore, problem (1.3) is strictly convex.*

To obtain the LCP, the inequality constraints in (1.3) are first converted into equality constraints $A\mathbf{x} + \mathbf{s} = \mathbf{b}$ by adding a slack variable $\mathbf{s} \geqq \mathbf{0}$. Let $\mathbf{u}$ and $\mathbf{r}$ be the Lagrange multipliers (dual variables) associated with the constraints $\mathbf{s} \geqq \mathbf{0}$ and $\mathbf{x} \geqq \mathbf{0}$, respectively. The KKT necessary conditions for optimality of problem (1.3) are

$$\begin{aligned} \mathbf{r} \quad - Q\mathbf{x} - A^T\mathbf{u} \quad &= \mathbf{p} \\ \mathbf{s} \quad + A\mathbf{x} \quad &= \mathbf{b} \\ \mathbf{r}^T\mathbf{x} \quad &= 0 \\ \mathbf{s}^T\mathbf{u} \quad &= 0 \\ \mathbf{x}, \mathbf{s}, \mathbf{u}, \mathbf{r} \quad &\geqq \mathbf{0}. \end{aligned} \tag{1.4}$$

Proposition 1.6.2. *If Q is PD in (1.3), the KKT necessary conditions (1.4) are sufficient for optimality of problem (1.3).*

Proof. The reader may refer to Bazaraa et al. [12]. $\square$

We can rewrite system (1.4) in matrix form:

$$\begin{bmatrix} \mathbf{r} \\ \mathbf{s} \end{bmatrix} - \begin{bmatrix} Q & A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{p} \\ \mathbf{b} \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{r} & \mathbf{s} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} = \mathbf{0} \tag{1.5}$$

$$\begin{bmatrix} \mathbf{r} \\ \mathbf{s} \end{bmatrix}, \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} \geqq \mathbf{0}.$$

Let $n+m = h$ and then let $M = \begin{bmatrix} Q & A^T \\ -A & 0 \end{bmatrix} \in R^{h \times h}$, $\mathbf{q} = \begin{bmatrix} \mathbf{p} \\ \mathbf{b} \end{bmatrix} \in \mathbb{R}^h$, $\mathbf{w} = \begin{bmatrix} \mathbf{r} \\ \mathbf{s} \end{bmatrix} \in \mathbb{R}^h$, and $\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} \in \mathbb{R}^h$.

For a given M and $\mathbf{q}$, the LCP is to find vectors $\mathbf{w}$ and $\mathbf{z}$ which satisfy:

$$\begin{aligned} \mathbf{w} - M\mathbf{z} &= \mathbf{q} \\ \mathbf{w}^T \mathbf{z} &= 0 \\ \mathbf{w}, \mathbf{z} &\geqq \mathbf{0}. \end{aligned} \tag{1.6}$$

Proposition 1.6.3. *The matrix M in LCP* (1.6) *is PD.*

Proof. Let $\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} \in \mathbb{R}^h$ such that $\mathbf{y}_1 \neq \mathbf{0}$. Then,

$$
\begin{aligned}
\mathbf{y}^T M \mathbf{y} &= \mathbf{y}^T \begin{bmatrix} Q & A^T \\ -A & 0 \end{bmatrix} \mathbf{y} \\
&= \begin{bmatrix} \mathbf{y}_1^T & \mathbf{y}_2^T \end{bmatrix} \begin{bmatrix} Q & A^T \\ -A & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix} \\
&= \mathbf{y}_1^T Q \mathbf{y}_1 - \mathbf{y}_2^T A \mathbf{y}_1 + \mathbf{y}_1^T A^T \mathbf{y}_2 \\
&= \mathbf{y}_1^T Q \mathbf{y}_1 \\
&> 0 \qquad \text{from Assumption } \mathbf{1.6.1}
\end{aligned}
$$

□

Proposition 1.6.4. *A solution to LCP* (1.6) *provides an optimal solution to QP* (1.3).

Proof. The proof is immediate with Proposition 1.6.2 □

1.7 Dissertation Overview

The remainder of the dissertation is organized as follows. In Chapter 2 we discuss the proposed branch and bound algorithm for solving biobjective quadratic programs. The content of this chapter include material from a paper entitled "A Branch and Bound Algorithm for Biobjective Mixed Integer Quadratic Programs" by Pubudu L.W. Jayasekara and Margaret M. Wiecek which will be soon submitted for publication.

In Chapter 3 we review methods for solving convex quadratic multiobjective programs with an application to the portfolio optimization problem. The contents of this chapter include material from a paper entitled "On Convex Multiobjective Programs with Application to Portfolio Optimization" by Pubudu L.W. Jayasekara, Nathan Adelgren and Margaret M. Wiecek which has been published in the Journal of Multi Criteria Decision Analysis.

In Chapter 4 we study multiparametric quadratic programs (mpQPs) and discuss algorithms for solving mpQPs. The contents of this chapter include material from a paper entitled "On Solving Parametric Multiobjective Quadratic Programs with Parameters in General Locations" by Pubudu L.W. Jayasekara, Andrew C. Pangia and Margaret M. Wiecek which has been submitted in the Annals of Operations Research. A modified version of the paper is included in Chapter 4 and consists of the work related to spLCP and mpLCP methods.

The conclusion of the dissertation is included in Chapter 5 where we also discuss possible directions for future research.

Chapter 2

A Branch and Bound Algorithm for Biobjective Mixed Integer Quadratic Programs

[The contents of this chapter include material from a paper entitled "A Branch and Bound Algorithm for Biobjective Mixed Integer Quadratic Program" which will soon be submitted for publication and the authors are Pubudu L.W. Jayasekara and Margaret M. Wiecek.]

2.1 Introduction

Multiobjective programs (MOPs) with mixed-integer variables have recently been the objects of numerous studies since they model decision-making problems arising in many areas of human activity and their Pareto sets have interesting mathematical properties. Consequently, the development of algorithms for computing these sets has been the main goal of those studies. Multiobjective mixed-integer linear programs is a class of MOPs for which various algorithms have already been developed (refer to [5] for a concise review of these methods). Despite the linear case, the research have been performed mainly in the biobjective setting, while problems with more objectives have been addressed only in a few recent studies [6, 103, 104].

A natural next research step to study multiobjective mixed-integer convex programs has already been initiated. Branch and bound (BB) methods to approximate the Pareto set for such problems are proposed in [26, 105, 35], while the biobjective case is examined in [39]. A decomposition algorithm to compute an enclosure of the Pareto set is designed in [47] and a report on its implementation is contained in [48]. Some authors go further and assume nonconvexity of the functions on top of variable integrality. In [91], a branch-and-bound (BB) type algorithm is proposed for MOPs with nonconvex objectives and convex constraints, while in [25] the general case of bounded objective functions and disconnected feasible sets is addressed.

In this paper we continue the research direction to compute the Pareto set for the convex quadratic case. Multiobjective quadratic programs (MOQPs) are appealing since they have elegant mathematical properties and model important applications such as regression analysis [1], portfolio optimization [106, 134, 68], predictive control [15], and others. Integer variables are added in [34], while mixed-integer variables in the context of portfolio optimization are included in [111] and the original MOQP is scalarized and solved as a single objective problem.

We design and implement a BB algorithm for biobjective mixed-integer quadratic programs (BOMIQPs). In contrast to the studies above, in which the

Pareto set is approximated or represented by computing specific points, the proposed algorithm provides the exact Pareto set in closed form. We emphasize that there have been so far three classes of MOPs whose exact Pareto set can be computed in closed form, namely, multiobjective linear programs (e.g., [116, 81]) including the case of mixed-integer variables, and multiobjective quadratic programs (refer to [68] for a review). BOMIQPs now join this "elite" group of MOPs as the current paper appears to be the first study making it possible.

The algorithm relies on three fundamental modules of the BB scheme: solving node problems, branching, and fathoming; and a newly developed module of set dominance. Continuous relaxation of the BOMIQP are solved at the BB nodes with the recently developed mpLCP method, which solves MOQPs as multiparametric linear complementarity problems (mpLCPs) and provides the exact efficient solutions to MOQPs in parametric form [3]. The branching module is extended to be applicable to BOMIQPs and is integrated with the mpLCP method. Selected fathoming rules are implemented in a new way to account for the properties of the Pareto set of the BOMIQP. In the module of set dominance, Pareto sets are compared under incomplete information to yield the resulting nondominated set and eventually produce the Pareto set of the BOMIQP.

The paper consists of eight sections and an appendix. In Section 2.2, the BOMIQP is formulated and accompanied by related auxiliary biobjective quadratic programs (BOQPs), and the methods for solving these BOQPs are reviewed in the context of the BB algorithm. An overview and the modules of the algorithm are presented in Sections 2.3-2.6, while the complete algorithm and numerical results are given in Section 2.7. The paper is concluded in Section 2.8. In the appendix an example BOMIQP is solved with the BB algorithm and other supporting information is included.

2.2 Problem Statements and Solution Methods

We begin with basic notation and define nondominated points in an arbitrary set. We then formulate the BOMIQP, define the concepts, and present auxiliary optimization problems and solution methods that are needed for the BB algorithm.

Let $n, p \in \mathbb{N}$, $\mathbb{R}^n, \mathbb{R}^p \subset \mathbb{R}^n, 0 \leq p \leq n, \mathbb{R}^2$ be Euclidean vector spaces, and $\mathbb{Z}^{n-p} \subset \mathbb{R}^n$ be the set of all integer vectors. For $\mathbf{y}^1, \mathbf{y}^2 \in \mathbb{R}^2$ the following binary relations are defined: $\mathbf{y}^1 \leqq \mathbf{y}^2$ if and only if $y_k^1 \leq y_k^2$ for $k = 1, 2; \mathbf{y}^1 \leq \mathbf{y}^2$ if and only if $y_k^1 \leq y_k^2$ for $k = 1, 2$ and $\mathbf{y}^1 \neq \mathbf{y}^2; \mathbf{y}^1 < \mathbf{y}^2$ if and only if $y_k^1 < y_k^2$ for $k = 1, 2$.

Definition 2.2.1. *Let $S \subset \mathbb{R}^2$. A point $\boldsymbol{y}^1 \in S$ is said to dominate a point $\boldsymbol{y}^2 \in S$ provided $\boldsymbol{y}^1 \leq \boldsymbol{y}^2$. A point $\boldsymbol{y}^1 \in S$ is said to be nondominated provided there does not exist $\boldsymbol{y}^2 \in S$ such that $\boldsymbol{y}^2 \leq \boldsymbol{y}^1$. Let S_N denote the set of nondominated points in S and $N(\cdot)$ denote the operator on a set such that $S_N = N(S)$.*

We define $\mathbb{R}^2_{\geqq} = \{\mathbf{y} \in \mathbb{R}^2 : \mathbf{y} \geqq \mathbf{0}\}$ and the sets $\mathbb{R}^2_{\geq}, \mathbb{R}^2_{>}$ are defined accordingly. We also define $\mathbb{R}^2_{\substack{\geqq \\ \leqq}} = \{\mathbf{y} \in \mathbb{R}^2 : y_1 \geq 0, y_2 \leq 0\}$, $\mathbb{R}^2_{\substack{\geq \\ \leq}} = \mathbb{R}^2_{\substack{\geqq \\ \leqq}} \setminus \{\mathbf{0}\}$ and $\mathbb{R}^2_{\gtrless} = \{\mathbf{y} \in \mathbb{R}^2 : y_1 > 0, y_2 < 0\}$. For a subset $S \subseteq \mathbb{R}^2$ we define $S_{\geqq} = S + \mathbb{R}^2_{\geqq}$. The sets $S_{\geq}$ and $S_{>}$ are defined accordingly. In addition, $bd(S)$ and $|S|$ denote the boundary and cardinality of S, respectively.

Consider the BOMIQP:

$$\begin{aligned} \mathcal{P}: \quad \min \quad & \mathbf{f}(\mathbf{x}) = \left[f_1(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T Q_1 \mathbf{x} + \mathbf{p}_1^T \mathbf{x}, \;\; f_2(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T Q_2 \mathbf{x} + \mathbf{p}_2^T \mathbf{x}\right] \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{X} = \{\mathbf{x} \in \mathbb{R}^p \times \mathbb{Z}^{n-p} : A\mathbf{x} \leqq \mathbf{b}, \mathbf{x} \geqq \mathbf{0}\}, \end{aligned} \tag{2.1}$$

where $\mathbf{f}: \mathbb{R}^n \to \mathbb{R}^2$, $Q_1, Q_2 \in \mathbb{Q}^{n\times n}$, $\mathbf{p}_1, \mathbf{p}_2 \in \mathbb{Q}^n, A \in \mathbb{Q}^{m\times n}$, and $\mathbf{b} \in \mathbb{Q}^m$. The spaces $\mathbb{R}^p \times \mathbb{Z}^{n-p}$ and $\mathbb{R}^2$ are referred to as the decision space and the objective space to BOMIQP (2.1), respectively. Problem $\mathcal{P}$ (2.1), which includes p continuous and $n-p$ integer variables, is referred to as the original problem. We make the following assumptions.

Assumption 2.2.2. *1. The set $\mathcal{X}$ is nonempty and compact.*

2. The function f_i is strictly convex, i.e., matrix $Q_i, i = 1, 2,$ is positive definite (PD).

3. Each integer variable is bounded, i.e., $a_{l_i} \leq x_i \leq a_{u_i}$, for some $a_{l_i}, a_{u_i} \in \mathbb{Z}, i = p+1, \cdots, n$.

For BOMIQP (2.1), the outcome (attainable) set in the objective space is defined as the image of the feasible set $\mathcal{X}$ under the vector-valued mapping $\mathbf{f}$.

$$\mathcal{Y} = \mathbf{f}(\mathcal{X}) = \left\{\mathbf{y} \in \mathbb{R}^2 : \mathbf{y} = (f_1(\mathbf{x}), f_2(\mathbf{x})), \mathbf{x} \in \mathcal{X}\right\}.$$

The elements of $\mathcal{Y}$ are referred to as outcome or criterion vectors. Solution to BOMIQP (2.1) is understood as finding its efficient solutions in $\mathcal{X}$ and Pareto outcomes in $\mathcal{Y}$.

Definition 2.2.3. *A feasible solution $\boldsymbol{x}^1 \in \mathcal{X}$ is said to be an efficient solution to BOMIQP* (2.1) *provided there does not exist $\boldsymbol{x}^2 \in \mathcal{X}$ such that $\boldsymbol{f}(\boldsymbol{x}^2) \leq \boldsymbol{f}(\boldsymbol{x}^1)$. The* *outcome* $\boldsymbol{y}^1 = \boldsymbol{f}(\boldsymbol{x}^1)$ *is said to be a Pareto (or nondominated) outcome in* $\mathcal{Y}$*. The sets of efficient solutions and Pareto outcomes to* (2.1) *are denoted by* $\mathcal{X}_E$ *and* $\mathcal{Y}_P$*, respectively.*

Because the Pareto set of BOMIQP (2.1) is a nonconvex and disconnected curve that is neither open nor closed, the BOMIQP is a global optimization problem. More details on the structure of this set are given in the next section. For (2.1), the notions of Pareto and nondominated outcomes in $\mathcal{Y}$ is used interchangeably and therefore $Y_P = N(\mathcal{Y})$. We also require the following definitions:

Definition 2.2.4. *The point* $\boldsymbol{y}^I = (y_1^I, y_2^I) \in \mathbb{R}^2$*, where* $y_k^I := \min_{\boldsymbol{x}\in X} f_k(x) = \min_{\boldsymbol{y}\in\mathcal{Y}} \boldsymbol{y}_k$*, for* $k = 1, 2$*, is called the ideal point of BOMIQP* (2.1)*.*

Definition 2.2.5. *The point* $\boldsymbol{y}^{\mathcal{N}} = (y_1^{\mathcal{N}}, y_2^{\mathcal{N}}) \in \mathbb{R}^2$*, where* $y_k^{\mathcal{N}} := \max_{\boldsymbol{x}\in X_E} f_k(x) = \max_{\boldsymbol{y}\in\mathcal{Y}_P} \boldsymbol{y}_k$*, for* $k = 1, 2$*, is called the nadir point of BOMIQP* (2.1)*.*

In the next section we formulate auxiliary BOQPs for which we also define the efficient and Pareto sets. We maintain the notation we use in Definition 2.2.3. For each BOQP we denote the efficient set and the Pareto set, respectively, by attaching a subscript E to the symbol denoting this problem's feasible set and by attaching a subscript P to the symbol denoting this problem's image of the feasible set.

2.2.1 Auxiliary Biobjective Quadratic Programs

Given BOMIQP (2.1), four BOQPs, which are of interest for the BB algorithm, are introduced. Each of these four problems has the same two quadratic objective functions but a different feasible set that is a subset or a superset of the set $\mathcal{X}$.

First, consider first Problem $\tilde{\mathcal{P}}$ being the continuous relaxation of BOMIQP (2.1).

$$\begin{aligned} \tilde{\mathcal{P}}: \quad \min \quad & \mathbf{f}(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x} \in \tilde{\mathcal{X}} = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leqq \mathbf{b}, \mathbf{x} \geqq \mathbf{0}\}. \end{aligned} \tag{2.2}$$

Problem $\tilde{\mathcal{P}}$ is referred to as the relaxed BOMIQP. Based on the definition of $\tilde{\mathcal{X}}$ and Assumption 2.2.2.3, it is a strictly convex problem. Let $\tilde{\mathcal{Y}} = \mathbf{f}(\tilde{\mathcal{X}})$ denote the outcome set of $\tilde{\mathcal{P}}$.

Second, we introduce the so-called slice problem which is also a continuous BOQP obtained by fixing all the integer variables at some feasible values. Let $\bar{\mathbf{x}} \in \mathcal{X} \cap (\mathbb{R}^p_{\geqq} \times \mathbb{Z}^{n-p}_{\geqq})$, then the slice problem of BOMIQP (2.1) can be written as

$$\begin{aligned} \mathcal{P}(\bar{\mathbf{x}}): \quad \min \quad & \mathbf{f}(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{X}(\bar{\mathbf{x}}) = \{\mathbf{x} \in \mathbb{R}^p \times \mathbb{Z}^{n-p} : A\mathbf{x} \leqq \mathbf{b}, \mathbf{x} \geqq \mathbf{0}, x_i = \bar{x}_i, i = p+1, \cdots, n\}. \end{aligned} \tag{2.3}$$

The set $\mathcal{X}(\bar{\mathbf{x}})$ is referred to as a slice of the set $\mathcal{X}$. The slice problem is equivalent to a leaf node problem in the BB tree, because all integer variables have been fixed and no branching is necessary. Let $\mathcal{Y}(\bar{x}) = \mathbf{f}(\mathcal{X}(\bar{\mathbf{x}}))$ denote the outcome set of $\mathcal{P}(\bar{\mathbf{x}})$. Figure 2.1$a$ depicts the sets $\mathcal{Y}(\bar{\mathbf{x}}^i)$ for three slice problems $\mathcal{P}(\bar{\mathbf{x}}^i)$ formulated for three different points $\bar{\mathbf{x}}^i$, and the ideal point and nadir point for BOMIQP (2.1). Each $\mathcal{P}(\bar{\mathbf{x}})$ is a BOQP and its Pareto set is a strictly convex curve. The bold (red), nonconvex, and discontinuous curve is the Pareto set, $\mathcal{Y}_P$, for (2.1). This example reveals that the Pareto set to (2.1) can be constructed as the nondominated set of the union of the Pareto sets of all the slice problems. Consequently, a naive method to compute this Pareto set is to solve all the slice problems and perform this construction. The goal of this work is to design an algorithm to compute this Pareto set in a more efficient way.

The BB tree recursively subdivides the feasible set and creates new BOMIQPs. The third problem considered here is the BOMIQP that is associated with a node s of the BB tree and can be written as

$$\begin{aligned} \mathcal{P}^s: \quad \min \quad & \mathbf{f}(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{X}^s = \{\mathbf{x} \in \mathbb{R}^p \times \mathbb{Z}^{n-p} : A^s\mathbf{x} \leqq \mathbf{b}^s, \mathbf{x} \geqq \mathbf{0}\}. \end{aligned} \tag{2.4}$$

$\mathcal{P}^s$ is referred to as the node problem. The matrix A^s and vector b^s are obtained by augmenting the original linear constraints with additional constraints in the form of bounds on integer variables according to a branching rule.

Although $\mathcal{P}^s$ is associated with a node s, the continuous relaxation, $\tilde{\mathcal{P}}^s$, of this problem is the BOQP that is repetitively solved and therefore plays the role of the engine of the BB algorithm. $\tilde{\mathcal{P}}^s$ is the fourth problem we consider and formulate as follows:

$$\begin{aligned} \tilde{\mathcal{P}}^s: \quad \min \quad & \mathbf{f}(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x} \in \tilde{\mathcal{X}}^s = \{\mathbf{x} \in \mathbb{R}^n : A^s\mathbf{x} \leqq \mathbf{b}^s, \mathbf{x} \geqq \mathbf{0}\}. \end{aligned} \tag{2.5}$$

$\tilde{\mathcal{P}}^s$ is a strictly convex BOQP and is referred to as the relaxed node problem. Let $\tilde{\mathcal{Y}}^s = \mathbf{f}(\tilde{\mathcal{X}}^s)$, $\tilde{\mathcal{X}}^s_E$ and $\tilde{\mathcal{Y}}^s_P = N(\tilde{\mathcal{Y}}^s_P)$ be the outcome set, efficient set, and Pareto set of $\tilde{\mathcal{P}}^s$, respectively. $\tilde{\mathcal{Y}}^s_P$ is a continuous strictly convex curve with the end points $\tilde{\mathbf{y}}^{s1} = \big(\tilde{y}_1^{s1} = f_1(\tilde{\mathbf{x}}^{s1}), \tilde{y}_2^{s1} = f_2(\tilde{\mathbf{x}}^{s1})\big)$ and $\tilde{\mathbf{y}}^{s2} = \big(\tilde{y}_1^{s2} = f_1(\tilde{\mathbf{x}}^{s2}), \tilde{y}_2^{s2} = f_2(\tilde{\mathbf{x}}^{s2})\big)$, where $\tilde{\mathbf{x}}^{si} = arg\min\limits_{\mathbf{x}\in\tilde{\mathcal{X}}^s} f_i(\mathbf{x})$ for $i = 1, 2$ (see Figure 2.1b). Let $\tilde{\mathbf{y}}^{sI}$ and $\tilde{\mathbf{y}}^{s\mathcal{N}}$ be the ideal point and the nadir point of the relaxed node s problem, respectively, where $\tilde{\mathbf{y}}^{sI} = (\tilde{y}_1^{sI} = \tilde{y}_1^{s1}, \tilde{y}_2^{sI} = \tilde{y}_2^{s2}), \tilde{\mathbf{y}}^{s\mathcal{N}} = (\tilde{y}_1^{s\mathcal{N}} = \tilde{y}_1^{s2}, \tilde{y}_2^{s\mathcal{N}} = \tilde{y}_2^{s1})$.

The slice problem $\mathcal{P}(\bar{\mathbf{x}})$ and relaxed node problem $\tilde{\mathcal{P}}^s$ are convex BOQPs with continuous variables. The reader may refer to [68] for the properties of such MOQPs. In particular, the following property is immediate.

Proposition 2.2.6. *[68] The set $\tilde{\mathcal{Y}}^s$ is $\mathbb{R}^2_{\geqq}$-convex, i.e., the set $\tilde{\mathcal{Y}}^s + \mathbb{R}^2_{\geqq}$ is convex.*

For $\tilde{\mathcal{P}}^s$ we define additional sets in the objective space $\mathbb{R}^2$. The set of all convex combinations of the points $\tilde{\mathbf{y}}^{s1}$,$\tilde{\mathbf{y}}^{s2}$ and $\tilde{\mathbf{y}}^{sI}$ is defined as

$$T^s = \{\lambda_1 \tilde{\mathbf{y}}^{s1} + \lambda_2 \tilde{\mathbf{y}}^{s2} + \lambda_3 \tilde{\mathbf{y}}^{sI} : \sum_{i=1}^{3} \lambda_i = 1, \lambda_i \geq 0 \text{ for all } i = 1, 2, 3\}, \tag{2.6}$$

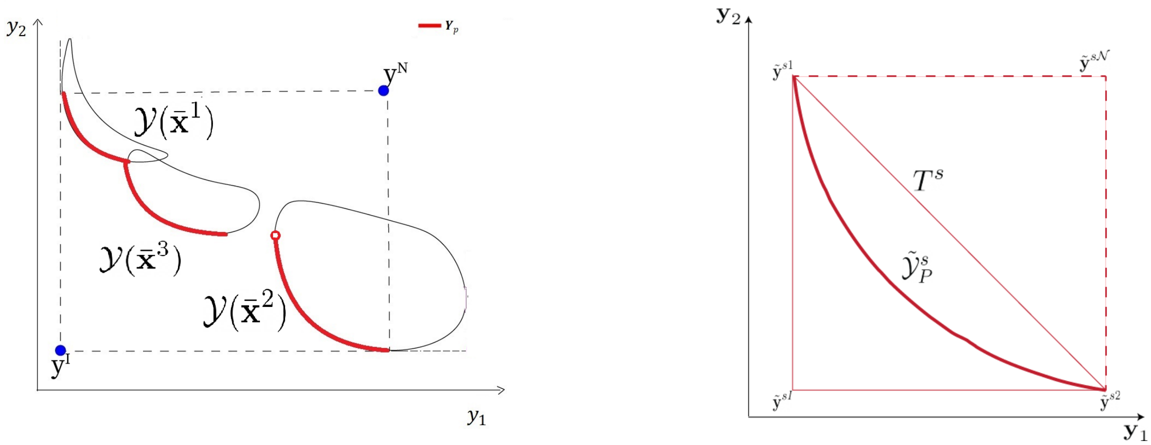
and creates a closed triangle in $\mathbb{R}^2$. Two subsets in $\mathbb{R}^2$ for $\kappa = s, l$ are defined as

$$\begin{aligned} C^{\kappa W} &= (\{\tilde{\mathbf{y}}^{\kappa 1}\}^{<} \setminus \{\tilde{\mathbf{y}}^{\kappa I}\}^{\leqq}) \cup L(\tilde{\mathbf{y}}^{\kappa 1}, \tilde{\mathbf{y}}^{\kappa I}), \\ C^{\kappa S} &= (\{\tilde{\mathbf{y}}^{\kappa 2}\}^{<} \setminus \{\tilde{\mathbf{y}}^{\kappa I}\}^{\leqq}) \cup L(\tilde{\mathbf{y}}^{\kappa 2}, \tilde{\mathbf{y}}^{\kappa I}), \end{aligned} \tag{2.7}$$

where $L(\tilde{\mathbf{y}}^{s1}, \tilde{\mathbf{y}}^{sI})$ and $L(\tilde{\mathbf{y}}^{s2}, \tilde{\mathbf{y}}^{sI})$ are open line segments joining the ideal point with each end point of the Pareto set. Figure 2.1b depicts the Pareto set with two end points, ideal point, nadir point, and triangle T^s for $\tilde{\mathcal{P}}^s$.

2.2.2 Solving Biobjective Quadratic Programs

The traditional approach to solve MOPs is by scalarization in which the problem is reformulated into a single objective program whose optimal solution is efficient to the original problem. In the BB algorithm we employ three well-established scalarizations to solve problems $\mathcal{P}$ and $\tilde{\mathcal{P}}^s$. This section can be safely skipped by the reader who has the knowledge of scalarization techniques in multiobjective optimization.



(a) Outcome sets $\mathcal{Y}(\bar{x}^i), i = 1, 2, 3$; ideal point, nadir point and Pareto set for BOMIQP (2.1).

(b) Pareto set with end points, ideal point, nadir point, triangle T^s for $\tilde{\mathcal{P}}^s$ (2.5).

Figure 2.1: Objective Space and Pareto Sets of $\mathcal{P}$, and Pareto Sets of $\tilde{\mathcal{P}}^s$.

2.2.2.1 The Weighted-Sum Problem associated with BOMIQP(2.1)

Consider the following weighted-sum problem associated with BOMIQP(2.1).

$$\begin{aligned} \mathcal{P}(\lambda) \quad \min \quad & \tfrac{1}{2}\mathbf{x}^T Q(\lambda)\mathbf{x} + \mathbf{p}(\lambda)^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{X}, \end{aligned} \tag{2.8}$$

where $\lambda \in [0, 1]$, $Q(\lambda) = \lambda Q_i + (1 - \lambda)Q_j$ and $\mathbf{p}(\lambda) = \lambda\mathbf{p}_i + (1 - \lambda)\mathbf{p}_j, i \neq j$.

Proposition 2.2.7. *[56] Let $\lambda \in (0, 1)$. If $\check{\boldsymbol{x}} = \check{\boldsymbol{x}}(\lambda)$ is an optimal solution to problem (2.8), then $\check{\boldsymbol{x}}$ is efficient to BOMIQP(2.1).*

In the BB algorithm, problem (2.8) is solved for a fixed set of values of the parameter λ to obtain an initial set of Pareto points of BOMIQP(2.1). For a fixed weight, problem (2.8) is a single objective mixed integer quadratic program (SOMIQP) and commercial software such as GUROBI can be used to solve it. We emphasize that these SOMIQPs are solved only at the initialization of the BB algorithm.

2.2.2.2 The Weighted-Sum Problem associated with BOQP(2.5)

Let $\lambda \in [0,1]$. In the algorithm, the relaxed weighted-sum problem associated with problem (2.5).

$$\begin{aligned} \tilde{\mathcal{P}}^s(\lambda): \quad \min \quad & \tfrac{1}{2}\mathbf{x}^T Q(\lambda)\mathbf{x} + \mathbf{p}(\lambda)^T\mathbf{x} \\ \text{s.t.} \quad & \mathbf{x} \in \tilde{\mathcal{X}}^s = \{\mathbf{x} \in \mathbb{R}^n : A^s\mathbf{x} \leqq \mathbf{b}^s, \mathbf{x} \geqq \mathbf{0}\}, \end{aligned} \tag{2.9}$$

is solved for a fixed parameter λ and also as a parametric optimization problem. Let $\hat{\mathbf{x}}(\lambda)$ denote an optimal solution $\hat{\mathbf{x}}(\lambda)$ to (2.9) and $\hat{\mathbf{y}}(\lambda)$ denote its image in the objective space of BOQP (2.5).

In the parametric case, the optimal solutions to (2.9) are functions of λ. Despite being a convex parametric optimization problem, (2.9) is a challenging problem because it has a parameter in the quadratic term of the objective function. In the literature, only two independently developed methods exist to solve (2.9) for its optimal parametric solutions, the spLCP method [124], and the mpLCP method [3, 2] with a *MATLAB* implementation available [2]. Both these methods rely on the well-known reformulation of the KKT conditions for quadratic programs into an LCP problem [12], which converts (2.9) into a single-parametric linear complementarity problem (spLCP). In both methods the parameter space $[0, 1]$ is partitioned into invariancy intervals over which the solutions to the mpLCP are computed prividing also optimal solution to (2.9).

The spLCP method is applicable to a specific class of spLCPs meaning that it can solve BOQPs satisfying certain assumptions. The solutions to the spLCP are computed at the end points of the invariancy intervals. The mpLCP method is applicable to a broad class of mpLCPs and therefore can solve all convex MOQPs.

The solutions to the mpLCP are computed over the entire parameter space meaning that a parametric solution as a function λ is computed for each invariancy interval. The spLCP method has been implemented and compared to the mpLCP method in [68]. Based on this comparison and the earlier comparisons in [68], the latter emerged as the most universal method to solve MOQPs and is used as a solver for (2.9) in the BB algorithm.

2.2.2.3 The Achievement Function Problem associated with BOQP (2.5)

An achievement function can be used to scalarize BOQP (2.5) [131, 130]. Consider the following problem that is formulated for the relaxed node s problem with a reference point $\mathbf{y}^R \in \mathbb{R}^2$:

$$\begin{aligned} \tilde{\mathcal{P}}(\mathbf{y}^R): \quad & \min_{\mathbf{x}} \max_{k=1,2} \quad \{f_k(\mathbf{x}) - y_k^R\} \\ \text{s.t.} \quad & \qquad \mathbf{x} \in \tilde{\mathcal{X}}^s. \end{aligned} \tag{2.10}$$

Theorem 2.2.8. *[131, 130] Let $\hat{\boldsymbol{x}} \in \tilde{\mathcal{X}}^s$. If $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}(\boldsymbol{y}^R)$ is an optimal solution to problem $\tilde{\mathcal{P}}(\boldsymbol{y}^R)$ (2.10), then $\hat{\boldsymbol{x}}$ is efficient to problem $\tilde{\mathcal{P}}^s$ (2.5).*

Problem $\tilde{\mathcal{P}}(\mathbf{y}^R)$, known as the achievement function problem, is used in the fathoming and set dominance modules of the BB algorithm to check the location of a reference point $\mathbf{y}^R$ with respect to the Pareto set of a relaxed node problem. The following result is immediate based on Proposition 2.2.6 (ii) and Theorem 2.2.8.

Theorem 2.2.9. *Let $\boldsymbol{y}^R \in \mathbb{R}^2 = (y_1^R, y_2^R)$. Let $\hat{\boldsymbol{x}}^s(\boldsymbol{y}^R)$ be an optimal solution to (2.10) and $\hat{\boldsymbol{y}}^s(\boldsymbol{y}^R) = \big(f_1(\hat{\boldsymbol{x}}^s(\boldsymbol{y}^R)), f_2(\hat{\boldsymbol{x}}^s(\boldsymbol{y}^R))\big) = (\hat{y}_1^s, \hat{y}_2^s)$, and let $\tilde{\mathcal{Y}}_P^s$ be the Pareto set of problem (2.5).*

(i) If $\hat{\boldsymbol{y}}^s(\boldsymbol{y}^R) \geq \boldsymbol{y}^R$ *then* $\boldsymbol{y}^R \in \tilde{\mathcal{Y}}_P^s - \mathbb{R}^2_>$*;*

(ii) If $\hat{\boldsymbol{y}}^s(\boldsymbol{y}^R) \leqq \boldsymbol{y}^R$ *then* $\boldsymbol{y}^R \in \tilde{\mathcal{Y}}_P^s + \mathbb{R}^2_{\geqq}$*;*

(iii) If $\hat{y}_1^s < y_1^R$ *and* $\hat{y}_2^s > y_1^R$ *then* $\boldsymbol{y}^R \in \tilde{\mathcal{Y}}_P^s - \mathbb{R}^2_{\gtrless}$*;*

(iv) If $\hat{y}_1^s > y_1^R$ *and* $\hat{y}_2^s < y_1^R$ *then* $\boldsymbol{y}^R \in \tilde{\mathcal{Y}}_P^s + \mathbb{R}^2_{\gtrless}$*.*

Problem $\tilde{\mathcal{P}}(\mathbf{y}^R)$ is a SOQP and commercial software such as *MATLAB* can be used to solve it.

2.2.2.4 The $\boldsymbol{\epsilon}$-Constraint Problem associated with BOQP (2.5)

Another commonly used scalariztion technique to solve MOQPs is the $\boldsymbol{\epsilon}$-constraint problem [59]. We use it for BOQP (2.5), where one quadratic objective function is minimzed while the other objective function generates an inequality constraint with a fixed parameter $\epsilon_j \in \mathbb{R}$ in the right-han-side (RHS). The i^{th} $\boldsymbol{\epsilon}$-constraint problem, $i = 1, 2$, can be formulated as:

$$\begin{aligned} \tilde{\mathcal{P}}(\epsilon_j): \quad \min \quad & f_i(\mathbf{x}) \\ \text{s.t.} \quad & f_j(\mathbf{x}) \leq \epsilon_j \quad j = \{1,2\},\ j \neq i \\ & \mathbf{x} \in \tilde{\mathcal{X}}^s. \end{aligned} \tag{2.11}$$

Theorem 2.2.10. *[59] Let* $\epsilon_j \in \mathbb{R}$ *be fixed and* $\hat{\boldsymbol{x}} \in \tilde{\mathcal{X}}^s(\epsilon_j) = \{\boldsymbol{x} \in \tilde{\mathcal{X}}^s : f_j(\boldsymbol{x}) \leq \epsilon_j, j = \{1,2\},\ j \neq i\}$*. If* $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}(\epsilon_j)$ *is an optimal solution to problem* $\tilde{\mathcal{P}}(\epsilon_j)$ *(3.3), then* $\hat{\boldsymbol{x}}$ *is efficient to problem* $\tilde{\mathcal{P}}^s$ *(2.5).*

Note that at optimality of $\tilde{\mathcal{P}}(\epsilon_j)$ (3.3) the ϵ constraint is active, i.e., $f_j(\hat{\mathbf{x}}(\epsilon_j)) =$

ϵ_j. Let $\hat{\mathbf{y}}(\epsilon_j)$ denote the image of $\hat{\mathbf{x}}(\epsilon_j)$ in the objective space of BOQP (2.5), i.e., $\hat{\mathbf{y}}(\epsilon_j) = (f_1(\hat{\mathbf{x}}(\epsilon_j)), f_2(\hat{\mathbf{x}}(\epsilon_j)))$.

Problem $\tilde{\mathcal{P}}(\epsilon_j)$ (3.3) is used in the set dominance module to discard the points of a Pareto set that are dominated by another Pareto set. The parameter ϵ is selected as a coordinate of an end point of one of the two Pareto sets. For $i, j = \{1, 2\},\ i \neq j$ and $\nu, \omega = \{s, l\}, \nu \neq \omega$, we formulate a quadratically constrained single objective quadratic program

$$\begin{aligned} \tilde{\mathcal{P}}(\tilde{y}_j^{\nu j}):\quad \min \quad & f_i(\mathbf{x}) \\ \text{s.t.} \quad & f_j(\mathbf{x}) \leq \tilde{y}_j^{\nu j} \quad j = \{1, 2\},\ j \neq i \\ & \mathbf{x} \in \tilde{\mathcal{X}}^{\omega}, \end{aligned} \tag{2.12}$$

where $\tilde{y}_j^{\nu j}$ is the j^{th} coordinate of the j^{th} end point of the ν^{th} node problem. Commercial software such as *MATLAB* can solve problem $\tilde{\mathcal{P}}(\tilde{y}_j^{\nu j})$. Let $\hat{\mathbf{x}}(\tilde{y}_j^{\nu j})$ denote an optimal solution to (2.12) and $\hat{\mathbf{y}}(\tilde{y}_j^{\nu j})$ denote its image in the objective space of BOQP (2.5). The following proposition shows how to retrieve the weight λ to be used in problem $\tilde{\mathcal{P}}^s(\lambda)$ (2.9) and obtain the same efficient solution that is obtained by solving $\tilde{\mathcal{P}}(\tilde{y}_j^{\nu j})$(2.12).

Proposition 2.2.11 ([76])**.** *Let $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}(\tilde{y}_j^{\nu j})$ be an optimal solution to $\tilde{\mathcal{P}}(\tilde{y}_j^{\nu j})$(2.12) and $\hat{u} = \hat{u}(\tilde{y}_j^{\nu j}) > 0$ be the Lagrange multiplier associated with the constraint $f_j(\boldsymbol{x}) \leq \tilde{y}_j^{\nu j}$. Then $\hat{\boldsymbol{x}}$ is also an optimal solution to problem $\tilde{\mathcal{P}}^s(\lambda)$ (2.9) for $\lambda = \frac{\hat{u}}{\hat{u}+1}$.*

Having prepared the foundations for the development of the BB algorithm, in the next section we give its overview and then focus on its modules.

2.3 Algorithm Overview

The following definitions and notations are used to develop the BB algorithm for BOMIQP (2.1).

1. Let $\mathcal{Y}_a \subset \mathcal{Y}$ be the set of points and curves found so far by the BB algorithm as candidates to be the elements in the Pareto set, $\mathcal{Y}_P$, for (2.1). $\mathcal{Y}_a$ is referred to as the incumbent (or provisionally nondominated) set.

2. Let $\mathcal{X}_a \subset \mathcal{X}$ be the set of preimages of the points in $\mathcal{Y}_a$.

3. Let $\mathcal{Y}_P^0 \subset \mathcal{Y}_P$ denote an initial subset of $\mathcal{Y}_P$ and $\mathcal{X}_a^0 \subset \mathcal{X}_E$ be the set of preimages of the points in $\mathcal{Y}_P^0$.

4. Let $\mathbf{y}^i = (y_1^i, y_2^i) \in \mathcal{Y}_a$ and $\mathbf{y}^j = (y_1^j, y_2^j) \in \mathcal{Y}_a$. The point $\mathbf{y}^{i,j} = (y_1^j, y_2^i)$ is said to be the nadir point and the point $\check{\mathbf{y}}^{i,j} = (y_1^i, y_2^j)$ is said to be the ideal point implied by the points $\mathbf{y}^i, \mathbf{y}^j \in \mathcal{Y}_a$. The set of all nadir points generated by some selected points in $\mathcal{Y}_a$ is denoted by $\mathcal{Y}^{\mathcal{N}}$.

5. Points $\mathbf{y}^i, \mathbf{y}^j \in \mathcal{Y}_a$ are said to be adjacent in $\mathcal{Y}_a$ if their nadir point $\mathbf{y}^{i,j}$ and ideal point $\check{\mathbf{y}}^{i,j}$ satisfy

$$\left(\{\mathbf{y}^{i,j}\}^{<}\right) \cap \left(\{\check{\mathbf{y}}^{i,j}\}^{>}\right) \cap \mathcal{Y}_a = \emptyset. \tag{2.13}$$

   A nadir point $\mathbf{y}^{i,j} \in \mathcal{Y}^{\mathcal{N}}$ is said to be adjacent if it is implied by two adjacent points $\mathbf{y}^i, \mathbf{y}^j \in \mathcal{Y}_a$.

6. Let $\mathcal{Y}_a^s \subset \mathcal{Y}_a$ be defined as

$$\mathcal{Y}_a^s = \mathcal{Y}_a \cap (T^s - \mathbb{R}^2_{\geqq}). \tag{2.14}$$

7. Let the set R^s be defined as the open rectangle spanned between $\tilde{\mathbf{y}}^{sI}$ and $\mathbf{y}^{i,j}$,

$$R^s = \{\tilde{\mathbf{y}}^{sI}\}^{>} \cap \{\mathbf{y}^{i,j}\}^{\leqq}$$

where $\mathbf{y}^{i,j} \in \mathcal{Y}^{\mathcal{N}}$.

In general, the BB algorithm consists of the initialization and the main step. The following information is available after the initialization step has been completed: (i) An initial subset of Pareto points $\mathcal{Y}_P^0$. These points are computed by solving the weighted sum problem (2.8) with a set of predetermined weights. Then $\mathcal{Y}_a = \mathcal{Y}_P^0$ at the initialization, and $\mathcal{Y}_P^0 \subset \mathcal{Y}_a \cap \mathcal{Y}_P$ during the execution of the algorithm. (ii) The set of nadir points implied by the adjacent Pareto points in $\mathcal{Y}_P^0$.

At every main step of the algorithm the relaxed BOQP (2.5) associated with a node s is solved for its efficient set, $\tilde{\mathcal{X}}_E^s = \{\mathbf{x}^s(\lambda) \in \mathbb{R}^n : \lambda \in [\lambda', \lambda'']\}_{0 \leq \lambda', \lambda'' \leq 1}$, which is as a collection of parametric efficient solution functions with the associated invariancy intervals. The Pareto set $\tilde{\mathcal{Y}}_P^s = \mathbf{f}(\tilde{\mathcal{X}}_E^s)$ (or its subset $\mathbf{y}^s(\lambda) = \mathbf{f}(\mathbf{x}^s(\lambda))$), is a strictly convex curve (or a strictly convex subcurve) that is available parametrically in the form $(f_1(\mathbf{x}(\lambda)), f_2(\mathbf{x}(\lambda)))$ for $\lambda \in [0, 1]$ (or for $\lambda \in [\lambda', \lambda'']$). Both sets, $\tilde{\mathcal{X}}_E^s$ and $\tilde{\mathcal{Y}}_P^s$, are stored. The coordinates of specific points in $\tilde{\mathcal{Y}}_P^s$: (i) the end points $\tilde{\mathbf{y}}^{s1} = \mathbf{f}(\mathbf{x}^s(1))$ and $\tilde{\mathbf{y}}^{s2} = \mathbf{f}(\mathbf{x}^s(0))$; (ii) the points $\tilde{\mathbf{y}}^s(\lambda') = \mathbf{f}(\mathbf{x}^s(\lambda'))$ and $\tilde{\mathbf{y}}^s(\lambda'') = \mathbf{f}(\mathbf{x}^s(\lambda''))$ associated with the end points of the invariancy intervals $[\lambda', \lambda'']$ for $0 < \lambda', \lambda'' < 1$, are actively used during the execution of the algorithm.

The algorithm proceeds differently depending on the properties of $\tilde{\mathcal{X}}_E^s$ and $\tilde{\mathcal{Y}}_P^s$. If the entire set $\tilde{\mathcal{X}}_E^s$ is feasible for (2.1), i.e., $\tilde{\mathcal{X}}_E^s \subset \mathcal{X}_E^s$, then its image, $\tilde{\mathcal{Y}}_P^s$, is added to Y_a and the resulting nondominated set is computed , i.e., $\mathcal{Y}_a = N(\mathcal{Y}_a \cup \tilde{\mathcal{Y}}_P^s)$.

This latter step is performed in the set dominance module. Additionally, if $\tilde{\mathcal{Y}}_P^s$ satisfies certain conditions executed in the fathoming module, then node s is fathomed. The set $\mathcal{Y}^{\mathcal{N}}$ of all nadir points generated by selected points in $\mathcal{Y}_a$ is used in this module.

In a similar way, if an efficient solution function $\mathbf{x}^s(\lambda) \in \tilde{\mathcal{X}}_E^s$ for some $\lambda \in [\lambda', \lambda'']$ is feasible for (2.1), i.e., $\mathbf{x}^s(\lambda) \in \mathcal{X}_E$ for $\lambda \in [\lambda', \lambda'']$, then its image, $\mathbf{y}^s(\lambda) = \mathbf{f}(\mathbf{x}^s(\lambda))$ for $\lambda \in [\lambda', \lambda'']$, is added to $\mathcal{Y}_a$ and the resulting nondominated set is computed in the set dominance module.

If there exists an interval $[\lambda', \lambda'']$ in the collection such that the corresponding parametric solution is not feasible for (2.1), then the algorithm initiates the branching module.

The incumbent set $\mathcal{Y}_a$ is a union of points and strictly convex curves. The objective space images of the newly obtained efficient solutions to relaxed node problems that are feasible for (2.1) are added to $\mathcal{Y}_a$ while this set remains nondominated. Because of the nondominance test, an interval $[\lambda', \lambda'']$ associated with the subcurve $\mathbf{y}(\lambda)$ for $\lambda \in [\lambda', \lambda'']$ may be partitioned into subintervals such that a subcurve $\mathbf{y}(\lambda)$ for $\lambda \in [\lambda^L, \lambda^R]$ is nondominated in $\mathcal{Y}_a$, where λ^L and λ^R are the parameter values associated with the end points of the subcurve that passed the test. Consequently, we have

$$\mathcal{Y}_a = \mathcal{Y}_P^0 \cup \{\mathbf{y}(\lambda) \in \mathbb{R}^2 : \lambda \in [\lambda^L, \lambda^R]\}_{0 \leq \lambda^L, \lambda^R \leq 1} \text{ and } \mathcal{Y}_a = N(\mathcal{Y}_a) \tag{2.15}$$

As the algorithm progresses, $\mathcal{Y}_a$ keeps changing. As new curves or points are added, some curves, subcurves, or points that have been elements of $\mathcal{Y}_a$ so far may be dropped.

In the fathoming module, a subset $\bar{\mathcal{Y}}_a \subset \mathcal{Y}_a$ containing only specific points in $\mathcal{Y}_a$ is used because the information about only these points is sufficient to run this module. We define

$$\bar{\mathcal{Y}}_a = \mathcal{Y}_P^0 \cup \{\mathbf{y}(\lambda) \in \mathbb{R}^2 : \lambda = \lambda^L, \lambda^R\}_{0 \leq \lambda^L, \lambda^R \leq 1} \text{ and } \bar{\mathcal{Y}}_a = N(\bar{\mathcal{Y}}_a) \tag{2.16}$$

In the subsequent sections, the three modules of the algorithm are presented in detail.

2.4 Branching

In the branching module the following strategy is used. At the parent node s, the mpLCP method solves BOQP (2.5) for the efficient set, $\tilde{\mathcal{X}}_E^s = \{\mathbf{x}(\lambda) : \lambda \in [\lambda', \lambda'']\}_{0 \leq \lambda', \lambda'' \leq 1}$.

If this efficient set is not feasible to the original problem (2.1), i.e., there exists an interval $[\lambda', \lambda'']$ and the corresponding parametric solution with an integer variable $x_i(\lambda)$, for some $i \in \{p+1, \ldots, n\}$, to which a function λ for $\lambda \in [\lambda', \lambda'']$ has been assigned, then this variable is selected for branching. In the next step, the range of values this variable takes for $\lambda \in [\lambda', \lambda'']$ is computed. Let $\bar{x}_i^{min}$ and $\bar{x}_i^{max}$ be the smallest and largest value the variable $x_i(\lambda)$ respectively assumes for $\lambda \in [\lambda', \lambda'']$. To obtain these bounds we solve $\bar{x}_i^{min} = \min\{x_i(\lambda) : \lambda \in [\lambda', \lambda'']\}$ and $\bar{x}_i^{max} = \max\{x_i(\lambda) : \lambda \in [\lambda', \lambda'']\}$. Since $x_i(\lambda)$ are rational functions of λ, these optimization problems are not easy to solve and we explain a solution approach later in this section.

Let $\phi_i' = \lfloor \bar{x}_i^{min} \rfloor$ and $\phi_i'' = \lceil \bar{x}_i^{max} \rceil$. Then the constraints of problem $\mathcal{P}^s(\lambda)$(2.9) are extended with bounds on the integer variable x_i, and two new node problems, $\mathcal{P}^{s+1}(\lambda)$ and $\mathcal{P}^{s+2}(\lambda)$, are created and solved for each integer $\phi_i \in [\phi_i', \phi_i'']$.

$$\begin{aligned} \mathcal{P}^{s+1}(\lambda): \quad \min \quad & \tfrac{1}{2}\mathbf{x}^T Q(\lambda)\mathbf{x} + \mathbf{p}(\lambda)^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{x} \in \tilde{\mathcal{X}}^s = \{\mathbf{x} \in \mathbb{R}^n : A^s\mathbf{x} \leqq \mathbf{b}^s, x_i \leq \phi_i, \mathbf{x} \geqq \mathbf{0}\} \\ & \lambda \in [0,1]. \end{aligned} \tag{2.17}$$

$$\begin{aligned} \mathcal{P}^{s+2}(\lambda): \quad \min \quad & \tfrac{1}{2}\mathbf{x}^T Q(\lambda)\mathbf{x} + \mathbf{p}(\lambda)^T \mathbf{x} \\ \text{s.t.} \quad & \mathbf{x} \in \tilde{\mathcal{X}}^s = \{\mathbf{x} \in \mathbb{R}^n : A^s\mathbf{x} \leqq \mathbf{b}^s, x_i \geq \phi_i + 1, \mathbf{x} \geqq \mathbf{0}\} \\ & \lambda \in [0,1]. \end{aligned} \tag{2.18}$$

The first child node problem that is associated with the values in the interval $[\phi_i', \phi_i'']$ has the constraint $x_i \leq \phi_i'$ while the last child node problem has the constraint $x_i \geq \phi_i''$. Using these additional constraints we ensure that no efficient solution to the original problem (2.1) is excluded.

This procedure is applied to every invariancy interval in the collection that carries the infeasibility. Once all such invariancy intervals associated with a node problem have been processed, the efficient set of another node problem is checked for feasibility and if needed, the above process is repeated. Note that a newly generated node problem may be identical to a previously obtained node problem. If so, such a new node problems is discarded. The branching process is started at the root node with problem (2.2) and may continue until leaf nodes have been reached with slice problem (2.3).

To obtain the bounds $\bar{x}_i^{min}$ and $\bar{x}_i^{max}$ by solving $\bar{x}_i^{min} = \min\{x_i(\lambda) : \lambda \in [\lambda', \lambda'']\}$ and $\bar{x}_i^{max} = \max\{x_i(\lambda) : \lambda \in [\lambda', \lambda'']\}$, we proceed as follows. Recall that the solution $x_i(\lambda)$ obtained from the mpLCP method is a rational function of λ, i.e.,

$x_i(\lambda) = \frac{\mathscr{N}_i(\lambda)}{\mathscr{D}_i(\lambda)}$, where $\mathscr{N}_i(\lambda)$ and $\mathscr{N}_i(\lambda)$ are the numerator and denominator functions respectively. If the denominator $\mathscr{D}_i(\lambda)$ is a constant function, then it is likely that the optimization problems are solvable with the *fmincon* function in *MATLAB*. Otherwise, when $\mathscr{D}(\lambda)$ is not a constant function, a specialized solution method for polynomial fractional optimization should be used [40]. Since adopting that method into the BB algorithm requires further research, in this first implementation the optimization problems are solved by discretization. The interval $[\lambda', \lambda'']$ is discretized, the values of $x_i(\lambda)$ are computed for each value of λ, and the minimum and the maximum values, $\bar{x}_i^{min}$ and $\bar{x}_i^{max}$, are found.

2.5 Fathoming

In the fathoming module it is not of interest to find efficient points to BOMIQP (2.1). Rather, it is of interest to decide whether a node problem can be discarded. A fathoming rule gives a condition for discarding a subset of the feasible set that contains no efficient points to (2.1). Until the termination of the BB algorithm, one may not know whether an efficient solution to a node problem, which is feasible to (2.1), is also efficient to (2.1). Regardless of whether that solution is efficient or not, one can still use the feasibility of that solution for dominance purposes.

Fathoming rules make use of the feasibility or infeasibility of a node problem and of dominance between bound sets. The first fathoming rule uses infeasibility. If node problem (2.5) is infeasible, then the corresponding mixed-integer problem (2.4) has no efficient solutions, i.e., if $\tilde{\mathcal{X}}^s = \emptyset$ then $\mathcal{X}^s = \emptyset$. The second fathoming rule is based on the slice problem. If the node problem is a slice problem, then this node problem can be fathomed. If these two rules are not applicable, the fathoming rules based on bound sets become relevant.

2.5.1 Bound Sets

In the biobjective case the bound sets are subsets of the objective space $\mathbb{R}^2$ and determine a region within which the Pareto set to BOMIQP (2.1) is located. In the literature, different bound sets have been proposed and generally presented as pairs (UB, LB^s) where $UB \subseteq \mathbb{R}^2$ stands for the upper bound set and $LB^s \subseteq \mathbb{R}^2$ stands for the lower bound set at node s. Reviews of bound sets in multiobjective programming are contained in [43, 99, 14]. Having examined the bound sets introduced in the literature, we select the pair in which the upper bound set is first proposed in [18], while the pair is also used in [14, 99, 26, 5]

$$UB = \mathcal{Y}^{\geqq} \;\text{ and }\; LB^s = \tilde{\mathcal{Y}}_P^{s\geqq} \;\text{ where }\; \tilde{\mathcal{Y}}_P^{s\geqq} = \tilde{\mathcal{Y}}_P^{s} + \mathbb{R}^2_{\geqq} \tag{2.19}$$

A general sufficient condition for a node s to be fathomed is that that node's lower bound set does not contains Pareto points of BOMIQP (2.1) [14]. In our work, using the sets in (2.19) we apply the equivalent condition [5]:

$$\text{If } LB^s \subset UB, \text{then node } s \text{ can be fathomed.} \tag{2.20}$$

Other similar conditions have been used for fathoming a node. The condition $LB^s \cap UB = \emptyset$ is checked in [14], while a hyperplane to separate the sets LB^s and UB is used in [37].

In the next section, we discuss how to make a fathoming decision based on bound sets (2.19). Since obtaining the lower bound set using $\tilde{\mathcal{Y}}_P^s$ is a challenging task, we introduce a method to make a fathoming decision without obtaining the complete $\tilde{\mathcal{Y}}_P^s$.

2.5.2 Practical Fathoming Rules

We develop fathoming rules based on the condition in (2.20), which, in the context of the BOMIQP, can be written as $\tilde{\mathcal{Y}}_P^{s\geqq} \subset \mathcal{Y}_a^{\geqq}$, or equivalently, $\tilde{\mathcal{Y}}_P^{s\geqq} \subset \mathcal{Y}_a^{s\geqq}$ meaning that each point in $\tilde{\mathcal{Y}}_P^s$ is dominated by at least one point in $\mathcal{Y}_a^s$ [5].

Recall that $\bar{\mathcal{Y}}_a$ is a set of points including the initial Pareto points in $\mathcal{Y}_P^0$ and the end points of the strictly convex curves and subcurves stored in $\mathcal{Y}_a$. Given the set $\bar{\mathcal{Y}}_a \subset \mathcal{Y}_a$ we define the set $\bar{\mathcal{Y}}_a^s \subset \mathcal{Y}_a^s$ as $\bar{\mathcal{Y}}_a^s = \bar{\mathcal{Y}}_a \cap (T^s - \mathbb{R}^2_{\geqq})$ and use it for fathoming.

Let $\mathbf{y}^i = (y_1^i, y_2^i) \in \bar{\mathcal{Y}}_a^s \cap C^{sW}$ and $\mathbf{y}^j = (y_1^j, y_2^j) \in \bar{\mathcal{Y}}_a^s \cap C^{sS}$. The points $\mathbf{y}^i$ and $\mathbf{y}^j$ are said to be the closest points to the point $\tilde{\mathbf{y}}^{sI} = (\tilde{y}_1^{s1}, \tilde{y}_2^{s2})$ if $y_1^i = \min\limits_{\mathbf{y} \in (\bar{\mathcal{Y}}_a^s \cap C^{sW})} (\tilde{y}_1^{s1} - y_1)$ and $y_2^j = \min\limits_{\mathbf{y} \in (\bar{\mathcal{Y}}_a^s \cap C^{sS})} (\tilde{y}_2^{s2} - y_2)$. Since $\bar{\mathcal{Y}}_a^s$ contains a finite number of points, we have the fathoming rule:

$$\text{If } \tilde{\mathcal{Y}}_P^s \subset \left(\bigcup_{\mathbf{y} \in \bar{\mathcal{Y}}_a^s} \{\mathbf{y}\}^{\geqq} \right), \text{ then node } s \text{ can be fathomed.} \tag{2.21}$$

Based on this rule, two practical fathoming rules are now introduced.

Rule 1: If there exists $\mathbf{y}^i \in \bar{\mathcal{Y}}_a^s$ such that $\mathbf{y}^i \in \{\tilde{\mathbf{y}}^{sI}\}^{\leqq}$, then node s can be fathomed.

In Rule 1, $\tilde{\mathcal{Y}}_P^s \subset \{\mathbf{y}^i\}^{\geqq} \subset \left(\bigcup_{\mathbf{y}\in\bar{\mathcal{Y}}_a^s} \{\mathbf{y}\}^{\geqq}\right)$ (see Figure 2.2a). Note that if Rule 1 does not hold and $\bar{\mathcal{Y}}_a^s \cap C^{sW} = \emptyset$ or $\bar{\mathcal{Y}}_a^s \cap C^{sS} = \emptyset$, then node s cannot be fathomed (see Figure 2.14).

Rule 2 is constructed using the following information. Assume $\mathcal{Y}^{\mathcal{N}} \cap T^s \neq \emptyset$. Let the nadir point $\mathbf{y}^{i,j} \in \mathcal{Y}^{\mathcal{N}}$ be implied by the points $\mathbf{y}^i \in \bar{\mathcal{Y}}_a^s \cap C^{sW}$ and $\mathbf{y}^j \in \bar{\mathcal{Y}}_a^s \cap C^{sS}$ that are the closest points to the point $\tilde{\mathbf{y}}^{sI}$. Note that these closest points $\mathbf{y}^i$ and $\mathbf{y}^j$ do not have to be adjacent because there might be other points $\mathbf{y} \in \bar{\mathcal{Y}}_a^s$ located in R^s. We examine the adjacent nadir points $\mathbf{y}^{\kappa,\eta}$ in R^s implied by the adjacent points $\mathbf{y}^{\kappa}, \mathbf{y}^{\eta} \in \left(\bar{\mathcal{Y}}_a^s \cap R^s\right) \cup \{\mathbf{y}^i\} \cup \{\mathbf{y}^j\}$. Note that the nadir point $\mathbf{y}^{i,j}$ is also included in R^s. Using these nadir points in R^s, Rule 2 can be written as follows (see Figure 2.13).

Rule 2: If $\mathbf{y}^{\kappa,\eta} \in \left(\tilde{\mathcal{Y}}_P^s - \mathbb{R}^2_{\geqq}\right)$ for all nadir points $\mathbf{y}^{\kappa,\eta} \in R^s$, then node s can be fathomed.

However, if there is at least one nadir point such that $\mathbf{y}^{\kappa,\eta} \in \left(\tilde{\mathcal{Y}}_P^s + \mathbb{R}^2_{>}\right)$ then node s cannot be fathomed. Implementing Rule 2 is not straightforward as the complete set $\tilde{\mathcal{Y}}_P^s$ is not available. To check the location of the nadir points with respect to $\tilde{\mathcal{Y}}_P^s$, we solve the achievement function problem (2.10) $\tilde{\mathcal{P}}(\mathbf{y}^R)$ with $\mathbf{y}^R = \mathbf{y}^{\kappa,\eta}$, where $\mathbf{y}^{\kappa,\eta}$ is a nadir point in R^s. A fathoming decision is then made based on the following proposition.

Proposition 2.5.1. *Let $\boldsymbol{y}^i = (y_1^i, y_2^i) \in \bar{\mathcal{Y}}_a^s \cap C^{sW}$ and $\boldsymbol{y}^j = (y_1^j, y_2^j) \in \bar{\mathcal{Y}}_a^s \cap C^{sS}$ be the points that are the closest to the point $\tilde{\boldsymbol{y}}^{sI}$, $\boldsymbol{y}^{\kappa,\eta} = (y_1^{\kappa,\eta}, y_2^{\kappa,\eta}) \in R^s$ be a nadir point, $\hat{\boldsymbol{x}}^s(\boldsymbol{y}^R)$ be an optimal solution to* (2.10) *$\tilde{\mathcal{P}}(\boldsymbol{y}^R)$ with $\boldsymbol{y}^R = \boldsymbol{y}^{\kappa,\eta}$, and $\hat{\boldsymbol{y}}^s(\boldsymbol{y}^R) = \left(f_1(\hat{\boldsymbol{x}}^s(\boldsymbol{y}^R)), f_2(\hat{\boldsymbol{x}}^s(\boldsymbol{y}^R))\right)$. If $\hat{\boldsymbol{y}}^s(\boldsymbol{y}^R) \in \{\boldsymbol{y}^{\kappa,\eta}\}^{\geq}$ for all $\boldsymbol{y}^{\kappa,\eta} \in R^s$ then node s can be fathomed.*

Proof. Assume $\hat{\mathbf{y}}^s(\mathbf{y}^R) \in \{\mathbf{y}^{\kappa,\eta}\}^{\geq}$ for all $\mathbf{y}^{\kappa,\eta} \in R^s$, then by Theorem 2.2.9(i) $\mathbf{y}^{\kappa,\eta} \in \tilde{\mathcal{Y}}_P^s - \mathbb{R}^2_>$ for all $\mathbf{y}^{\kappa,\eta} \in R^s$. Therefore,

$$\tilde{\mathcal{Y}}_P^s \subset \left(\{\mathbf{y}^i\}^{\geqq} \cup \{\mathbf{y}^j\}^{\geqq} \cup \bigcup_{\mathbf{y} \in \bar{\mathcal{Y}}_a^s \cap R^s} \{\mathbf{y}\}^{\geqq} \cup \bigcup_{\mathbf{y}^{\kappa,\eta} \in R^s} \{\mathbf{y}^{\kappa,\eta}\}^{\geq} \right) \tag{2.22}$$

Note that

$$\left(\bigcup_{\mathbf{y}^{\kappa,\eta} \in R^s} \{\mathbf{y}^{\kappa,\eta}\}^{\geq} \right) \subset \left(\{\mathbf{y}^i\}^{\geqq} \cup \{\mathbf{y}^j\}^{\geqq} \cup \bigcup_{\mathbf{y} \in \bar{\mathcal{Y}}_a^s \cap R^s} \{\mathbf{y}\}^{\geqq} \right) \tag{2.23}$$

From (2.22) and (2.23) we have

$$\tilde{\mathcal{Y}}_P^s \subset \left(\{\mathbf{y}^i\}^{\geqq} \cup \{\mathbf{y}^j\}^{\geqq} \cup \bigcup_{\mathbf{y} \in \bar{\mathcal{Y}}_a^s \cap R^s} \{\mathbf{y}\}^{\geqq} \right) \tag{2.24}$$

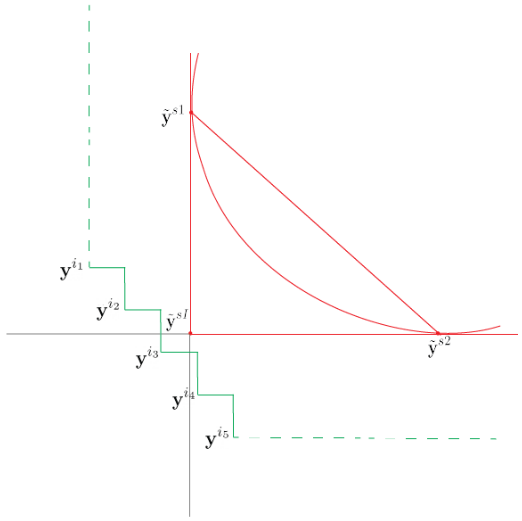
and then

$$\tilde{\mathcal{Y}}_P^s \subset \bigcup_{\mathbf{y} \in \bar{\mathcal{Y}}_a^s} \{\mathbf{y}\}^{\geqq}. \tag{2.25}$$

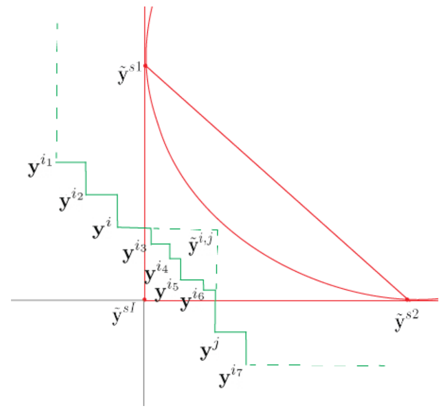
Hence by the rule in (2.21), node s can be fathomed. □

Based on Proposition 2.5.1, problem (2.10) has to be solved with each nadir point $\mathbf{y}^{\kappa,\eta} \in R^s$. Consequently, there are many but a finite number of optimization problems to be solved for one node. Although these problems are single objective problems, solving them may be time-consuming. The following strategy may allow one to speed up this process by solving fewer problems.

Consider the points $\mathbf{y}^i \in \bar{\mathcal{Y}}_a^s \cap C^{sW}$ and $\mathbf{y}^j \in \bar{\mathcal{Y}}_a^s \cap C^{sS}$, which we have already identified as the points closest to the point $\tilde{\mathbf{y}}^{sI}$, and the nadir point $\mathbf{y}^{i,j}$ implied by these two points. If $\mathbf{y}^{i,j} \in T^s$, then we check whether $\mathbf{y}^{i,j} \in \left(\tilde{\mathcal{Y}}_P^s - \mathbb{R}^2_{\geqq}\right)$. If this holds, then without considering the other nadir points we can conclude that



(a) Rule 1 - node s can be fathomed.



(b) The case which requires to solve only one achievement function problem to make a fathoming decision.

Figure 2.2: Fathoming Rules.

node s can be fathomed. To check whether $\mathbf{y}^{i,j} \in \left(\tilde{\mathcal{Y}}_P^s - \mathbb{R}^2_{\geqq}\right)$, we solve achievement function problem (2.10) with $\mathbf{y}^R = \mathbf{y}^{i,j}$ (see Figure 2.13). Otherwise, if $\mathbf{y}^{i,j} \notin T^s$ or $\mathbf{y}^{i,j} \notin \left(\tilde{\mathcal{Y}}_P^s - \mathbb{R}^2_{>}\right)$, the process cannot be sped up and problem (2.10) has to be solved with each nadir point in R^s.

A flowchart 5 of the resulting fathoming module is given bellow. The points in $\mathcal{Y}_a^s$ and the points $\tilde{\mathbf{y}}^{s1}$ and $\tilde{\mathbf{y}}^{s2}$ in $\tilde{\mathcal{Y}}_P^s$ are the input to this module. Using the input, the adjacent nadir points implied by the points in $\bar{\mathcal{Y}}_a^s$, the ideal point for node problem, $\tilde{\mathbf{y}}^{sI}$, and the sets T^s, C^{sW}, C^{sS}, R^s are computed. The condition follows a sequence of conditions that are checked.

First, by Rule 1, if there exists a point $\mathbf{y}^i$ in $\bar{\mathcal{Y}}_a^s$ such that $\mathbf{y}^i \in \{\tilde{\mathbf{y}}^{sI}\}^{\leqq}$, then node s can be fathomed. Otherwise, the sets C^{sW} or C^{sS} are checked whether they contain points in $\bar{\mathcal{Y}}_a^s$. If $\bar{\mathcal{Y}}_a^s \cap C^{sW} = \emptyset$ or $\bar{\mathcal{Y}}_a^s \cap C^{sS} = \emptyset$, then node s cannot be fathomed. Figures 2.14a and 2.14b depict this case.

If $\bar{\mathcal{Y}}_a^s \cap C^{sW} \neq \emptyset$ and $\bar{\mathcal{Y}}_a^s \cap C^{sS} \neq \emptyset$, then the points in $\bar{\mathcal{Y}}_a^s \cap C^{sW}$ and $\bar{\mathcal{Y}}_a^s \cap C^{sS}$ that are the closest to $\tilde{\mathbf{y}}^{sI}$ are found. Let $\mathbf{y}^i \in \bar{\mathcal{Y}}_a^s \cap C^{sW}$ and $\mathbf{y}^j \in \bar{\mathcal{Y}}_a^s \cap C^{sS}$ be the closest points to $\tilde{\mathbf{y}}^{sI}$. The nadir point $\mathbf{y}^{i,j} = (y_1^{i,j}, y_2^{i,j})$ implied by $\mathbf{y}^i$ and $\mathbf{y}^j$ is computed and its location with respect to the set T^s is checked. If $\mathbf{y}^{i,j} \in T^s$, the achievement function problem (2.10) $\tilde{\mathcal{P}}(\mathbf{y}^R)$ is solved with $\mathbf{y}^R = \mathbf{y}^{i,j}$. Let $\hat{\mathbf{y}}^s(\mathbf{y}^R)$ be the image of an optimal solution to problem (2.10). If $\hat{\mathbf{y}}^s(\mathbf{y}^R) > \mathbf{y}^{i,j}$, then node s can be fathomed. If this condition does not hold or $\mathbf{y}^{i,j} \notin T^s$ (see Figures 2.15a and 2.15b), problem (2.10) is solved with all adjacent nadir points $\mathbf{y}^{\kappa,\eta} = (y_1^{\kappa,\eta}, y_2^{\kappa,\eta}) \in R^s$ and the condition $\hat{\mathbf{y}}^s(\mathbf{y}^R) > \mathbf{y}^{\kappa,\eta}$ for all $\mathbf{y}^{\kappa,\eta} \in R^s$, where $\hat{\mathbf{y}}^s(\mathbf{y}^R)$ is the image of an optimal solution to problem (2.10) for a given nadir point. If this condition holds for every nadir point, then by Rule 2, node s can be fathomed. Otherwise, node s cannot be fathomed (see Figure 2.16).

Flowchart - Fathoming Rules

Input: All points in $\bar{\mathcal{Y}}_a^s$, $\tilde{\boldsymbol{y}}^{s1}, \tilde{\boldsymbol{y}}^{s2} \in \tilde{\mathcal{Y}}_P^s$

Obtain: Adjacent nadir points implied by points in $\bar{\mathcal{Y}}_a^s$, ideal point $\tilde{\boldsymbol{y}}^{sI}$, sets T^s, C^{sW}, C^{sS}, R^s

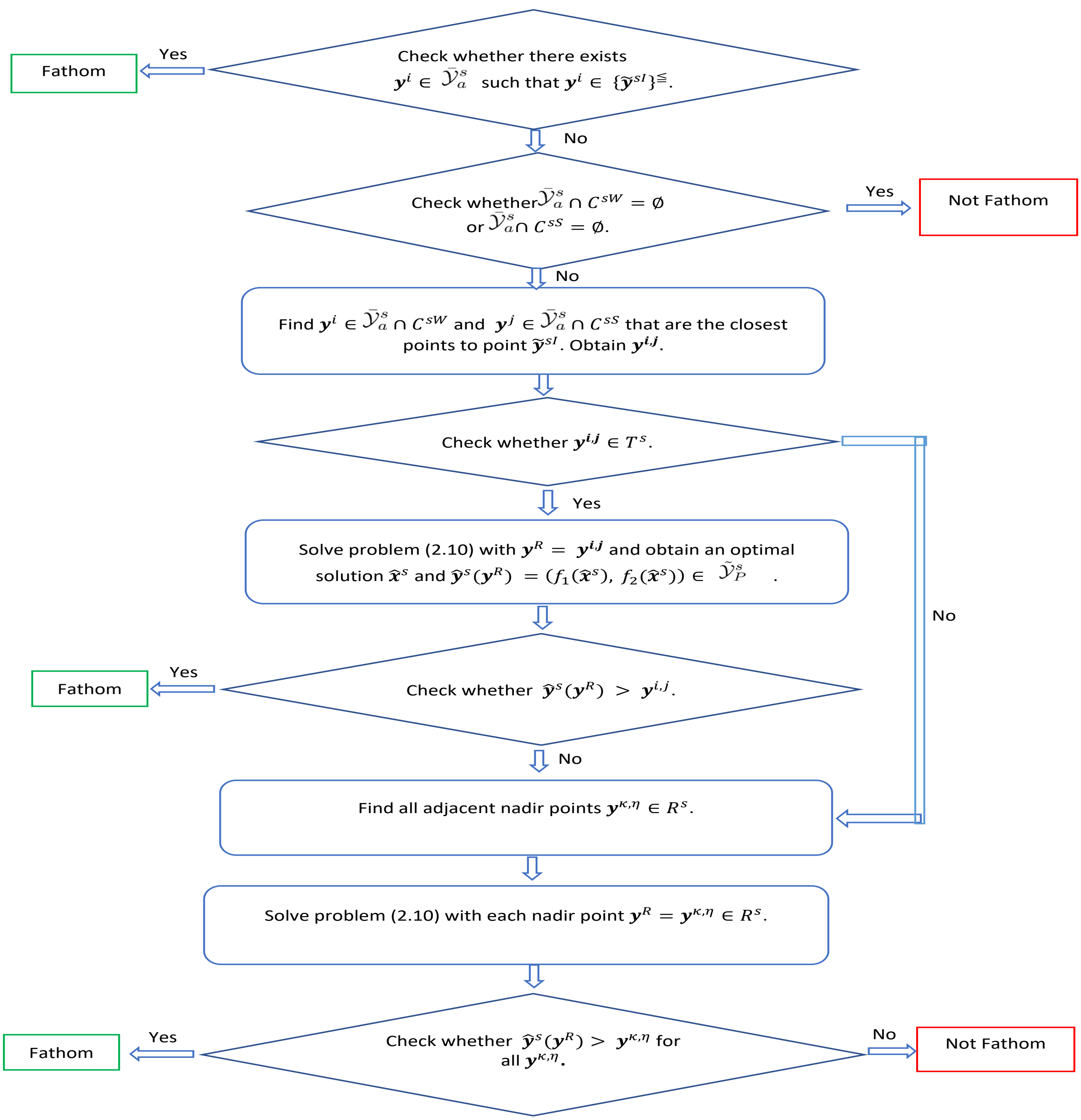


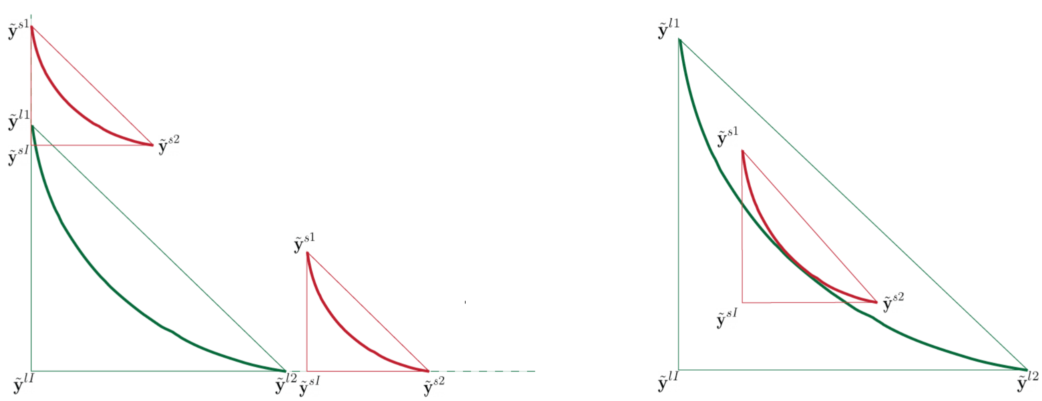
Figure 2.3: Flowchart for the fathoming module

2.6 Dominance Between Sets

The goal of this section is to address the dominance between two Pareto sets, $\tilde{\mathcal{Y}}_P^l$ and $\tilde{\mathcal{Y}}_P^s$ in $\mathbb{R}^2$, that are associated with the corresponding relaxed node problems as given in (2.5). We start with general definitions pertaining to two sets S_1 and S_2 in $\mathbb{R}^2$. Assume $N(S_i) \neq \emptyset$ for $i = 1, 2$. Following [61, 141, 142], the following definitions of (non)dominance for sets are used.

Definition 2.6.1. *Let S_1 and S_2 be two nonempty sets in $\mathbb{R}^2$. S_1 is said to (strictly, weakly) dominate $S_2 \subset \mathbb{R}^2$, or equivalently, S_2 is said to be (strictly, weakly) dominated by S_1, denoted by $S_1(<, \leqq) \leq S_2$, provided for each point $\boldsymbol{y}^2 \in S_2$ there exists a point $\boldsymbol{y}^1 \in S_1$, such that $\boldsymbol{y}^1(<, \leqq) \leq \boldsymbol{y}^2$.*

For two Pareto sets $\tilde{\mathcal{Y}}_P^l, \tilde{\mathcal{Y}}_P^s$, Figures 2.4(a) and 2.4(b) depict the dominance and the weak dominance, respectively while Figure 2.17 depicts the strict dominance.



(a) $\tilde{\mathcal{Y}}_P^l \leq \tilde{\mathcal{Y}}_P^s$. (b) $\tilde{\mathcal{Y}}_P^l \leqq \tilde{\mathcal{Y}}_P^s$.

Figure 2.4: Dominance and weak dominance between two Pareto sets.

Definition 2.6.2. *Let S_1 and S_2 be two nonempty sets in $\mathbb{R}^2$. S_1 is said to be nondominated with respect to S_2 provided there does not exists a point $\boldsymbol{y}^2 \in S_2$ such that $\boldsymbol{y}^2 \leq \boldsymbol{y}^1$ for each $\boldsymbol{y}^1 \in S_1$.*

For our purposes we introduce a definition of partial dominance between two sets.

Definition 2.6.3. *Let S_1 and S_2 be two nonempty sets in $\mathbb{R}^2$. $S_1 \subset \mathbb{R}^2$ is said to partially (weakly) dominate $S_2 \subset \mathbb{R}^2$, or equivalently, S_2 is said to be partially (weakly) dominated by S_1, denoted by $S_1(\leqq_\mathfrak{p}) \leq_\mathfrak{p} S_2$, provided there exists a nonempty subset $S_2' \subset S_2$ such that $S_1(\leqq) \leq S_2'$ and the subset $S_2 \setminus S_2'$ is nondominated with respect to S_1.*

The partially strictly dominated sets are not defined as if there exists a subset $S_2' \subset S_2$ such that $S_1 < S_2'$, then there does not exist a subset $S_2 \setminus S_2'$ that is nondominated with respect to S_1. For two Pareto sets $\tilde{\mathcal{Y}}_P^l, \tilde{\mathcal{Y}}_P^s$, Figures 2.5 and 2.6 depict the partial dominance and partial weak dominance, respectively.

The subsets that are associated with the partially dominated sets and nondominated are defined as follows.

Definition 2.6.4. *Let $S_1, S_2 \subset \mathbb{R}^2$, and $N(S_1 \cup S_2) \neq \emptyset$. A set $S_{iN} \subset S_i$ is called a nondominated subset of S_i for $i = 1, 2$ provided $S_{iN} = N(S_1 \cup S_2) \cap N(S_i)$.*

Based on these definitions the following properties hold.

Proposition 2.6.5. *Let $S \subset \mathbb{R}^2$, $N(S) \neq \emptyset$ and externally stable [42]. Then $N(S) \leqq S$.*

Proof. By Definition 2.6.1 we show for each $\mathbf{y} \in S$ there exists $\mathbf{y}^1 \in N(S)$ such that $\mathbf{y}^1 \leqq \mathbf{y}$. Since $\mathbf{y} \in S$, we consider two cases: (1) Let $\mathbf{y} \in N(S)$. Then $\mathbf{y} \leqq \mathbf{y}$. (2) Let $\mathbf{y} \in S \setminus N(S)$. Then since $N(S)$ is externally stable, there exists $\mathbf{y}^1 \in N(S)$ such that $\mathbf{y}^1 \leqq \mathbf{y}$. Hence $N(S) \leqq S$. $\square$

Proposition 2.6.6. *Let $S_1, S_2 \subset \mathbb{R}^2$, $N(S_i + \mathbb{R}^2_{\geqq}) \neq \emptyset$ and externally stable for $i = 1, 2$.*

(i) The set S_1 dominates the set S_2, $S_1 \leq S_2$, if and only if $S_2 + \mathbb{R}^2_{\geqq} \subset S_1 + \mathbb{R}^2_{\geq}$.

(ii) The set S_1 weakly dominates the set S_2, $S_1 \leqq S_2$, if and only if $S_2 + \mathbb{R}^2_{\geqq} \subseteq S_1 + \mathbb{R}^2_{\geqq}$.

(iii) The set S_1 strictly dominates the set S_2, $S_1 < S_2$, if and only if $S_2 + \mathbb{R}^2_{\geqq} \subset S_1 + \mathbb{R}^2_{>}$.

We prove part (i) of Proposition 2.6.6 here. The proofs of parts (ii) and (iii) are similar to the proof of part (i) and omitted.

Proof. $\Rightarrow$ Assume $S_1 \leq S_2$. By Definition 2.6.1, for each $\mathbf{y}^2 \in S_2$ there exists $\mathbf{y}^1 \in S_1$ such that $\mathbf{y}^1 \leq \mathbf{y}^2$. Let $\mathbf{z}^2 \in S_2 + \mathbb{R}^2_{\geqq}$ such that $\mathbf{z}^2 = \mathbf{y}^2 + \mathbf{d}^2$ where $\mathbf{y}^2 \in S_2$ and $\mathbf{d}^2 \geqq \mathbf{0}$. We have $\mathbf{y}^1 + \mathbf{d}^2 \leq \mathbf{y}^2 + \mathbf{d}^2 = \mathbf{z}^2$, which implies $\mathbf{z}^2 = \mathbf{y}^1 + \mathbf{d}^2 + \bar{\mathbf{d}}^2$ where $\bar{\mathbf{d}}^2 \geq \mathbf{0}$. Then $\mathbf{z}^2 = \mathbf{y}^1 + \mathbf{d}^1$, where $\mathbf{d}^1 = \mathbf{d}^2 + \bar{\mathbf{d}}^2 \geq \mathbf{0}$. Then $\mathbf{z}^2 \in S_1 + \mathbb{R}^2_{\geq}$ and hence $S_2 + \mathbb{R}^2_{\geqq} \subset S_1 + \mathbb{R}^2_{\geq}$. Note that by Definition 2.6.1, $S_2 + \mathbb{R}^2_{\geqq} \neq S_1 + \mathbb{R}^2_{\geq}$.

$\Leftarrow$ Assume $S_2 + \mathbb{R}^2_{\geqq} \subset S_1 + \mathbb{R}^2_{\geq}$. By Proposition 2.6.5, $N(S_1 + \mathbb{R}^2_{\geqq}) \leqq S_1 + \mathbb{R}^2_{\geqq}$, or equivalently, for each $\mathbf{y} \in S_1 + \mathbb{R}^2_{\geqq}$ there exists $\mathbf{y}^1 \in N(S_1 + \mathbb{R}^2_{\geqq})$ such that $\mathbf{y}^1 \leqq \mathbf{y}$. Then for each $\mathbf{y} \in S_1 + \mathbb{R}^2_{\geq}$ there exists $\mathbf{y}^1 \in N(S_1 + \mathbb{R}^2_{\geqq})$ such that $\mathbf{y}^1 \leq \mathbf{y}$. By Proposition 2.3 in [42], we have $N(S_1 + \mathbb{R}^2_{\geqq}) = N(S_1)$. Then for each $\mathbf{y} \in S_1 + \mathbb{R}^2_{\geq}$ there exists $\mathbf{y}^1 \in N(S_1)$ such that $\mathbf{y}^1 \leq \mathbf{y}$. Because $S_2 + \mathbb{R}^2_{\geqq} \subset S_1 + \mathbb{R}^2_{\geq}$, for each $\mathbf{y} \in S_2 + \mathbb{R}^2_{\geqq}$ there exists $\mathbf{y}^1 \in N(S_1)$ such that $\mathbf{y}^1 \leq \mathbf{y}$. Because $S_2 \subset S_2 + \mathbb{R}^2_{\geqq}$, for each $\mathbf{y} \in S_2$ there exists $\mathbf{y}^1 \in N(S_1)$ such that $\mathbf{y}^1 \leq \mathbf{y}$. Since $N(S_1) \subseteq S_1$, for each $\mathbf{y} \in S_2$ there exists $\mathbf{y}^1 \in S_1$ such that $\mathbf{y}^1 \leq \mathbf{y}$ and thus $S_1 \leq S_2$. $\square$

2.6.1 Dominance Between Two Pareto Sets: Theory

We now analyze the mutual location of two Pareto sets, $\tilde{\mathcal{Y}}_P^l$ and $\tilde{\mathcal{Y}}_P^s$, in $\mathbb{R}^2$ to conclude about the (partial) dominance between them. We assume that each Pareto set is a strictly convex curve for which only limited information is available in the form of the end points $\tilde{\mathbf{y}}^{\kappa 1}$ and $\tilde{\mathbf{y}}^{\kappa 2}$, and the ideal point $\tilde{\mathbf{y}}^{\kappa I} \notin \tilde{\mathcal{Y}}_P^{\kappa}$, $\kappa = l, s, l \neq s$. We therefore define the triangles T^{κ}, $\kappa = l, s, l \neq s$, as given in (2.6).

The first result is associated with the efficient and Pareto sets of relaxed node problems of the same branch in the BB tree but pertain to the case a general multiobjective problem with the original set reduced to its subset.

Proposition 2.6.7. *Let $\tilde{\mathcal{P}}^s, \tilde{\mathcal{P}}^l$ be two relaxed node problems* (2.5) *such that $\tilde{\mathcal{X}}^l \subseteq \tilde{\mathcal{X}}^s$.*

1. *If $\tilde{\mathcal{X}}_E^s \cap \tilde{\mathcal{X}}^l = \emptyset$, then*
   - *(i) $\tilde{\mathcal{X}}_E^s \cap \tilde{\mathcal{X}}_E^l = \emptyset$,*
   - *(ii) $\tilde{\mathcal{Y}}_P^s \leq \tilde{\mathcal{Y}}_P^l$.*
2. *If $\tilde{\mathcal{X}}_E^s \cap \tilde{\mathcal{X}}^l \neq \emptyset$,*
   - *(i) then $\tilde{\mathcal{X}}_E^s \cap \tilde{\mathcal{X}}^l \subseteq \tilde{\mathcal{X}}_E^l$,*
   - *(ii) and $\tilde{\mathcal{X}}_E^s \cap \tilde{\mathcal{X}}^l \subset \tilde{\mathcal{X}}_E^l$, then $\tilde{\mathcal{Y}}_P^s \leq_{\mathfrak{p}} \tilde{\mathcal{Y}}_P^l$.*

Proof. 1. (i) By definition $\tilde{\mathcal{X}}_E^l \subseteq \tilde{\mathcal{X}}^l$, but $\tilde{\mathcal{X}}_E^s \cap \tilde{\mathcal{X}}^l = \emptyset$, therefore $\tilde{\mathcal{X}}_E^s \cap \tilde{\mathcal{X}}_E^l = \emptyset$.

(ii) From (i), if $\mathbf{x} \in \tilde{\mathcal{X}}_E^l$, then $\mathbf{x} \notin \tilde{\mathcal{X}}_E^s$. By Definition 2.2.3, there exists $\mathbf{x}^1 \in \tilde{\mathcal{X}}^s$ such that $\mathbf{f}(\mathbf{x}^1) \leq \mathbf{f}(\mathbf{x})$ and then there exists $\mathbf{x}^2 \in \tilde{\mathcal{X}}_E^s$ such that $\mathbf{f}(\mathbf{x}^2) \leq \mathbf{f}(\mathbf{x})$ with $\mathbf{f}(\mathbf{x}^2) = \mathbf{y}^2 \in \tilde{\mathcal{Y}}_P^s$. Since $\mathbf{x} \in \tilde{\mathcal{X}}_E^l$, $\mathbf{f}(\mathbf{x}) = \mathbf{y} \in \tilde{\mathcal{Y}}_P^l$ and therefore $\mathbf{y}^2 \leq \mathbf{y}$. Since $\mathbf{x} \in \tilde{\mathcal{X}}_E^l$ is arbitrary, so is $\mathbf{y} \in \tilde{\mathcal{Y}}_P^l$. Then for each

$\mathbf{y} \in \tilde{\mathcal{Y}}_P^l$ there exists a point $\mathbf{y}^2 \in \tilde{\mathcal{Y}}_P^s$ such that $\mathbf{y}^2 \leq \mathbf{y}$. Therefore by Definition 2.6.1, $\tilde{\mathcal{Y}}_P^s \leq \tilde{\mathcal{Y}}_P^l$.

2. (i) By contradiction, assume $\mathbf{x} \in \tilde{\mathcal{X}}_E^s \cap \tilde{\mathcal{X}}^l$ and $\mathbf{x} \notin \tilde{\mathcal{X}}_E^l$. The former implies $\mathbf{x} \in \tilde{\mathcal{X}}_E^s$ and $\mathbf{x} \in \tilde{\mathcal{X}}^l$. The latter implies either (a) $\mathbf{x} \notin \tilde{\mathcal{X}}^l$, which is a contradiction, or (b) $\mathbf{x} \in \tilde{\mathcal{X}}^l$ and there exists $\mathbf{x}^1 \in \tilde{\mathcal{X}}_E^l$ such that $\mathbf{f}(\mathbf{x}^1) \leq \mathbf{f}(\mathbf{x})$. Then $\mathbf{x}^1 \in \tilde{\mathcal{X}}^l$ and hence $\mathbf{x}^1 \in \tilde{\mathcal{X}}^s$. This implies $\mathbf{x} \notin \tilde{\mathcal{X}}_E^s$, a contradiction.

   (ii) We need to show $\tilde{\mathcal{Y}}_P^s \leq_{\mathfrak{p}} \tilde{\mathcal{Y}}_P^l$. By Definition 2.6.3, we show there exists a nonempty subset $\tilde{\mathcal{Y}}_P^{l^1} \subset \tilde{\mathcal{Y}}_P^l$ such that (a) $\tilde{\mathcal{Y}}_P^s \leq \tilde{\mathcal{Y}}_P^{l^1}$; and (b) $\tilde{\mathcal{Y}}_P^l \setminus \tilde{\mathcal{Y}}_P^{l^1}$ is nondominated with respect to $\tilde{\mathcal{Y}}_P^s$. By Definition 2.6.1, part (a) becomes that for each $\mathbf{y}^1 \in \tilde{\mathcal{Y}}_P^{l^1}$ there exists $\mathbf{y} \in \tilde{\mathcal{Y}}_P^s$ such that $\mathbf{y} \leq \mathbf{y}^1$. By Definition 2.6.2, part (b) becomes that there does not exist $\mathbf{y} \in \tilde{\mathcal{Y}}_P^s$ such that $\mathbf{y} \leq \mathbf{y}'$ for each $\mathbf{y}' \in \tilde{\mathcal{Y}}_P^l \setminus \tilde{\mathcal{Y}}_P^{l^1}$. Below we continue part (a) and (b) separately.

   (a) From (i), if $\mathbf{x} \in \tilde{\mathcal{X}}_E^s \cap \tilde{\mathcal{X}}^l$ then $\mathbf{x} \in \tilde{\mathcal{X}}_E^l$, and therefore if $\mathbf{f}(\mathbf{x}) = \mathbf{y} \in \tilde{\mathcal{Y}}_P^s \cap \tilde{\mathcal{Y}}^l$ then $\mathbf{y} \in \tilde{\mathcal{Y}}_P^l$. Because of the strict containment in the assumption, (a1) there exists $\mathbf{x}^1 \in \tilde{\mathcal{X}}_E^l$ such that $\mathbf{x}^1 \notin \tilde{\mathcal{X}}_E^s \cap \tilde{\mathcal{X}}^l$; and (a2) we can define $\tilde{\mathcal{Y}}_P^{l^1} = \tilde{\mathcal{Y}}_P^l \setminus (\tilde{\mathcal{Y}}_P^s \cap \tilde{\mathcal{Y}}_P^l) \neq \emptyset$ and $\mathbf{f}(\mathbf{x}^1) = \mathbf{y}^1 \in \tilde{\mathcal{Y}}_P^{l^1}$. In (a1), the former implies $\mathbf{x}^1 \in \tilde{\mathcal{X}}^l$ which contradicts the latter. We are left with $\mathbf{x}^1 \notin \tilde{\mathcal{X}}_E^s$ meaning there exists $\mathbf{x}^2 \in \tilde{\mathcal{X}}^s$ such that $\mathbf{f}(\mathbf{x}^2) \leq \mathbf{f}(\mathbf{x}^1)$. Then there also exists $\mathbf{x}^3 \in \tilde{\mathcal{X}}_E^s$ such that $\mathbf{f}(\mathbf{x}^3) \leq \mathbf{f}(\mathbf{x}^1)$, which using (a2) can be written that there exists $\mathbf{f}(\mathbf{x}^3) = \mathbf{y}^3 \in \tilde{\mathcal{Y}}_P^s$ such that $\mathbf{y}^3 \leq \mathbf{y}^1$. Since $\mathbf{x}^1 \in \tilde{X}_E^l \setminus (\tilde{X}_E^s \cap \tilde{X}^l)$ is arbitrary, so is $\mathbf{y}^1 \in \tilde{\mathcal{Y}}_P^{l1}$. Therefore, for all $\mathbf{y}^1 \in \tilde{\mathcal{Y}}_P^{l1}$ there exists $\mathbf{y}^3 \in \tilde{\mathcal{Y}}_P^s$ such that $\mathbf{y}^3 \leq \mathbf{y}^1$.

   (b) Using the identities of relative complements [60] in the set theory, $\tilde{\mathcal{Y}}_P^l \setminus \tilde{\mathcal{Y}}_P^{l1} = \tilde{\mathcal{Y}}_P^l \setminus (\tilde{\mathcal{Y}}_P^l \setminus (\tilde{\mathcal{Y}}_P^s \cap \tilde{\mathcal{Y}}_P^l)) = \tilde{\mathcal{Y}}_P^s \cap \tilde{\mathcal{Y}}_P^l$. Then, by Definition 2.2.3, there does not exist a point $\mathbf{y} \in \tilde{\mathcal{Y}}_P^s$ such that $\mathbf{y} \leq \mathbf{y}'$ for each

$\mathbf{y}' \in \tilde{\mathcal{Y}}_P^s \cap \tilde{\mathcal{Y}}_P^l$, as desired.

$\square$

The subsequent results do not refer to the decision space but address numerous mutual locations of two Pareto sets in the objectve space.

Proposition 2.6.8. *Let $T^l \cap T^s = \emptyset$. The triangle T^l (strictly) dominates the triangle T^s, $T^l(<) \leq T^s$, if and only if the Pareto set $\tilde{\mathcal{Y}}_P^l$ (strictly) dominates the Pareto set $\tilde{\mathcal{Y}}_P^s$, $\tilde{\mathcal{Y}}_P^l(<) \leq \tilde{\mathcal{Y}}_P^s$.*

Proof. $\Rightarrow$ Assume $T^l(<) \leq T^s$. Then for each $\mathbf{y}^s \in T^s$ there exists $\mathbf{y}^l \in T^l$ such that $\mathbf{y}^l(<) \leq \mathbf{y}^s$. We have $\tilde{\mathcal{Y}}_P^l \subset T^l$ and $\tilde{\mathcal{Y}}_P^s \subset T^s$ and therefore for each $\tilde{\mathbf{y}}^s \in \tilde{\mathcal{Y}}_P^s$ there exists $\tilde{\mathbf{y}}^l \in \tilde{\mathcal{Y}}_P^l$ such that $\tilde{\mathbf{y}}^l(<) \leq \tilde{\mathbf{y}}^s$. Thus, by definition $\tilde{\mathcal{Y}}_P^l(<) \leq \tilde{\mathcal{Y}}_P^s$.

$\Leftarrow$ Assume $\tilde{\mathcal{Y}}_P^l(<) \leq \tilde{\mathcal{Y}}_P^s$. Then from Proposition 2.6.6 ($\tilde{\mathcal{Y}}_P^s + \mathbb{R}^2_{\geqq} \subset \tilde{\mathcal{Y}}_P^l + \mathbb{R}^2_{>}$) $\tilde{\mathcal{Y}}_P^s + \mathbb{R}^2_{\geqq} \subset \tilde{\mathcal{Y}}_P^l + \mathbb{R}^2_{\geq}$. Because $T^l \cap T^s = \emptyset$, we have ($T^s + \mathbb{R}^2_{\geqq} \subset T^l + \mathbb{R}^2_{>}$) $T^s + \mathbb{R}^2_{\geqq} \subset T^l + \mathbb{R}^2_{\geq}$. Then from Proposition 2.6.6((iii)) (i) we have $T^l(<) \leq T^s$. $\square$

In the following propositions, the cases conceiving the possible mutual locations of $\tilde{\mathcal{Y}}_P^l$ and $\tilde{\mathcal{Y}}_P^s$ are listed. Although the statements are immediate and therefore not proved, they are helpful in the development of a set dominance procedure.

Proposition 2.6.9. *Let $\tilde{\mathcal{Y}}_P^l$, $\tilde{\mathcal{Y}}_P^s$ be the Pareto sets of two instances of BOQP* (2.5).

(i) If $\tilde{\boldsymbol{y}}^{sI} \in \tilde{\mathcal{Y}}_P^l + \mathbb{R}^2_{\geqq}$, then $\tilde{\mathcal{Y}}_P^l \leq \tilde{\mathcal{Y}}_P^s$ (Figure 2.17).

(ii) If $\tilde{\boldsymbol{y}}^{s2} \in C^{lW}$ or $\tilde{\boldsymbol{y}}^{s1} \in C^{lS}$, then $\tilde{\mathcal{Y}}_P^s \leq_{\mathfrak{p}} \tilde{\mathcal{Y}}_P^l$ (Figure 2.5).

Proposition 2.6.10. *Let $\tilde{\mathcal{Y}}_P^l$, $\tilde{\mathcal{Y}}_P^s$ be the Pareto sets of two instances of BOQP* (2.5)*. If one of the following holds*

(i) $\tilde{\boldsymbol{y}}^{si} \in \tilde{\mathcal{Y}}_P^l - \mathbb{R}^2_>$ *and* $\tilde{\boldsymbol{y}}^{sj} \in \tilde{\mathcal{Y}}_P^l + \mathbb{R}^2_>$ *for* $i, j \in \{1, 2\}, i \neq j$ *(Figure 2.6),*

(ii) $\tilde{\boldsymbol{y}}^{sI} \in C^{lW}(C^{lS})$, $\tilde{\boldsymbol{y}}^{s2}(\tilde{\boldsymbol{y}}^{s1}) \in \tilde{\mathcal{Y}}_P^l + \mathbb{R}^2_>$ *and* $\tilde{\boldsymbol{y}}^{l1}(\tilde{\boldsymbol{y}}^{l2}) \in \tilde{\mathcal{Y}}_P^s + \mathbb{R}^2_>$ *(Figure 2.18),*

(iii) $\tilde{\boldsymbol{y}}^{sI} \in C^{lW}(C^{lS})$, $\tilde{\boldsymbol{y}}^{s1} \in \tilde{\boldsymbol{y}}^{l1} - \mathbb{R}^2_{\geqslant}(\tilde{\boldsymbol{y}}^{s2} \in \tilde{\boldsymbol{y}}^{l2} + \mathbb{R}^2_{\geqslant})$, $\tilde{\boldsymbol{y}}^{s2}(\tilde{\boldsymbol{y}}^{s1}) \in \tilde{\mathcal{Y}}_P^l - \mathbb{R}^2_>$ *and* $\tilde{\boldsymbol{y}}^{l1}(\tilde{\boldsymbol{y}}^{l2}) \in \tilde{\mathcal{Y}}_P^s - \mathbb{R}^2_>$ *(Figure 2.19),*

then the Pareto sets intersect and each Pareto set is partially weakly dominated by the other one.

The following two propositions cover special cases. Proposition 2.6.11 addresses a special case of Proposition 2.6.10 (i).

Proposition 2.6.11. *Let $\tilde{\mathcal{Y}}_P^l$, $\tilde{\mathcal{Y}}_P^s$ be the Pareto sets of two instances of BOQP* (2.5)*. If* $\tilde{\boldsymbol{y}}^{si} \in \tilde{\mathcal{Y}}_P^l$, $\tilde{\boldsymbol{y}}^{sj} \in \tilde{\mathcal{Y}}_P^l + \mathbb{R}^2_>$ *and* $\left(\tilde{\mathcal{Y}}_P^s \setminus \tilde{\boldsymbol{y}}^{si}\right) \cap \tilde{\mathcal{Y}}_P^l = \emptyset$ *for* $i, j \in \{1, 2\}, i \neq j$*, then* $\tilde{\mathcal{Y}}_P^l \leqq \tilde{\mathcal{Y}}_P^s$ *(Figure 2.20).*

Proposition 2.6.12. *Let $\tilde{\mathcal{Y}}_P^l$, $\tilde{\mathcal{Y}}_P^s$ be the Pareto sets of two instances of BOQP* (2.5)*. If* $\tilde{\boldsymbol{y}}^{s1} \in bd\left(\tilde{\boldsymbol{y}}^{l2} + \mathbb{R}^2_{\gtreqless}\right)\left(\tilde{\boldsymbol{y}}^{s2} \in bd\left(\tilde{\boldsymbol{y}}^{l1} - \mathbb{R}^2_{\gtreqless}\right)\right)$*, then* $\tilde{\mathcal{Y}}_P^l \leq_{\mathfrak{p}} \tilde{\mathcal{Y}}_P^s$ *(Figure 2.21).*

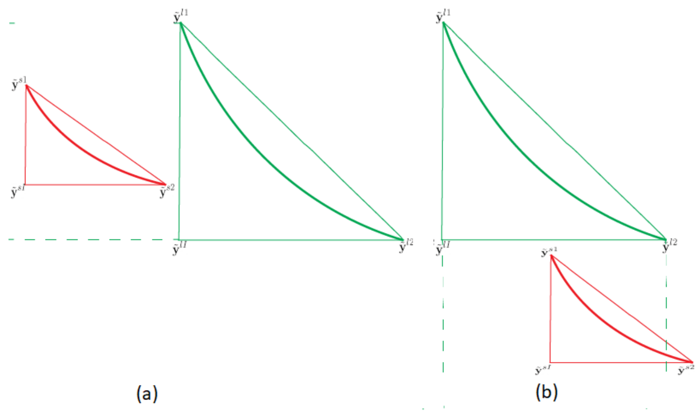


Figure 2.5: Two cases of Proposition 2.6.9(ii): $\tilde{\mathcal{Y}}_P^s \leq_{\mathfrak{p}} \tilde{\mathcal{Y}}_P^l$.

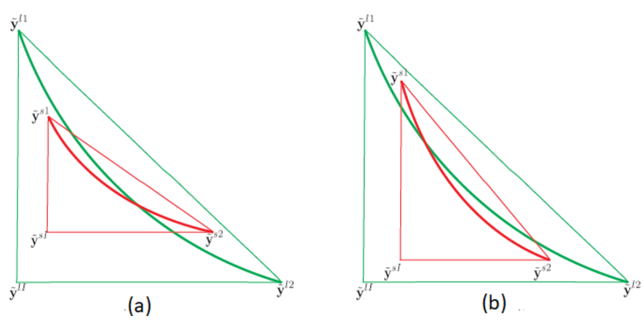


Figure 2.6: Two cases of Proposition 2.6.10(i): Pareto sets intersect, $\tilde{\mathcal{Y}}_P^s \leqq_{\mathfrak{p}} \tilde{\mathcal{Y}}_P^l$ and $\tilde{\mathcal{Y}}_P^l \leqq_{\mathfrak{p}} \tilde{\mathcal{Y}}_P^s$.

2.6.2 Set Dominance Procedure

In this section, we develop a set dominance procedure to make the dominance decision between two Pareto sets and compute the resulting nondominated set. We first give an overview of this procedure and then presents its components. Optional improvements to this procedure are given in the Appendix.

2.6.2.1 Overview

A parametric representation of each efficient set in the form of rational functions is available from the mpLCP method before the set dominance procedure is started. One could use this representation and solve the resulting polynomial equations to find the intersection points between two Pareto sets or decide there is none. However, solving polynomial equations is computationally costly while finding the intersection points is not sufficient to determine the nondominated set resulting from two Pareto sets. Additionally, there are many cases of mutual location of these sets with no intersection points and solving the polynomial equations is obviously unnecessary. Keeping this mind, in the proposed dominance procedure we postpone solving the polynomial equations and first investigate the mutual location of the Pareto sets. When the existence of the intersection point is confirmed or at least is highly probable, we solve the polynomial equations.

The inputs to the procedure are $\tilde{\mathbf{y}}^{l1}, \tilde{\mathbf{y}}^{l2} \in \tilde{\mathcal{Y}}_P^l$ and $\tilde{\mathbf{y}}^{s1}, \tilde{\mathbf{y}}^{s2} \in \tilde{\mathcal{Y}}_P^s$. We then obtain the associated ideal points $\tilde{\mathbf{y}}^{lI}$, $\tilde{\mathbf{y}}^{sI}$, and the triangles T^l and T^s. In this description, we assume the Pareto set $\tilde{\mathcal{Y}}_P^l$ has already been stored in the set $\mathcal{Y}_a$, while the Pareto set $\tilde{\mathcal{Y}}_P^s$ is being introduced to the set $\mathcal{Y}_a$.

We use the achievement function problem (2.10) to check the location of a point with respect to the Pareto set of (2.5) and the $\epsilon-$constraint problem (2.12) to recognize the nondominated subsets to store. If the Pareto sets are partially dominated, then only their nondominated subsets are stored in $\mathcal{Y}_a$ and the $\epsilon-$constraint problem (2.12) is used to identify these subsets. In general, the nondominated set to store is the nondominated set of the union of the two original Pareto sets. Let $\tilde{\mathcal{Y}}_N^{sl}$ denote the nondominated set obtained after applying the set dominance procedure to Pareto sets $\tilde{\mathcal{Y}}_P^s$ and $\tilde{\mathcal{Y}}_P^l$. Then $\tilde{\mathcal{Y}}_N^{sl} = N(\tilde{\mathcal{Y}}_P^s \cup \tilde{\mathcal{Y}}_P^l)$.

The set dominance procedure leads to one of the three decisions, one Pareto set (strictly, weakly) dominates the other, or one Pareto set partially (weakly) dominates the other, or both Pareto sets are nondominated. The nondominated set resulting from the first decision is $\tilde{\mathcal{Y}}_N^{sl} = \tilde{\mathcal{Y}}_P^\kappa$ with $\kappa = s$ or $\kappa = l$. The second decision is made either with $\tilde{\mathcal{Y}}_P^l \cap \tilde{\mathcal{Y}}_P^s = \emptyset$ or $\tilde{\mathcal{Y}}_P^l \cap \tilde{\mathcal{Y}}_P^s \neq \emptyset$. The nondominated set associated with the latter decision is $\tilde{\mathcal{Y}}_N^{sl} = \tilde{\mathcal{Y}}_P^l \cup \tilde{\mathcal{Y}}_P^s$.

To report the subset of the Pareto set to store, the following notation is used. For any two points $\mathbf{y}^1$, $\mathbf{y}^2$ and the associated Pareto set $\tilde{\mathcal{Y}}_P^\kappa$, $\kappa = s, l$, let $\mathrm{Z}^\kappa[\mathbf{y}^1, \mathbf{y}^2] \subseteq \tilde{\mathcal{Y}}_P^\kappa$ denote the closed strictly convex curve and $\mathrm{Z}^\kappa(\mathbf{y}^1, \mathbf{y}^2) \subseteq \tilde{\mathcal{Y}}_P^\kappa$ denote the open strictly convex curve from $\mathbf{y}^1$ to $\mathbf{y}^2$. Also, let $\mathrm{Z}^\kappa(\mathbf{y}^1, \mathbf{y}^2] = \mathrm{Z}^\kappa[\mathbf{y}^1, \mathbf{y}^2] \setminus \{\mathbf{y}^1\}$ and $\mathrm{Z}^\kappa[\mathbf{y}^1, \mathbf{y}^2) = \mathrm{Z}^\kappa[\mathbf{y}^1, \mathbf{y}^2] \setminus \{\mathbf{y}^2\}$ such that $\mathrm{Z}^\kappa(\mathbf{y}^1, \mathbf{y}^2], \mathrm{Z}^\kappa[\mathbf{y}^1, \mathbf{y}^2) \subseteq \tilde{\mathcal{Y}}_P^\kappa$.

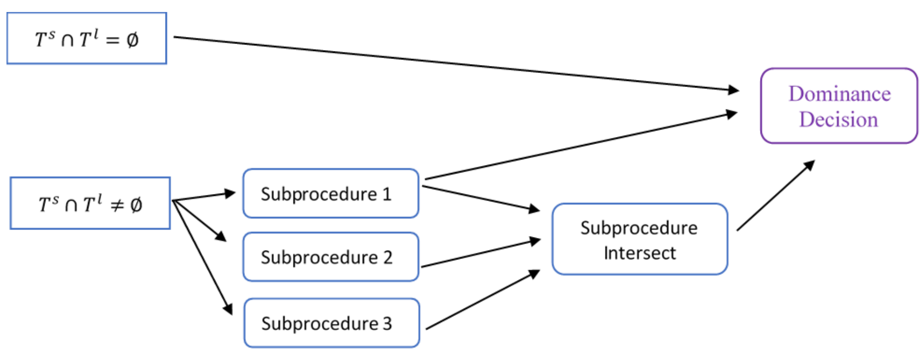


Figure 2.7: Flowchart for the set dominance procedure

The set dominance procedure consists of four subprocedures as depicted in Figure 2.7 and starts with checking whether the triangles T^s and T^l intersect or not. If the triangles do not intersect, then a dominance decision is made accordingly. Otherwise, the mutual locations of the end points of the Pareto sets $\tilde{\mathcal{Y}}_P^s$ and

$\tilde{\mathcal{Y}}_P^l$ are examined in Subprocedures 1-3 that are independently initiated and rely on solving the achievement function problem. In some cases, a dominance decision can be made directly from Subprocedure 1. Otherwise, the three subprocedures continue to Subprocedure Intersect to check for and compute intersection points between the two Pareto sets. In all cases at the end of the entire process, a dominance decision is made and the resulting nondominated set is computed based on solving the ϵ-constraint problem.

2.6.2.2 Subprocedures

In this section, the subprocedures of the set dominance procedure given in Figure 2.7 are presented in detail. These subprocedures are leading to several different cases based on the locations of the end points of two Pareto sets. In each case, the locations of end points, the associated dominance decision, and the nodminated set $\mathcal{Y}_N^{sl}$ are discussed.

1. Let $T^l \cap T^s = \emptyset$.

   I. If $T^s \in \{\tilde{\mathbf{y}}^{l1}\} - \mathbb{R}^2_{\gtrless}$ or $T^s \in \{\tilde{\mathbf{y}}^{l2}\} + \mathbb{R}^2_{\gtrless}$, then both $\tilde{\mathcal{Y}}_P^s$ and $\tilde{\mathcal{Y}}_P^l$ are nondominated. Hence $\tilde{\mathcal{Y}}_N^{sl} = \tilde{\mathcal{Y}}_P^l \cup \tilde{\mathcal{Y}}_P^s$. Figure 2.22 depicts this case. In particular, if $\tilde{\mathbf{y}}^{s2} \in bd\left(\tilde{\mathbf{y}}^{l1} - \mathbb{R}^2_{\substack{\geqq \\ <}}\right)$ $\left(\tilde{\mathbf{y}}^{s1} \in bd\left(\tilde{\mathbf{y}}^{l2} + \mathbb{R}^2_{\substack{\geqq \\ <}}\right)\right)$, then by Proposition 2.6.12, $\tilde{\mathcal{Y}}_P^l \leq_{\mathfrak{p}} \tilde{\mathcal{Y}}_P^s$ and $\tilde{\mathcal{Y}}_N^{sl} = (\tilde{\mathcal{Y}}_P^s \setminus \tilde{\mathbf{y}}^{s2}) \cup \tilde{\mathcal{Y}}_P^l$ $\left(\tilde{\mathcal{Y}}_N^{sl} = (\tilde{\mathcal{Y}}_P^s \setminus \tilde{\mathbf{y}}^{s1}) \cup \tilde{\mathcal{Y}}_P^l\right)$. Figure 2.21 depicts this case.

   II. Otherwise, if $T^s \subset T^l + \mathbb{R}^2_{\geq}$ $(T^l \subset T^s + \mathbb{R}^2_{\geq})$, then $T^l \leq T^s$ $(T^s \leq T^l)$, and by Proposition 2.6.8, $\tilde{\mathcal{Y}}_P^l \leq \tilde{\mathcal{Y}}_P^s$ $(\tilde{\mathcal{Y}}_P^s \leq \tilde{\mathcal{Y}}_P^l)$. Then $\tilde{\mathcal{Y}}_N^{sl} = \tilde{\mathcal{Y}}_P^l$ $(\tilde{\mathcal{Y}}_P^s)$ and $\tilde{\mathcal{Y}}_P^s$ $(\tilde{\mathcal{Y}}_P^l)$ is discarded. Figure 2.23 depicts this case.

III. If the above conditions do not hold, then by Proposition 2.6.9(ii), $\tilde{\mathcal{Y}}_P^s \leq_{\mathfrak{p}} \tilde{\mathcal{Y}}_P^l$. Figure 2.5 depicts this case. The complete Pareto set $\tilde{\mathcal{Y}}_P^s$ and a subset of the Pareto set $\tilde{\mathcal{Y}}_P^l$ are stored as the nondominated set. To obtain this subset of $\tilde{\mathcal{Y}}_P^l$, $\mathcal{P}(\tilde{y}_2^{s2})$ (2.12) is solved and the point $\hat{\mathbf{y}}(\tilde{y}_2^{s2})$ is obtained. Then $\tilde{\mathcal{Y}}_N^{sl} = \tilde{\mathcal{Y}}_P^s \cup \mathrm{Z}^l(\hat{\mathbf{y}}(\tilde{y}_2^{s2}), \tilde{\mathbf{y}}^{l2}]$. Alternatively, $\mathcal{P}(\tilde{y}_1^{s1})$ (2.12) is solved and $\tilde{\mathcal{Y}}_N^{sl} = \tilde{\mathcal{Y}}_P^s \cup \mathrm{Z}^l[\tilde{\mathbf{y}}^{l1}, \hat{\mathbf{y}}(\tilde{y}_1^{s1}))$ is stored.

2. Let $T^l \cap T^s \neq \emptyset$.

I. Without loss of generality, first check whether $T^s \subset T^l$. If this holds, apply Subprocedure 1.

II. Otherwise, check whether $\tilde{\mathbf{y}}^{sI} \in T^l$. If this does not hold, continue to Subprocedure 2. If this condition holds, check whether $\tilde{\mathbf{y}}^{sI} \in \tilde{\mathcal{Y}}_P^l + \mathbb{R}^2_{\geqq}$ by solving problem (2.10) with point $\mathbf{y}^R = \tilde{\mathbf{y}}^{sI}$. If $\tilde{\mathbf{y}}^{sI} \in \tilde{\mathcal{Y}}_P^l + \mathbb{R}^2_{\geqq}$, then by Proposition 2.6.9(i), $\tilde{\mathcal{Y}}_P^l \leq \tilde{\mathcal{Y}}_P^s$. Figure 2.17 depicts this case. Then $\tilde{\mathcal{Y}}_N^{sl} = \tilde{\mathcal{Y}}_P^l$ and $\tilde{\mathcal{Y}}_P^s$ is discarded. Otherwise, apply Subprocedure 3.

Subprocedure 1: Let $T^s \subset T^l$.

Solving problem (2.10) with $\mathbf{y}^R = \tilde{\mathbf{y}}^{sI}$, first check whether $\tilde{\mathbf{y}}^{sI} \in \tilde{\mathcal{Y}}_P^l + \mathbb{R}^2_{\geqq}$. If this holds, by Proposition 2.6.9(i), $\tilde{\mathcal{Y}}_P^l \leq \tilde{\mathcal{Y}}_P^s$. Then $\tilde{\mathcal{Y}}_N^{sl} = \tilde{\mathcal{Y}}_P^l$ and $\tilde{\mathcal{Y}}_P^s$ is discarded. Figure 2.24 depicts this case. Else, check whether $\tilde{\mathbf{y}}^{si} \in \tilde{\mathcal{Y}}_P^l + \mathbb{R}^2_{>}$ for $i = 1, 2$. If both $\tilde{\mathbf{y}}^{s1}, \tilde{\mathbf{y}}^{s2} \in \tilde{\mathcal{Y}}_P^l + \mathbb{R}^2_{>}$, then continue directly to Subprocedure Intersect. Figure 2.25 depicts this case. If an intersect point does not exist, that is, $\tilde{\mathcal{Y}}_P^l \cap \tilde{\mathcal{Y}}_P^s = \emptyset$, then $\tilde{\mathcal{Y}}_N^{sl} = \tilde{\mathcal{Y}}_P^l$. (Figure 2.25(a)). Otherwise, report the set $\tilde{\mathcal{Y}}_N^{sl}$ accordingly with the intersection points (Figure 2.25(b)).

If $\tilde{\mathbf{y}}^{si} \in \tilde{\mathcal{Y}}_P^l - \mathbb{R}^2_{>}$ and $\tilde{\mathbf{y}}^{sj} \in \tilde{\mathcal{Y}}_P^l + \mathbb{R}^2_{>}$ for i $\neq j$, then by Proposition 2.6.10(i), $\tilde{\mathcal{Y}}_P^l$

and $\tilde{\mathcal{Y}}_P^s$ intersect and each Pareto set is partially weakly dominated by the other one. That is, $\tilde{\mathcal{Y}}_P^l \leqq_{\mathfrak{p}} \tilde{\mathcal{Y}}_P^s$ and $\tilde{\mathcal{Y}}_P^s \leqq_{\mathfrak{p}} \tilde{\mathcal{Y}}_P^l$. Figures 2.6 (a) and (b) depict this case. Problem $\mathcal{P}(\tilde{y}_1^{s1})$(2.12) or $\mathcal{P}(\tilde{y}_2^{s2})$(2.12) is solved to identify the nondominated subsets of $\tilde{\mathcal{Y}}_P^l$. Subprocedure Intersect is used to find an intersection point. Otherwise, that is, if both $\tilde{\mathbf{y}}^{s1}, \tilde{\mathbf{y}}^{s2} \in \tilde{\mathcal{Y}}_P^l - \mathbb{R}^2_>$, then continue directly to Subprocedure Intersect. Figure 2.27 depicts this case. If Subprocedure Intersect concludes $\tilde{\mathcal{Y}}_P^l \cap \tilde{\mathcal{Y}}_P^s = \emptyset$, then $\tilde{\mathcal{Y}}_P^s \leq_{\mathfrak{p}} \tilde{\mathcal{Y}}_P^l$. For the case depicted in Figure 2.27(a), problems $\mathcal{P}(\tilde{y}_1^{s1})$(2.12) and $\mathcal{P}(\tilde{y}_2^{s2})$(2.12) are solved and the points $\hat{\mathbf{y}}(\tilde{y}_1^{s1})$, $\hat{\mathbf{y}}(\tilde{y}_2^{s2})$ in $\tilde{\mathcal{Y}}_P^l$ are obtained, respectively. Then $\tilde{\mathcal{Y}}_N^{sl} = \mathrm{Z}^l[\tilde{\mathbf{y}}^{l1}, \hat{\mathbf{y}}(\tilde{y}_1^{s1})) \cup \tilde{\mathcal{Y}}_P^s \cup \mathrm{Z}^l(\hat{\mathbf{y}}(\tilde{y}_2^{s2}), \tilde{\mathbf{y}}^{l2}]$. If Subprocedure Intersect concludes $\tilde{\mathcal{Y}}_P^l \cap \tilde{\mathcal{Y}}_P^s \neq \emptyset$, then $\tilde{\mathcal{Y}}_P^s \leqq_{\mathfrak{p}} \tilde{\mathcal{Y}}_P^l$ and $\tilde{\mathcal{Y}}_P^l \leqq_{\mathfrak{p}} \tilde{\mathcal{Y}}_P^s$. For the case depicted in Figure 2.27(b), problems $\mathcal{P}(\tilde{y}_1^{s1})$(2.12) and $\mathcal{P}(\tilde{y}_2^{s2})$(2.12) are solved and the set $\tilde{\mathcal{Y}}_N^{sl}$ is reported accordingly with the two intersection points.

Subprocedure 2: Let $T^s \not\subset T^l$ and $\tilde{\mathbf{y}}^{sI} \notin T^l$.

I. Let $\tilde{\mathbf{y}}^{sI}, \tilde{\mathbf{y}}^{s1} \in C^{lW}$ $(\tilde{\mathbf{y}}^{sI}\tilde{\mathbf{y}}^{s2} \in C^{lS})$.

   i. If $\tilde{\mathbf{y}}^{s2}(\tilde{\mathbf{y}}^{s1}) \in \tilde{\mathcal{Y}}_P^l + \mathbb{R}^2_>$, then by Proposition 2.6.10(i), the Pareto sets intersect and each Pareto set is partially weakly dominated by the other one. Figure 2.28 depicts this case. Subprocedure Intersect is used to find the intersection point;

   ii. else, that is, $\tilde{\mathbf{y}}^{s2}(\tilde{\mathbf{y}}^{s1}) \in \tilde{\mathcal{Y}}_P^l - \mathbb{R}^2_>$, continue to Subprocedure Intersect. Figure 2.8 depicts this case. If Subprocedure Intersect concludes $\tilde{\mathcal{Y}}_P^l \cap \tilde{\mathcal{Y}}_P^s = \emptyset$, then $\tilde{\mathcal{Y}}_P^s \leq_{\mathfrak{p}} \tilde{\mathcal{Y}}_P^l$. For the case depicted in Figure 2.8(a), problem $\mathcal{P}(\tilde{y}_2^{s2})$(2.12) is solved and point $\hat{\mathbf{y}}(\tilde{y}_2^{s2})$ in $\tilde{\mathcal{Y}}_P^l$ is obtained. Then $\tilde{\mathcal{Y}}_N^{sl} = \tilde{\mathcal{Y}}_P^s \cup \mathrm{Z}^l(\hat{\mathbf{y}}(\tilde{y}_2^{s2}), \tilde{\mathbf{y}}^{l2}]$. If Subprocedure Intersect concludes $\tilde{\mathcal{Y}}_P^l \cap \tilde{\mathcal{Y}}_P^s \neq \emptyset$, then each Pareto set is partially weakly dominated by the other one. For

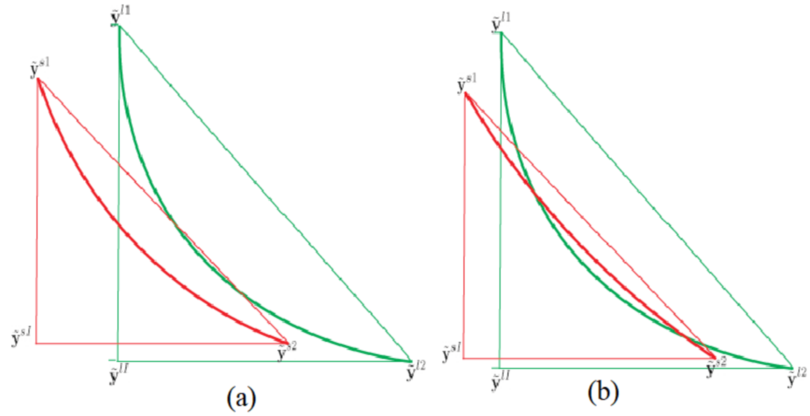


Figure 2.8: Subprocedure 2: (a) $\tilde{\mathcal{Y}}_P^l \cap \tilde{\mathcal{Y}}_P^s = \emptyset$, (b) $\tilde{\mathcal{Y}}_P^l \cap \tilde{\mathcal{Y}}_P^s \neq \emptyset$.

the case depicted in Figure 2.8(b), problem $\mathcal{P}(\tilde{y}_2^{s2})$(2.12) is solved and the set $\tilde{\mathcal{Y}}_N^{sl}$ is reported accordingly with the two intersection points.

II. Let $\tilde{\mathbf{y}}^{sI} \in C^{lW}$ ($\tilde{\mathbf{y}}^{sI} \in C^{lS}$) and $\tilde{\mathbf{y}}^{s1} \in \tilde{\mathcal{Y}}_P^l - \mathbb{R}^2_{\gtrless}(\tilde{\mathbf{y}}^{s2} \in \tilde{\mathcal{Y}}_P^l + \mathbb{R}^2_{\gtrless})$.

i. If $\left(\tilde{\mathbf{y}}^{l1}(\tilde{\mathbf{y}}^{l2}) \in \tilde{\mathcal{Y}}_P^s + \mathbb{R}^2_> \text{ and } \tilde{\mathbf{y}}^{s2}(\tilde{\mathbf{y}}^{s1}) \in \tilde{\mathcal{Y}}_P^l + \mathbb{R}^2_>\right)$ or $\left(\tilde{\mathbf{y}}^{l1}(\tilde{\mathbf{y}}^{l2}) \in \tilde{\mathcal{Y}}_P^s - \mathbb{R}^2_> \text{ and } \tilde{\mathbf{y}}^{s2}(\tilde{\mathbf{y}}^{s1}) \in \tilde{\mathcal{Y}}_P^l - \mathbb{R}^2_>\right)$, then by Propositions 2.6.10(ii) and (iii), the Pareto sets intersect and each Pareto set is partially dominated by the other one. Figures 2.18 and 2.19 depict these cases. Subprocedure Intersect is used to find the intersection points. For the case depicted in Figure 2.19 (a), problems $\mathcal{P}(\tilde{y}_1^{l1})$(2.12) and $\mathcal{P}(\tilde{y}_2^{s2})$(2.12) are solved, while for the case depicted in Figure 2.19 (b), problems $\mathcal{P}(\tilde{y}_1^{s1})$(2.12) and $\mathcal{P}(\tilde{y}_2^{l2})$(2.12) are solved to identify the nondominated subsets of $\tilde{\mathcal{Y}}_P^l$;

ii. else continue to Subprocedure Intersect. Figures 2.26 and 2.29 depict some of these cases. If Subprocedure Intersect concludes $\tilde{\mathcal{Y}}_P^l \cap \tilde{\mathcal{Y}}_P^s = \emptyset$, and (i) $\tilde{\mathcal{Y}}_P^s \leq_{\mathfrak{p}} \tilde{\mathcal{Y}}_P^l$, then for the case depicted in Figure 2.26(a), problem $\mathcal{P}(\tilde{y}_2^{s2})$(2.12) is solved and point $\hat{\mathbf{y}}(\tilde{y}_2^{s2})$ in $\tilde{\mathcal{Y}}_P^l$ is obtained. Then $\tilde{\mathcal{Y}}_N^{sl} = \tilde{\mathcal{Y}}_P^s \cup \mathrm{Z}^l(\hat{\mathbf{y}}(\tilde{y}_2^{s2}), \tilde{\mathbf{y}}^{l2}]$.

(ii) $\tilde{\mathcal{Y}}_P^l \leq_{\mathfrak{p}} \tilde{\mathcal{Y}}_P^s$, then for the case depicted in Figure 2.29(a), problem $\mathcal{P}(\tilde{y}_1^{l1})$(2.12) is solved and point $\hat{\mathbf{y}}(\tilde{y}_1^{l1})$ in $\tilde{\mathcal{Y}}_P^s$ is obtained. Then $\tilde{\mathcal{Y}}_N^{sl} = \mathrm{Z}^s[\tilde{\mathbf{y}}^{s1}, \hat{\mathbf{y}}(\tilde{y}_1^{l1})) \cup \tilde{\mathcal{Y}}_P^l$.

If Subprocedure Intersect concludes $\tilde{\mathcal{Y}}_P^l \cap \tilde{\mathcal{Y}}_P^s \neq \emptyset$ (Figures 2.26(b), 2.29(b)), then each Pareto set is partially weakly dominated by the other one. Problem $\mathcal{P}(\tilde{y}_1^{l1})$(2.12) or $\mathcal{P}(\tilde{y}_2^{s2})$(2.12) is solved and the set $\tilde{\mathcal{Y}}_N^{sl}$ is reported accordingly with the intersection points.

Subprocedure 3: Let $T^s \not\subset T^l$, $\tilde{\mathbf{y}}^{sI} \in T^l$ and $\tilde{\mathbf{y}}^{sI} \in \tilde{\mathcal{Y}}_P^l - \mathbb{R}^2_{>}$.

Check the locations of $\tilde{\mathbf{y}}^{s1}$ and $\tilde{\mathbf{y}}^{s2}$ with respect to $\tilde{\mathcal{Y}}_P^l$ by solving problem (2.10) with $\mathbf{y}^R = \tilde{\mathbf{y}}^{s1}$ and $\mathbf{y}^R = \tilde{\mathbf{y}}^{s2}$. If both conditions $\tilde{\mathbf{y}}^{si} \in \tilde{\mathcal{Y}}_P^l + \mathbb{R}^2_{>}$ for $i = 1, 2$ are satisfied, then proceed directly to Subprocedure Intersect. Figure 2.30 depicts this case. If Subprocedure Intersect concludes $\tilde{\mathcal{Y}}_P^l \cap \tilde{\mathcal{Y}}_P^s = \emptyset$, then $\tilde{\mathcal{Y}}_N^{sl} = \tilde{\mathcal{Y}}_P^l$ (Figure 2.30(a)). If Subprocedure Intersect concludes $\tilde{\mathcal{Y}}_P^l \cap \tilde{\mathcal{Y}}_P^s \neq \emptyset$, then each Pareto set is partially weakly dominated by the other one (Figure 2.30(b)) and the set $\tilde{\mathcal{Y}}_N^{sl}$ is reported accordingly with the intersection points. Otherwise, i.e., if $\tilde{\mathbf{y}}^{si} \in \tilde{\mathcal{Y}}_P^l + \mathbb{R}^2_{>}$ for $i \in \{1, 2\}$, then by Proposition 2.6.10 (i) the Pareto sets intersect and each Pareto set is partially weakly dominated by the other one. Subprocedure Intersect is used to find the intersection point.The case in which both points satisfy $\tilde{\mathbf{y}}^{si} \in \tilde{\mathcal{Y}}_P^l - \mathbb{R}^2_{>}$ for $i = 1, 2$ is addressed in Subprocedure 1.

When the dominance decision taken in Subprocedures 1-3 implies that two Pareto sets intersect or are likely to intersect, Subprocedure Intersect is called.

Subprocedure Intersect: This subprocedure is used to compute the intersection points between two Pareto sets. When this subprocedure is initiated, the existence of these points may be unknown. Each of the two Pareto sets, $\tilde{\mathcal{Y}}_P^s$ and $\tilde{\mathcal{Y}}_P^l$, being the input to this suprocedure, is available parametrically in the form $(f_1(\tilde{\mathbf{x}}(\lambda)), f_2(\tilde{\mathbf{x}}(\lambda)))$ for $\lambda \in [0,1]$, where $\tilde{\mathbf{x}}(\lambda)$ is the parametric optimal solution to problem (2.9) provided by the mpLCP method for each invariancy interval in $[0,1]$.

Since an intersection point may be located in any invariancy interval of each Pareto set, all pairs of invariancy intervals shall be checked. Let $[\lambda_1^l, \lambda_2^l] \subseteq [0,1]$ and $[\lambda_1^s, \lambda_2^s] \subseteq [0,1]$ be two invariancy intervals found for $\tilde{\mathcal{Y}}_P^l$ and $\tilde{\mathcal{Y}}_P^s$, respectively. For the pair $([\lambda_1^l; \lambda_2^l], [\lambda_1^s; \lambda_2^s])$, the following system of two polynomial equations is solved to identify the parameter values, $\lambda^l \in [\lambda_1^l; \lambda_2^l]$ and $\lambda^s \in [\lambda_1^s; \lambda_2^s]$, that determine the intersection point(s).

$$\begin{aligned} & f_i(\tilde{\mathbf{x}}(\lambda^s)) - f_i(\tilde{\mathbf{x}}(\lambda^l)) = 0 \quad i =, 1, 2 \\ & \lambda^s \in [\lambda_1^s, \lambda_2^s],\ \lambda^l \in [\lambda_1^l, \lambda_2^l] \end{aligned} \tag{2.26}$$

Let $(\hat{\lambda}^s, \hat{\lambda}^l)$ be a solution to system (2.26). Then the intersection point is given by $\tilde{\mathbf{y}}^{int} = \Big(f_1(\tilde{\mathbf{x}}(\hat{\lambda^\kappa})), f_2(\tilde{\mathbf{x}}(\hat{\lambda^\kappa}))\Big)$ for $\kappa = s, l$. Note that one invariancy interval may contain more than one intersection point or even infinitely many intersection points if the two curves (partially) coincide.

The polynomial equation solver *roots* in *MATLAB* is used to solve (2.26). While the solutions to (2.26) can be real or complex numbers, only the real solutions are reported. If a real solution to (2.26) is not found or all solutions found are complex numbers for the examined pair of invariancy intervals, we conclude that the Pareto sets do not intersect in that pair [119].

With the intersection points found from Subprocedure Intersect, the nondominated set $\tilde{\mathcal{Y}}_N^{sl}$ can be constructed. For example, consider the case depicted in

Figure 2.28 which has one intersection point and let $\tilde{\mathbf{y}}^{int}$ denote that point. Then $\tilde{\mathcal{Y}}_N^{sl} = \mathrm{Z}^s[\tilde{\mathbf{y}}^{s1}, \tilde{\mathbf{y}}^{int}] \cup \mathrm{Z}^l[\tilde{\mathbf{y}}^{int}, \tilde{\mathbf{y}}^{l2}]$.

2.7 Complete BB Algorithm and Numerical Experiments

Algorithm 1 The branch-and-bound algorithm for BOMIQPs.

1: INPUT: Problem $\mathcal{P}$
2: $S = \emptyset$, $\Lambda^{\ell} = \emptyset$, $\mathcal{X}_a = \emptyset$, $\mathcal{Y}_a = \emptyset$
3: Calculate $\mathcal{X}_E^0$ and $\mathcal{Y}_P^0$ (solve (2.8))
4: $\mathcal{X}_a \leftarrow \mathcal{X}_E^0$, $\mathcal{Y}_a \leftarrow \mathcal{Y}_P^0$
5: Compute $\tilde{\mathcal{X}}_E^0$ and $\tilde{\mathcal{Y}}_P^0$ (solve (2.2))
6: **if** $\mathbf{x}_i(\lambda) \in \mathbb{Z}\ \forall\ i = p+1, \cdots, n,\ \lambda \in [0,1]$ **then**
7:     $\mathcal{X}_E = \tilde{\mathcal{X}}_E^0$, $\mathcal{Y}_P = \tilde{\mathcal{Y}}_P^0$
8: **else**
9:     Add all invariancy intervals to Λ^0
10:     **while** $\Lambda^0 \neq \emptyset$ **do**
11:         Select an invariancy interval $[\lambda', \lambda''] \subseteq \Lambda^0$
12:         **if** $\mathbf{x}_i(\lambda) \in \mathbb{Z}\ \forall\ i = p+1, \cdots, n,\ \lambda \in [\lambda', \lambda'']$ **then**
13:             $\mathcal{Y}_a \leftarrow \mathbf{y}(\lambda)$, $\mathcal{Y}_a = N(\mathcal{Y}_a)$
14:             $\mathcal{X}_a \leftarrow \mathbf{x}(\lambda)$ for $\lambda \in [\lambda^L, \lambda^R] \subseteq [\lambda', \lambda'']$
15:         **else**
16:             Branch: Create node problems on each $x_i(\lambda) \in \tilde{\mathcal{X}}_E^0$ for $i = \{p+1, \cdots, n\}$ s.t. $x_i(\lambda) \notin \mathbb{Z}$
17:             Add new node problems to S
18:         **end if**
19:         Delete the invariancy interval
20:     **end while**
21:     **while** $S \neq \emptyset$ **do**
22:         Select problem $\tilde{\mathcal{P}}^s$ from S and compute $\tilde{\mathcal{X}}_E^s$, $\tilde{\mathcal{Y}}_P^s$
23:         **if** $\tilde{\mathcal{P}}^s$ infeasible **then**
24:             Delete $\tilde{\mathcal{P}}^s$ from S
25:             Goto line 21
26:         **else if** node fathomed due to rule (2.21) **then**
27:             Delete $\tilde{\mathcal{P}}^s$ from S
28:             Goto line 21
29:         **else if** $\mathbf{x}_i^s(\lambda) \in \mathbb{Z},\ \forall\ i = p+1, \cdots, n,\ \lambda \in [0,1]$ **then**
30:             $\mathcal{Y}_a \leftarrow \mathbf{y}^s(\lambda)$, $\mathcal{Y}_a = N(\mathcal{Y}_a)$
31:             $\mathcal{X}_a \leftarrow \mathbf{x}^s(\lambda)$ for $\lambda \in [\lambda^L, \lambda^R] \subseteq [0,1]$
32:             Delete $\tilde{\mathcal{P}}^s$ from S
33:             Goto line 21
34:         **else**
35:             Add all invariancy intervals to Λ^s
36:             **while** $\Lambda^s \neq \emptyset$ **do**
37:                 Select an invariancy interval from $[\lambda', \lambda''] \subseteq \Lambda^s$
38:                 **if** $\mathbf{x}_i^s(\lambda) \in \mathbb{Z}\ \forall\ i = p+1, \cdots, n,\ \lambda \in [\lambda', \lambda'']$ **then**
39:                     $\mathcal{Y}_a \leftarrow \mathbf{y}^s(\lambda)$, $\mathcal{Y}_a = N(\mathcal{Y}_a)$
40:                     $\mathcal{X}_a \leftarrow \mathbf{x}^s(\lambda)$ for $\lambda \in [\lambda^L, \lambda^R] \subseteq [\lambda', \lambda'']$
41:                 **else**
42:                     Branch: Create node problems on each $x_i^s(\lambda) \in \tilde{\mathcal{X}}_E^s$ for $i = \{p+1, \cdots, n\}$ s.t. $x_i^s(\lambda) \notin \mathbb{Z}$
43:                     Add new node problems to S
44:                 **end if**
45:                 Delete the invariancy interval
46:             **end while**
47:         **end if**
48:     **end while**
49: **end if**
50: OUTPUT: $\mathcal{X}_E = \mathcal{X}_a$ and $\mathcal{Y}_P = \mathcal{Y}_a$

The four modules presented in the prior sections are now integrated into a BB Algorithm 1 that computes efficient solutions and Pareto points to BOMIQP (2.1). The algorithm is presented in the form of pseudo-code, its properties and complexity are discussed, and numerical results are presented. In the Appendix the steps of Algorithm 1 are applied to an example BOMIQP.

2.7.1 Algorithm 1

The data of problem $\mathcal{P}$ is the input to Algorithm 1. Four sets are initiated: the set S contains all new node problems, the set Λ^ℓ contains all invariancy intervals associated with the solution to node ℓ problem, and the incumbent sets, $\mathcal{X}_a$ and $\mathcal{Y}_a$, in the decision and objective space respectively, as defined in Section 2.3 (line 2).

The BB algorithm begins by computing an initial set of efficient and Pareto solutions, $\mathcal{X}_E^0$ and $\mathcal{Y}_P^0$, as described in Section 2.3 (line 3).

At the root node 0, all integer variables are relaxed and problem $\tilde{\mathcal{P}}^0$ is solved for the parametric sets $\tilde{\mathcal{X}}_E^0$ and $\tilde{\mathcal{Y}}_P^0$ (line 5). If the set $\tilde{\mathcal{X}}_E^0$ is feasible to $\mathcal{P}$, then $\mathcal{P}$ has been solved: $\mathcal{X}_E = \tilde{\mathcal{X}}_E^0$ and $\mathcal{Y}_P = \tilde{\mathcal{Y}}_P^0$ (lines 6-7).

Else, all the invariancy intervals of $\tilde{\mathcal{P}}^0$ are added to Λ^0 and examined one at a time (lines 10-19). Consider an invariancy interval $[\lambda', \lambda'']$, and the associated solution $\mathbf{x}(\lambda)$ and its image $\mathbf{y}(\lambda)$ for $\lambda \in [\lambda', \lambda'']$. If this solution is feasible to $\mathcal{P}$, then $\mathbf{y}(\lambda)$ is added to the set $\mathcal{Y}_a$ that is updated in the set dominance module to satisfy $\mathcal{Y}_a = N(\mathcal{Y}_a)$. Due to this update, $\mathcal{Y}_a$ may contain $\mathbf{y}(\lambda)$ for $\lambda \in [\lambda^L, \lambda^R] \subseteq [\lambda', \lambda'']$. The set $\mathcal{X}_a$ is then updated accordingly to contain the preimage of the current $\mathcal{Y}_a$.

Otherwise, the branching is performed, i.e., node problems are created for all variables $\mathbf{x}_i(\lambda) \notin \mathbb{Z}$, $i = \{p+1, \cdots, n\}$ and added to the set S. If there are more than one candidate for a branching variable, one can select a variable based on the index of that variable choosing the one with the lowest (or the largest) index.

In the main step, a node problem, say node s problem from the set S, is solved for $\tilde{\mathcal{X}}_E^s$ and $\tilde{\mathcal{Y}}_P^s$ (line 22). The fathoming is applied next (lines 23-33). If the node is fathomed due to infeasibility or based on rule (2.21), then it is deleted from S. Rule (2.21) is implemented using Rule 1 and Rule 2 (see Section 2.5).

If the node is fathomed due to integer feasibility, then $\mathbf{y}^s(\lambda)$ for $\lambda \in [0,1]$ is added to the set $\mathcal{Y}_a$ that is updated in the set dominance module to satisfy $\mathcal{Y}_a = N(\mathcal{Y}_a)$. Due to this update, $\mathcal{Y}_a$ may contain $\mathbf{y}^s(\lambda)$ for $\lambda \in [\lambda^L, \lambda^R] \subseteq [0,1]$. The set $\mathcal{X}_a$ is then updated accordingly to contain the preimage of the current $\mathcal{Y}_a$ and the node is deleted from S.

If the node is not fathomed, all the invariancy intervals of $\tilde{\mathcal{P}}^s$ are added to Λ^s and examined one at a time (lines 35-45). Consider an invariancy interval $[\lambda', \lambda'']$, and the associated solution $\mathbf{x}^s(\lambda)$ and its image $\mathbf{y}^s(\lambda)$ for $\lambda \in [\lambda', \lambda'']$. If this solution is feasible to $\mathcal{P}$, then $\mathbf{y}^s(\lambda)$ is added to the set $\mathcal{Y}_a$ that is updated in the set dominance module to satisfy $\mathcal{Y}_a = N(\mathcal{Y}_a)$. Due to this update, $\mathcal{Y}_a$ may contain $\mathbf{y}^s(\lambda)$ for $\lambda \in [\lambda^L, \lambda^R] \subseteq [\lambda', \lambda'']$. The set $\mathcal{X}_a$ is then updated accordingly to contain the preimage of the current $\mathcal{Y}_a$.

Else, the branching is performed on fractional variables (line 42). The current node is deleted from S and the new node problems are added to S (lines 43-45). This process is repeated until all the node problems in S have been examined. At termination, the current sets $\mathcal{X}_a$ and $\mathcal{Y}_a$ yield the solution sets, $\mathcal{X}_E$ and $\mathcal{Y}_P$, to $\mathcal{P}$

respectively.

2.7.2 Properties of Algorithm 1

We now prove that all Pareto points to BOMIQP (2.1) are computed upon termination of Algorithm 1.

Theorem 2.7.1. *Upon termination, the BB Algorithm 1 returns the complete Pareto set for $\mathcal{P}$ (2.1), i.e., $\mathcal{Y}_P = \mathcal{Y}_a$.*

Proof. The proof is based on the properties of the three main modules of Algorithm 1, the branching, fathoming and set dominance, which are responsible for performing the steps of the BB scheme. During the execution of the algorithm, the incumbent sets $\mathcal{Y}_a$ and $\mathcal{X}_a$ are dynamically updated by adding elements that are feasible to problem $\mathcal{P}$. These elements are computed by solving the node problems that are created by the branching and fathoming modules.

At the initialization of the algorithm, an initial set of of efficient solutions, $\mathcal{X}_E^0$, and their images, $\mathcal{Y}_P^0$, are computed and stored in the incumbent sets $\mathcal{X}_a$ and $\mathcal{Y}_a$, respectively.

Note that the root node 0 is a special node of the BB tree. The following discussion addresses the main step of the algorithm in which an arbitrary node s (including node 0) of the BB tree is examined. At node s, problem $\tilde{\mathcal{P}}^s$ is solved and the fathoming module is applied.

If the node problem is infeasible or the set $\tilde{\mathcal{Y}}_P^s$ satisfies the condition in rule (2.21), then this node can be fathomed. This guarantees that the infeasible solutions or the solutions dominated by the incumbent set are excluded from the search.

If the set (a subset of) $\tilde{\mathcal{X}}_E^s$ is feasible to $\mathcal{P}$, then $\tilde{\mathcal{Y}}_P^s$ (a subset of $\tilde{\mathcal{Y}}_P^s$) is added to $\mathcal{Y}_a$ so that $\mathcal{Y}_a$ remains nondominated. $\tilde{\mathcal{X}}_E^s$ is added to $\mathcal{X}_a$ that is updated to be the preimage of the current $\mathcal{Y}_a$. This guarantees that no efficient solutions or the associated Pareto points are excluded from the search.

If the node is not fathomed, the branching module is applied to all invariancy intervals whose integer variables $x_i, i \in \{p+1, \cdots, n\}$, have fractional values. While this module creates more node problems, some of them are new and some others may have been obtained earlier. Only the new node problems are considered.

The branching and fathoming procedures are repeated until all the node problems have been examined. Since every node is generated through branching, no feasible solutions to $\mathcal{P}$ are eliminated during the search. Each time new elements are added to $\mathcal{Y}_a$, the set dominance module is executed to filter and discard the dominated points in $\mathcal{Y}_a$ and keep $\mathcal{Y}_a = N(\mathcal{Y}_a)$.

The three main modules guarantee that only the Pareto points to $\mathcal{P}$ remain in $\mathcal{Y}_a$ and that$\mathcal{Y}_a$ contains all Pareto points to $\mathcal{P}$. The associated set $\mathcal{X}_a$ is also updated accordingly to contain only the efficient solutions to $\mathcal{P}$. $\square$

The complexity of Algorithm 1 originates mainly from solving three types of single objective quadratic programs (QPs).

First, the mpLCP method employed at the nodes as the solver for $\tilde{\mathcal{P}}^s$ solves parametric QPs and primarily contributes to the complexity. As recognized in [3], the number of invariancy intervals is exponential in $(n+m)$, where n and m are the numbers of variables and constraints in $\tilde{\mathcal{P}}^s$ respectively, and a polynomial-time algorithm can never be developed to solve the node problem. The total number of node problems can be determined before the algorithm runs based on the number of slice problems, while the actual number of these problems being solved results from the branching module and remains unknown.

Second, mixed integer QPs are solved to compute an initial set of Pareto points and efficient solutions at the initialization of Algorithm 1, and it can take exponential time to solve them [97]. The number of these problems is decided by the user.

Third, quadratically constrained QPs (QCQPs) are solved in the fathoming and set dominance modules. The number of such problems is unknown at the beginning of the algorithm. QCQP is NP-hard [80].

Additionally, single-variable fractional optimization problems are solved in the branching module and systems of two polynomial equations with two-variables are solved in the set dominance module. Solving all these problems contributes to the total run time of Algorithm 1. Single-variable fractional optimization problem is NP-hard [98].

2.7.3 Preliminary numerical experiments

A variation of Algorithm 1 is implemented in the MATLAB programming language and preliminary numerical experiments are performed. The node problem solver, which has been implemented by Adelgren [2], is integrated with the algorithm.

In this rudimentary implementation, the initialization follows lines 3-4 of the pseudo-code, i.e., an initial set of Pareto points to the BOMIQP instance is computed. However, the four modules of the BB algorithm do not follow the pseudo-code because they have not been integrated with each other in the main step but are performed sequentially. The root node problem is first solved and its efficient solutions in all invariancy intervals are examined to identify the integer variables with fractional values. The naive branching method (see the Appendix) is used to create and solve all slice problems at the leaf nodes. The fathoming module is then executed. At the last stage, the set dominance procedure is applied to the incumbent set to delete the dominated points and find the complete Pareto set of the instance.

Since BOMIQPs have previously been unsolved, there are no instances in the literature to use, and there is no other algorithm to compare with. A set of randomly generated strictly convex BOMIQPs are generated and solved and the obtained numerical results are summarized in Table 2.1. The tests have been performed on a Lenovo Ideapad FLEX 4 with a 256 GB SSD storage, 6th Generation Intel Core i5-6200U, 2.30GHz, 2401 Mhz, 2 Cores, 4 Logical Processors and 8GB memory.

The results are reported in Table 2.1. In the columns from the first to the last one the following items are displayed: the instance number, dimension of the decision space, number of integer or binary variables, number of invariancy intervals (IIs) in the root node problem, total number of nodes in the BB tree without the root node, total number of invariancy intervals observed, and the CPU time for solving the instances. This time aggregates the total CPU time it takes for the rudimentary implementation to solve an instance.

Since the implementation of the mpLCP method is recent and still at the

Instance	n	$n-p$	No. of IIs in root node	Total no. of nodes	Total no. off IIs examined	Time (seconds)
1	2	1	2	2	3	101.59
2	2	1	3	2	3	81.07
3	3	1	2	2	4	96.31
4	3	1	2	2	4	110.64
5	3	1	3	2	4	90.37
6	3	1	3	3	9	115.39
7	3	1	3	5	16	276.53
8	3	2	2	4	6	150.64
9	3	2	4	4	7	196.42
10	3	2	4	8	15	398.39

Table 2.1: Summary of the results for BOMIQP instances solved with a rudimentary implementation of Algorithm 1

stage of being rudimentary, it is sensitive to adding branching constraints. Therefore only small-sized instances with 1 or 2 binary or integer variables out of 2 or 3 all variables have been solved. One can observe that as the number of integer variables increases, the total computation time also increases as expected. Because solving the node problems takes a big portion of the total computational time, as the number of node problems increases, the run time of the algorithm also increases. However, the total number of node problems seems to be related to the number of examined invariancy intervals.

2.8 Conclusion

We have developed the first algorithm to compute the complete Pareto set of BOMIQP (2.1). The algorithm computes two solution sets in parametric form: the Pareto set in the objective space and its pre-image, the efficient set, in the decision space. Since (2.1) is a global optimization problem, the algorithm follows the BB scheme. The branching module is integrated with the mpLCP method, a state-of-the-art algorithm employed as a node problem solver. The new practical fathoming rules introduced in the fathoming module are based on the bound sets established in a multiobjective setting. The new set domination module filters the incumbent set to become the Pareto set of (2.1) at termination of the algorithm.

The appealing and valuable feature of providing the exact solution sets to BOMIQPs is contrasted with exponential complexity of the algorithm. This complexity is reflected in a numerical study showing an increase in computational time with an increase of the number of integer variables as well as node problems even for BOMIQPs with 2 or 3 variables. This study is based on a rudimentary implementation of a variation of the pseudo-code given in Section 2.7.

This work immediately opens up several avenues for future research. While the complexity of the mpLCP method cannot be reduced, it is desirable to make its implementation more stable when solving the node problems along a branch of the BB tree and adding constraints. Adopting methods to solve fractional programs will enhance the performance of the branching module, while using a more robust solver for polynomial equations will improve the set dominance module. More numerical studies are needed to determine the tradeoff in the set dominance module between solving the polynomial equations to determine intersection points between two Pareto sets and solving single objective quadratic programs to examine their mutual location.

A reverse research direction is also possible since the set dominance procedure can be applicable to biobjective nonconvex programs that are decompsoed into biobjective convex subproblems such as the location problem in [73]. In general, since the proposed algorithm relies on solving four types of optimization problems as well as systems of polynomial equations, improvements in each of the five directions will affect not only the algorithm performance but will also advance polynomial, mixed-integer, integer, and fractional optimization.

Supporting Information

This section contains additional information in support of the methodology presented in the paper. We first describe a branching approach based on slice problems (2.3) associated with BOMIQP (2.1). In the second part we present the steps of Algorithm 1 on an example BOMIQP.

A Slice branching

Branching based on slice problems is a branching strategy that is used in the example problem. In this strategy all slice problems associated with the original BOMIQP are created. Since the integer variables are bounded from below and above, there is a finite number of combinations of the integer values these variables may assume so that the linear constraints hold, and therefore there is a finite number of slice problems that are the leaf node problems of the BB tree in Algorithm 1.

Since in this method all combinations of feasible values for integer variables are checked, this method of branching seems naive and is clearly not suitable for BOMIQPs with many variables. However, for instances with few variables the number of such combinations is small, which makes this method useful during algorithmic developments and testing, as indicated in [26]. In fact, we observe that for small instances the number of leaf nodes may be less than or equal to the number of nodes created in the branching module of Algorithm 1. We refer to this strategy as "slice branching". Since the example solved in the next section lends itself to slice branching, we include it in the description.

In the context of the pseudo-code of Algorithm 1, if the slice branching is used, the number of nodes in the set S is known. The node problems are solved and all their solutions are feasible to (2.1) because the constraint $x_i \in \mathbb{Z}$ for $i = p+1, \cdots, n$ is always satisfied. All these solutions can be passed to the set dominance module to compute the solution sets $\mathcal{Y}_p$ and $\mathcal{X}_E$. However, to reduce the computational effort in the set dominance module, we first pass these solutions to the fathoming module to check whether a node can be fathomed. If so, that node's solutions will be discarded and will not be passed to the dominance module.

B Example

Consider the following BOMIQP with one integer variable.

$$\begin{aligned} \mathcal{P}: \quad min \quad & \mathbf{f}(\mathbf{x}) = \left[f_1(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T Q_1 \mathbf{x} + \boldsymbol{p}_1^T \mathbf{x}, f_2(\mathbf{x}) \tfrac{1}{2}\mathbf{x}^T Q_2 \mathbf{x} + \boldsymbol{p}_2^T \mathbf{x} \right] \\ s.t. \quad & \mathbf{x} \in \mathcal{X} = \{\mathbf{x} \in \mathbb{R}^2 \times \mathbb{Z} : A\mathbf{x} \leqq \mathbf{b}, \mathbf{x} \geqq \mathbf{0}, x_2 \in \mathbb{Z}, x_2 \leq 4\}, \end{aligned} \tag{2.27}$$

where

$$Q_1 = \begin{bmatrix} 6 & -6 & 6 \\ -6 & 14 & -10 \\ 6 & -10 & 8 \end{bmatrix}, \; \boldsymbol{p}_1 = \begin{bmatrix} 0 \\ -3 \\ 0 \end{bmatrix}, \; Q_2 = \begin{bmatrix} 2 & -4 & 2 \\ -4 & 16 & -2 \\ 2 & -2 & 4 \end{bmatrix}, \; \boldsymbol{p}_2 = \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix},$$

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \text{ and } \mathbf{b} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}.$$

Initialization The set $\mathcal{Y}_P^0$ for $\lambda = 0, 0.1, 0.2, \cdots, 1$ where $|\mathcal{Y}_P^0| = 11$, is first computed. Figure 2.9 depicts the set $\mathcal{Y}_P^0$. The sets $\mathcal{Y}_a = \mathcal{Y}_P^0$ and $\mathcal{X}_a = \mathcal{X}_E^0$ are initialized.

Solving the root node problem The relaxed BOQP of (2.27), which is the root node problem of the BB tree, assumes the form:

$$\begin{aligned} \tilde{\mathcal{P}}^1: \quad min \quad & \mathbf{f}(\mathbf{x}) \\ s.t. \quad & \mathbf{x} \in \tilde{\mathcal{X}} = \tilde{\mathcal{X}}^1 = \{\mathbf{x} \in \mathbb{R}^3 : A\mathbf{x} \leqq \mathbf{b}, \mathbf{x} \geqq \mathbf{0}\}, \end{aligned} \tag{2.28}$$

The weighted-sum problem associated with (2.28) is formulated

$$\begin{aligned} \tilde{\mathcal{P}}^1(\lambda): \quad min \quad & \lambda f_1(\mathbf{x}) + (1-\lambda) f_2(\mathbf{x}) \\ s.t. \quad & \mathbf{x} \in \tilde{\mathcal{X}}^1 = \{\mathbf{x} \in \mathbb{R}^3 : A\mathbf{x} \leqq \mathbf{b}, \mathbf{x} \geqq \mathbf{0}\}, \\ & \lambda \in [0,1]. \end{aligned} \tag{2.29}$$

and solved with the mpLCP method that provides the optimal solution functions and the associated invariancy intervals. At optimality of (2.29), the parameter space is partitioned into three invariancy. These intervals and the optimal solution functions, which are also efficient solution functions to (2.28), are given in Table 2.2.

Branching Consider first the solution function in the first invariancy interval, $[0, 0.6747]$. To obtain the range of the values for the integer variable x_2, the following polynomial fractional programs need to be solved; $x_2^{min} = \min\{x_2 = \frac{8\lambda+3\lambda^2+1}{3(-6\lambda^2+2\lambda+5)} : \lambda \in [0, 0.6747]\}$ and $x_2^{max} = \max\{x_2 = \frac{8\lambda+3\lambda^2+1}{3(-6\lambda^2+2\lambda+5)} : \lambda \in [0, 0.6747]\}$. Using discretization of the intervals, $x_2^{min} = 0.5902$ and $x_2^{max} = 0.75$ are obtained and therefore $x_2 \in [0.5902, 0.75]$. Then $[\phi_2', \phi_2''] = [0, 1]$. With this range of x_2, two new node problems are created.

Table 2.2: Efficient solution functions for $\tilde{\mathcal{P}}^1$

$$\hat{\mathbf{x}}(\lambda) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{3\lambda^2-4\lambda+10}{-3\lambda^2+11\lambda+4} \\ x_2 = \frac{3\lambda^2+3\lambda+3}{-3\lambda^2+11\lambda+4} \\ x_3 = 0 \end{array} \text{ for } \lambda \in [0, 0.6747] \right\}$$

$$\hat{\mathbf{x}}(\lambda) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{4\lambda^3-\lambda-14\lambda^2+8}{-6\lambda^3+5\lambda+1} \\ x_2 = \frac{12\lambda^3-7\lambda^2-\lambda+5}{-6\lambda^3+5\lambda+1} \\ x_3 = \frac{3\lambda^3+34\lambda^2-8\lambda-11}{3(-6\lambda^3+5\lambda+1)} \end{array} \text{ for } \lambda \in [0.6747, 0.8182] \right\}$$

$$\hat{\mathbf{x}}(\lambda) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 0 \\ x_2 = \frac{8\lambda+3\lambda^2+1}{3(-6\lambda^2+2\lambda+5)} \\ x_3 = \frac{7\lambda^2+15\lambda-7}{3(-6\lambda^2+2\lambda+5)} \end{array} \text{ for } \lambda \in [0.8182, 1] \right\}$$

The first one, $\tilde{\mathcal{P}}^2$, with the feasible set $\tilde{\mathcal{X}}^2 = \{\mathbf{x} \in \mathbb{R}^3 : A\mathbf{x} \leqq \mathbf{b}, x_2 \leq 0, \mathbf{x} \geqq \mathbf{0}\}$, and the second one, $\tilde{\mathcal{P}}^3$, with the feasible set $\tilde{\mathcal{X}}^3 = \{\mathbf{x} \in \mathbb{R}^3 : A\mathbf{x} \leqq \mathbf{b}, x_2 \geq 1, \mathbf{x} \geqq \mathbf{0}\}$. Both problems are solved with the mpLCP method. The solution to $\tilde{\mathcal{P}}^2$ is given in Table 2.3 while the solution to $\tilde{\mathcal{P}}^3$ is given in Table 2.4. In Table 2.3, $x_2 = 0$ and

Table 2.3: Efficient solution functions for $\tilde{\mathcal{P}}^2$

$$\hat{\mathbf{x}}(\lambda) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{1-\lambda}{2\lambda+1} \\ x_2 = 0 \\ x_3 = 0 \end{array} \text{ for } \lambda \in [0, 1] \right\}$$

hence the entire efficient set to $\tilde{\mathcal{P}}^2$ is feasible to $\mathcal{P}$. Based on the feasibility, this node is fathomed and the associated Pareto points are saved in the set $\mathcal{Y}_a$. Then the set dominance procedure is applied to satisfy the condition $\mathcal{Y}_a = N(\mathcal{Y}_a)$. Note also that this node is a leaf node and branching cannot continue.

The solutions to $\tilde{\mathcal{P}}^3$ in Table 2.4 are now examined. In the first two invariancy intervals $x_2 = 1$ and hence the associated efficient solutions to $\tilde{\mathcal{P}}^3$ are feasible to

Table 2.4: Efficient solution functions for $\tilde{\mathcal{P}}^3$

$$\hat{\mathbf{x}}(\lambda) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{3}{2\lambda+1} \\ x_2 = 1 \\ x_3 = 0 \end{array} \quad \text{for } \lambda \in [0, 0.6] \right\}$$

$$\hat{\mathbf{x}}(\lambda) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{3}{2\lambda+1} - (5\lambda - 3) \\ x_2 = 1 \\ x_3 = 5\lambda - 3 \end{array} \quad \text{for } \lambda \in [0.6, 0.7974] \right\}$$

$$\hat{\mathbf{x}}(\lambda) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{4\lambda^3 - 14\lambda^2 - \lambda + 8}{-6\lambda^3 + 5\lambda + 1} \\ x_2 = \frac{12\lambda^3 - 7\lambda^2 - \lambda + 5}{-6\lambda^3 + 5\lambda + 1} \\ x_3 = \frac{3\lambda^3 - 34\lambda^2 - 8\lambda + 11}{3(-6\lambda^3 + 5\lambda + 1)} \end{array} \quad \text{for } \lambda \in [0.7974, 0.8182] \right\}$$

$$\hat{\mathbf{x}}(\lambda) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 0 \\ x_2 = \frac{8\lambda^2 + 3\lambda + 1}{3(-6\lambda^2 + 2\lambda + 5)} \\ x_3 = \frac{7\lambda^2 + 15\lambda - 7}{3(-6\lambda^2 + 2\lambda + 5)} \end{array} \quad \text{for } \lambda \in [0.8182, 1] \right\}$$

problem $\mathcal{P}$. These efficient solutions and the associated Pareto outcomes are saved in sets $\mathcal{X}_E$ and $\mathcal{Y}_a$, respectively. Then the set dominance procedure is applied to satisfy the condition $\mathcal{Y}_a = N(\mathcal{Y}_a)$. Branching is applied to the third and fourth invariancy intervals and is summarized in Table 2.5.

Table 2.5: New branching nodes for invariancy intervals 3 and 4 in $\tilde{\mathcal{P}}^3$

	Branching process	
Invariancy intervals $[\lambda', \lambda'']$	$[0.7974, 0.8182]$	$[0.8182, 1]$
x_2	$\frac{12\lambda^3 - 7\lambda^2 - \lambda + 5}{-6\lambda^3 + 5\lambda + 1}$	$\frac{8\lambda^2 + 3\lambda + 1}{3(-6\lambda^2 + 2\lambda + 5)}$
x_2^{min}	1	1.1858
x_2^{max}	1.1651	3.9861
New child node problem constraints	$x_2 \leq 1,\ x_2 \geq 2$	$x_2 \leq 1,\ x_2 \geq 2$ $x_2 \leq 2,\ x_2 \geq 3$ $x_2 \leq 3,\ x_2 \geq 4$

In Table 2.5, the second row shows the invariancy intervals $[\lambda', \lambda'']$; the third row shows the solution function x_2; the third and fourth rows show the minimum and maximum values assumed by $x_2(\lambda)$ in each invariancy interval; the last row shows the pairs of branching constraints that are generated for $x_2 \in [x_2^{min}, x_2^{max}]$. There are eight child node problems but two pairs have identical constraints, which makes six new node problems.

Going back to $\tilde{\mathcal{P}}^1$, the first invariancy interval in Table 2.2 has been explored. The second and third invariancy intervals, $[0.6747, 0.8182]$ and $[0.8182, 1]$, are now examined and a summary is given in Table 2.6.

Table 2.6: New branching nodes for invariancy intervals 2 and 3 in $\tilde{\mathcal{P}}^1$

	Branching process	
Invariancy intervals $[\lambda', \lambda'']$	$[0.6747, 0.8182]$	$[0.8182, 1]$
x_2	$\frac{12\lambda^3-7\lambda^2-\lambda+5}{-6\lambda^3+5\lambda+1}$	$\frac{8\lambda+3\lambda^2+1}{3(-6\lambda^2+2\lambda+5)}$
x_2^{min}	0.6355	1.0986
x_2^{max}	1.121	3.918
New child node problem constraints	$x_2 \leq 0,\ x_2 \geq 1$ $x_2 \leq 1,\ x_2 \geq 2$	$x_2 \leq 1,\ x_2 \geq 2$ $x_2 \leq 2,\ x_2 \geq 3$ $x_2 \leq 3,\ x_2 \geq 4$

In this table there are ten child node problems. Among them the node problems associated with the constraints $x_2 \leq 0$ and $x_2 \geq 1$ for $\lambda \in [0.6747, 0.8182]$ have already been added to the BB tree as problems $\tilde{\mathcal{P}}^2$ and $\tilde{\mathcal{P}}^3$. The remaining eight problems are identical to those in Table 2.5. Based on Tables 2.5 and 2.6 and to avoid duplication, the following six new node problems are formulated: $\tilde{\mathcal{P}}^4$ with the feasible set $\tilde{\mathcal{X}}^4 = \{\mathbf{x} \in \mathbb{R}^3 : A\mathbf{x} \leqq \mathbf{b}, x_2 \leq 1, \mathbf{x} \geqq \mathbf{0}\}$; $\tilde{\mathcal{P}}^5$ with $\tilde{\mathcal{X}}^5 = \{\mathbf{x} \in \mathbb{R}^3 : A\mathbf{x} \leqq \mathbf{b}, x_2 \geq 2, \mathbf{x} \geqq \mathbf{0}\}$; $\tilde{\mathcal{P}}^6$ with $\tilde{\mathcal{X}}^6 = \{\mathbf{x} \in \mathbb{R}^3 : A\mathbf{x} \leqq \mathbf{b}, x_2 \leq 2, \mathbf{x} \geqq \mathbf{0}\}$; $\tilde{\mathcal{P}}^7$ with $\tilde{\mathcal{X}}^7 = \{\mathbf{x} \in \mathbb{R}^3 : A\mathbf{x} \leqq \mathbf{b}, x_2 \geq 3, \mathbf{x} \geqq \mathbf{0}\}$; $\tilde{\mathcal{P}}^8$ with $\tilde{\mathcal{X}}^8 = \{\mathbf{x} \in \mathbb{R}^3 : A\mathbf{x} \leqq \mathbf{b}, x_2 \leq 3, \mathbf{x} \geqq \mathbf{0}\}$; and $\tilde{\mathcal{P}}^9$ with $\tilde{\mathcal{X}}^9 = \{\mathbf{x} \in \mathbb{R}^3 : A\mathbf{x} \leqq \mathbf{b}, x_2 \geq 4, \mathbf{x} \geqq \mathbf{0}\}$.

Slice branching The slice branching strategy is now illustrated on Example (2.27). Because the range of feasible values of the integer variable x_2 is $[0,4]$, five node problems are created by fixing x_2 at each integer value while the linear constraints remain feasible: $\tilde{P}^1$ with the feasible set $\tilde{\mathcal{X}}^1 = \{\mathbf{x} \in \mathbb{R}^3 : A\mathbf{x} \leqq \mathbf{b}, x_2 = 0, \mathbf{x} \geq \mathbf{0}\}$; $\tilde{P}^2$ with $\tilde{\mathcal{X}}^2 = \{\mathbf{x} \in \mathbb{R}^3 : A\mathbf{x} \leqq \mathbf{b}, x_2 = 1, \mathbf{x} \geq \mathbf{0}\}$; $\tilde{P}^3$ with $\tilde{\mathcal{X}}^3 = \{\mathbf{x} \in \mathbb{R}^3 : A\mathbf{x} \leqq \mathbf{b}, x_2 = 2, \mathbf{x} \geq \mathbf{0}\}$; $\tilde{P}^4$ with $\tilde{\mathcal{X}}^4 = \{\mathbf{x} \in \mathbb{R}^3 : A\mathbf{x} \leqq \mathbf{b}, x_2 = 3, \mathbf{x} \geq \mathbf{0}\}$; and $\tilde{P}^5$ with $\tilde{\mathcal{X}}^5 = \{\mathbf{x} \in \mathbb{R}^3 : A\mathbf{x} \leqq \mathbf{b}, x_2 = 4, \mathbf{x} \geq \mathbf{0}\}$. Note that the superscripts denoting these node problems are the same as those used for the problems emerging from the first branching method but the node problems are different. The solutions to these problems are given Table 2.7, 2.8, 2.9, 2.10 and 2.11, respectively.

Table 2.7: Efficient solution functions for $\tilde{P}^1$

$$\hat{\mathbf{x}}(\lambda) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{1-\lambda}{2\lambda+1} \\ x_2 = 0 \\ x_3 = 0 \end{array} \quad \text{for } \lambda \in [0,1] \right\}$$

Table 2.8: Efficient solution functions for $\tilde{P}^2$

$$\hat{\mathbf{x}}(\lambda) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{3}{2\lambda+1} \\ x_2 = 1 \\ x_3 = 0 \end{array} \quad \text{for } \lambda \in [0,0.6] \right\}$$

$$\hat{\mathbf{x}}(\lambda) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{3}{2\lambda+1} - 5\lambda + 3 \\ x_2 = 1 \\ x_3 = 5\lambda - 3 \end{array} \quad \text{for } \lambda \in [0.6,1] \right\}$$

Figure 2.10 depicts the points in $\mathcal{Y}_P^0$ and the Pareto points of the five slice problems. The sets $\mathcal{Y}_P^0$ and $\tilde{\mathcal{Y}}_P^\ell, \ell = 1, \cdots, 5$ can be passed to the set dominance module to compute the final solution sets. However, we first pass these sets to the fathoming module to check whether a node can be fathomed and its solution set can

Table 2.9: Efficient solution functions for $\tilde{P}^3$

$$\hat{\mathbf{x}}(\lambda) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 3 \\ x_2 = 2 \\ x_3 = 0 \end{array} \text{ for } \lambda \in [0, 0.4] \right\}$$

$$\hat{\mathbf{x}}(\lambda) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{\lambda+5}{2\lambda+1} \\ x_2 = 2 \\ x_3 = 0 \end{array} \text{ for } \lambda \in [0.4, 0.5] \right\}$$

$$\hat{\mathbf{x}}(\lambda) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{-16\lambda^2+\lambda+9}{2\lambda+1} \\ x_2 = 2 \\ x_3 = 8\lambda - 4 \end{array} \text{ for } \lambda \in [0.5, 0.7819] \right\}$$

$$\hat{\mathbf{x}}(\lambda) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 0 \\ x_2 = 2 \\ x_3 = \frac{9\lambda+1}{2(\lambda+1)} \end{array} \text{ for } \lambda \in [0.7819, 1] \right\}$$

be discarded.

Fathoming The fathoming module is now applied to all the node problems obtained from the slice branching. Since the solutions to all the node problems are feasible to $\mathcal{P}$, fathoming based on bound sets with rule 2.21 is used and the procedure presented in Figure 5 is followed.

We have $\mathcal{Y}_a = \bar{\mathcal{Y}}_a = \mathcal{Y}_P^0$. First, consider $\tilde{\mathcal{Y}}_P^1$, the Pareto set associated with $\tilde{P}^1$. The ideal point $\tilde{\mathbf{y}}^{1I} = (0, -0.25)$ for $\tilde{P}^1$ is computed and the triangle T^1 with vertices $\tilde{\mathbf{y}}^{11} = (0, 0)$, $\tilde{\mathbf{y}}^{12} = (0.75, -0.25)$, and $\tilde{\mathbf{y}}^{1I} = (0, -0.25)$ is constructed. Here $\tilde{\mathbf{y}}^{11}, \tilde{\mathbf{y}}^{12}$ are the end points of $\tilde{\mathcal{Y}}_P^1$ for $\lambda = 1$ and $\lambda = 0$, respectively. Then the set $\bar{\mathcal{Y}}_a^1 = \bar{\mathcal{Y}}_a \cap (T^1 - \mathbb{R}^2_{\geqq})$ is obtained. Now Rule 1 (cf. Section 2.5) is checked. Since there does not exist a point $\mathbf{y}^i \in \bar{\mathcal{Y}}_a^1$ such that $\mathbf{y}^i \in \{\tilde{\mathbf{y}}^{1I}\}^{\leqq}$, Rule 1 does not hold. Then

Table 2.10: Efficient solution functions for $\tilde{P}^4$

$$\hat{\mathbf{x}}(\lambda) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 3 \\ x_2 = 3 \\ x_3 = \frac{15\lambda+1}{2(\lambda+1)} \end{array} \quad \text{for } \lambda \in [0, 0.5947] \right\}$$

$$\hat{\mathbf{x}}(\lambda) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{-34\lambda^2+\lambda+18}{2\lambda 1+1} \\ x_2 = 3 \\ x_3 = 17\lambda - 7 \end{array} \quad \text{for } \lambda \in [0.5947, 0.7425] \right\}$$

$$\hat{\mathbf{x}}(\lambda) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 0 \\ x_2 = 3 \\ x_3 = \frac{21\lambda+4}{2(\lambda+1)} \end{array} \quad \text{for } \lambda \in [0.7425, 1] \right\}$$

the conditions $\bar{\mathcal{Y}}_a^1 \cap C^{1W} = \emptyset$ and $\mathcal{Y}_a^1 \cap C^{1S} = \emptyset$ are checked. We have $\mathcal{Y}_a^1 \cap C^{1S} = \emptyset$. Therefore, node 1 cannot be fathomed.

Similarly, the triangles T^ℓ for $\tilde{\mathcal{Y}}_P^\ell$ (Pareto set of $\tilde{P}^\ell$), and consequently the sets $\bar{\mathcal{Y}}_a^\ell$ for $\ell = 2, \cdots, 5$, are constructed. Rule 1 does not hold for nodes 2 and 3, but $\mathcal{Y}_a^\ell \cap C^{\ell S} = \emptyset$ for $\ell = 2, 3$ and nodes 2 and 3 cannot be fathomed. However, Rule 1 holds for nodes 4 and 5 and they are fathomed, i.e., dropped from the BB tree. We now proceed with nodes 1,2, and 3 to the set dominance module.

Set Dominance The goal of this module is to add the sets $\tilde{\mathcal{Y}}_P^\ell, \ell = 1, \cdots, 3$, to $\mathcal{Y}_a$ and have $\mathcal{Y}_a = N(\mathcal{Y}_a)$. Due to the structure of the set $\mathcal{Y}_P^0$, which is depicted in Figure 2.9, this module is executed as follows. Initially, $\mathcal{Y}_a = \mathcal{Y}_P^0$ and the associated $\mathcal{X}_a = \mathcal{X}_a^0$. Based on Figure 2.9, three clusters of the Pareto points in $\mathcal{Y}_P^0$ can be identified. These clusters are associated with the integer solutions $x_2 = 0$ (southeast cluster), $x_2 = 1$ (middle cluster) and $x_2 = 3$ (northwest cluster). Let $\mathcal{Y}_P^{SE}, \mathcal{Y}_P^M, \mathcal{Y}_P^{NW} \subset \mathcal{Y}_P^0$ denote the sets of points that are contained in the southeast, middle and northwest cluster,

Table 2.11: Efficient solution functions for $\tilde{P}^5$

$$\hat{\mathbf{x}}(\lambda) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 3 \\ x_2 = 4 \\ x_3 = 0 \end{array} \text{ for } \lambda \in [0, 0.5714] \right\}$$

$$\hat{\mathbf{x}}(\lambda) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{-28*\lambda^2+\lambda+15}{2\lambda+1} \\ x_2 = 4 \\ x_3 = 14\lambda - 6 \end{array} \quad \text{ for } \lambda \in [0.5714, 0.75] \right\}$$

$$\hat{\mathbf{x}}(\lambda) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 0 \\ x_2 = 4 \\ x_3 = \frac{17\lambda+3}{2(\lambda+1)} \end{array} \quad \text{ for } \lambda \in [0.75, 1] \right\}$$

respectively. We have $\mathcal{Y}_a = \mathcal{Y}_P^0 = \mathcal{Y}_P^{SE} \cup \mathcal{Y}_P^M \cup \mathcal{Y}_P^{NW}$.

Recall that triangles T^ℓ for $\ell = 1, 2, 3$ have already been constructed. To add $\tilde{\mathcal{Y}}_P^1$ to $\mathcal{Y}_a$ and have $\mathcal{Y}_a = N(\mathcal{Y}_a)$, the location of $\tilde{\mathcal{Y}}_P^1$ and T^1 with respect to the current $\mathcal{Y}_a$ is examined. By solving $\tilde{\mathcal{P}}(\mathbf{y}^R)$ (2.10) with $\mathbf{y}^R = \mathbf{y}$ for all $\mathbf{y} \in \mathcal{Y}_P^{SE}$, we obtain that all $\mathbf{y} \in \mathcal{Y}_P^{SE}$ are in $\tilde{\mathcal{Y}}_P^1$. By solving $\tilde{\mathcal{P}}(\mathbf{y}^R)$ (2.10) with $\mathbf{y}^R = \mathbf{y}$ for all $\mathbf{y} \in \mathcal{Y}_P^M$, and later for all $\mathbf{y} \in \mathcal{Y}_P^{NW}$, we discover that the points in $\mathcal{Y}_P^M, \mathcal{Y}_P^{NW}$, and $\tilde{\mathcal{Y}}_P^1$ are nondominated. Therefore, $\mathcal{Y}_a$ is updated as $\mathcal{Y}_a = \mathcal{Y}_P^M \cup \mathcal{Y}_P^{NW} \cup \tilde{\mathcal{Y}}_P^1$, and the set $\mathcal{X}_a$ is updated accordingly.

To add $\tilde{\mathcal{Y}}_P^2$ to $\mathcal{Y}_a$ and have $\mathcal{Y}_a = N(\mathcal{Y}_a)$, the location of $\tilde{\mathcal{Y}}_P^2$ and T^2 with respect to the current $\mathcal{Y}_a$ is examined. By solving $\tilde{\mathcal{P}}(\mathbf{y}^R)$ (2.10) with $\mathbf{y}^R = \mathbf{y}$ for all $\mathbf{y} \in \mathcal{Y}_P^M$, we obtain that all $\mathbf{y} \in \mathcal{Y}_P^M$ are in $\tilde{\mathcal{Y}}_P^2$. Because $\tilde{\mathcal{Y}}_P^1 \subset \mathcal{Y}_a$, the mutual location of T^1 and T^2 is checked to obtain $T^1 \cap T^2 = \emptyset$. Since $\tilde{\mathbf{y}}^{11} \in C^{2s}$, based on Proposition 2.6.9(ii), $\tilde{\mathcal{Y}}_P^1 \leq_{\mathfrak{p}} \tilde{\mathcal{Y}}_P^2$. To find the nondominated set resulting from $\tilde{\mathcal{Y}}_P^1$ and $\tilde{\mathcal{Y}}_P^2$, $\mathcal{P}(\tilde{\mathbf{y}}_1^{11})$ (2.12) is solved using the first coordinate of the northwest end

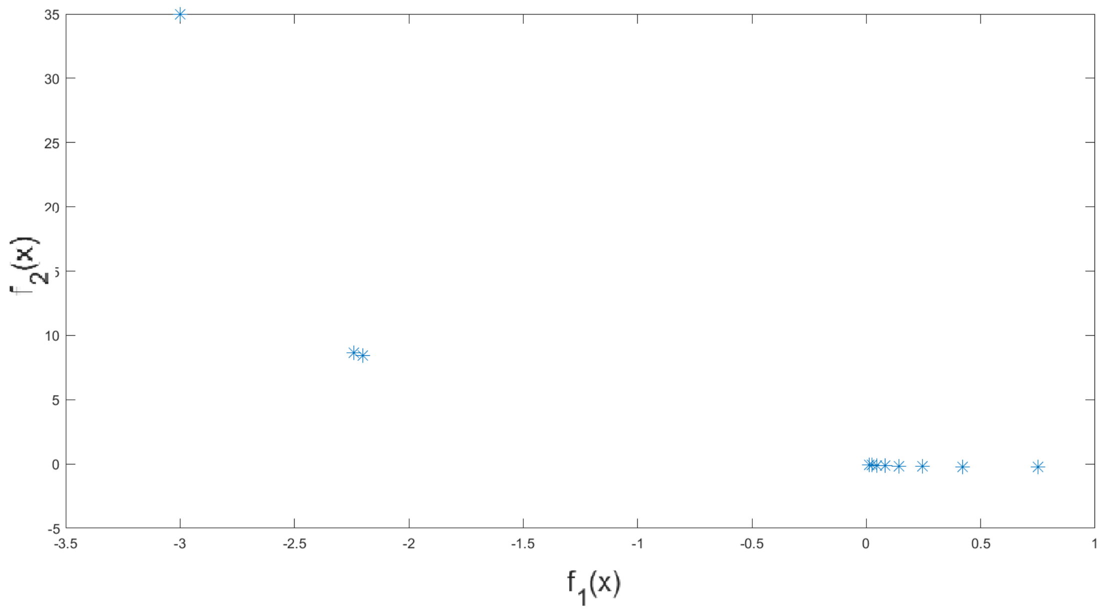


Figure 2.9: $\mathcal{Y}_P^0$

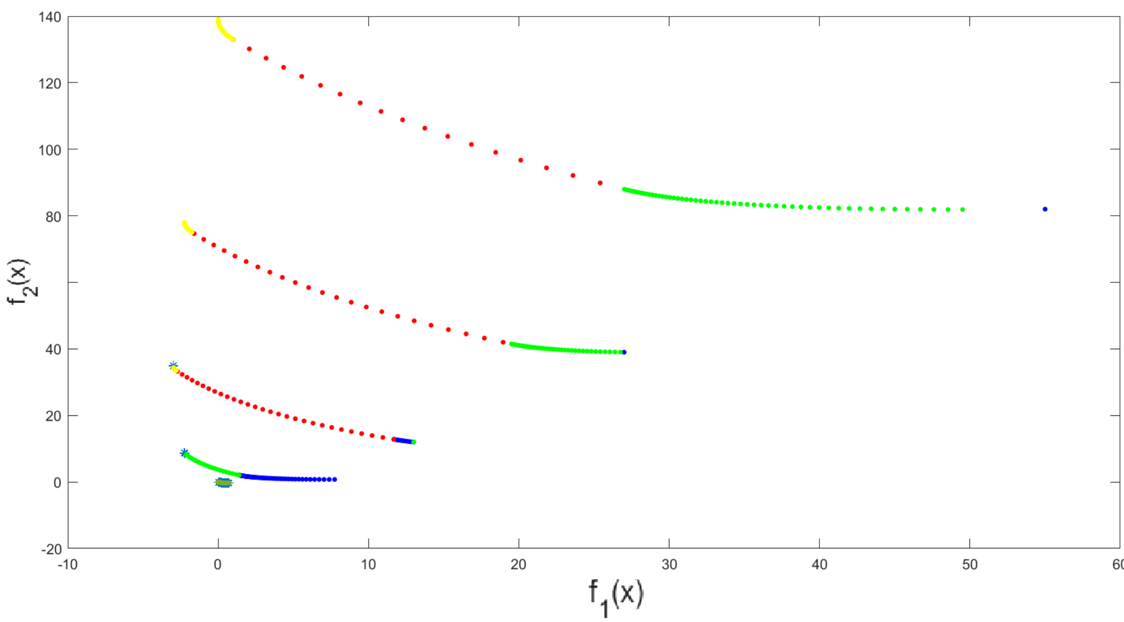


Figure 2.10: $\mathcal{Y}_P^0$ and $\tilde{\mathcal{Y}}_P^\ell, \ell = 1, \cdots, 5$ of slice problems

Figure 2.11: Pareto points

point, $\tilde{\mathbf{y}}^{11}$, of $\tilde{\mathcal{Y}}_P^1$ obtained for $\lambda = 1$, and the point $\hat{\mathbf{y}}(\tilde{\mathbf{y}}_1^{11}) = (0, 3.7008) \in \tilde{\mathcal{Y}}_P^2$ is obtained. This point is dominated by the northwest end point, $\tilde{\mathbf{y}}^{11}$, of $\tilde{\mathcal{Y}}_P^1$. Then $\tilde{\mathcal{Y}}_N^{12} = N(\tilde{\mathcal{Y}}_P^1 \cup \tilde{\mathcal{Y}}_P^2) = \tilde{\mathcal{Y}}_P^1 \cup \mathrm{Z}^2[\tilde{\mathbf{y}}^{21}, \hat{\mathbf{y}}(\tilde{\mathbf{y}}_1^{11}))$ is obtained, where $\tilde{\mathbf{y}}^{21}$ is the northwest end point of $\tilde{\mathcal{Y}}_P^2$ with $\lambda = 1$ and $\mathrm{Z}^2[\tilde{\mathbf{y}}^{21}, \hat{\mathbf{y}}(\tilde{\mathbf{y}}_1^{11}))$ is a subset of $\tilde{\mathcal{Y}}_P^2$. $\mathcal{Y}_a$ is updated as $\mathcal{Y}_a = \mathcal{Y}_P^{NW} \cup \tilde{\mathcal{Y}}_N^{12}$. To update $\mathcal{X}_a$, by Proposition2.2.11, the parameter $\lambda = 0.85$ is calculated. This value of λ allows to calculate the efficient solution $\mathbf{x}(\lambda = 0.85)$ for $\tilde{P}^2$ such that $\mathbf{f}(\mathbf{x}(\lambda = 0.85)) = \hat{\mathbf{y}}(\tilde{\mathbf{y}}_1^{11}) = (0, 3.7008)$. This solution is used to update $\mathcal{X}_a$.

To add $\tilde{\mathcal{Y}}_P^3$ to $\mathcal{Y}_a$ and have $\mathcal{Y}_a = N(\mathcal{Y}_a)$, the location of $\tilde{\mathcal{Y}}_P^3$ and triangle T^3 with respect to the current $\mathcal{Y}_a$ is now examined. By solving $\tilde{\mathcal{P}}(\mathbf{y}^R)$ (2.10) with $\mathbf{y}^R = \mathbf{y}$ for all $\mathbf{y} \in \mathcal{Y}_P^{NW}$, we obtain that all $\mathbf{y} \in \mathcal{Y}_P^{NW}$ are in $\tilde{\mathcal{Y}}_P^3$. A similar processes to that at node 2 follows. Using the first coordinate of the northwest end point, $\tilde{\mathbf{y}}_1^{21}$, of $\tilde{\mathcal{Y}}_P^2$, $\mathcal{P}(\tilde{\mathbf{y}}_1^{21})$ (2.12) is solved to obtain the point $\hat{\mathbf{y}}(\tilde{\mathbf{y}}_1^{21}) = (-2.25, 31.9457) \in \tilde{\mathcal{Y}}_P^3$. Then $\tilde{\mathcal{Y}}_N^{123} = N(\tilde{\mathcal{Y}}_N^{12} \cup \tilde{\mathcal{Y}}_P^3) = \tilde{\mathcal{Y}}_N^{12} \cup \mathrm{Z}^3[\tilde{\mathbf{y}}^{31}, \hat{\mathbf{y}}(\tilde{\mathbf{y}}_1^{21}))$ is obtained, where $\tilde{\mathbf{y}}^{31}$ is the northwest end point of $\tilde{\mathcal{Y}}_P^3$ with $\lambda = 1$ and $\mathrm{Z}^3[\tilde{\mathbf{y}}^{31}, \hat{\mathbf{y}}(\tilde{\mathbf{y}}_1^{21}))$ is a subset of $\tilde{\mathcal{Y}}_P^3$. The incumbent set is updated as $\mathcal{Y}_a = \tilde{\mathcal{Y}}_N^{123}$. To update $\mathcal{X}_a$, by Proposition2.2.11, the parameter $\lambda = 0.7317$ is calculated and used to identify the efficient solutions whose images are in the updated $\mathcal{Y}_a$.

Since all the nodes have been examined, $\mathcal{Y}_P = \tilde{\mathcal{Y}}_N^{123}$ and $\mathcal{X}_E = \mathcal{X}_a$. Table 2.12 contains the efficient solutions making the set $\mathcal{X}_E$ and Figure 2.12 depicts the Pareto set, $\mathcal{Y}_P$ for example (2.27). Note that in $\mathcal{X}_E$, only one invariancy interval comes from each $\tilde{P}^1$ and $\tilde{P}^2$, while two invariancy intervals come from $\tilde{P}^3$. As expected, the Pareto set is a disconnected nonconvex curve consisting of convex subcurves. Two of the three subcurves are neither open nor closed.

Table 2.12: Efficient solution functions of BOMIQP (2.27)

$$\hat{\mathbf{x}}(\lambda) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{1-\lambda}{2\lambda+1} \\ x_2 = 0 \\ x_3 = 0 \end{array} \quad \text{for } \lambda \in [0,1] \text{ in } \tilde{P}^1 \right\}$$

$$\hat{\mathbf{x}}(\lambda) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{3}{2\lambda+1} - 5\lambda + 3 \\ x_2 = 1 \\ x_3 = 5\lambda - 3 \end{array} \quad \text{for } \lambda \in (0.85, 1] \text{ in } \tilde{P}^2 \right\}$$

$$\hat{\mathbf{x}}(\lambda) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{-16\lambda^2+\lambda+9}{2\lambda+1} \\ x_2 = 2 \\ x_3 = 8\lambda 1 - 4 \end{array} \quad \text{for } \lambda \in (0.7317, 0.7819] \text{ in } \tilde{P}^3 \right\}$$

$$\hat{\mathbf{x}}(\lambda) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 0 \\ x_2 = 2 \\ x_3 = \frac{9\lambda+1}{2(\lambda 1+1)} \end{array} \quad \text{for } \lambda \in [0.7819, 1] \text{ in } \tilde{P}^3 \right\}$$

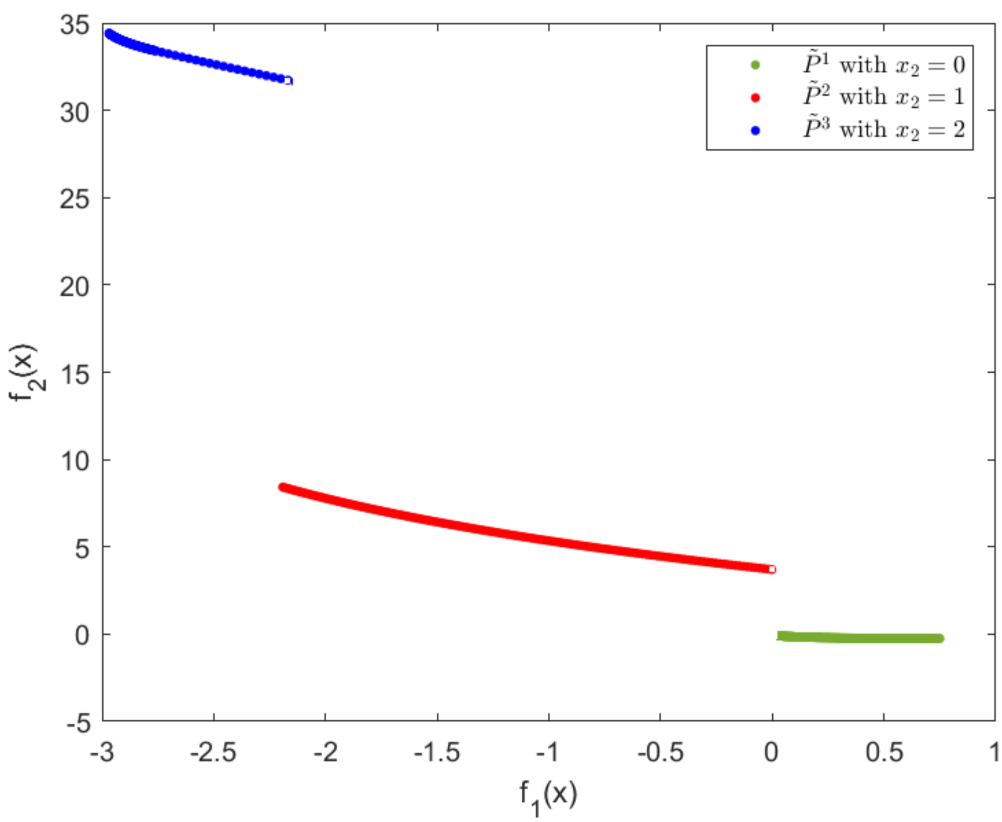


Figure 2.12: Pareto set, $\mathcal{Y}_P$, of BOMIQP (2.27)

C Figures: Fathoming Rules

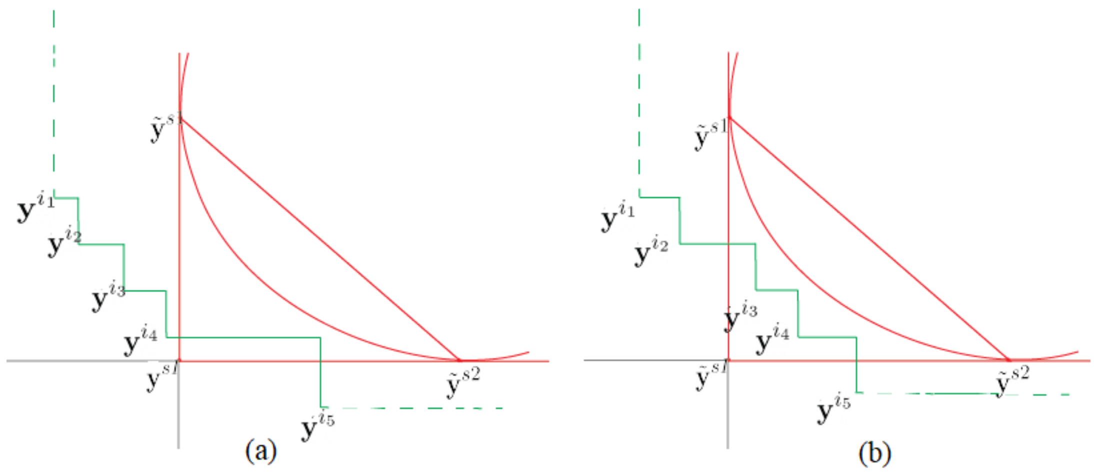
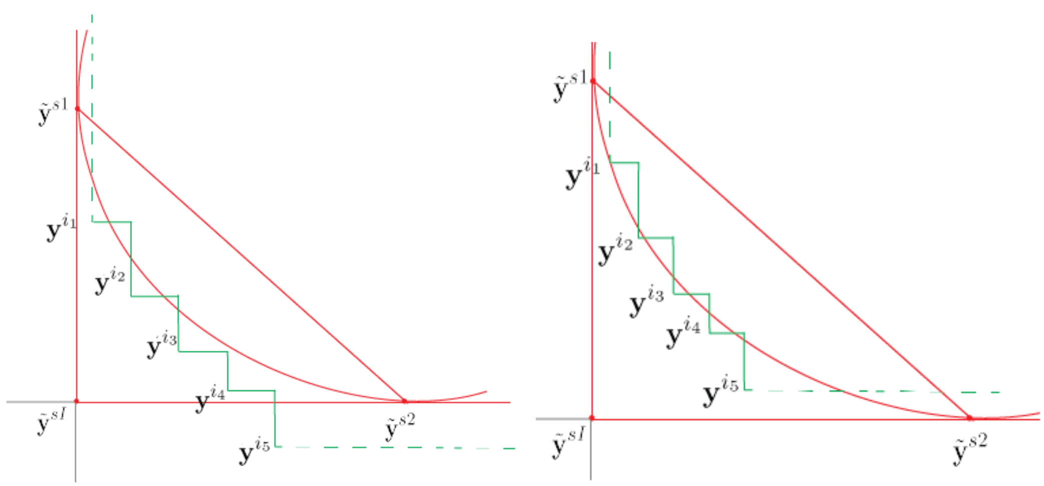
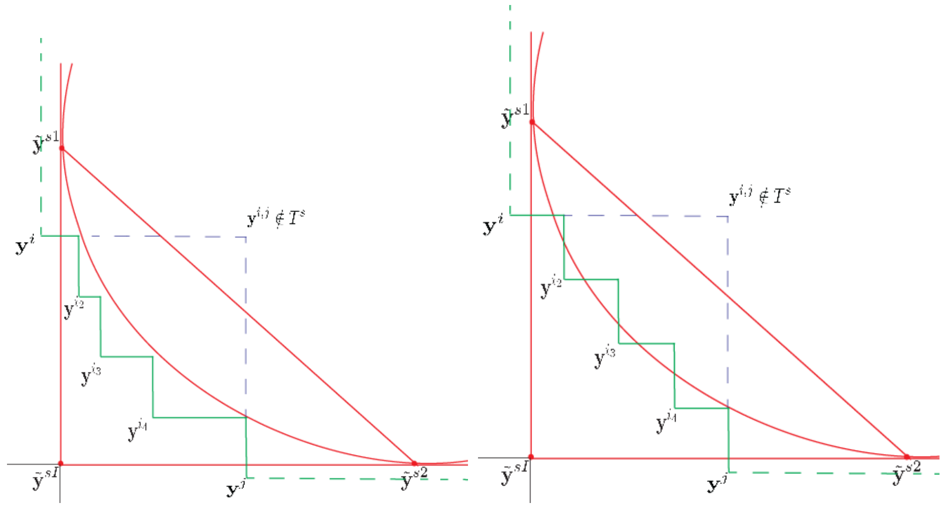


Figure 2.13: Two instances of Case 2 - node s can be fathomed



(a) (b)

Figure 2.14: Node s cannot be fathomed since $\mathcal{Y}_a^s \cap C^{sW} = \emptyset$ or $\mathcal{Y}_a^s \cap C^{sS} = \emptyset$



(a) Can be fathomed (b) Cannot be fathomed

Figure 2.15: The fathoming decision is not immediate when the nadir point implied by the closest nondominated points to $\tilde{\mathbf{y}}^{sI}$ is not in T^s

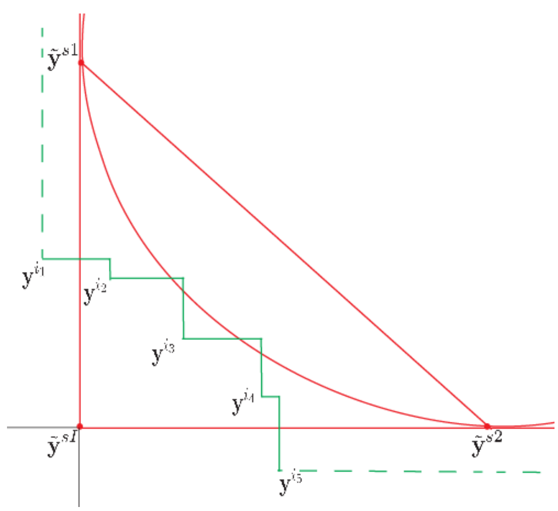


Figure 2.16: Node s cannot be fathomed since $\hat{y}^s_1 \not> y^{\kappa,\eta}_1$ or $\hat{y}^s_2 \not> y^{\kappa,\eta}_2$ for at least one $\mathbf{y}^{\kappa,\eta}$

D Figures: Set Dominance

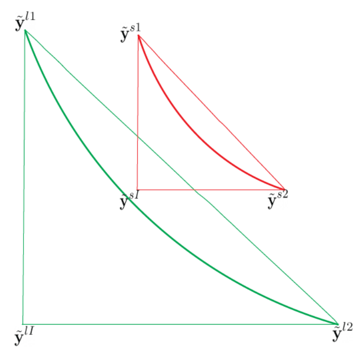


Figure 2.17: Proposition 2.6.9(i): $\tilde{\mathcal{Y}}_P^l < \tilde{\mathcal{Y}}_P^s$.

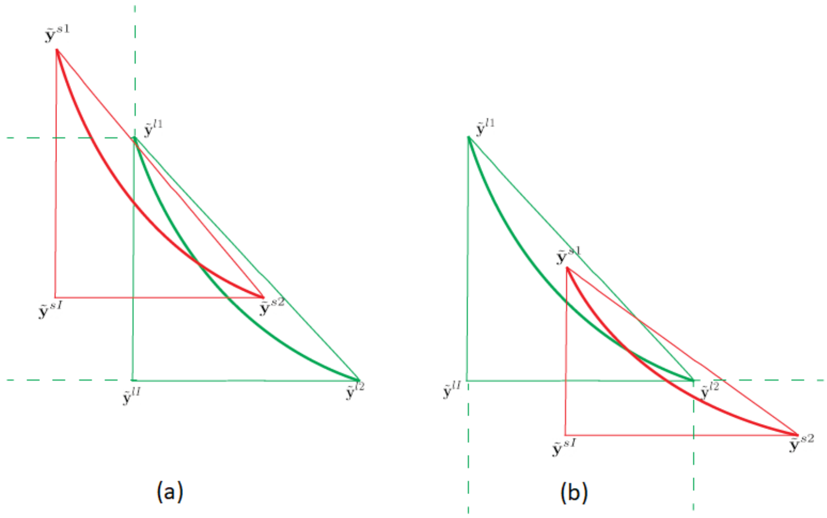


Figure 2.18: Two cases of Proposition 2.6.10(ii): Pareto sets intersect (a) $\tilde{\mathbf{y}}^{sI} \in C^{lW}$ and $\tilde{\mathbf{y}}^{l1} \in \tilde{\mathcal{Y}}_P^s + \mathbb{R}^2_>$, (b) $\tilde{\mathbf{y}}^{sI} \in C^{lS}$ and $\tilde{\mathbf{y}}^{l2} \in \tilde{\mathcal{Y}}_P^s + \mathbb{R}^2_>$.

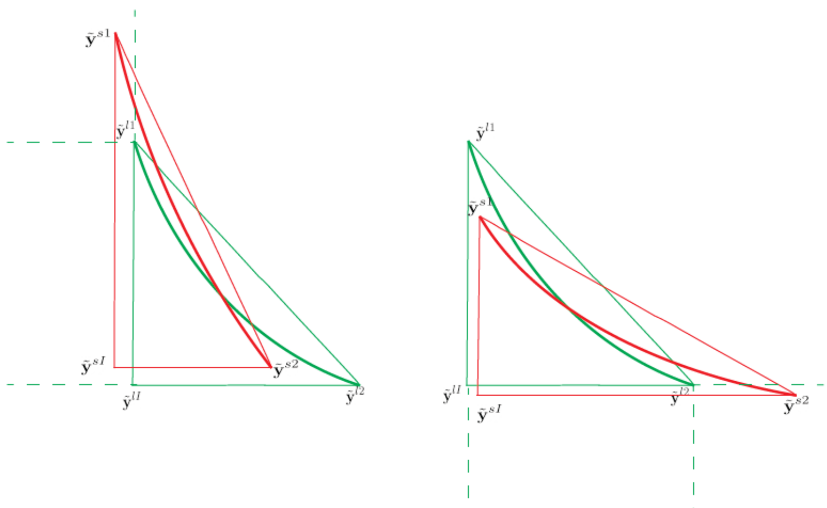


Figure 2.19: Two cases of Proposition 2.6.10(iii): Pareto sets intersect (a) $\tilde{\mathbf{y}}^{sI} \in C^{lW}$ and $\tilde{\mathbf{y}}^{l1} \in \tilde{\mathcal{Y}}_P^s - \mathbb{R}^2_>$, (b) $\tilde{\mathbf{y}}^{sI} \in C^{lS}$ and $\tilde{\mathbf{y}}^{l2} \in \tilde{\mathcal{Y}}_P^s - \mathbb{R}^2_>$.

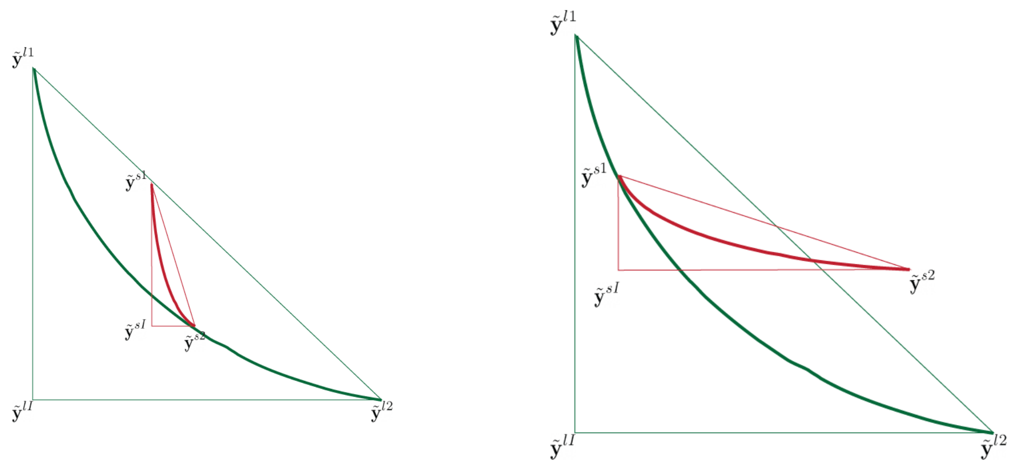


Figure 2.20: Proposition 2.6.11: $\tilde{\mathcal{Y}}_P^l \leqq \tilde{\mathcal{Y}}_P^s$.

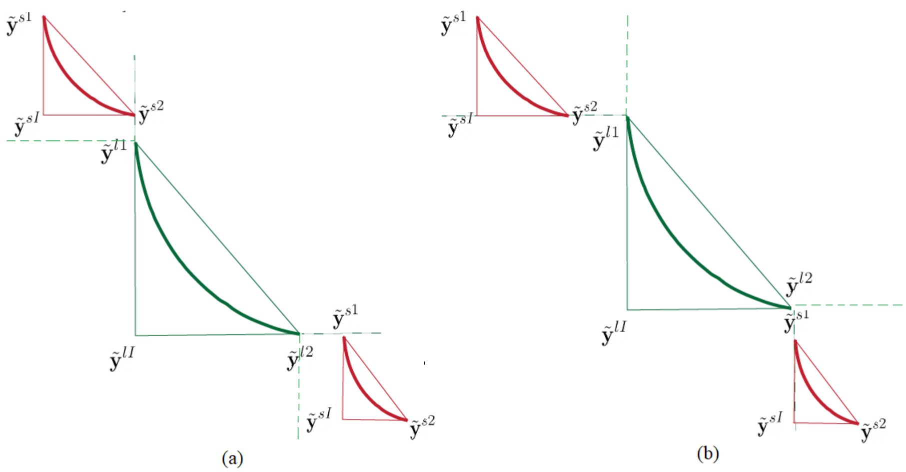


Figure 2.21: Proposition 2.6.12: $\tilde{\mathcal{Y}}_P^l \leq_P \tilde{\mathcal{Y}}_P^s$.

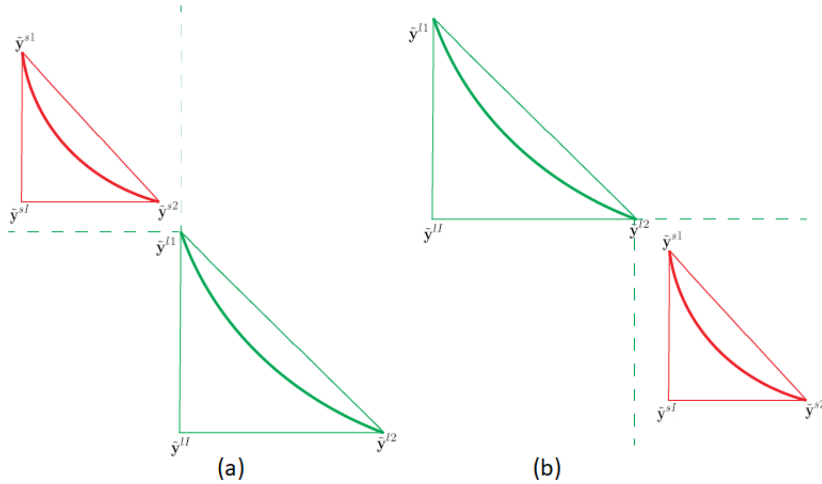
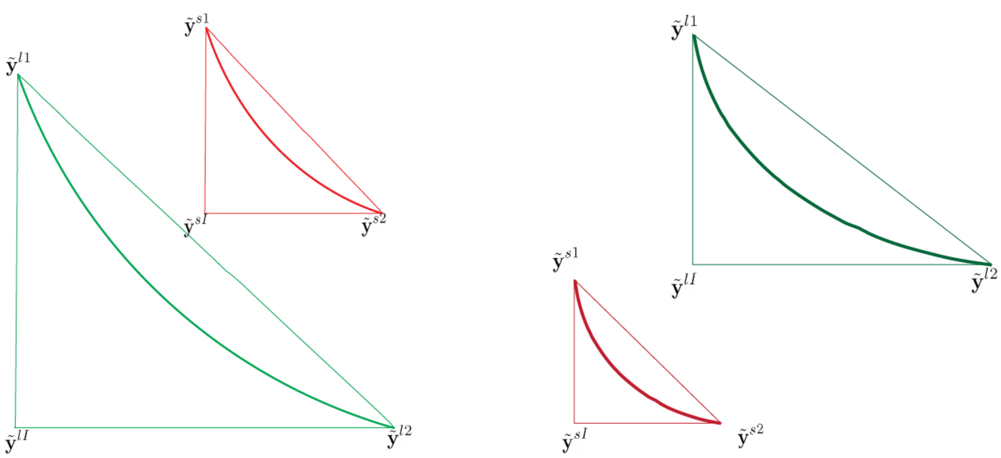


Figure 2.22: Two cases of $T^l \cap T^s = \emptyset$ and $\tilde{\mathcal{Y}}_N^{sl} = \tilde{\mathcal{Y}}_P^s \cup \tilde{\mathcal{Y}}_P^l$
(a) $T^s \in \{\tilde{\mathbf{y}}^{l1}\} - \mathbb{R}^2_{\gtrless}$, (b) $T^s \in \{\tilde{\mathbf{y}}^{l1}\} + \mathbb{R}^2_{\gtrless}$.



(a) $T^s \subset T^l + \mathbb{R}^2_{\geqq}$, $\tilde{\mathcal{Y}}_P^l \leq \tilde{\mathcal{Y}}_P^s$ and $\tilde{\mathcal{Y}}_N^{sl} = \tilde{\mathcal{Y}}_P^l$ (b) $T^l \subset T^s + \mathbb{R}^2_{\geqq}$, $\tilde{\mathcal{Y}}_P^s \leq \tilde{\mathcal{Y}}_P^l$ and $\tilde{\mathcal{Y}}_N^{sl} = \tilde{\mathcal{Y}}_P^s$

Figure 2.23: One Pareto set dominates the other.

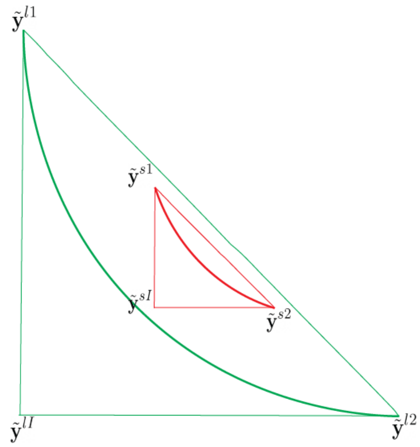


Figure 2.24: Subprocedure 1: $T^s \subset T^l$, $\tilde{\mathcal{Y}}_P^l \leq \tilde{\mathcal{Y}}_P^s$ and $\tilde{\mathcal{Y}}_N^{sl} = \tilde{\mathcal{Y}}_P^l$.

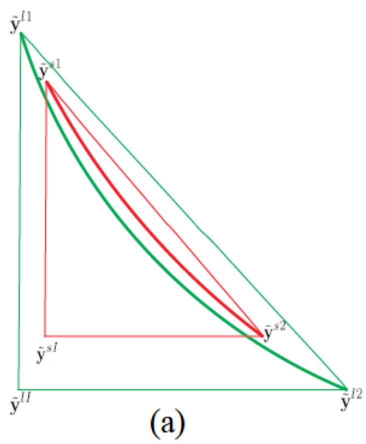


Figure 2.25: Subprocedure 1: (a) $\tilde{\mathcal{Y}}_P^l \cap \tilde{\mathcal{Y}}_P^s = \emptyset$, (b) $\tilde{\mathcal{Y}}_P^l \cap \tilde{\mathcal{Y}}_P^s \neq \emptyset$.

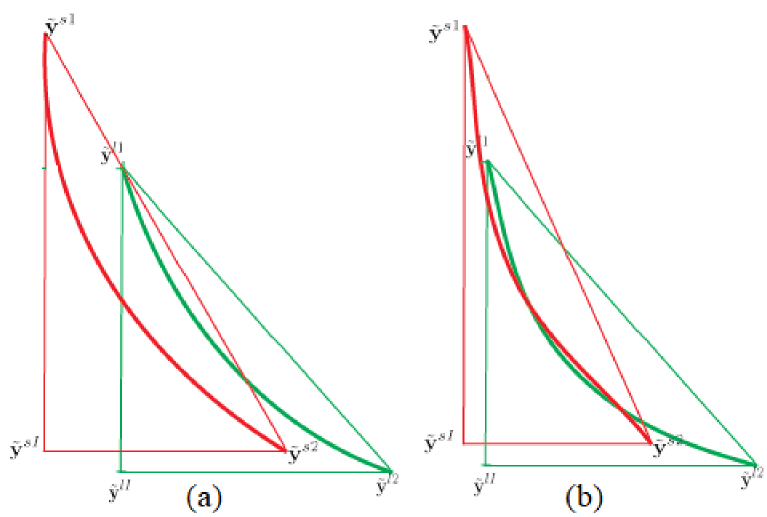


Figure 2.26: Subprocedure 2: (a) $\tilde{\mathcal{Y}}_P^l \cap \tilde{\mathcal{Y}}_P^s = \emptyset$, (b) $\tilde{\mathcal{Y}}_P^l \cap \tilde{\mathcal{Y}}_P^s \neq \emptyset$.

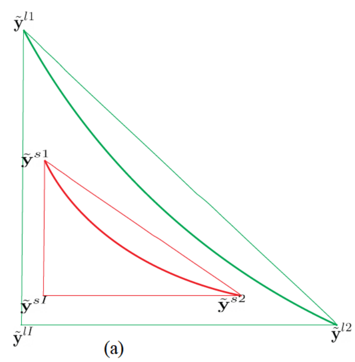


Figure 2.27: Subprocedure 1: (a) $\tilde{\mathcal{Y}}_P^l \cap \tilde{\mathcal{Y}}_P^s = \emptyset$, (b) $\tilde{\mathcal{Y}}_P^l \cap \tilde{\mathcal{Y}}_P^s \neq \emptyset$.

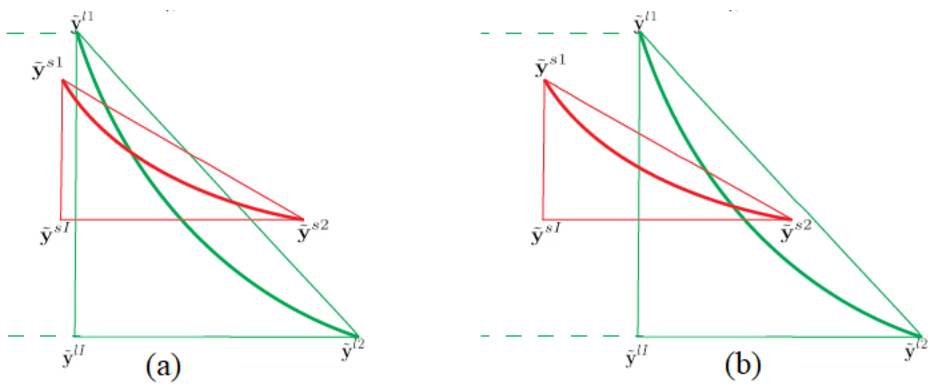


Figure 2.28: Subprocedure 2: $T^s \not\subset T^l$, $\tilde{\mathbf{y}}^{sI} \notin T^l$, $\tilde{\mathcal{Y}}_P^l \leq_P \tilde{\mathcal{Y}}_P^s$ and $\tilde{\mathcal{Y}}_P^s \leq_P \tilde{\mathcal{Y}}_P^l$.

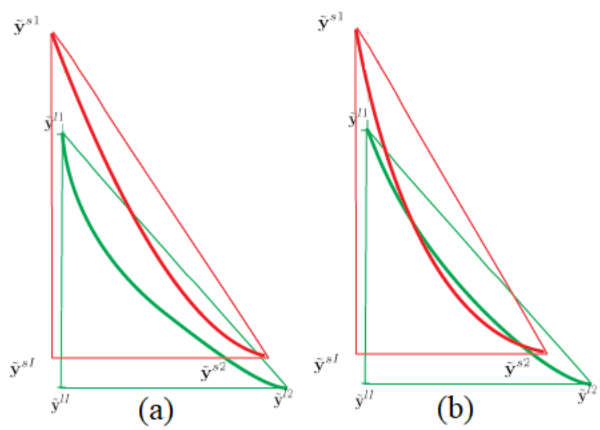


Figure 2.29: Subprocedure 2: (a) $\tilde{\mathcal{Y}}_P^l \cap \tilde{\mathcal{Y}}_P^s = \emptyset$, (a) $\tilde{\mathcal{Y}}_P^l \cap \tilde{\mathcal{Y}}_P^s \neq \emptyset$.

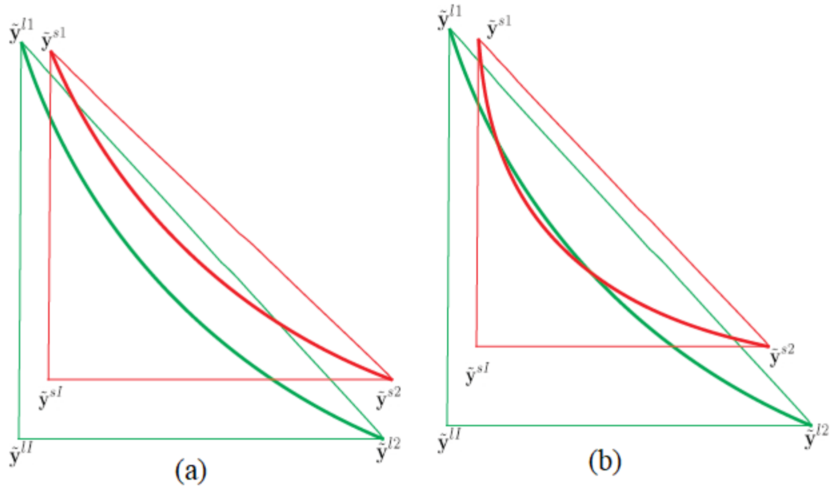


Figure 2.30: Subprocedure 3: $\tilde{\mathbf{y}}^{s1}, \tilde{\mathbf{y}}^{s2} \in \tilde{\mathcal{Y}}_P^l + \mathbb{R}^2_>$ (a) $\tilde{\mathcal{Y}}_P^l \cap \tilde{\mathcal{Y}}_P^s = \emptyset$, (b) $\tilde{\mathcal{Y}}_P^l \cap \tilde{\mathcal{Y}}_P^s \neq \emptyset$.

Chapter 3

Multiobjective Programs with Application to Portfolio Optimization

[The contents of this chapter include material from the paper entitled “On Convex Multiobjective Programs with Application to Portfolio Optimization” which has been published in the Journal of Multi Criteria Decision Analysis and the authors are Pubudu L.W. Jayasekara, Nathan Adelgren and Margaret M. Wiecek.]

3.1 Introduction

Multiobjective programs (MOPs) have proven to be an important modeling and methodological tool to solve a variety of decision making problems. On top of abundant applications in business and management [46, 44], applications of multicriteria decision making (MCDM) can be found in chemical engineering [102], engineering design [10], environmental engineering [66, 139], energy systems [66, 89], financial engineering [138], and other domains of human activity. Among various MOPs, convex problems play a specific role due to their elegant mathematical properties, easier solvability, and relevance to real-life applications. A major class within convex MOPs is made up by multiobjective quadratic programs (MOQPs) that have come as an extension of well-established and useful single objective quadratic programs (QPs). QPs have a special structure that is amenable to analytical derivations and algorithmic developments, but more importantly, to mathematical and real-life applications. MOQPs model decision making problems in science, business, and engineering such as regression analysis [1], finance [138], predictive control [15], and others.

In finance, biobjective portfolio optimization was initiated by Markowitz [84] who proposed to minimize the predicted variance of portfolio return as a measure of risk and to maximize the expected value of the portfolio return. Since then many researchers have been studying portfolio optimization in a multiobjective setting from various perspectives. Zopounidis et al. [143], Xinodas and Mavrotas [135, 136], and Smimou [113] propose various models of this problem. Sawik [111] concentrates on mixed-integer models and uses different risk measures. Li and Xu [77] account for uncertainty with a fuzzy approach, while Xinodas et al. [137] use a robust approach. Some authors choose to compute the best efficient solution but many others address the computation of the efficient set (also known as the efficient frontier), which is done with evolutionary or exact algorithms. Reviews of evolutionary algorithms applied to portfolio optimization are provided by Anagnostopoulos and Mamanis [7] and Metaxiotis and Liagkouras [88]. Hirschberger et al. [63] and Romanko et al. [107] apply parametric optimization, while Utz et al. [123] develop an inverse optimization framework. Steuer et al. [117] solve large-scale problems and conclude that parametric quadratic programming is the best technique for the efficient set computation. Approximation of the Pareto set and a search in the regions of investors' interests are suggested by Juszczuk et al. [70].

Another group of papers contains analytical derivations of the efficient set. The first one is perhaps the derivation for the biobjective case by Merton [87], which is followed by Qi et al. [101] for the triobjective case with one quadratic and two linear objective functions. This work is extended by Qi [100] to any number of quadratic and linear functions and is the first analytical study addressing the multiobjective portfolio optimization problem (MOPOP) with more than one quadratic objective function. This goes along with Boyd et al. [22] and Salas-Molina et al. [109] who recognize the need to model different types of risk measures, which naturally may lead to several quadratic functions in the MOPOP model. We are not aware of any study addressing the computation of the efficient set for MOPOPs with more than one quadratic objective function.

In this chapter, we recognizing the importance of convex MOPs, and collect their theoretical properties from the perspective of parametric convex optimization. We also propose a new scalarization suitable for a class of specially structured convex MOPs. Recognizing also the wide applicability of convex MOQPs, we review available exact algorithms for computing their efficient solutions and choose one of them that suits our needs best. We merge these two lines of investigation to solve the MOPOP with two or more quadratic objective functions, a class of problems that has not been solved before.

In Section 3.2 we recast three well-established scalarizations for convex

MOPs and propose the new one, while in Section 3.3 we review the algorithms for convex MOQPs. The multiobjective portfolio problem is solved in Section 3.4. We include numerical examples and conclude the paper in Section 4.6. A Supporting Information section contains detailed numerical and graphical results.

In the next section, we discuss the properties of (strictly) convex MOPs from the point of view of parametric optimization.

3.2 A Parametric Optimization Perspective on Convex MOPs

Scalarization, which involves reformulation of the original MOP into a single objective program (SOP) by means of scalarizing parameters such as weights, right-hand-side values, etc., is a common and well-established methodology for computing efficient solutions to MOPs [128]. It is expected that the optimal solutions to the SOP are (weakly) efficient solutions to the MOP. When solving the SOP, the scalarizing parameters may be known and therefore assume specific values for which the (weakly) efficient solutions are computed. Alternatively and more generally, these parameters may remain unknown and the optimal solutions to the SOP may be determined as functions of these parameters. The latter setting leads to parametric optimization that involves two groups of unknowns: the optimization variables for which the SOP is solved and the parameters that represent unknown problem data. An optimal solution and the optimal objective value to a parametric SOP (par-SOP) come in the form of functions of the parameters, and the solution to a par-SOP also includes a partition of the parameter space into invariancy regions for which a specific optimal solution function and the optimal objective value function are valid. Refer to Bank et al. [8] and Fiacco [52] who provide in-depth studies of parametric nonlinear optimization.

In the context of multiobjective optimization, the obtained optimal solutions to the par-SOP become parametric (weakly) efficient solutions to the MOP and come as functions of the scalarizing parameters, while the partition of the parameter space provides a full range of the (weakly) efficient solutions for a full range of potential values of the scalarizing parameters. Refer to [50] for a study of the weighted-sum method in the context of parametric optimization.

In this section, we review three well-established scalarization techniques from the perspective of parametric convex optimization and emphasize their applicability to (strictly) convex MOPs. For a specially structured class of convex MOPs, we propose a scalarization whose applicability is demonstrated in Section 3.4.

3.2.1 The weighted-sum problem

One of the most common scalarizing approaches is the weighted-sum method that reformulates the MOP into a single objective problem (SOP) by using a convex combination of the objective functions. The weighted-sum problem assumes the form:

$$\begin{aligned} \min \quad & \textstyle\sum_{i=1}^{r} \lambda_i f_i(\mathbf{x}) \\ s.t. \quad & \mathbf{x} \in \mathcal{X}, \end{aligned} \tag{3.1}$$

where $\boldsymbol{\lambda} \in \Lambda$ is a vector of weights that can be considered as parameters.

The weight or parameter space $\Lambda \subseteq \mathbb{R}^r$ is defined as

$$\Lambda = \{\boldsymbol{\lambda} \in \mathbb{R}^r : \boldsymbol{\lambda} \geq \mathbf{0}, \sum_{i=1}^{r} \lambda_i = 1\} \tag{3.2}$$

Problem (3.1) is a convex par-SOP whose optimal solution $\hat{\mathbf{x}}$ is a function of the parameter $\boldsymbol{\lambda}, \hat{\mathbf{x}} = \hat{\mathbf{x}}(\boldsymbol{\lambda})$. The following result is well-known.

Theorem 3.2.1. *[56] Let MOP (1.1) be convex. Then $\hat{\boldsymbol{x}}$ is weakly efficient to MOP (1.1) if and only if $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}(\boldsymbol{\lambda})$ is an optimal solution to problem (3.1) for some $\boldsymbol{\lambda} \in \Lambda$.*

In view of the properties of strictly convex MOPs, this result can be rewritten as follows.

Corollary 3.2.1.1. *Let MOP (1.1) be strictly convex and $\boldsymbol{\lambda} \in \Lambda$. Then $\hat{\boldsymbol{x}}$ is efficient to MOP (1.1) if and only if $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}(\boldsymbol{\lambda})$ is an optimal solution to problem (3.1).*

3.2.2 The $\boldsymbol{\epsilon}$-constraint problem

Another common approach to solving MOPs is the $\boldsymbol{\epsilon}$-constraint problem that involves minimizing an objective considered as primary and expressing the other objectives in the form of inequality constraints with a parameter $\boldsymbol{\epsilon} \in \mathbb{R}^{r-1}$. The i^{th} $\boldsymbol{\epsilon}$-constraint SOP is formulated as:

$$\begin{aligned} \hat{f}_i^{\leq}(\boldsymbol{\epsilon}) = \quad \min \quad & f_i(\mathbf{x}) \\ s.t. \quad & f_j(\mathbf{x}) \leq \epsilon_j \quad j = 1, \ldots, r \text{ and } j \neq i \\ & \mathbf{x} \in \mathcal{X}. \end{aligned} \tag{3.3}$$

Let $\mathcal{X}^{\leq}(\boldsymbol{\epsilon}) = \Big\{\mathbf{x} \in \mathcal{X} : f_j(\mathbf{x}) \leq \epsilon_j, j = 1, \ldots, r, ; j \neq i\Big\}$ and $\hat{\mathcal{X}}^{\leq}(\boldsymbol{\epsilon}) = \Big\{\mathbf{x} \in \mathcal{X}^{\leq}(\boldsymbol{\epsilon}) : f_i(\mathbf{x}) = \hat{f}_i^{\leq}(\boldsymbol{\epsilon})\Big\}$. A suitable choice of $\boldsymbol{\epsilon}$ ensures that a feasible solution to this SOP exists. Let $\mathbb{E}^{\leq}$ be the set of all $\boldsymbol{\epsilon}$ for which problem (3.3) is feasible, that is,

$$\mathbb{E}^{\leq} = \{\boldsymbol{\epsilon} \in \mathbb{R}^{r-1} : \mathcal{X}^{\leq}(\boldsymbol{\epsilon}) \neq \emptyset\}. \tag{3.4}$$

Problem (3.3) is a convex par-SOP whose optimal solution $\hat{\mathbf{x}}$ is a function of the parameter $\boldsymbol{\epsilon}, \hat{\mathbf{x}} = \hat{\mathbf{x}}(\boldsymbol{\epsilon})$, and the optimal objective value, $\hat{f}_i^{\leq}(\boldsymbol{\epsilon})$, is also a function of this parameter. The following result is also well-known.

Theorem 3.2.2. *[59] Let $\boldsymbol{\epsilon} \in \mathbb{E}^{\leq}$. If $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}(\boldsymbol{\epsilon})$ is a unique optimal solution to problem (3.3) then $\hat{\boldsymbol{x}}$ is an efficient solution to MOP (1.1).*

Since for strictly convex MOPs, the second order necessary and sufficient conditions for optimality [12] hold for problem (3.3), this result can be rewritten as follows.

Corollary 3.2.2.1. *Let $\boldsymbol{\epsilon} \in \mathbb{E}^{\leq}$ and $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}(\boldsymbol{\epsilon})$ be a an optimal solution to problem (3.3).*

1. *Let MOP (1.1) be convex. If the second order sufficient conditions for optimality hold at $\hat{\boldsymbol{x}}$ then $\hat{\boldsymbol{x}}$ is an efficient solution to MOP (1.1).*

2. *Let MOP (1.1) be strictly convex. Then $\hat{\boldsymbol{x}}$ is an efficient solution to MOP (1.1).*

Proof. 1. If the second order necessary and sufficient conditions for optimality hold at $\hat{\mathbf{x}}$ then $\hat{\mathbf{x}}$ is a unique optimal solution to MOP (1.1), which gives the desired result.

2. Under the strict convexity, the second order optimality conditions hold.

□

Treating problem (3.3) as a par-SOP, we obtain more insight into the properties of its solution.

Corollary 3.2.2.2. *Let $\boldsymbol{\epsilon} \in \mathbb{E}^{\leq}$. The optimal value function $\hat{f}_i^{\leq}(\boldsymbol{\epsilon})$ to problem (3.3) is nonincreasing on $\mathbb{E}^{\leq}$.*

Proof. Refer to page 260 in [12] . □

Corollary 3.2.2.3. *Let MOP (1.1) be convex, the set $\mathbb{E}^{\leq}$ be convex, and $\hat{f}_i^{\leq}(\boldsymbol{\epsilon})$ be the optimal value function to problem (3.3).*

1. *Then $\hat{f}_i^{\leq}(\boldsymbol{\epsilon})$ is convex on $\mathbb{E}^{\leq}$.*

2. *If f_i is strictly convex, the set $\hat{\mathcal{X}}^{\leq}(\boldsymbol{\epsilon}) \neq \emptyset$ for all $\boldsymbol{\epsilon} \in \mathbb{E}^{\leq}$, and $\hat{\mathcal{X}}^{\leq}(\boldsymbol{\epsilon^1}) \neq \hat{\mathcal{X}}^{\leq}(\boldsymbol{\epsilon^2})$ for $\boldsymbol{\epsilon^1}, \boldsymbol{\epsilon^2} \in \mathbb{E}^{\leq}, \boldsymbol{\epsilon^1} \neq \boldsymbol{\epsilon^2}$, then $\hat{f}_i^{\leq}(\boldsymbol{\epsilon})$ is strictly convex on $\mathbb{E}^{\leq}$.*

Proof. Parts 1 and 2 follow from Corollary 2.2 and Proposition 2.10, respectively, in [53]. □

3.2.3 The proper equality constraint problem

The proper equality constraint method is a modification of the $\boldsymbol{\epsilon}$-constraint method. The main difference between these two methods is that the additional $\boldsymbol{\epsilon}$-constraints are equalities rather than inequalities. The i^{th} proper equality constraint

SOP is formulated as:

$$
\begin{aligned}
\hat{f}_i^{=}(\boldsymbol{\epsilon}) = \quad \min \quad & f_i(\mathbf{x}) \\
s.t. \quad & f_j(\mathbf{x}) = \epsilon_j \quad j = 1, \ldots, r \ \text{ and } \ j \neq i \\
& \mathbf{x} \in \mathcal{X}.
\end{aligned}
\tag{3.5}
$$

Let $\mathcal{X}^{=}(\boldsymbol{\epsilon}) = \Big\{ \mathbf{x} \in \mathcal{X} : f_j(\mathbf{x}) = \epsilon_j, j = 1, \ldots, r, ; j \neq i \Big\}$ and $\hat{\mathcal{X}}^{=}(\boldsymbol{\epsilon}) = \Big\{ \mathbf{x} \in \mathcal{X}^{=}(\boldsymbol{\epsilon}) : f_i(\mathbf{x}) = \hat{f}_i^{=}(\boldsymbol{\epsilon}) \Big\}$. A suitable choice of $\boldsymbol{\epsilon}$ ensures that a feasible solution to this SOP exists. Let $\mathbb{E}^{=}$ be the set of all $\boldsymbol{\epsilon}$ for which problem (3.5) is feasible, that is,

$$
\mathbb{E}^{=} = \{ \boldsymbol{\epsilon} \in \mathbb{R}^{r-1} : \mathcal{X}^{=}(\boldsymbol{\epsilon}) \neq \emptyset \}. \tag{3.6}
$$

Problem (3.5) is a convex par-SOP whose optimal solution $\hat{\mathbf{x}}$ is a function of the parameter $\boldsymbol{\epsilon}, \hat{\mathbf{x}} = \hat{\mathbf{x}}(\boldsymbol{\epsilon})$ and the optimal objective value, $\hat{f}_i^{=}(\boldsymbol{\epsilon})$, is also a function of this parameter. The following result is also well-known.

Theorem 3.2.3. *[78] Let $\boldsymbol{\epsilon} \in \mathbb{E}^{=}$ and $\hat{\boldsymbol{x}}^i = \hat{\boldsymbol{x}}^i(\hat{\boldsymbol{\epsilon}})$ be an optimal solution to problem* (3.5). *Then $\hat{\boldsymbol{x}}^i$ is efficient for MOP (1.1) if and only if*

$$
\hat{f}_i^{=}(\boldsymbol{\epsilon}) \geq \hat{f}_i^{=}(\hat{\boldsymbol{\epsilon}}) \text{ for any } \boldsymbol{\epsilon} \in \mathbb{E}^{=} \text{ such that } \boldsymbol{\epsilon} \leqq \hat{\boldsymbol{\epsilon}} \tag{3.7}
$$

and

$$
\hat{f}_i^{=}(\boldsymbol{\epsilon}) > \hat{f}_i^{=}(\hat{\boldsymbol{\epsilon}}) \text{ for any } \boldsymbol{\epsilon} \in \mathbb{E}^{=} \text{ such that } \boldsymbol{\epsilon} \leq \hat{\boldsymbol{\epsilon}}. \tag{3.8}
$$

While conditions (3.7) and (3.8) are difficult to check to verify efficiency, it is possible to establish a relationship between problems (3.3) and (3.5) to get more insight. Let $\mathbb{E}^{\leq} = \mathbb{E}^{=}$. If at optimality of problem (3.3) all $\boldsymbol{\epsilon}$-constraints are active, then this problem assumes the form of problem (3.5) and the properties of the former carry over to the latter.

Corollary 3.2.3.1. *Let $\hat{\boldsymbol{\epsilon}} \in \mathbb{E}^{\leq} = \mathbb{E}^{=}$ and $\hat{\boldsymbol{x}}^i = \hat{\boldsymbol{x}}^i(\hat{\boldsymbol{\epsilon}})$ be an optimal solution to problem* (3.3) *with all $\hat{\boldsymbol{\epsilon}}$-constraints active at optimality. Then*

1. *$\hat{\boldsymbol{x}}^i$ is an optimal solution to problem* (3.5)*;*

2. *$\hat{\boldsymbol{x}}^i$ is an efficient solution to MOP (1.1) if and only if $\hat{f}_i^{\leq}(\boldsymbol{\epsilon}) > \hat{f}_i^{\leq}(\hat{\boldsymbol{\epsilon}})$ for all $\boldsymbol{\epsilon} \in \mathbb{E}^{\leq}$ such that $\boldsymbol{\epsilon} \leq \hat{\boldsymbol{\epsilon}}$.*

Proof. 1. Under the given assumption, $\hat{\mathbf{x}}^i$ is feasible and optimal to (3.5).

2. Refer to Theorem 3.1 in [79].

□

Under the assumptions of Corollary 3.2.3.1, Theorem 3.2.2 and Corollary 3.2.2.1 are valid for problem (3.5). A corollary for problem (3.5) that is analogous to Corollary 3.2.2.3 for problem (3.3) can be obtained when the objective functions creating the inequality constraints in (3.5) are affine.

Corollary 3.2.3.2. *Let MOP (1.1) be convex, the objective functions $f_j, j = 1, \ldots, r, j \neq i$, be affine, the set $\mathbb{E}^{=}$ be convex, and $\hat{f}_i^{=}(\boldsymbol{\epsilon})$ be the optimal value function to problem (3.5).*

1. *Then* $\hat{f}_i^{=}(\boldsymbol{\epsilon})$ *is convex on* $\mathbb{E}^{=}$.

2. *If* f_i *is strictly convex, the set* $\hat{\mathcal{X}}^{=}(\boldsymbol{\epsilon}) \neq \emptyset$ *for all* $\boldsymbol{\epsilon} \in \mathbb{E}^{=}$, *and* $\hat{\mathcal{X}}^{=}(\boldsymbol{\epsilon_1}) \neq \hat{\mathcal{X}}^{=}(\boldsymbol{\epsilon_2})$ *for* $\boldsymbol{\epsilon_1}, \boldsymbol{\epsilon_2} \in \mathbb{E}^{=}, \boldsymbol{\epsilon_1} \neq \boldsymbol{\epsilon_2}$, *then* $\hat{f}_i^{=}(\boldsymbol{\epsilon})$ *is strictly convex on* $\mathbb{E}^{=}$.

Proof. Parts 1 and 2 follow from Corollary 2.2 and Proposition 2.10, respectively, in [53]. □

3.2.4 The modified hybrid problem

We now combine the concepts of the three scalarizations above into an SOP that we refer to as the modified hybrid problem. Consider a scalarization of the original MOP (1.1) by means of a partial weighted-sum applied only to the first s objective functions:

$$\begin{array}{ll} \min & \mathbf{f}(\mathbf{x}) = \left[\sum_{i=1}^{s} \gamma_i f_i(\mathbf{x}), f_{s+1}(\mathbf{x}), \ldots, f_r(\mathbf{x})\right] \\ s.t. & \mathbf{x} \in \mathcal{X}, \end{array} \tag{3.9}$$

where $\boldsymbol{\gamma} \in \mathbb{R}^s$ such that $\sum_{i=1}^{s} \gamma_i = 1$ and $\gamma_i \geq 0$ for $i = 1, \ldots, s$. Let $(\mathcal{X}_{wE}(\boldsymbol{\gamma}))\mathcal{X}_E(\boldsymbol{\gamma})$ denote the set of (weakly) efficient solutions to MOP (3.9).

The following result is immediate.

Theorem 3.2.4. *Let MOP (1.1) be convex. A solution* $\hat{\boldsymbol{x}} \in \mathcal{X}$ *is weakly efficient to MOP (1.1) if and only if* $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}(\boldsymbol{\gamma})$ *is weakly efficient to MOP (3.9) for some* $\boldsymbol{\gamma} \geq \mathbf{0}$.

Proof. Let $\hat{\mathbf{x}} \in \mathcal{X}_{wE}$ to MOP (1.1). Equivalently, $\hat{\mathbf{x}}$ is an optimal solution to the weighted-sum problem (3.1) for some $\boldsymbol{\lambda} \in \Lambda$. Choose ρ and $\rho_i, i = s+1, \ldots, r$, where $\boldsymbol{\rho} = (\rho, \rho_{s+1}, \ldots, \rho_r) \geq \mathbf{0}$, and $\boldsymbol{\gamma} \in \mathbb{R}^s, \boldsymbol{\gamma} \geq \mathbf{0}$ such that

$$\rho\gamma_i = \lambda_i, i = 1, \ldots, s, \rho_i = \lambda_i, i = s+1, \ldots, r, \rho + \sum_{i=s+1}^{r} \rho_i = 1. \tag{3.10}$$

The objective function in (3.1) can equivalently be rewritten using the new parameters so this problem assumes the form

$$\begin{array}{ll} \min & \rho \sum_{i=1}^{s} \gamma_i f_i(\mathbf{x}) + \sum_{i=s+1}^{r} \rho_i f_i(\mathbf{x}) \\ s.t. & \mathbf{x} \in \mathcal{X}, \end{array} \tag{3.11}$$

where $\sum_{i=1}^{s} \gamma_i = 1$. Hence $\hat{\mathbf{x}}$ is also an optimal solution to the weighted-sum-problem (3.11) for some $\boldsymbol{\rho} \geq \mathbf{0}$ and $\boldsymbol{\gamma} \geq \mathbf{0}$, and also a weakly efficient solution to MOP (3.9). □

For strictly convex MOPs, this result can be rewritten as follows.

Corollary 3.2.4.1. *Let MOP (1.1) be strictly convex. A solution $\hat{\boldsymbol{x}} \in \mathcal{X}$ is efficient to MOP (1.1) if and only if $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}(\boldsymbol{\gamma})$ is efficient to MOP (3.9) for some $\boldsymbol{\gamma} \geq \mathbf{0}$.*

MOP (3.9) can be scalarized in the same way as MOP (1.1). Applying the $\boldsymbol{\epsilon}$-constraint (or the proper equality constraint) approach to (3.9), we obtain two types of the modified hybrid problem for MOP (1.1)

$$\begin{array}{rll} \hat{f}^{\leq(=)}(\boldsymbol{\gamma}, \boldsymbol{\epsilon}) & = \min & \sum_{i=1}^{s} \gamma_i f_i(\mathbf{x}) \\ & s.t. & f_i(\mathbf{x}) \leq (=) \epsilon_i \quad i = s+1, s+2, \ldots, r \\ & & \mathbf{x} \in \mathcal{X}, \end{array} \tag{3.12}$$

where $\gamma_i \in \Gamma = \{\boldsymbol{\gamma} \in \mathbb{R}^s : \gamma_i \geq 0, \sum_{i=1}^{s} \gamma_i = 1\}$. Define $\mathcal{X}_h^{\leq(=)}(\boldsymbol{\epsilon}) = \big\{\mathbf{x} \in \mathcal{X} : f_j(\mathbf{x}) \leq (=)\epsilon_j, \quad j = s+1, \dots r\big\}$, and let $\mathbb{E}_h^{\leq(=)}$ be the set of all $\boldsymbol{\epsilon} \in \mathbb{R}^{r-s}$ for which problem (3.12) is feasible, $\mathbb{E}_h^{\leq(=)} = \big\{\boldsymbol{\epsilon} \in \mathbb{R}^{r-s} : \mathcal{X}_h^{\leq(=)}(\boldsymbol{\epsilon}) \neq \emptyset\big\}$. Problem (3.12) is a convex par-SOP whose optimal solution $\hat{\mathbf{x}}$ is a function of the parameters $(\boldsymbol{\gamma}, \boldsymbol{\epsilon}), \hat{\mathbf{x}} = \hat{\mathbf{x}}(\boldsymbol{\gamma}, \boldsymbol{\epsilon})$, and the optimal objective value, $\hat{f}^{\leq(=)}(\boldsymbol{\gamma}, \boldsymbol{\epsilon})$, is also a function of these parameters.

Having established the equivalence between MOP (1.1) and MOP (3.9), the results for MOP (1.1) presented earlier in this section are applicable to MOP (3.9). In particular, the efficient solutions to MOP (1.1) are found in the following circumstances.

Corollary 3.2.4.2. *Let $\boldsymbol{\gamma} \in \Gamma, \boldsymbol{\epsilon} \in \mathbb{E}_h^{\leq}$, and $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}(\boldsymbol{\gamma}, \boldsymbol{\epsilon}) \in \mathcal{X}$ be a an optimal solution to problem (3.12) with the inequality $\boldsymbol{\epsilon}$-constraints.*

1. *Let MOP (1.1) be convex. If the second order sufficient conditions for optimality hold at $\hat{\boldsymbol{x}}$ then $\hat{\boldsymbol{x}}$ is an efficient solution to MOP (1.1).*

2. *Let MOP (1.1) be strictly convex. Then $\hat{\boldsymbol{x}}$ is an efficient solution to MOP (1.1).*

Proof. Parts 1 and 2 follow from Corollary 3.2.2.1 and Theorem 3.2.4. □

Problem (3.12) with the inequality constraints is similar to the formulation of Wendell and Lee [126] and Corley [28] known in the literature as the hybrid problem. The latter combines the weighted-sum method with the $\boldsymbol{\epsilon}$-constraint method with the distinctions that *all* objectives are put into the weighted-sum as well as into the additional inequality constraints.

The modified hybrid problem is useful when the objective functions of the original MOP come in two groups with different real life meanings. One group is associated with a parameter $\boldsymbol{\gamma} \in \Gamma$ while the other group with a parameter $\boldsymbol{\epsilon} \in \mathbb{E}$, which allow the decision maker for an independent analysis of tradeoffs among the criteria within each group. A multiobjective portfolio optimization problem in which the objectives associated with the risk of a portfolio are minimized while maximizing the portfolio's return or liquidity is an example of MOPs with two groups of objectives. In Section 3.4 we discuss this case in more detail. However, since portfolio optimization involves quadratic objective functions that are typically (strictly) convex, in the next section we turn our attention to convex MOQPs as a broad class of problems to which the results of this sections can be applied.

3.3 State-of-the-Art Algorithms for Convex MOQPs

In this section, we review and compare the algorithms for solving (strictly) convex MOQPs. Considering MOP (1.1) with its assumptions and with quadratic objective functions, we have the following MOQP:

$$\begin{aligned} \min \quad & \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T Q_1 \mathbf{x} + \mathbf{p}_1^T \mathbf{x}, \ldots, f_r(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T Q_r \mathbf{x} + \mathbf{p}_r^T \mathbf{x}] \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{X} \end{aligned} \tag{3.13}$$

where $Q_i \in \mathbb{R}^{n \times n}$, $\boldsymbol{p}_i \in \mathbb{R}^n$ for $i = 1, \ldots, r$, $\mathbf{x} \in \mathbb{R}^n$, $\mathcal{X} \subseteq \mathbb{R}^n$.

Assumption 3.3.1. *Matrix $Q_i \in \mathbb{R}^{n \times n}$ is a symmetric positive semidefinite (PSD) matrix for all $i = 1, \ldots, r$. Therefore, MOQP (3.13) is convex.*

Since we are concerned with the algorithms for solving MOQPs and the state-of-the-art methods use the weighted-sum scalarization, we provide the formulation below.

$$\begin{aligned} \min \quad & \tfrac{1}{2}\mathbf{x}^T Q(\boldsymbol{\lambda})\mathbf{x} + \mathbf{p}^T(\boldsymbol{\lambda})\mathbf{x} \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{X}, \end{aligned} \tag{3.14}$$

where $Q(\boldsymbol{\lambda}) = \sum_{i=1}^{r} \lambda_i Q_i, \mathbf{p}(\boldsymbol{\lambda}) = \sum_{i=1}^{r} \lambda_i \mathbf{p}_i$ and the set of parameters is rewritten as

$$\Lambda' = \{\boldsymbol{\lambda} \in \mathbb{R}^{r-1} : \lambda_i \geq 0, \sum_{i=1}^{r-1} \lambda_i \leq 1\} \text{ with } \lambda_r = 1 - \sum_{i=1}^{r-1} \lambda_i \geq 0. \tag{3.15}$$

Proposition 3.3.2. *Under Assumption 3.3.1, matrix $Q(\boldsymbol{\lambda})$ is PSD for every $\boldsymbol{\lambda} \geq \mathbf{0}$.*

Proof. For all nonzero $\mathbf{x} \in \mathbb{R}^n$, calculate

$$\begin{aligned} \mathbf{x}^T Q(\boldsymbol{\lambda})\mathbf{x} &= \mathbf{x}^T \Big(\sum_{i=1}^{r} \lambda_i Q_i\Big)\mathbf{x} \\ &= \sum_{i=1}^{r} \lambda_i \mathbf{x}^T Q_i \mathbf{x} \\ &\geq 0 \qquad \text{since } Q_i \text{ is a PSD matrix for all } i = 1, \ldots, r. \end{aligned}$$

□

Based on the proposition above, matrix $Q(\boldsymbol{\lambda})$ is PSD for every $\boldsymbol{\lambda} \in \Lambda'$.

We have found four prominent algorithms to solve MOQP (3.13) with a polyhedral feasible set $\mathcal{X}$. We present them chronologically in the order they have been published. All algorithms but one make use of the Karush-Kuhn-Tucker (KKT) optimality conditions applied to problem (3.14) that is treated as a par-SOP with the parameter $\boldsymbol{\lambda}$. Since this par-SOP is convex, the KKT conditions are also sufficient and guarantee the optimality. However, in each of the three algorithms, the KKT conditions are differently processed.

In this section, to stay in agreement with the terminology used by the authors of the presented algorithms, we use the term "multiparametric" whenever we refer to a vector parameter.

3.3.1 The active set method [57]

In the active set method the MOQP has the feasible set in the form

$$\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n : A_1\mathbf{x} - \mathbf{b}_1 \leqq \mathbf{0}, A_2\mathbf{x} - \mathbf{b}_2 = \mathbf{0}\},$$

where $A_1 \in \mathbb{R}^{m_1 \times n}$, $A_2 \in \mathbb{R}^{m_2 \times n}$, and $\mathbf{x}$ is a free variable. It is assumed that the matrix $Q(\boldsymbol{\lambda})$ is positive-definite (PD) for each $\boldsymbol{\lambda} \in \{\boldsymbol{\lambda} \in \mathbb{R}^{r-1} : \lambda_i > 0, \sum_{i=1}^{r-1} \lambda_i < 1\}$ and the rows of matrix $[A_1^T \ A_2^T]^T$ are linearly independent. Using the KKT conditions, the explicit analytical primal and dual parametric solutions to problem (3.14) are computed for any value of $\boldsymbol{\lambda} \in \Lambda'$, and the initial set of active constraints at the primal solution is identified. A new set of active constraints is found by computing an allowable increase in $\boldsymbol{\lambda}$ which determines a subset of Λ' for which new explicit analytical primal and dual parametric solutions are computed.

3.3.2 The multiparametric pivoting method [64]

The multiparametric pivoting method can be applied to MOQPs with one quadratic objective function and any number of linear objective functions. The feasible set comes in the form

$$\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n : A_1\mathbf{x} - \mathbf{b}_1 \leqq \mathbf{0}, A_2\mathbf{x} - \mathbf{b}_2 = \mathbf{0}, \mathbf{x} - \mathbf{x}^u \leqq \mathbf{0}, \mathbf{x} - \mathbf{x}^l \geqq \mathbf{0}\},$$

where $\mathbf{x}^u$ and $\mathbf{x}^l \in \mathbb{R}^n$ are upper and lower bounds respectively. The KKT optimality conditions applied to the original SOP are reduced to a system of linear parametric equations whose solutions are functions of $\boldsymbol{\lambda} \in \Lambda'$ and also the (weakly) efficient solutions of the original MOQP. The system is solved with a parametric pivoting procedure that requires a basis matrix to be constant rather than parametric and therefore this method cannot handle more than one quadratic objective function.

3.3.3 The approximation method [92]

In the approximation method, MOQPs have the feasible set of the form

$$\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} - \mathbf{b} \leqq \mathbf{0}\},$$

where $A \in \mathbb{R}^{m\times n}$ and $\mathbf{b} \in \mathbb{R}^m$. In this method, matrices Q_i, $i = 1, \ldots, r$ are assumed to be PD and $r-1$ objective functions are approximated by auxiliary affine functions that are constructed based on the first order Taylor expansion of the original $r-1$ functions. The original MOQP is scalarized with the $\boldsymbol{\epsilon}$-constraint method in which the affine functions are put in the $\boldsymbol{\epsilon}$-constraints.

3.3.4 The mpLCP method [3]

The multiparametric linear complementarity problem (mpLCP) method solves the MOQP by solving the mpLCP resulting from applying the KKT conditions to problem (3.14) with the feasible set defined as

$$\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} - \mathbf{b} \leqq \mathbf{0}, \mathbf{x} \geqq \mathbf{0}\}. \tag{3.16}$$

To obtain the mpLCP one proceeds as follows. The inequality constraints in $\mathcal{X}$ are first converted into equality constraints $A\mathbf{x} + \mathbf{s} = \mathbf{b}$ by adding a slack variable $\mathbf{s} \geqq \mathbf{0}$. The KKT conditions yield the following system of equations with nonnegative variables:

$$\begin{aligned}
\begin{bmatrix} \mathbf{s} \\ \boldsymbol{u_2} \end{bmatrix} - \begin{bmatrix} \mathbf{0} & -A \\ A^T & Q(\boldsymbol{\lambda}) \end{bmatrix} \begin{bmatrix} \boldsymbol{u_1} \\ \mathbf{x} \end{bmatrix} &= \begin{bmatrix} \mathbf{b} \\ \mathbf{p}(\boldsymbol{\lambda}) \end{bmatrix} \\
\begin{bmatrix} \mathbf{s} & \boldsymbol{u_2} \end{bmatrix} \begin{bmatrix} \boldsymbol{u_1} \\ \mathbf{x} \end{bmatrix} &= \mathbf{0} \\
\begin{bmatrix} \mathbf{s} \\ \boldsymbol{u_2} \end{bmatrix}, \begin{bmatrix} \boldsymbol{u_1} \\ \mathbf{x} \end{bmatrix} &\geqq \mathbf{0},
\end{aligned} \tag{3.17}$$

where $\boldsymbol{u_1}$ and $\boldsymbol{u_2}$ are the dual variables associated with the linear constraint and the nonnegativity constraint respectively. Defining $M(\boldsymbol{\lambda}) = \begin{bmatrix} 0 & -A \\ A^T & Q(\boldsymbol{\lambda}) \end{bmatrix}$, $\mathbf{q}(\boldsymbol{\lambda}) = \begin{bmatrix} \mathbf{b} \\ \mathbf{p}(\boldsymbol{\lambda}) \end{bmatrix}$, $\mathbf{w} = \begin{bmatrix} \mathbf{s} \\ \boldsymbol{u_2} \end{bmatrix}$, $\mathbf{z} = \begin{bmatrix} \boldsymbol{u_1} \\ \mathbf{x} \end{bmatrix}$, we obtain the mpLCP

$$\begin{aligned} \mathbf{w} - M(\boldsymbol{\lambda})\mathbf{z} &= \mathbf{q}(\boldsymbol{\lambda}) \\ \mathbf{w}^T\mathbf{z} &= \mathbf{0} \\ \mathbf{w}, \mathbf{z} &\geqq \mathbf{0}, \end{aligned} \tag{3.18}$$

where the elements of matrix $M(\boldsymbol{\lambda}) \in \Omega^{(n+m)\times(n+m)}$ and vector $\mathbf{q}(\boldsymbol{\lambda}) \in \Omega^{(n+m)}$ are affine functions of $\boldsymbol{\lambda}$, $\boldsymbol{\lambda} \in \Lambda'$, from a set $\Omega = \{\boldsymbol{\alpha}^T\boldsymbol{\lambda} + \beta : \boldsymbol{\alpha} \in \mathbb{R}^{r-1}, \beta \in \mathbb{R}\}$. Since Q_i are PSD for all $i = 1\ldots, r$, it is easy to verify that $M(\boldsymbol{\lambda})$ is PSD for each $\boldsymbol{\lambda} \in \Lambda'$. Solving the MOQP reduces to solving the mpLCP with parameters in general locations. The first ever algorithm to solve this class of LCPs was only recently developed in [3]. In the algorithm, the parameter space Λ' is partitioned into $r-1$ dimensional invariancy regions over which the mpLCP solution functions $\mathbf{w}(\boldsymbol{\lambda})$ and $\mathbf{z}(\boldsymbol{\lambda})$ are computed as functions of $\boldsymbol{\lambda}$. The method consists of two phases. In Phase I, an initial $r-1$ dimensional invariancy region in Λ' is found. In Phase II, a partition of Λ' is found by considering the invariancy regions only of dimension $(r-2)$ or $(r-1)$. Given the mpLCP optimal solution functions, $(\mathbf{w}(\boldsymbol{\lambda}), \mathbf{z}(\boldsymbol{\lambda}))$, for the invariancy regions discovered, the optimal solution functions $\mathbf{x}(\boldsymbol{\lambda})$ to problem (3.14), and at the same time, the (weakly) efficient solution functions to MOQP (3.13) are obtained.

3.3.5 Comparison

To compare the four methods we apply the following criteria: (i) the number of quadratic objective functions in (3.13); (ii) special assumptions made; (iii) the type of solutions, exact or approximate, computed; and (iv) the type of scalarization used. We note that all but the multiparametric pivoting method can handle MOQPs with any number of quadratic functions. However, the active set method uses explicit formulas for the primal and dual parametric solutions of problem (3.14) and is numerically very costly. The approximation method is flexible since it can also be used with the $\boldsymbol{\epsilon}$-constraint scalarization but, by design, it does not yield the exact solutions. The active set method works under the linear independence condition which, for example, may not allow one to include nonnegativity constraints in the feasible set. As a result, we conclude that the mpLCP method emerges as a winner across all four categories and the most universal approach to solving MOQPs. We use this method to solve MOQPs in the remaining part of this paper.

3.3.6 Example 1

We illustrate the mpLCP method on an example problem. Consider MOQP (3.13) with three quadratic objective functions and the feasible set in form (3.16) with the following data:

$$Q_1 = \begin{bmatrix} 1 & -1 & -1 \\ -1 & 7 & -1 \\ -1 & -1 & 3 \end{bmatrix}, Q_2 = \begin{bmatrix} 7 & -5 & 0 \\ -5 & 5 & 1 \\ 0 & 1 & 1 \end{bmatrix}, Q_3 = \begin{bmatrix} 4 & 3 & 1 \\ 3 & 4 & 0 \\ 1 & 0 & 2 \end{bmatrix}.$$

$$\mathbf{p}_1 = \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{p}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \mathbf{p}_3 = \begin{bmatrix} -1 \\ -2 \\ 0 \end{bmatrix}.$$

$$A = \begin{bmatrix} -1 & 1 & 0 \\ 2 & -2 & 1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

We calculate

$$Q(\boldsymbol{\lambda}) = \lambda_1 Q_1 + \lambda_2 Q_2 + (1 - \lambda_1 - \lambda_2) Q_3$$

$$= \begin{bmatrix} -3\lambda_1 + 3\lambda_2 + 4 & -4\lambda_1 - 8\lambda_2 + 3 & -2\lambda_1 - \lambda_2 + 1 \\ -4\lambda_1 - 8\lambda_2 + 3 & 3\lambda_1 + 2\lambda_2 + 4 & -\lambda_1 + \lambda_2 \\ -2\lambda_1 - \lambda_2 + 1 & -\lambda_1 + \lambda_2 & \lambda_1 - \lambda_2 + 2 \end{bmatrix}.$$

$$\mathbf{p}(\boldsymbol{\lambda}) = \lambda_1 p_1 + \lambda_2 p_2 + (1 - \lambda_1 - \lambda_2) p_3$$

$$= \begin{bmatrix} \lambda_2 - 1 \\ 3\lambda_1 + 2\lambda_2 - 2 \\ \lambda_1 + \lambda_2 \end{bmatrix}.$$

where $\Lambda' = \{\boldsymbol{\lambda} \in \mathbb{R}^2 : \lambda_1, \lambda_2 \geq 0, \lambda_1 + \lambda_2 \leq 1\}$. The associated mpLCP given in (3.18) is as follows:

$$\mathbf{w} - \begin{bmatrix} 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & -2 & 2 & -1 \\ -1 & 2 & -3\lambda_1 + 3\lambda_2 + 4 & -4\lambda_1 - 8\lambda_2 + 3 & -2\lambda_1 - \lambda_2 + 1 \\ 1 & -2 & -4\lambda_1 - 8\lambda_2 + 3 & 3\lambda_1 + 2\lambda_2 + 4 & -\lambda_1 + \lambda_2 \\ 0 & 1 & -2\lambda_1 - \lambda_2 + 1 & -\lambda_1 + \lambda_2 & \lambda_1 - \lambda_2 + 2 \end{bmatrix} \mathbf{z} = \begin{bmatrix} 0 \\ 1 \\ \lambda_2 - 1 \\ 3\lambda_1 + 2\lambda_2 - 2 \\ \lambda_1 + \lambda_2 \end{bmatrix}$$

$$\mathbf{w}^T \mathbf{z} = 0$$

$$\mathbf{w}, \mathbf{z} \geqq \mathbf{0}.$$

Using the mpLCP method coded in MATLAB [3], four invariancy regions in the parameter space and the associated efficient solutions are computed and given in Supplementary Table 3.4. Figure 3.1 depicts the invariancy regions in the parameter space Λ' for the efficient solution functions $\mathbf{x}(\lambda_1, \lambda_2)$, and Figure 3.2 depicts the Pareto set in the three-dimensional objective space. In Figure 2 (as well as in Figures 3.6 and 3.7 in Section 3.4), the points in the Pareto set were obtained by mapping the points in the parameter space into the decision space and then into the objective space. The colors help to see the parts of the Pareto set that are associated with particular invariancy regions in Figure 3.1 (or Figure 3.5 in Section 3.4).

In the next section, we merge the two lines of investigation presented in Sections 3.2 and 3.3 to study multiobjective portfolio optimization. We use the modified hybrid problem for scalarization, solve the resulting par-SOP with the mpLCP method and discuss the benefits of this approach in comparison to the classical weighted-sum approach.

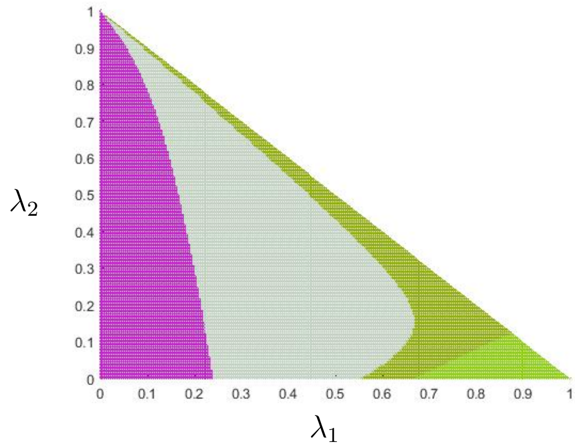


Figure 3.1: Invariancy regions in the parameter space for Example 1

3.4 Portfolio Optimization with Multiple Quadratic Objective Functions

The Markowitz mean-variance portfolio optimization problem [84] can be formulated as an MOQP. The problem involves minimizing quadratic objective functions modeling the variance of return and other measures of risk [22] and maximizing linear functions such as the expected return, liquidity, growth in sales, sustainability, and others [45, 64]. The goal is to find the set of efficient portfolios and the set of their Pareto performances.

Consider the multiobjective portfolio optimization problem (MOPOP) with s quadratic functions and $r-s$ linear functions:

$$\begin{aligned} \min \quad & \left[\tfrac{1}{2}\mathbf{x}^T Q_1 \mathbf{x}, \tfrac{1}{2}\mathbf{x}^T Q_2 \mathbf{x}, \ldots, \tfrac{1}{2}\mathbf{x}^T Q_s \mathbf{x}, -\boldsymbol{p}_{s+1}^T \mathbf{x}, -\boldsymbol{p}_{s+2}^T \mathbf{x}, \ldots, -\boldsymbol{p}_r^T \mathbf{x}\right] \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{1}^T \mathbf{x} - 1 = 0, \mathbf{x} \geqq \mathbf{0}\}, \end{aligned} \tag{3.19}$$

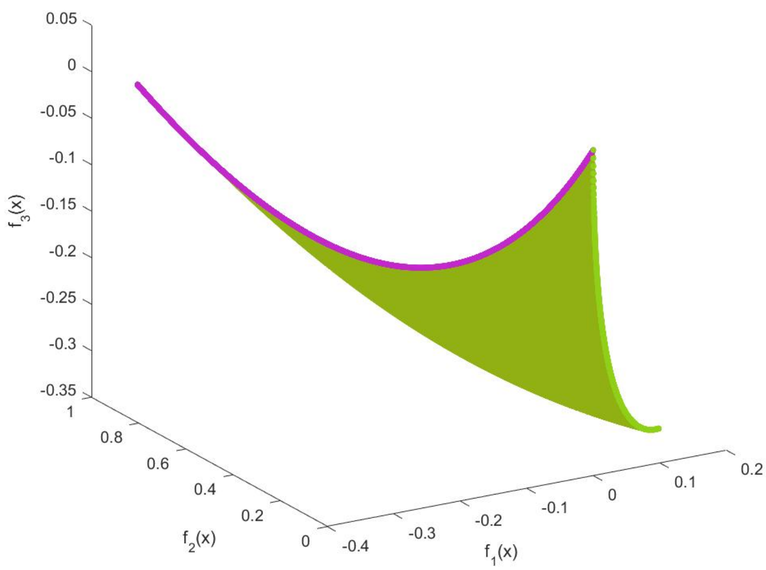


Figure 3.2: The Pareto set in the three-dimensional objective space for Example 1

where $Q_i \in \mathbb{R}^{n\times n}, i = 1, \ldots, s$, $\mathbf{p}_j \in \mathbb{R}^n, j = s+1, \ldots, r$, $\mathbf{1} \in R^n$ is a vector of ones. The components of decision variable $\mathbf{x} \in \mathbb{R}^n$ are defined as the proportions of capital to be allocated to different securities, and the linear constraint models the investment of the entire capital.

As discussed in Section 3.1, in the literature, the MOPOP (3.19) with only one quadratic function has been solved so far, that is, the efficient and Pareto sets have been computed. Using the mpLCP method, we can now solve this problem with any number of quadratic functions. To apply the mpLCP method, MOPOP needs to be scalarized, which can be accomplished with any of the scalarizations presented in Section 3.2. In Section 3.4.1, we apply the modified hybrid method and, in order to gain more insight into the MOPOP, we first solve the resulting par-SOP analytically. In Section 3.4.2, we solve this par-SOP numerically with the mpLCP method. We then scalarize the MOPOP with the weighted-sum method and solve the resulting par-SOP also with the mpLCP method. Finally, we compare the numerical solutions obtained from solving the two par-SOPs.

3.4.1 Analytical Solution to the Modified Hybrid Problem for the MOPOP

To obtain an analytical solution to MOPOP (3.19) we make the following assumptions.

Assumption 3.4.1. *1. Matrix Q_i is symmetric and PD for all $i = 1, \ldots, s$.*

2. Vectors $\boldsymbol{p}_j$ and $\mathbf{1}$ are linearly independent for all $j = s+1, \ldots, r$.

The modified hybrid problem with the inequality constraints for MOPOP (3.19) assumes the form of a parametric QP (par-QP)

$$\hat{\sigma}(\boldsymbol{\gamma}, \boldsymbol{\epsilon}) = \min \tfrac{1}{2}\mathbf{x}^T Q(\boldsymbol{\gamma})\mathbf{x}$$

$$\text{s.t. } \mathbf{x} \in \ \mathcal{X}_h^{\leq}(\boldsymbol{\epsilon}) = \left\{\mathbf{x} \in \mathbb{R}^n : -\boldsymbol{p}_j^T\mathbf{x} \leq \epsilon_j, \ \ j = s+1, \ldots, r, \mathbf{1}^T\mathbf{x} = 1, \mathbf{x} \geqq \mathbf{0}\right\}, \tag{3.20}$$

where $\boldsymbol{\gamma} \in \Gamma' = \left\{\boldsymbol{\gamma} \in \mathbb{R}^{s-1} : \boldsymbol{\gamma} \geq \mathbf{0}, \sum_{i=1}^{s-1} \gamma_i \leq 1\right\}$, $\boldsymbol{\epsilon} \in \mathbb{E}_h^{\leq} = \left\{\boldsymbol{\epsilon} \in \mathbb{R}^{r-s} : \mathcal{X}_h^{\leq}(\boldsymbol{\epsilon}) \neq \emptyset\right\}$, and $Q(\boldsymbol{\gamma}) = \sum_{i=1}^{s-1} \gamma_i Q_i + \left(1 - \sum_{i=1}^{s-1} \gamma_i\right) Q_s$ is an $n \times n$ symmetric PD matrix for $\boldsymbol{\gamma} \in \Gamma'$.

By definition of $\mathbb{E}_h^{\leq}$, par-QP (3.20) is feasible for all $(\boldsymbol{\gamma}, \boldsymbol{\epsilon}) \in \Gamma' \times \mathbb{E}_h^{\leq}$. The optimal solution to (3.20) includes a partition of the parameter space $\Gamma' \times \mathbb{E}_h^{\leq}$ into invariancy regions, and the optimal solution function $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}(\boldsymbol{\gamma}, \boldsymbol{\epsilon})$ and the optimal objective value functions $\hat{\sigma}(\boldsymbol{\gamma}, \boldsymbol{\epsilon})$ for each region. Due to Assumption 3.4.1.1 and based on Corollary 3.2.4.2.2, an optimal solution to problem (3.20) is efficient to problem (3.19).

To keep the analysis simple, we assume that at optimality of problem (3.20) for a subset Θ of the parameter space $\Gamma' \times \mathbb{E}_h^{\leqq}$, all the inequality $\boldsymbol{\epsilon}$-constraints are active (i.e., all the investments bring the anticipated level of expected return) and all the nonnegativity constraints are inactive (i.e., there are no zero investments).

$$\Theta = \{(\boldsymbol{\gamma}, \boldsymbol{\epsilon}) \in \Gamma' \times \mathbb{E}_h^{\leqq} : \hat{\mathbf{x}}(\boldsymbol{\gamma}, \boldsymbol{\epsilon}) > \mathbf{0}, -\boldsymbol{p}_j^T \hat{\mathbf{x}} = \epsilon_j, \;\; j = s+1, \ldots, r\} \tag{3.21}$$

Having the $\boldsymbol{\epsilon}$-constraints active, based on Corollary 3.2.3.1.1, the optimality of problem (3.20) with equality constraints $\mathbf{x} \in \mathcal{X}_h^{=}(\boldsymbol{\epsilon}) = \left\{\mathbf{x} \in \mathbb{R}^n : -\boldsymbol{p}_j^T \mathbf{x} = \epsilon_j, \;\; j = s+1, \ldots, r, \mathbf{1}^T \mathbf{x} = 1, \mathbf{x} \geqq \mathbf{0}\right\}$ has also been achieved. Applying the KKT optimality conditions to par-QP (3.20) treated as a proper equality constrained problem (3.5), we obtain the following system of parametric linear equations:

$$Q(\boldsymbol{\gamma})\mathbf{x} - \textstyle\sum_{j=s+1}^{r} w_j(-\boldsymbol{p}_j) - v\mathbf{1} \;\; = \mathbf{0} \tag{3.22}$$

$$\epsilon_j - (-\boldsymbol{p}_j)^T \mathbf{x} = 0 \quad j = s+1, \ldots, r \tag{3.23}$$

$$1 - \mathbf{1}^T \mathbf{x} \;\; = 0, \tag{3.24}$$

where $w_j, j = s+1, \ldots, r$, and v are the dual variables associated with $\boldsymbol{\epsilon}$-constraints and the linear constraint respectively. Solving (3.22) for $\mathbf{x}$, we get

$$\mathbf{x} \quad = \textstyle\sum_{j=s+1}^{r} w_j Q(\boldsymbol{\gamma})^{-1}(-\boldsymbol{p}_j) + vQ(\boldsymbol{\gamma})^{-1}\mathbf{1}. \tag{3.25}$$

Substituting (3.25) into (3.23) and (3.24) yields another system of parametric linear equations

$$\Psi(\boldsymbol{\gamma}) \begin{bmatrix} w_{s+1} \\ \vdots \\ w_r \\ v \end{bmatrix} = \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_r \\ 1 \end{bmatrix}, \tag{3.26}$$

where

$$\Psi(\boldsymbol{\gamma}) = \begin{bmatrix} \boldsymbol{p}_{s+1}^T Q(\boldsymbol{\gamma})^{-1}\boldsymbol{p}_{s+1} & \boldsymbol{p}_{s+1}^T Q(\boldsymbol{\gamma})^{-1}\boldsymbol{p}_{s+2} & \dots & \boldsymbol{p}_{s+1}^T Q(\boldsymbol{\gamma})^{-1}\boldsymbol{p}_r & -\boldsymbol{p}_{s+1}^T Q(\boldsymbol{\gamma})^{-1}\mathbf{1} \\ \boldsymbol{p}_{s+1}^T Q(\boldsymbol{\gamma})^{-1}\boldsymbol{p}_{s+2} & \boldsymbol{p}_{s+2}^T Q(\boldsymbol{\gamma})^{-1}\boldsymbol{p}_{s+2} & \dots & \boldsymbol{p}_{s+2}^T Q(\boldsymbol{\gamma})^{-1}\boldsymbol{p}_r & -\boldsymbol{p}_{s+2}^T Q(\boldsymbol{\gamma})^{-1}\mathbf{1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \boldsymbol{p}_{s+1}^T Q(\boldsymbol{\gamma})^{-1}\boldsymbol{p}_r & \boldsymbol{p}_{s+2}^T Q(\boldsymbol{\gamma})^{-1}\boldsymbol{p}_r & \dots & \boldsymbol{p}_r^T Q(\boldsymbol{\gamma})^{-1}\mathbf{p}_r & -\boldsymbol{p}_r^T Q(\boldsymbol{\gamma})^{-1}\mathbf{1} \\ -\boldsymbol{p}_{s+1}^T Q(\boldsymbol{\gamma})^{-1}\mathbf{1} & -\mathbf{p}_{s+2}^T Q(\boldsymbol{\gamma})^{-1}\mathbf{1} & \dots & -\boldsymbol{p}_r^T Q(\boldsymbol{\gamma})^{-1}\mathbf{1} & \mathbf{1}^T Q(\boldsymbol{\gamma})^{-1}\mathbf{1} \end{bmatrix}$$

is an $(r-s+1)\times(r-s+1)$ matrix, which we show is PD.

Proposition 3.4.2. *Matrix $\Psi(\boldsymbol{\gamma})$ is PD for $\boldsymbol{\gamma} \in \Gamma'$.*

Proof. Note that $\Psi(\boldsymbol{\gamma}) = M^T Q(\boldsymbol{\gamma})^{-1} M$ with $M = [-\boldsymbol{p}_{s+1}, -\boldsymbol{p}_{s+2}, \dots, -\boldsymbol{p}_r, \mathbf{1}] \in \mathbb{R}^{n\times(r-s+1)}$.

Let $\mathbf{t} \in \mathbb{R}^{r-s+1}$ such that $\mathbf{t} \neq \mathbf{0}$, then

$$\begin{aligned}\mathbf{t}^T\Psi(\boldsymbol{\gamma})\mathbf{t} &= \mathbf{t}^T M^T Q(\boldsymbol{\gamma})^{-1} M\mathbf{t} \\ &= (M\mathbf{t})^T Q(\boldsymbol{\gamma})^{-1}(M\mathbf{t}) \\ &> 0,\end{aligned}$$

where the inequality results from the facts that, under Assumption 3.4.1.2, $M\mathbf{t} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$, and $Q(\boldsymbol{\gamma})^{-1}$ exists and is PD for $\boldsymbol{\gamma} \in \Gamma'$. □

The solution to system (3.26) is given as

$$\begin{bmatrix} \hat{w}_{s+1} \\ \vdots \\ \hat{w}_r \\ \hat{v} \end{bmatrix} = \Psi(\gamma)^{-1} \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_r \\ 1 \end{bmatrix} \tag{3.27}$$

and can be substituted into (3.25) to obtain a unique optimal solution $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}(\boldsymbol{\gamma}, \boldsymbol{\epsilon})$ to problem (3.20) and an efficient solution to MOPOP (3.19) in explicit parametric form. Notice that $\hat{\mathbf{x}}, \hat{\boldsymbol{w}}$ and $\hat{v}$ obtained from (3.25) and (3.27) are linear functions of $\boldsymbol{\epsilon}$ and polynomial functions of $\boldsymbol{\gamma}$.

We calculate the value of the objective function $\hat{\sigma}(\boldsymbol{\gamma}, \boldsymbol{\epsilon})$ at the optimal solution $\hat{\boldsymbol{x}}$. Multiplying (3.22) by $\frac{1}{2}\mathbf{x}^T$ we have

$$\tfrac{1}{2}\mathbf{x}^T Q(\boldsymbol{\gamma})\mathbf{x} \;\; = \tfrac{1}{2}\big(\textstyle\sum_{j=s+1}^{r} w_j \mathbf{x}^T(-\mathbf{p}_j) + v\mathbf{x}^T\mathbf{1}\big), \tag{3.28}$$

and then using (3.23) and (3.24) we obtain

$$\hat{\sigma}(\boldsymbol{\gamma}, \boldsymbol{\epsilon}) = \frac{1}{2}\left(\sum_{j=s+1}^{r} \hat{w}_j(\boldsymbol{\gamma}, \boldsymbol{\epsilon})\epsilon_j + \hat{v}(\boldsymbol{\gamma}, \boldsymbol{\epsilon}) \right). \tag{3.29}$$

Rewriting equation (3.29) we have $\hat{\sigma}(\boldsymbol{\gamma}, \boldsymbol{\epsilon}) = \frac{1}{2}\begin{bmatrix} \boldsymbol{\epsilon} & 1 \end{bmatrix}\begin{bmatrix} \hat{\boldsymbol{w}} \\ \hat{v} \end{bmatrix}$, and using (3.26) we calculate

$$\hat{\sigma}(\boldsymbol{\gamma}, \boldsymbol{\epsilon}) = \frac{1}{2}\begin{bmatrix} \boldsymbol{\epsilon} & 1 \end{bmatrix} \Psi(\gamma)^{-1} \begin{bmatrix} \boldsymbol{\epsilon} \\ 1 \end{bmatrix}, \tag{3.30}$$

which gives the value of the minimum weighted risk associated with the efficient portfolio $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}(\boldsymbol{\gamma}, \boldsymbol{\epsilon})$ for every $(\boldsymbol{\gamma}, \boldsymbol{\epsilon}) \in \Theta$.

Note that $\hat{\sigma}$ is a strictly convex quadratic function of $\boldsymbol{\epsilon}$ since $\Psi(\gamma)^{-1}$ is PD. Equating to zero the partial derivative of $\hat{\sigma}(\boldsymbol{\gamma}, \boldsymbol{\epsilon})$ with respect to $\epsilon_j, j = s+1, \ldots, r$, we calculate $\hat{\epsilon}_j$ from $\frac{\partial \hat{\sigma}(\boldsymbol{\gamma}, \boldsymbol{\epsilon})}{\partial \epsilon_j} = 0$ and obtain $\hat{\epsilon}_j = \hat{\epsilon}_j(\boldsymbol{\gamma}), j = s+1, \ldots, r$, which is a function representing the relationship between the parameters $\gamma_i, i = 1, \ldots, s-1$, weighing the risk objective functions and the parameters $\epsilon_j, j = s+1, \ldots, r$, determining the level of the expected return.

Our analysis leads to two corollaries describing the meaning of the optimal parametric solutions to par-QP (3.20) in the context of MOPOP (3.19).

Corollary 3.4.2.1. *Let Assumption 3.4.1 hold and* $\boldsymbol{\gamma} = \bar{\boldsymbol{\gamma}}$. *The function* $\hat{\boldsymbol{x}}(\bar{\boldsymbol{\gamma}}, \boldsymbol{\epsilon})$ *is the efficient portfolio to MOPOP (3.19) yielding the expected return* $\boldsymbol{\epsilon} \in \Theta(\bar{\boldsymbol{\gamma}}) = \{(\boldsymbol{\gamma}, \boldsymbol{\epsilon}) \in \Theta : \boldsymbol{\gamma} = \bar{\boldsymbol{\gamma}}\}$ *and generating the minimum weighted risk* $\hat{\sigma}(\bar{\boldsymbol{\gamma}}, \boldsymbol{\epsilon})$. *The portfolio* $\hat{\boldsymbol{x}}(\bar{\boldsymbol{\gamma}}, \hat{\boldsymbol{\epsilon}}), \hat{\boldsymbol{\epsilon}} = \arg\min_{\boldsymbol{\epsilon} \in \Theta(\bar{\boldsymbol{\gamma}})} \hat{\sigma}(\bar{\boldsymbol{\gamma}}, \boldsymbol{\epsilon})$, *is the efficient portfolio that generates the lowest minimum weighted risk* $\hat{\sigma}(\bar{\boldsymbol{\gamma}}, \hat{\boldsymbol{\epsilon}})$.

Corollary 3.4.2.2. *Let Assumption 3.4.1 hold and* $\boldsymbol{\epsilon} = \bar{\boldsymbol{\epsilon}}$*. The function* $\hat{\boldsymbol{x}}(\boldsymbol{\gamma}, \bar{\boldsymbol{\epsilon}})$ *is the efficient portfolio to MOPOP (3.19) yielding the expected return* $\bar{\boldsymbol{\epsilon}}$ *and generating the minimum weighted risk of* $\hat{\sigma}(\boldsymbol{\gamma}, \bar{\boldsymbol{\epsilon}})$ *for every* $\boldsymbol{\gamma} \in \Theta(\bar{\boldsymbol{\epsilon}}) = \{(\boldsymbol{\gamma}, \boldsymbol{\epsilon}) \in \Theta : \boldsymbol{\epsilon} = \bar{\boldsymbol{\epsilon}}\}$.

We illustrate our observations on a triobjective example having the form of MOPOP (3.19).

3.4.2 Example 2: A Triobjective Portfolio Optimization Problem

We apply the mpLCP method [3] to a portfolio optimization problem that we scalarize with the modified hybrid method and with the weighted-sum method, and compare the obtained efficient solutions. Consider the following triobjective portfolio example with two quadratic objectives modeling two different types of risk and one linear objective modeling the expected return.

$$\begin{array}{ll} \min & \Big[f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})\Big] = \Big[\frac{1}{2}\mathbf{x}^T Q_1 \mathbf{x}, \frac{1}{2}\mathbf{x}^T Q_2 \mathbf{x}, -\boldsymbol{p}_3^T \mathbf{x}\Big] \\ \text{s.t.} & \mathbf{x} \in \mathcal{X}, \end{array} \tag{3.31}$$

where $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^3 : \mathbf{1}^T \mathbf{x} - 1 = 0, \mathbf{x} \geqq \mathbf{0}\}$, and

$$Q_1 = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 2 & 0 \\ -1 & 0 & 2.5 \end{bmatrix}, \quad Q_2 = \begin{bmatrix} 3 & -1 & 0 \\ -1 & 4 & 1 \\ 0 & 1 & 3.5 \end{bmatrix}, \quad \boldsymbol{p}_3 = \begin{bmatrix} -13.5 \\ 20 \\ 16 \end{bmatrix}.$$

3.4.2.1 Solution with the modified hybrid method

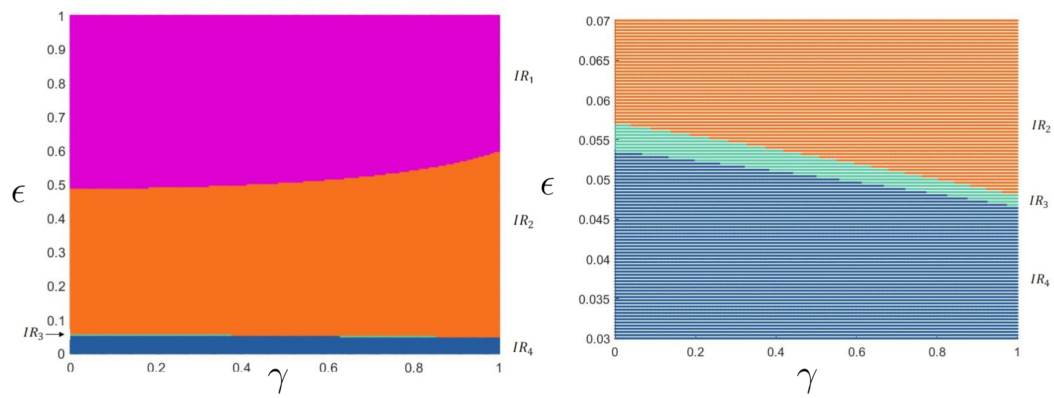
Applying the modified hybrid method with the inequality constraint, the resulting par-SOP assumes the form:

$$\begin{aligned} \hat{\sigma}(\boldsymbol{\gamma}, \boldsymbol{\epsilon}) = \min \quad & \tfrac{1}{2}\mathbf{x}^T Q(\gamma)\mathbf{x} \\ \text{s.t.} \quad & \mathbf{x} \in \mathcal{X}_h^{\leq}(\epsilon), \end{aligned} \tag{3.32}$$

where $\mathcal{X}_h^{\leq}(\epsilon) = \{\mathbf{x} \in \mathbb{R}^3 : \ -\boldsymbol{p}_3^T\mathbf{x} \leq \epsilon, \ \mathbf{1}^T\mathbf{x} = 1, \mathbf{x} \geqq \mathbf{0}\}$, $Q(\gamma) = \begin{bmatrix} 3-2\gamma & \gamma-1 & -\gamma \\ \gamma-1 & 4-2\gamma & 1-\gamma \\ -\gamma & 1-\gamma & 7/2-\gamma \end{bmatrix}$, $\gamma \in \Gamma' = [0,1]$ and $\epsilon \in \mathbb{E}_h^{\leq} = \{\epsilon \in \mathbb{R} : \mathcal{X}_h^{\leq}(\epsilon) \neq \emptyset\}$.

To be able to solve this problem with the mpLCP method, we reformulate the equality constraints into inequalities to have the form of $A\mathbf{x} \leq \mathbf{b}(\epsilon)$, where $A = \begin{bmatrix} -\boldsymbol{p}_3 & \mathbf{1} & -\mathbf{1} \end{bmatrix}^T$, $\mathbf{b}(\epsilon) = \begin{bmatrix} \epsilon & 1 & -1 \end{bmatrix}^T$, and $\mathbf{1} \in \mathbb{R}^3$. We emphasize that any choice of epsilon such that $\mathbb{E}_h^{\leq} \neq \emptyset$ is good since, by Corollary 3.2.2.1 (2), the optimal solutions to (3.32) are efficient to (4.34). To obtain the bounds for ϵ that are consistent with γ we use normalization such that $\epsilon^{nor} = \frac{\epsilon - \epsilon^{min}}{\epsilon^{max} - \epsilon^{min}}$, where ϵ^{min} and ϵ^{max} are computed by solving two linear programs: $\epsilon^{min} = \min\{-\mathbf{p}_3^T\mathbf{x} : \mathbf{1}^T\mathbf{x} = 1, \mathbf{x} \geqq \mathbf{0}\}$ and $\epsilon^{max} = \max\{-\mathbf{p}_3^T\mathbf{x} : \mathbf{1}^T\mathbf{x} = 1, \mathbf{x} \geqq \mathbf{0}\}$. We obtain $\epsilon^{min} = -20$ and $\epsilon^{max} = 13.5$ and $\mathbb{E}_h^{\leq} = \{\epsilon \in \mathbb{R} : -20 \leq \epsilon \leq 13.5\}$ which models a maximum average gain of 20 and a maximum average loss of 13.5. Hence, $\epsilon^{nor} = \frac{\epsilon+20}{33.5}$ and $\mathbb{E}_h^{\leq}$ can be written as $\mathbb{E} = \{\epsilon^{nor} \in \mathbb{R} : 0 \leq \epsilon^{nor} \leq 1\}$. The parameter space can now be written as $\Gamma' \times \mathbb{E} = \{(\gamma, \epsilon^{nor}) : \gamma \in [0,1] \text{ and } \epsilon^{nor} \in [0,1]\}$.

At optimality of problem (3.32), the parameter space is partitioned into four invariancy regions $IR_i, i = 1, \ldots, 4$, that are depicted in Figure 3.3a. Since region IR_3 is not visible, it is enlarged in Figure 3.3b (the colors in Figure 3.3b are lighter than in Figure 3.3a due to the magnification). The analytical descriptions of the invariancy regions and the efficient solution functions are given in Supplementary Table 3.5. In Figures 3.3a, 3.3b, and Supplementary Table 3.5, to keep the notation simple we write ϵ but mean ϵ^{nor}. Note that in this example the reported optimal solution functions in the invariancy regions IR_3 and IR_4 are identical, while the optimal solution functions $(\hat{\mathbf{w}}(\gamma, \epsilon^{nor}), \hat{\mathbf{z}}(\gamma, \epsilon^{nor}))$ to the solved mpLCP, which are not reported, are not identical.



(a) (b)

Figure 3.3: (a) Four invariancy regions $IR_i, i = 1, \ldots, 4$, in the parameter space of Example 2 scalarized with the modified hybrid method ($\epsilon = \epsilon^{norm}$); (b) Enlarged IR_3.

Based on Corollaries 3.4.2.1 and 3.4.2.2 we analyze the obtained solutions in two decision-making scenarios to examine the role of the parameters. The corollaries can only be used for $(\gamma, \epsilon) \in \Theta$. Based on the obtained results reported in Supplementary Table 3.5, $\hat{x}_i(\gamma, \epsilon^{nor}) > 0, i = 1, 2, 3$ in the invariancy regions IR_1, and IR_2 but not in $IR_2 \cap IR_3$. To check whether the $\epsilon-$constraint is active we can proceed in two different ways. We may quickly check the optimal value of the slack variable s that was added to the the $\epsilon-$constraint when problem (3.32) was reformulated to an mpLCP, as described in Section 3.3.4. The $\epsilon-$constraint is active in an invariancy region if $\hat{s}(\gamma, \epsilon^{nor}) = 0$ at optimality in this region. Alternatively, we adjust the $\epsilon-$constraint for ϵ^{nor} as follows

$$-\mathbf{p}_3^T\mathbf{x} \leq 33.5\epsilon^{nor} - 20 \tag{3.33}$$

and use the optimal solution functions listed in Supplementary Table 3.5 to examine whether the inequality in (3.33) becomes active. We observe that the $\epsilon-$constraint is active in the invariancy regions IR_2, IR_3 and IR_4. We obtain $\Theta = \{(\gamma, \epsilon) \in \Gamma' \times \mathbb{E} : (\gamma, \epsilon) \in IR_2 \setminus (IR_2 \cap IR_3)\}$.

In the invariancy region IR_1, ϵ is large enough so that the $\epsilon-$constraint is not active while the optimal solution in this region rightly does not depend on ϵ. However, Corollary 3.2.2.1 applies and this optimal solution is efficient to (4.34). We also solved (3.32) for ϵ in the smaller interval of $-20 \leq \epsilon \leq 0.04$ that we obtained by calculating $f_3(\arg\min_{\mathbf{x}\in X} f_1(\mathbf{x})) = f_3(0.56, 0.12, 0.32) = 0.04$ and $f_3(\arg\min_{\mathbf{x}\in X} f_2(\mathbf{x})) = f_3(0.46, 0.33, 0.21) = -3.66$ and dropping -3.66 because the interval $[-20, -3.66]$ would not contain 0.04. The resulting partition of the parameter space is almost the same as that depicted in Figure 3.3a with the only difference that the invariancy regions cover proportionally different portions of this space because of a different normalization formula $\frac{\epsilon+20}{20.04}$.

Due to the normalization we also have to adjust formula (4.36) to later correctly calculate the optimal objective function. We have

$$\hat{\sigma}(\gamma, \epsilon^{nor}) = 1/2 \begin{bmatrix} (33.5\epsilon^{nor} - 20) & 1 \end{bmatrix} \Psi(\gamma)^{-1} \begin{bmatrix} (33.5\epsilon^{nor} - 20) \\ 1 \end{bmatrix}, \tag{3.34}$$

where $\Psi(\gamma) = \begin{bmatrix} \boldsymbol{p}_3^T Q(\gamma)^{-1} \boldsymbol{p}_3 & -\boldsymbol{p}_3^T Q((\gamma)^{-1} \mathbf{1} \\ -\boldsymbol{p}_3^T Q(\gamma)^{-1} \mathbf{1} & \mathbf{1} Q(\gamma)^{-1} \mathbf{1} \end{bmatrix}$
and $\Psi(\gamma)^{-1} = \frac{2}{(3314\gamma - 41955)} \begin{bmatrix} 2(6\gamma^2 + 36\gamma - 67) & (192\gamma^2 + 301\gamma - 491) \\ (192\gamma^2 + 301\gamma - 491) & (1415\gamma^2 + 17977\gamma - 24029) \end{bmatrix}$.

In the first scenario, assume that the decision maker (DM) applies equal weights to both risk measures, that is, $\bar{\gamma} = 0.5$. Based on the solutions reported in Figure 3.3a and Supplementary Table 3.5, the associated optimal solution $\hat{\mathbf{x}} = \hat{\mathbf{x}}(0.5, \epsilon^{nor})$ passes through IR_2 and is given in Table 3.1.

Table 3.1: Efficient solution functions for Example 2 and $\bar{\gamma} = 0.5$ in IR_2

$$\hat{\mathbf{x}}(0.5, \epsilon^{nor}) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{c} x_1 = (21105\epsilon^{nor})/20149 - 1116/20149 \\ x_2 = 23837/40298 - (26197\epsilon^{nor})/40298 \\ x_3 = 18693/40298 - (16013\epsilon^{nor})/40298 \end{array} \right\} \text{ for } \epsilon^{nor} \in (0.0529; 0.5051] \text{ in } IR_2.$$

Using (3.34), we calculate the optimal objective function:

$$\hat{\sigma}(0.5, \epsilon^{nor}) = 0.5 \begin{bmatrix} (33.5\epsilon^{nor} - 20) & 1 \end{bmatrix} \begin{bmatrix} 95/20149 & 131/9024 \\ 131/9024 & 527/723 \end{bmatrix} \begin{bmatrix} (33.5\epsilon^{nor} - 20) \\ 1 \end{bmatrix}$$

$$= (426455(\epsilon^{nor})^2)/161192 - (215405\epsilon^{nor})/80596 + 163947/161192,$$

which gives the minimum weighted risk assumed by the efficient solution $\hat{\mathbf{x}}(0.5, \epsilon^{nor})$ for each $\epsilon^{nor} \in (0.0529, 0.5051]$. By Corollary 3.4.2.1, among these efficient portfolios we find the one that yields the lowest minimum risk. We calculate $\hat{\epsilon}^{nor}(\bar{\gamma}) = \hat{\epsilon}^{nor}(0.5) = 0.5051$, and note that this efficient portfolio is located on the boundary of IR_2 with IR_1 and is given by $(\hat{x}_1, \hat{x}_2, \hat{x}_3) = (9/19, 5/19, 5/19)$. We then obtain $\hat{\epsilon} = -3.0789$ and the lowest minimum risk value of $\hat{\sigma}(\gamma = 0.5, \hat{\epsilon} = -3.0789) = 0.3421$. The DM concludes that weighing the two types of risk equally, the proportions of the capital allocated to the three securities $(9/19, 5/19, 5/19)$ give the expected return of 3.0789 and assume the lowest minimum weighted risk of 0.3421. The DM can repeat this scenario by weighing the risk with any value of $\bar{\gamma}$ available in Θ and computing the associated efficient solution functions $\hat{\mathbf{x}}(\bar{\gamma}, \epsilon)$ and the minimum weighted risk value $\hat{\sigma}(\bar{\gamma}, \epsilon)$ for all $\epsilon \in \Theta(\bar{\gamma})$, and identifying the efficient portfolio $\hat{\mathbf{x}}(\bar{\gamma}, \hat{\epsilon})$ that yields the lowest minimum risk.

In the second scenario, assume that the DM wants to investigate the model hoping to make a much higher expected return of 9 which makes $\bar{\epsilon} = -9$ and $\bar{\epsilon}^{nor} = 0.3284$. The corresponding efficient solutions are located in IR_2 and are listed in Table 3.2.

Table 3.2: Efficient solution functions for Example 2 and $\bar{\epsilon}^{nor} = 0.3284$ in IR_2

$$\hat{\mathbf{x}}(\gamma, 0.3284) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{c} x_1 = (1204\gamma - 12230)/(3314\gamma - 41955) \\ x_2 = (3080\gamma - 16775)/(3314\gamma - 41955) \\ x_3 = (-970\gamma + 12950)/(3314\gamma - 41955) \end{array} \right\} \text{ for } \gamma \in [0,1] \text{ in } IR_2.$$

We again use (3.34) to calculate the optimal objective function:

$$\hat{\sigma}(\gamma, 0.3284) = -(1069\gamma^2 - 18391\gamma + 26045)/(3314\gamma - 41955) \qquad \text{for } \gamma \in [0,1], \tag{3.35}$$

which, by Corollary 3.4.2.2, gives the minimum risk of the efficient solution $\hat{\mathbf{x}}(\gamma, 0.3284)$ for each $\gamma \in [0,1]$. The DM concludes that assuming the expected return of 9, the proportions of the capital allocated to the three securities in the efficient portfolio $\hat{\mathbf{x}}(\gamma, 0.3284)$ give the minimum weighted risk calculated in (3.35) for $\gamma \in [0,1]$. For example, for $\bar{\gamma} = 0.5$, the minimum risk $\hat{\sigma}(0.5, 0.3284) = 0.4248$, which is higher than the risk in the first scenario, as expected. The DM can repeat the second scenario by fixing the expected return at any value of $\bar{\epsilon}$ available in Θ and computing the associated efficient solution functions $\hat{\mathbf{x}}(\gamma, \bar{\epsilon})$ and the minimum weighted risk value $\hat{\sigma}(\gamma, \bar{\epsilon})$ for any $\gamma \in \Theta(\bar{\epsilon})$.

The Supporting Information section includes figures depicting the graphs of all optimal solution functions for Example 2 solved with the modified hybrid method. In all these figures, the colors match the colors used in Figure 3.3a to help the reader to associate the invariancy regions with the corresponding parts of the function graphs. The graph of the optimal objective function $\hat{\sigma}(\gamma, \epsilon^{nor})$ of problem (3.32), which is the minimum weighted risk function for the efficient solutions to problem (4.34), is depicted in Supplementary Figure 3.8. The graph of the optimal constraint function $-\mathbf{p}_3^T \hat{\mathbf{x}}(\gamma, \epsilon^{nor})$ for problem (3.32), which is a Pareto expected return function for problem (4.34), is depicted in Supplementary Figure 3.9. These graphs can be analyzed by fixing one of the two parameters, either γ or ϵ and tracing the resulting path to obtain the Pareto expected return for a given weighted risk or to get the minimum weighted risk for a given expected return. One can also observe that for a fixed γ, the weighted risk and expected return remain constant in IR_1 (cf. $\hat{\mathbf{x}}(0.5, \epsilon^{nor}) = (9/19, 5/19, 5/19)$ in IR_1) but they both increase in the other regions, as expected. Figure 3.4 depicts the graph of the optimal objective value function for problem (3.32) for $\bar{\gamma} = 0.5$ and $\epsilon \in IR_2 \cap \Theta(\bar{\gamma})$, which is a subset of the Pareto set of the biobjective problem for which (3.32) is the ϵ-constraint scalarization. Supplementary Figures 3.10, 3.11, and 3.12 show the graphs of the optimal solution functions to problem (3.32), which are the efficient portfolios to problem (4.34). In Supplementary Figures 3.8, 3.11, and 3.12, parts of the graphs have very steep gradients that cause numerical difficulties and incomplete coloring.

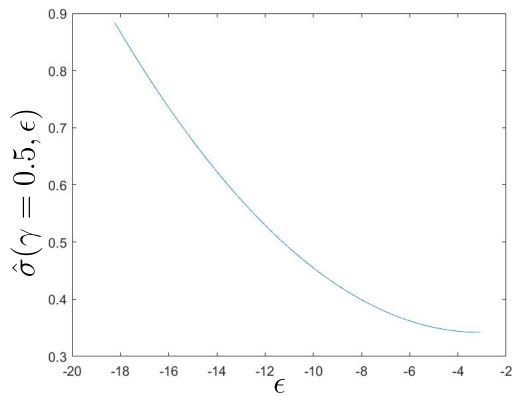


Figure 3.4: Minimum weighted risk function for $\bar{\gamma} = 0.5$ and $\epsilon \in IR_2 \cap \Theta(\bar{\gamma})$ for Example 2 scalarized with the modified hybrid method

3.4.2.2 Solution with the weighted-sum method

Applying the weighted-sum scalarization to problem (4.34), the resulting par-SOP assumes the form

$$
\begin{aligned}
\min \quad & \tfrac{1}{2}\mathbf{x}^T Q(\boldsymbol{\lambda})\mathbf{x} - (1-\lambda_1-\lambda_2)\boldsymbol{p}_3^T\mathbf{x} \\
\text{s.t.} \quad & A\mathbf{x} \leqq \mathbf{b} \\
& \mathbf{x} \geqq \mathbf{0}
\end{aligned}
\tag{3.36}
$$

with

$$
Q(\boldsymbol{\lambda}) = \begin{bmatrix} \lambda_1 + 3\lambda_2 & -\lambda_2 & -\lambda_1 \\ -\lambda_2 & 2\lambda_1 + 4\lambda_2 & \lambda_2 \\ -\lambda_1 & \lambda_2 & (5\lambda_1)/2 + (7\lambda_2)/2 \end{bmatrix},
$$

$$
\boldsymbol{\lambda} \in \Lambda' = \{\boldsymbol{\lambda} \in \mathbb{R}^2 : \lambda_1, \lambda_2 \geq 0, \lambda_1 + \lambda_2 \leq 1\},
$$

$$
A = \begin{bmatrix} 1 & 1 & 1 \\ -1 & -1 & -1 \end{bmatrix}, \text{ and } \mathbf{b} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}.
$$

Solving problem (3.36) with the mpLCP method, we obtain the solution to problem (4.34) in the form of four invariancy regions in the parameter space of (λ_1, λ_2) (see Figure 3.5) and the associated efficient solution functions $\mathbf{x}(\lambda_1, \lambda_2)$. The analytical descriptions of the invariancy regions and the efficient solution functions are given in Supplementary Table 4.8. Similar to the previous case, the reported optimal solutions functions in the invariancy regions IR_2 and IR_3 are identical, while the optimal solution functions $(\hat{\mathbf{w}}(\boldsymbol{\lambda}), \hat{\mathbf{z}}(\boldsymbol{\lambda}))$ to the solved mpLCP, which are not reported, are not identical. Given these results, the DM may choose any feasible values of the weighting parameters and compute the resulting efficient portfolio and its optimal objective which, however, is not meaningful because different types of objective functions have been aggregated into one objective. Obviously, the weighted-sum formulation does not support the decision making process at the level the modified hybrid formulation does.

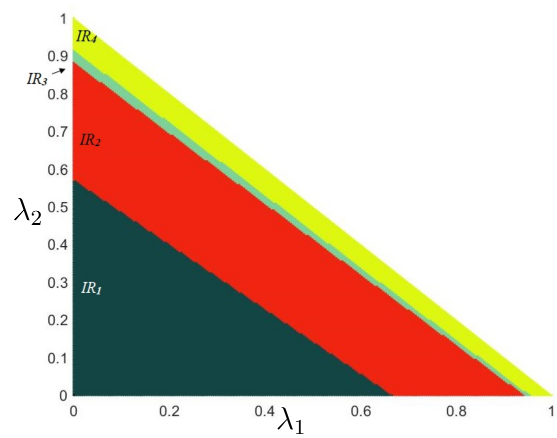


Figure 3.5: Invariancy regions in the parameter space of Example 2 scalarized with the weighted-sum method

3.4.2.3 Comparison

We now compare the results obtained when solving problem (4.34) with the modified hybrid method and the weighted-sum method. Comparing Figures 3.3a and 3.5, and Supplementary Tables 3.5 and 4.8, we observe that the two partitions of the parameters space and the analytical description of the efficient solution functions are very different. Figures 3.6 and 3.7 depict the Pareto set in the three-dimensional objective space of problem (4.34) obtained by solving problem (3.32) and problem (3.36), respectively. To provide some evidence that the two graphs depict the same Pareto set, we selected points A, B, C, and D in each graph (see Figures 3.6 and 3.7) and report their coordinates in Table 3.3. Due to different computations the numbers are not identical but very close to each other.

In Supplementary Table 4.8 we additionally observe that the denominator of the efficient solutions $\mathbf{x}(\lambda)$ can reach zero for $\lambda_1 = \lambda_2 = 0$. To overcome this situation when making the objective space graph using the weighted-sum method, we assumed that $\lambda_1 = \lambda_2 = \delta > 0$ where δ is a relatively small number.

Table 3.3: Coordinates of four Pareto points for Example 2 computed with the modified hybrid method and the weighted-sum method

Pareto point	Modified hybrid method	Weighted-sum method
A	(1, 2, -20)	(1, 2, -20)
B	(0.557, 1.226, -18.32)	(0.5558, 1.216, -18.27)
C	(0.1728, 0.5299, -3.664)	(0.1728, 0.5298, -3.664)
D	(0.12, 0.6496, 0.04)	(0.12, 0.6496, 0.04001)

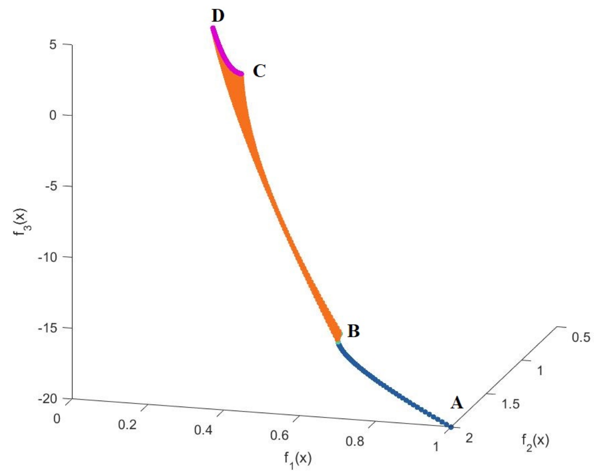


Figure 3.6: The Pareto set for Example 2 solved with the modified hybrid method

3.5 Conclusion

In this Chapter, we focused on (strictly) convex MOPs and reviewed properties of three scalarization methods for such problems. Treating the resulting (strictly) convex SOPs as parametric problems, we carried over the results from parametric optimization that are relevant to the MOPs. We also proposed a modified hybrid method, a variation of the hybrid scalarization, which is useful for MOPs whose objective functions come in groups. Believing that MOQPs make up an important class of MOPs, we reviewed the-state-of-the-art-algorithms for solving convex MOQPs and concluded that the mpLCP method [3] is currently the best method for computing exact solutions to convex MOQPs.

Finally, we applied the modified hybrid scalarization and the mpLCP method to a multiobjective portfolio optimization problem. The scalarization allows for a

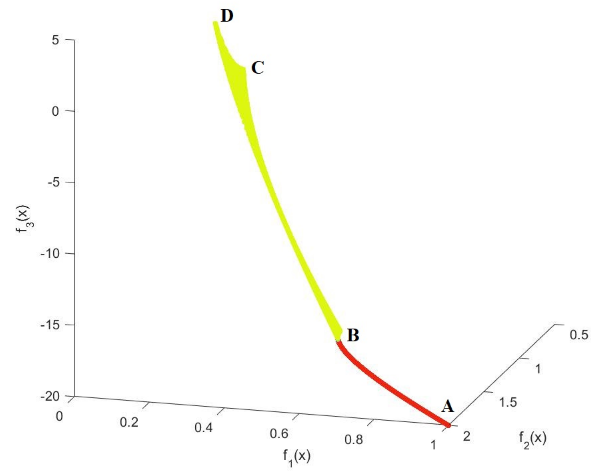


Figure 3.7: The Pareto set for Example 2 solved with the weighted-sum method

parametric analysis that reveals interplay between two independent parameters: one of them aggregates the quadratic risk functions and the other sets a level of the expected return. The mpLCP method can solve the portfolio problem with any number of quadratic risk functions, a feature that has not yet been reported in the literature.

In our future research, we intend to apply the modified hybrid scalarization and the mpLCP algorithm to other real-life decision problems that are modeled as convex MOQPs and are likely to benefit from two independent scalarizing parameters.

Supporting Information

A Graphs of the optimal objective value functions for Example 2 solved with the modified hybrid method

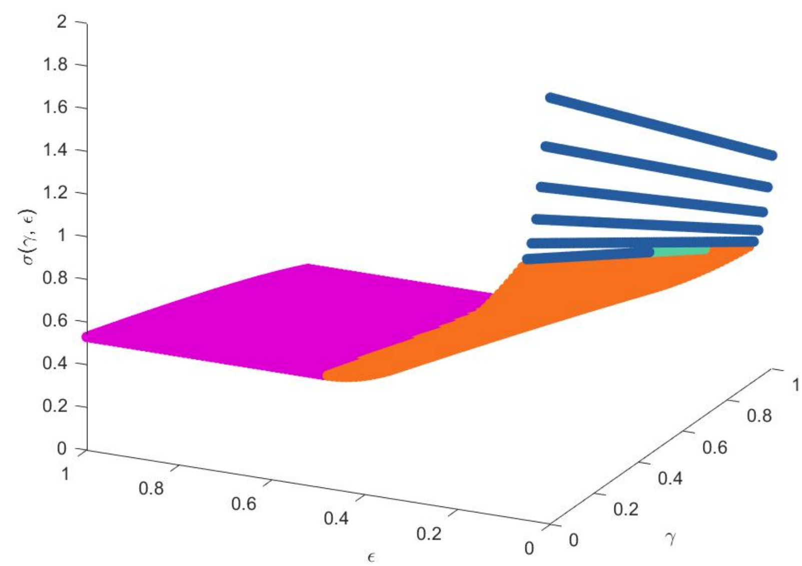


Figure 3.8: Optimal objective value function (minimum weighted risk) $\hat{\sigma}(\gamma, \epsilon^{nor})$ for problem (3.32)

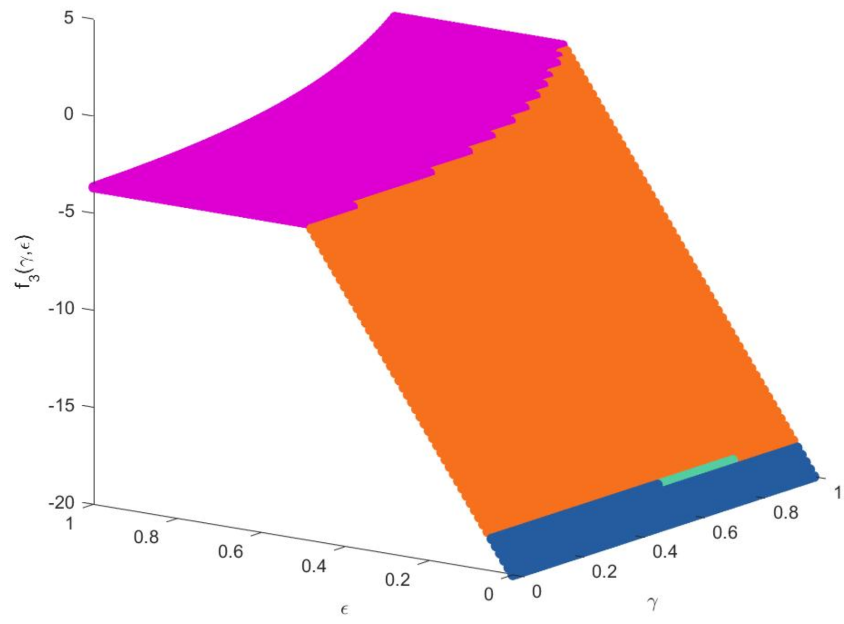


Figure 3.9: Pareto objective value function $-\mathbf{p}_3^T\hat{\mathbf{x}}(\gamma, \epsilon^{nor})$ for problem (4.34)

B Graphs of the efficient solution functions for Example 2 solved with the modified hybrid method

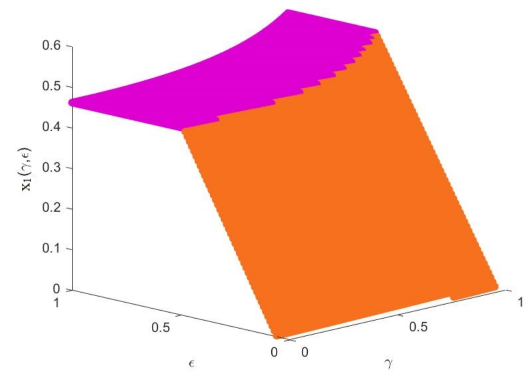


Figure 3.10: Efficient solution function $\hat{x}_1(\gamma, \epsilon^{nor})$ for problem (4.34)

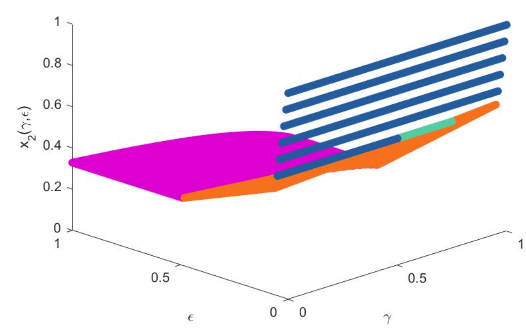


Figure 3.11: Efficient solution function $\hat{x}_2(\gamma, \epsilon^{nor})$ for problem (4.34)

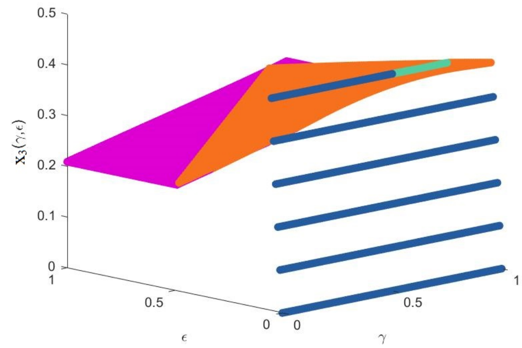


Figure 3.12: Efficient solution function $\hat{x}_3(\gamma, \epsilon^{nor})$ for problem (4.34)

C Parametric solutions for Example 1 and Example 2

Table 3.4: Invariancy regions and efficient solution functions for Example 1 scalarized with the weighted-sum method

$$I\mathcal{R}_1 = \left\{ \boldsymbol{\lambda} \in \Lambda : \begin{array}{l} -(\lambda_2 - 1)/(3\lambda_2 - 3\lambda_1 + 4) \geq 0 \\ (5\lambda_2 - 3\lambda_1 + 2)/(3\lambda_2 - 3\lambda_1 + 4) \geq 0 \\ -(\lambda_2 - 1)/(3\lambda_2 - 3\lambda_1 + 4) \geq 0 \\ (14\lambda_1 - 9\lambda_2 + 7\lambda_1\lambda_2 - 9\lambda_1^2 + 14\lambda_2^2 - 5)/(3\lambda_2 - 3\lambda_1 + 4) \geq 0 \\ (2\lambda_1 + 2\lambda_2 + 2\lambda_1\lambda_2 - 3\lambda_1^2 + 4\lambda_2^2 + 1)/(3\lambda_2 - 3\lambda_1 + 4) \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\boldsymbol{\lambda}) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = -(\lambda_2 - 1)/(3\lambda_2 - 3\lambda_1 + 4) \\ x_2 = 0 \\ x_3 = 0 \end{array} \right\} \text{ for } \boldsymbol{\lambda} \in I\mathcal{R}_1$$

$$IR_2 = \left\{ \boldsymbol{\lambda} \in \Lambda : \begin{array}{l} 3\lambda_1 - 5\lambda_2 - 2 \geq 0 \\ \lambda_2 - \lambda_1 + 1 \geq 0 \\ 3\lambda_1 - 3\lambda_2 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\boldsymbol{\lambda}) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 1/2 \\ x_2 = 0 \\ x_3 = 0 \end{array} \right\} \text{ for } \boldsymbol{\lambda} \in IR_2$$

$$IR_3 = \left\{ \boldsymbol{\lambda} \in \Lambda : \begin{array}{l} -(28\lambda_1\lambda_2 - 10\lambda_2 - 34\lambda_1 + 21\lambda_1^2 + 3\lambda_2^2 + 7)/(24\lambda_1 + 64\lambda_2 \\ \qquad -58\lambda_1\lambda_2 - 25\lambda_1^2 - 61\lambda_2^2 + 7) \geq 0 \\ -(44\lambda_1 - 44\lambda_2 + 2\lambda_1\lambda_2 - 17\lambda_1^2 + 55\lambda_2^2 - 21)/(24\lambda_1 + 64\lambda_2 \\ \qquad -58\lambda_1\lambda_2 - 25\lambda_1^2 - 61\lambda_2^2 + 7) \geq 0 \\ -(35\lambda_1\lambda_2 - 19\lambda_2 - 20\lambda_1 + 12\lambda_1^2 + 17\lambda_2^2 + 2)/(24\lambda_1 + 64\lambda_2 \\ \qquad -58\lambda_1\lambda_2 - 25\lambda_1^2 - 61\lambda_2^2 + 7) \geq 0 \\ -(14\lambda_1 - 9\lambda_2 + 7\lambda_1\lambda_2 - 9\lambda_1^2 + 14\lambda_2^2 - 5)(24\lambda_1 + 64\lambda_2 \\ \qquad -58\lambda_1\lambda_2 - 25\lambda_1^2 - 61\lambda_2^2 + 7) \geq 0 \\ -(28\lambda_1\lambda_2 - 33\lambda_2 - 26\lambda_1 + 43\lambda_1\lambda_2^2 - 15\lambda_1^2\lambda_2 + 14\lambda_1^2 + 10\lambda_1^3 - 37\lambda_2^2 \\ \quad +58\lambda_2^3 + 2)/(24\lambda_1 + 64\lambda_2 - 58\lambda_1\lambda_2 - 25\lambda_1^2 - 61\lambda_2^2 + 7) \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\boldsymbol{\lambda}) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = -(35\lambda_1\lambda_2 - 19\lambda_2 - 20\lambda_1 + 12\lambda_1^2 + 17\lambda_2^2 + 2)/(24\lambda_1 \\ \quad +64\lambda_2 - 58\lambda_1\lambda_2 - 25\lambda_1^2 - 61\lambda_2^2 + 7) \\ x_2 = -(14\lambda_1 - 9\lambda_2 + 7\lambda_1\lambda_2 - 9\lambda_1^2 + 14\lambda_2^2 - 5)/(24\lambda_1 \\ \quad +64\lambda_2 - 58\lambda_1 x - 25\lambda_1^2 - 61\lambda_2^2 + 7) \\ x_3 = 0 \end{array} \right\} \text{ for } \boldsymbol{\lambda} \in IR_3$$

$$IR_4 = \left\{ \boldsymbol{\lambda} \in \Lambda : \begin{array}{l} -(28\lambda_1\lambda_2 - 10\lambda_2 - 34\lambda_1 + 21\lambda_1^2 + 3\lambda_2^2 + 7)/(2(4\lambda_1 + 6\lambda_2 - 7)) \geq 0 \\ (3(\lambda_1 + \lambda_2 - 1))/(2(4\lambda_1 + 6\lambda_2 - 7)) \geq 0 \\ (3(\lambda_1 + \lambda_2 - 1))/(2(4\lambda_1 + 6\lambda_2 - 7)) \geq 0 \\ -(2\lambda_1 + 11\lambda_2 - 11\lambda_1\lambda_2 + \lambda_1^2 - 12\lambda_2^2 + 3)/(2(4\lambda_1 + 6\lambda_2 - 7)) \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\boldsymbol{\lambda}) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = (3(\lambda_1 + \lambda_2 - 1))/(2(4\lambda_1 + 6\lambda_2 - 7)) \\ x_2 = (3(\lambda_1 + \lambda_2 - 1))/(2(4\lambda_1 + 6\lambda_2 - 7)) \\ x_3 = 0 \end{array} \right\} \text{ for } \boldsymbol{\lambda} \in IR_4$$

Table 3.5: Invariancy regions and efficient solution functions in the parameter space for Example 2 scalarized with the modified hybrid method ($\epsilon = \epsilon^{nor}$.)

$$IR_1 = \left\{ \gamma, \epsilon \in [0,1] \times [0,1] : \begin{array}{l} (1139\gamma + 4489\epsilon - 2412\gamma\epsilon - 402\gamma^2\epsilon + 48\gamma^2 - 2189) \geq 0 \\ (15\gamma^2 - 86\gamma + 6\gamma^3 + 71) \geq 0 \\ -17\gamma + 31 \geq 0 \\ -19\gamma + 22 \geq 0 \\ -3\gamma^2 + 7 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\gamma, \epsilon) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = (17\gamma - 31)/(36\gamma + 6\gamma^2 - 67) \\ x_2 = (19\gamma - 22)/(36\gamma + 6\gamma^2 - 67) \\ x_3 = (2(3\gamma^2 - 7))/(36\gamma + 6\gamma^2 - 67) \end{array} \right\} \text{ for } (\gamma, \epsilon) \in IR_1$$

$$IR_2 = \left\{ \gamma, \epsilon \in [0,1] \times [0,1] : \begin{array}{l} -(1139\gamma + 4489\epsilon - 2412\gamma\epsilon - 402\gamma^2\epsilon + 48\gamma^2 - 2189) \geq 0 \\ -(23914\gamma - 32897\epsilon + 20167\gamma\epsilon + 12864\gamma^2\epsilon \\ \qquad\qquad -4850\gamma^2 - 28418) \geq 0 \\ (688\gamma + 45091\epsilon - 5762\gamma\epsilon - 2576) \geq 0 \\ (1760\gamma - 18827\epsilon - 14740\gamma\epsilon + 22957) \geq 0 \\ -(43\gamma + 196\epsilon - 153\gamma\epsilon - 161) \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\gamma, \epsilon) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = -(688\gamma + 45091\epsilon - 5762\gamma\epsilon \\ \qquad\qquad -2576)/(3314\gamma - 41955) \\ x_2 = -(1760\gamma - 18827\epsilon - 14740\gamma\epsilon \\ \qquad\qquad +22957)/(3314\gamma - 41955) \\ x_3 = (134(43\gamma + 196\epsilon - 153\gamma\epsilon \\ \qquad\qquad -161))/(3314\gamma - 41955) \end{array} \right\} \text{ for } (\gamma, \epsilon) \in IR_2$$

$$IR_3 = \left\{ \gamma, \epsilon \in [0,1] \times [0,1] : \begin{array}{l} (67\gamma\epsilon)/32 - (737\epsilon)/64 - \gamma/4 + 3/4 \geq 0 \\ 3\gamma + (3283\epsilon)/16 - (67\gamma\epsilon)/2 - 11 \geq 0 \\ (2881\gamma\epsilon)/64 - (45091\epsilon)/128 - (43\gamma)/8 + 161/8 \geq 0 \\ 1 - (67\epsilon)/8 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\gamma, \epsilon) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 0 \\ x_2 = 1 - (67\epsilon)/8 \\ x_3 = (67\epsilon)/8 \end{array} \right\} \text{ for } (\gamma, \epsilon) \in IR_3$$

$$
\mathit{IR}_4 = \left\{ \gamma, \epsilon \in [0,1] \times [0,1] : \begin{array}{l} (67\gamma\epsilon)/32 - (737\epsilon)/64 - \gamma/4 + 3/4 \geq 0 \\ (67\gamma\epsilon)/32 - (3283\epsilon)/64 - 3\gamma + 11 \geq 0 \\ (2881\gamma\epsilon)/64 - (45091\epsilon)/128 - (43\gamma)/8 + 161/8 \geq 0 \\ 1 - (67\epsilon)/8 \geq 0 \end{array} \right\}
$$

$$
\hat{\mathbf{x}}(\gamma, \epsilon) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 0 \\ x_2 = 1 - (67\epsilon)/8 \\ x_3 = (67\epsilon)/8 \end{array} \right\} \text{ for } (\gamma, \epsilon) \in \mathit{IR}_4
$$

Table 3.6: Invariancy regions and efficient solution functions for Example 2 scalarized with the weighted-sum method

$$I\mathcal{R}_1 = \left\{ \boldsymbol{\lambda} \in \Lambda : \begin{array}{l} 20 - 24\lambda_2 - 22\lambda_1 \geq 0 \\ 67/2 - (77\lambda_2)/2 - (71\lambda_1)/2 \geq 0 \\ 4 - 7\lambda_2 - 6\lambda_1 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\boldsymbol{\lambda}) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 0 \\ x_2 = 1 \\ x_3 = 0 \end{array} \right\} \text{ for } \boldsymbol{\lambda} \in I\mathcal{R}_1$$

$$I\mathcal{R}_2 = \left\{ \boldsymbol{\lambda} \in \Lambda : \begin{array}{l} -(197\lambda_1\lambda_2 - 98\lambda_2 - 82\lambda_1 + 87\lambda_1^2 + 111\lambda_2^2) \geq 0 \\ -(1350\lambda_1\lambda_2 - 673\lambda_2 - 587\lambda_1 + 615\lambda_1^2 + 735\lambda_2^2) \geq 0 \\ -(3\lambda_1 + 3\lambda_2 - 8) \geq 0 \\ 12\lambda_1 + 14\lambda_2 - 8 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\boldsymbol{\lambda}) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 0 \\ x_2 = -(3\lambda_1 + 3\lambda_2 - 8)/(9\lambda_1 + 11\lambda_2) \\ x_3 = (12\lambda_1 + 14\lambda_2 - 8)/(9\lambda_1 + 11\lambda_2) \end{array} \right\} \text{ for } \boldsymbol{\lambda} \in I\mathcal{R}_2$$

$$I\mathcal{R}_3 = \left\{ \lambda \in \Lambda : \begin{array}{l} (197\lambda_1\lambda_2 - 98\lambda_2 - 82\lambda_1 + 87\lambda_1^2 + 111\lambda_2^2) \geq 0 \\ -(1350\lambda_1\lambda_2 - 673\lambda_2 - 587\lambda_1 + 615\lambda_1^2 + 735\lambda_2^2) \geq 0 \\ -(3\lambda_1 + 3\lambda_2 - 8) \geq 0 \\ 6\lambda_1 + 7\lambda_2 - 4 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\boldsymbol{\lambda}) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 0 \\ x_2 = -(3\lambda_1 + 3\lambda_2 - 8)/(9\lambda_1 + 11\lambda_2) \\ x_3 = (12\lambda_1 + 14\lambda_2 - 8)/(9\lambda_1 + 11\lambda_2) \end{array} \right\} \text{ for } \boldsymbol{\lambda} \in I\mathcal{R}_3$$

$$
I\mathcal{R}_4 = \left\{ \lambda \in \Lambda : \begin{array}{l} 1426\lambda_1\lambda_2^2 - 681\lambda_1\lambda_2 + 791\lambda_1^2\lambda_2 + 2\lambda_1^2 + 10\lambda_1^3 - 491\lambda_2^2 + 633\lambda_2^3 \geq 0 \\ 1350\lambda_1\lambda_2 - 673\lambda_2 - 587\lambda_1 + 615\lambda_1^2 + 735\lambda_2^2 \geq 0 \\ -(732\lambda_1\lambda_2 - 281\lambda_2 - 501\lambda_1 + 495\lambda_1^2 + 237\lambda_2^2) \geq 0 \\ -(211\lambda_1\lambda_2 - 196\lambda_2 - 43\lambda_1 + 35\lambda_1^2 + 182\lambda_2^2) \geq 0 \end{array} \right\}
$$

$$
\hat{\mathbf{x}}(\boldsymbol{\lambda}) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = (1350\lambda_1\lambda_2 - 673\lambda_2 - 587\lambda_1 \\ \qquad\qquad +615\lambda_1^2 + 735\lambda_2^2)/(2(98\lambda_1\lambda_2 + 25\lambda_1^2 + 67\lambda_2^2)) \\ x_2 = -(732\lambda_1\lambda_2 - 281\lambda_2 - 501\lambda_1 \\ \qquad\qquad +495\lambda_1^2 + 237\lambda_2^2)/(2(98\lambda_1\lambda_2 + 25\lambda_1^2 + 67\lambda_2^2)) \\ x_3 = -(211\lambda_1\lambda_2 - 196\lambda_2 - 43\lambda_1 \\ \qquad\qquad +35\lambda_1^2 + 182\lambda_2^2)/(98\lambda_1\lambda_2 + 25\lambda_1^2 + 67\lambda_2^2) \end{array} \right\} \text{ for } \boldsymbol{\lambda} \in I\mathcal{R}_4
$$

Chapter 4

On Solving Parametric MOQPs with Parameters in General Locations

[The contents of this chapter include material from the paper entitled "On solving parametric multiobjective quadratic programs with parameters in general locations" which has been submitted to Annals of Operations Research, and the authors are Pubudu L.W. Jayasekara, Andrew Pangia and Margaret M. Wiecek.]

4.1 Introduction

Many real-life problems in engineering, business, and management are characterized by multiple conflicting criteria such as cost, performance, reliability, safety, productivity, affordability, and the field of multiobjective optimization provides models, theories and methods to address these types of applications [44]. In addition to conflict between objective functions, uncertainty (from unknown or imprecise data, inaccurate measurements, or inadequate models) is another important characteristic of many real-life decision problems. In the operations research literature, there are three classical paradigms to model uncertainty: probabilistic, possibilistic, and deterministic. The latter approach, using crisp sets to define domains within which uncertainties vary, has given foundation to robust optimization and parametric optimization.

In addition to constants and variables, a parametric optimization problem also contains parameters which model uncertainty because their values are neither known nor unknown, and not being solved for. Parameters fundamentally change the problem: the parametric single objective program (mpSOP) is solved to obtain a solution vector-valued function and the corresponding optimal value function, both of which map the parameters to the solution space. In contrast, the nonparametric SOP is solved to obtain a specific solution vector and corresponding optimal value. Similarly, the parametric multiobjective program (mpMOP) is solved to obtain a parametrized collection of efficient (Pareto) sets, as opposed to a specific efficient (Pareto) set. Parametric multiobjective optimization offers a bridge to robust multiobjective optimization, which uses various concepts to yield robust efficient solutions arguably preferred under the conditions of uncertainty. Since robust efficient solutions correspond to specific values of uncertainty, the robust approach is subsumed in the parametric approach and the latter emerges as a more universal methodology [127].

Studies on mpMOPs go back to 1979 when Naccache [90] examined the stability of solution sets due to perturbations in the feasible set. Since then, researchers have worked on parametrization of feasible sets, objective functions and domination structures, and related stability properties [13, 49, 67, 82, 94, 95, 110]. Parametric linear programs are analyzed in [16, 17], while the polyhedral structure of the efficient set for such problems is more recently examined in [51, 120]. Theoretical works have been accompanied by applied studies. Methods to compute a family of solution sets for unconstrained problems with a scalar parameter are developed in [36, 132, 75] and applied to mechatronic systems in which the parameter plays the role of time [133]. The theoretical framework for a solution method based on a technique which subdivides the solution and parameter spaces is proposed in [108]. In engineering design, genetic algorithms are tailored to the parametric case to allow for design exploration in the presence of exogenous factors [55, 62].

Independently of parametric multiobjective optimization, parametrization of the efficient set of MOPs can be conducted as a result of treating scalarized MOPs as mpSOPs [128]. This point of view is theoretically examined in [50, 58] and used for computational work in [63, 65, 68, 107, 117] to obtain different types of parametric descriptions of the efficient set for MOPs arising in portfolio optimization. In [92], such a description is obtained for multiobjective quadratic programs (MOQPs) whose objective functions are linearized.

While theoretical and computational studies on mpMOPs have been steadily progressing, the latter have been rather limited despite the fact that parametric optimization can provide a complete parametric description of the efficient set regardless of uncertainty in the model. Because mpMOPs can be solved before their efficient sets are actually needed, in time-sensitive situations, the only computations required are function evaluations at the specific parameter values stemming from the situation. This benefit, however, comes at the cost of the increased computational complexity which parametric optimization causes. This paper puts forward the premise that parametrization of the efficient set can naturally be combined with solving mpMOPs because the algorithms performing the former can also be used to achieve the latter. The purpose is to examine the state-of-the-art in algorithmic development for parametric multiobjective quadratic programs (mpMOQPs). Based on this premise, mpMOQPs are scalarized to be solved as parametric (single objective) quadratic programs by means of suitable algorithms designed for this class of problems. We develop a generalized weighted-sum scalarization method that subsumes several established scalarizations and leads to several related SOPs that can be matched with different solution algorithms. In a computational study, we compare the performance of three parametric optimization algorithms on mpQPs with linear and/or quadratic constraints that result from different scalarizations. The algorithms are also applied to mpMOQPs modeling decision-making problems in statistics and portfolio optimization. By means of these applications, the interplay between the scalarizations is disclosed and additional insight into the parametric efficient solution sets is obtained.

The paper is structured as follows. In Section 4.2, we formulate the mpMOQP and define solution concepts. The generalized weighted-sum method and related scalarizations for MOPs are developed in Section 4.3. In the subsequent two sections we present algorithms for solving different types of mpQPs that result from various scalarizations of mpMOQPs. In Section 4.4 we provide algorithms for mpQPs with linear constraints and results from computational tests performed on synthetic problems are included. Section 4.5 contains the applications and the paper is concluded in Section 4.6.

4.2 Problem Statement

We define the mpMOQP and the solution concepts used to solve this class of problems. We also introduce the assumptions that are needed by the solution algorithms we present in the subsequent sections.

Let $\kappa, n, r, \tilde{r} \in \mathbb{N}$, $\tilde{r} \leq r$, and $\mathbb{R}^\kappa, \mathbb{R}^n, \mathbb{R}^r$ be Euclidean spaces that are related to the parameter space, decision or solution space, and objective or outcome space, respectively. Let $\Theta \subseteq \mathbb{R}^\kappa$ be a parameter space, $\mathcal{X} : \mathbb{R}^\kappa \to \mathbb{R}^n$ be a point-to-set map such that $\mathcal{X}(\Theta) \subseteq \mathbb{R}^n$

and $\mathcal{X}(\boldsymbol{\theta}) \neq \emptyset$ for all $\boldsymbol{\theta} \in \Theta$. We investigate the following mpMOQP:

$$\min_{\mathbf{x}} \mathbf{f}(\mathbf{x};\boldsymbol{\theta}) = [f_1(\mathbf{x};\boldsymbol{\theta}) = \frac{1}{2}\mathbf{x}^T Q_1(\boldsymbol{\theta})\mathbf{x} + \mathbf{p}_1^T(\boldsymbol{\theta})\mathbf{x} + c_1(\boldsymbol{\theta}),\ \ldots,\ f_{\tilde{r}}(\mathbf{x};\boldsymbol{\theta}) = \frac{1}{2}\mathbf{x}^T Q_{\tilde{r}}(\boldsymbol{\theta})\mathbf{x} + \mathbf{p}_{\tilde{r}}^T(\boldsymbol{\theta})\mathbf{x} + c_{\tilde{r}}(\boldsymbol{\theta}),$$
$$f_{\tilde{r}+1}(\mathbf{x};\boldsymbol{\theta}) = \mathbf{p}_{\tilde{r}+1}^T(\boldsymbol{\theta})\mathbf{x} + c_{\tilde{r}+1}(\boldsymbol{\theta}),\ \ldots,\ f_r(\mathbf{x};\boldsymbol{\theta}) = \mathbf{p}_r^T(\boldsymbol{\theta})\mathbf{x} + c_r(\boldsymbol{\theta})] \tag{MOQP($\boldsymbol{\theta}$)}$$

$$\text{s.t. } \mathbf{x} \in \mathcal{X}(\boldsymbol{\theta}) = \{\mathbf{x} \in \mathbb{R}^n : A(\boldsymbol{\theta})\mathbf{x} \leqq \boldsymbol{b}(\boldsymbol{\theta}), \mathbf{x} \geqq \mathbf{0}\}$$
$$\boldsymbol{\theta} \in \Theta,$$

where $A : \Theta \to \mathbb{R}^{m\times n}$, $\boldsymbol{b} : \Theta \to \mathbb{R}^m$. The vector valued-objective $\mathbf{f} : \mathbb{R}^n \times \Theta \to \mathbb{R}^r$ is composed of functions $f_i : \mathbb{R}^n \times \Theta \to \mathbb{R}$ such that $Q_i : \Theta \to \mathbb{R}^{n\times n}$, $\mathbf{p}_i : \Theta \to \mathbb{R}^n$ and $c_i : \Theta \to \mathbb{R}$ for all $i = 1, \ldots, r$. The vector of parameters $\boldsymbol{\theta}$ models quantities that are unknown due to lack of knowledge at the time of the MOP construction. Examples include road capacity, interest rate, selling price, air humidity, material density or other application-specific values. Throughout this paper we make the following assumptions.

Assumption 4.2.1. *1. The parameter space $\Theta \subseteq \mathbb{R}^\kappa$ is a nonempty compact and polyhedral set.*

2. The feasible set $\mathcal{X}(\boldsymbol{\theta})$ for all $\boldsymbol{\theta} \in \Theta$ is nonempty compact and convex set.

To perform optimization, we need to be able to compare the outcomes of MOQP($\boldsymbol{\theta}$).

Definition 4.2.2. *1. Let $\boldsymbol{\theta} \in \Theta$ and $\boldsymbol{x}^1, \boldsymbol{x}^2 \in \mathcal{X}(\boldsymbol{\theta})$. Then $\boldsymbol{f}(\boldsymbol{x}^1;\boldsymbol{\theta})(<)(\leqq) \leq \boldsymbol{f}(\boldsymbol{x}^2;\boldsymbol{\theta})$ if and only if $f_i(\boldsymbol{x}^1;\boldsymbol{\theta})(<) \leq f_i(\boldsymbol{x}^2;\boldsymbol{\theta})$ for all $i = 1, \ldots, r$, where $\leq$ requires strict inequality for at least one index i, while $\leqq$ allows equality for all i.*

2. Let $\boldsymbol{x}^1, \boldsymbol{x}^2 \in \mathcal{X}(\Theta)$ and $i \in \{1, \ldots, r\}$. Then $f_i(\boldsymbol{x}^1;\boldsymbol{\theta}) \leq f_i(\boldsymbol{x}^2;\boldsymbol{\theta})$ if and only if $f_i(\boldsymbol{x}^1;\boldsymbol{\theta}) \leq f_i(\boldsymbol{x}^2;\boldsymbol{\theta})$ for all $\boldsymbol{\theta} \in \Theta$.

3. Let $\boldsymbol{x}^1, \boldsymbol{x}^2 \in \mathcal{X}(\Theta)$. Then $\boldsymbol{f}(\boldsymbol{x}^1;\boldsymbol{\theta})(<)(\leqq) \leq \boldsymbol{f}(\boldsymbol{x}^2;\boldsymbol{\theta})$ if and only if $\boldsymbol{f}(\boldsymbol{x}^1;\boldsymbol{\theta})(<)(\leqq) \leq \boldsymbol{f}(\boldsymbol{x}^2;\boldsymbol{\theta})$ for all $\boldsymbol{\theta} \in \Theta$.

Solving MOQP($\boldsymbol{\theta}$) for a fixed parameter $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}} \in \Theta$ is defined as finding the set of (weakly) efficient solutions.

Definition 4.2.3. *A feasible solution $\hat{\boldsymbol{x}} \in \mathcal{X}(\bar{\boldsymbol{\theta}})$ is called (weakly) efficient to MOQP($\boldsymbol{\theta}$) for $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}} \in \Theta$ if there exists no other solution $\boldsymbol{x} \in \mathcal{X}(\bar{\boldsymbol{\theta}})$ such that $\boldsymbol{f}(\boldsymbol{x};\bar{\boldsymbol{\theta}})(<) \leq \boldsymbol{f}(\hat{\boldsymbol{x}};\bar{\boldsymbol{\theta}})$. Let $\mathcal{X}_{(w)E}(\bar{\boldsymbol{\theta}})$ denote the set of (weakly) efficient solutions for $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}}$.*

We assume that $\mathcal{X}_E(\bar{\boldsymbol{\theta}}) \neq \emptyset$ for each $\bar{\boldsymbol{\theta}} \in \Theta$. Solving MOQP($\boldsymbol{\theta}$) for all $\boldsymbol{\theta} \in \Theta$ is defined as finding the (weakly) efficient set $\mathcal{X}_E(\boldsymbol{\theta}) \subseteq \mathcal{X}(\boldsymbol{\theta})$ for each $\boldsymbol{\theta} \in \Theta$.

Definition 4.2.4. *The set $\mathcal{X}_{(w)E} \subseteq \mathcal{X}(\Theta)$, defined as the collection of the (weakly) efficient sets $\mathcal{X}_{(w)E}(\boldsymbol{\theta})$, $\mathcal{X}_{(w)E} := \{X_{(w)E}(\boldsymbol{\theta})\}_{\boldsymbol{\theta}\in\Theta}$, is called the set of (weakly) efficient solutions to MOQP($\boldsymbol{\theta}$) for all $\boldsymbol{\theta} \in \Theta$.*

For each $\boldsymbol{\theta} \in \Theta$, we define the attainable set, $\mathcal{Y}(\boldsymbol{\theta})$, as the image of the feasible set $\mathcal{X}(\boldsymbol{\theta})$ under the vector-valued objective function mapping $\mathbf{f}$

$$\mathcal{Y}(\boldsymbol{\theta}) := \{\mathbf{y} \in \mathbb{R}^r : \mathbf{y} = \mathbf{f}(\mathbf{x},\boldsymbol{\theta}), \mathbf{x} \in \mathcal{X}(\boldsymbol{\theta})\}.$$

The image of a (weakly) efficient solution to MOQP($\boldsymbol{\theta}$) for $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}} \in \Theta$ is called a (weak) Pareto outcome. Let $\mathcal{Y}_{(w)P}(\bar{\boldsymbol{\theta}})$ denote the set of all (weak) Pareto outcomes for $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}} \in \Theta$.

Definition 4.2.5. *The set $\mathcal{Y}_{(w)P} \subseteq \mathbb{R}^r$, defined as the collection of the (weak) Pareto sets $\mathcal{Y}_{(w)P}(\boldsymbol{\theta})$, $\mathcal{Y}_{(w)P} := \{Y_{(w)P}(\boldsymbol{\theta})\}_{\boldsymbol{\theta}\in\Theta}$, is called the set of (weak) Pareto outcomes to MOQP($\boldsymbol{\theta}$) for all $\boldsymbol{\theta} \in \Theta$.*

Scalarization methods can reformulate MOQP($\boldsymbol{\theta}$) into mpSOPs using parameters that are specific to each method. In effect, the resulting mpSOPs have two types of parameters, those from the original model, $\boldsymbol{\theta}$, and the auxiliary parameters, $\boldsymbol{\epsilon}$ and $\boldsymbol{\lambda}$, needed for scalarization. We refer to the former as modeling parameters and to the latter as scalarization parameters. We further discuss the scalarization parameters in Section 4.3. The mpSOPs are solved with parametric optimization algorithms that partition the augmented parametric space into subsets called invariancy regions (critical regions or validity sets) and compute optimal solution functions defined on these regions. Under some conditions, the latter can provide the efficient solution functions making up the set $X_{(w)E}$ for the original MOQP($\boldsymbol{\theta}$).

Based on the state of the art in parametric optimization, we believe that the weighted-sum scalarization, epsilon-constraint scalarization, and their variations are the most useful for solving mpMOQPs. In the next section, therefore, we propose a generalized weighted sum scalarization for the nonparametric MOP and reduce it to a variety of SOPs whose optimal solutions are at least weakly efficient to the MOP. We then apply these scalarizations to mpMOQPs with suitable parametric optimization algorithms.

4.3 Generalized weighted sum scalarization

[This section is written jointly with Andrew Pangia and Margaret M. Wiecek.]

In this section, to keep the notation simple we depart from the parametric setting, and deal with a standard (nonparametric) MOP to prepare the ground for the algorithms we present in subsequent sections. We develop a weighted-sum scalarization encompassing several other scalarizing approaches. This scalarization employs several sets of parameters whose interplay allows for formulating variants of SOPs that will be useful for scalarizing mpMOQPs.

In this section only, let $g : \mathbb{R}^n \to \mathbb{R}^r$ denote the vector-valued objective function and let $\mathcal{X} \subseteq \mathbb{R}^n$ denote the feasible set. Then consider the MOP

$$\min_{\mathbf{x}\in\mathcal{X}} \ [g_1(\mathbf{x}),\ g_2(\mathbf{x}), \ldots, g_r(\mathbf{x})] \tag{MOP}$$

and define the following sets and parameters for the index set $\{1, \ldots, r\}$.

Definition 4.3.1. *Let $t \in \mathbb{N}$, $t \leq r$. Let the index set $\{1, \ldots, r\}$ be given. Define $t+1$ sets J, J_1, $\ldots$, J_t, where $J \subseteq \{1, \ldots, r\}$, $J_j \subseteq \{1, \ldots, r\}$ for $j = 1, \ldots, t$. Define the sets of parameters*

$$\Lambda := \left\{ \boldsymbol{\lambda} \in \mathbb{R}^{|J|} : \lambda_i \geq 0, i \in J, \sum_{i \in J} \lambda_i = 1 \right\}$$

and

$$\mathcal{M}^j := \left\{ \boldsymbol{\mu}^j \in \mathbb{R}^{|J_j|} : \mu_i^j \geq 0, i \in J_j, \sum_{i \in J_j} \mu_i^j = 1 \right\}$$

for all $j = 1, \ldots, t$. For convenience, also define

$$\boldsymbol{\mu} := \left[\boldsymbol{\mu}^1, \ldots, \boldsymbol{\mu}^t \right] \in \mathcal{M} := \mathcal{M}^1 \times \cdots \times \mathcal{M}^t.$$

Making use of Definition 4.3.1, consider another MOP in which every objective function is a weighted sum of the objective functions corresponding to each subset:

$$\min_{\mathbf{x} \in \mathcal{X}} \left[\sum_{i \in J} \lambda_i g_i(\mathbf{x}), \ \sum_{i \in J_1} \mu_i^1 g_i(\mathbf{x}), \ldots, \sum_{i \in J_t} \mu_i^t g_i(\mathbf{x}) \right] \tag{MOP'}$$

These two MOPs have the following relationship.

Proposition 4.3.2. *Let* (MOP) *be convex. Then $\hat{\boldsymbol{x}} \in \mathcal{X}$ is a weakly efficient solution to* (MOP) *if and only if $\hat{\boldsymbol{x}}$ is a weakly efficient solution to* (MOP') *for some $\boldsymbol{\lambda} \in \Lambda$ and $\boldsymbol{\mu} \in \mathcal{M}$.*

Proof. Let $\hat{\mathbf{x}} \in \mathcal{X}$ be a weakly efficient solution to (MOP). Then, by [56], $\hat{\mathbf{x}}$ is an optimal solution to the SOP

$$\min_{x \in \mathcal{X}} \sum_{i=1}^{r} \gamma_i g_i(\mathbf{x}) \tag{4.1}$$

for some $\boldsymbol{\gamma} \in \Gamma$, where $\Gamma := \{\boldsymbol{\gamma} \in \mathbb{R}^r : \gamma_i \geq 0, i = 1, \ldots, r, \sum_{i=1}^r \gamma_i = 1\}$. Consider now the following weighted sum problem which derives from (MOP'):

$$\min_{\mathbf{x}\in\mathcal{X}} \ \rho_0 \sum_{i\in J} \lambda_i g_i(\mathbf{x}) + \rho_1 \sum_{i\in J_1} \mu_i^1 g_i(\mathbf{x}) + \cdots + \rho_t \sum_{i\in J_t} \mu_i^t g_i(\mathbf{x}), \tag{4.2}$$

where

$$\boldsymbol{\rho} := [\rho_0,\ \rho_1,\ \ldots,\ \rho_t] \in \mathcal{P} := \left\{\boldsymbol{\rho} \in \mathbb{R}^{t+1} : \rho_i \geq 0, i = 1,\ \ldots,\ t+1, \sum_{i=0}^{t} \rho_i = 1\right\}. \tag{4.3}$$

If one can find $\boldsymbol{\rho}$, $\boldsymbol{\lambda}$, and $\boldsymbol{\mu}$ such that $\hat{\mathbf{x}}$ is an optimal solution to (4.2), then, by [56], $\hat{\mathbf{x}}$ will equivalently be weakly efficient to (MOP'). Similar to the proof of Theorem 4 in [68], take

$$\begin{aligned} &\rho_0\lambda_i = \gamma_i, i \in J \\ &\rho_j\mu_i^j = \gamma_i, i \in J_j, \quad j = 1, \ldots, t \end{aligned}$$

which ensures that $\boldsymbol{\rho} \in \mathcal{P}$ as defined in (4.3). Then, $\hat{\mathbf{x}}$ is an optimal solution to (4.2), and equivalently, it is a weakly efficient solution to (MOP'). □

Depending on the definitions of the sets J and J_j, $j = 1, \ldots, t$, different variants of (MOP') can be formulated and different SOPs obtained, which may be useful depending on the needs of the optimization solver being used or the context of the decision situation being modeled. Below six scalarizations are listed among which the first four are already established in the literature and whose optimal solutions are known to be (weakly) efficient to (MOP') and, by Proposition 4.3.2, are also weakly efficient to (MOP). Because scalarizations (4.8) and (4.9) are new, in the subsequent propositions their relationships with (MOP) are examined.

Corollary 4.3.2.1. *Let* (MOP) *and* (MOP') *be given, and the sets J and J_j, $j = 1, \ldots, t$ be defined as in Def. 4.3.1. Let $\mathcal{E}$ be a hypercube contained in a Euclidean space of the dimension as specified below.*

1. *Taking* $J = \{1, \ldots, r\}$ *converts* (MOP') *into the* ***weighted sum SOP*** *associated with* (MOP) *[56]:*

$$\begin{aligned} &\min_{\boldsymbol{x} \in \mathcal{X}} \sum_{i=1}^{r} \lambda_i g_i(\boldsymbol{x}) \\ &s.t.\ \boldsymbol{\lambda} \in \Lambda \subseteq \mathbb{R}^r. \end{aligned} \tag{4.4}$$

2. *Let* $J = \{i\}$, $J_j = \{j\}$ *for all* $j = 1, \ldots, r, j \neq i$. *Then the* ϵ*-constraint scalarization converts* (MOP') *into the* ***ϵ-constraint*** $SOP_i, i = 1, \ldots, r$, *associated with* (MOP) *[59]:*

$$\begin{aligned} &\min_{\boldsymbol{x} \in \mathcal{X}} g_i(\boldsymbol{x}) \\ &s.t.\ \ g_j(\boldsymbol{x}) \leq \epsilon_j,\ j = 1, \ldots, r, j \neq i \\ &\quad \boldsymbol{\epsilon} \in \mathcal{E} \subseteq \mathbb{R}^{r-1}. \end{aligned} \tag{4.5}$$

3. *Let* $J = \{1, \ldots, r\}$, $J_j = \{j\}$ *for all* $j = 1, \ldots, r$. *Then the* ϵ*-constraint scalarization converts* (MOP') *into the* ***hybrid SOP*** *associated with* (MOP) *[58]:*

$$\begin{aligned} &\min_{\boldsymbol{x} \in \mathcal{X}} \sum_{i=1}^{r} \lambda_i g_i(\boldsymbol{x}) \\ &s.t.\ \ g_j(\boldsymbol{x}) \leq \epsilon_j,\ j = 1, \ldots, r \\ &\quad \boldsymbol{\lambda} \in \Lambda \subseteq \mathbb{R}^r,\ \boldsymbol{\epsilon} \in \mathcal{E} \subseteq \mathbb{R}^r. \end{aligned} \tag{4.6}$$

4. *Let* $p \in \mathbb{N}$, $p < r$, $J \subset \{1, \ldots, r\}$, $|J| = p$, $J_j = \{j\}$ *for all* $j \notin J$. *Then the* ϵ*-constraint scalarization converts* (MOP') *into the* ***modified hybrid SOP*** *associated*

with (MOP) *[68]:*

$$\begin{aligned}
&\min_{\boldsymbol{x}\in X} \sum_{i\in J} \lambda_i g_i(\boldsymbol{x}) \\
&s.t.\;\; g_j(\boldsymbol{x}) \le \epsilon_j,\; j \notin J \\
&\qquad \boldsymbol{\lambda} \in \Lambda \subseteq \mathbb{R}^p,\; \boldsymbol{\epsilon} \in \mathcal{E} \subseteq \mathbb{R}^{r-p}.
\end{aligned} \tag{4.7}$$

5. *If* $0 \le t < r$*, and* J *and* J_j *are defined as in Def. 4.3.1, then the* ϵ*-constraint scalarization converts* (MOP') *into a variant of the hybrid SOP called the* ***weighted hybrid SOP****:*

$$\begin{aligned}
&\min_{\boldsymbol{x}\in X} \sum_{i\in J} \lambda_i g_i(\boldsymbol{x}) \\
&s.t.\;\; \sum_{i\in J_j} \mu_i^j g_i(\boldsymbol{x}) \le \epsilon_j,\; j = 1,\ldots,t \\
&\qquad \boldsymbol{\lambda} \in \Lambda \subseteq \mathbb{R}^{|J|},\; \boldsymbol{\mu} \in \mathcal{M},\; \boldsymbol{\epsilon} \in \mathcal{E} \subseteq \mathbb{R}^{t}.
\end{aligned} \tag{4.8}$$

6. *Let* $p \in \mathbb{N}$*,* $p < r$*,* $J \subseteq \{1,\ldots,r\}$*,* $|J| = p$*,* $J_j = \{j\}$ *for all* $j \notin J$*. Then the* ϵ*-constraint scalarization converts* (MOP') *into a variant of the* ϵ*-constraint* SOP_i *called the* ***reduced*** $\boldsymbol{\epsilon}$***-constraint****,* SOP_i*,* $i \notin J$*:*

$$\begin{aligned}
&\min_{\boldsymbol{x}\in X} g_i(\boldsymbol{x}) \\
&s.t.\;\; \sum_{j\in J} \lambda_j g_j(\boldsymbol{x}) \le \epsilon \\
&\qquad g_j(\boldsymbol{x}) \le \epsilon_j,\; j \notin J,\; j \ne i \\
&\qquad \boldsymbol{\lambda} \in \Lambda \subseteq \mathbb{R}^p,\; \boldsymbol{\epsilon} \in \mathcal{E} \subseteq \mathbb{R}^{r-p}.
\end{aligned} \tag{4.9}$$

The weighted hybrid SOP (4.8) allows for weighing all objective functions in the new objective and constraints.

Proposition 4.3.3. *Let $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}(\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\epsilon})$ be an optimal solution to the weighted hybrid SOP* (4.8) *for some $\boldsymbol{\lambda} \in \Lambda$, $\boldsymbol{\mu} \in \mathcal{M}$, and $\boldsymbol{\epsilon} \in \mathcal{E}$. Then $\hat{\boldsymbol{x}}$ is a weakly efficient solution to* (MOP).

Proof. Let $\hat{\mathbf{x}} = \hat{\mathbf{x}}(\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\epsilon})$ be an optimal solution to (4.8). Therefore, $\hat{\mathbf{x}}$ is feasible to (4.8), that is,

$$\textstyle\sum_{i \in J_j} \mu_i^j g_i(\mathbf{x}) \leq \epsilon_j, \; j = 1, \ldots, t. \tag{4.10}$$

Assume $\hat{\mathbf{x}} \notin \mathcal{X}_{wE}$. Then there exists a point $\bar{\mathbf{x}} \in \mathcal{X}$ such that

$$g_i(\bar{\mathbf{x}}) < g_i(\hat{\mathbf{x}}) \quad \text{for all } \; i = 1, \ldots, r. \tag{4.11}$$

Applying $\mu_i^j \geq 0$ not all 0, we have $\mu_i^j g_i(\bar{\mathbf{x}}) \leq \mu_i^j g_i(\hat{\mathbf{x}}), i \in J_j$ with at least one strict inequality, for $j = 1, \ldots, t$. Then

$$\sum_{i \in J_j} \mu_i^j g_i(\bar{\mathbf{x}}) < \sum_{i \in J_j} \mu_i^j g_i(\hat{\mathbf{x}})$$

for $j = 1, \ldots, t$, which makes $\bar{\mathbf{x}}$ feasible to (4.8) with $\epsilon_j = \sum_{i \in J_j} \mu_i^j g_i(\hat{\mathbf{x}}), \; j = 1, \ldots, t$. From (4.11), we obtain $\lambda_i g_i(\bar{\mathbf{x}}) \leq \lambda_i g_i(\hat{\mathbf{x}})$ for $i \in J$ with at least one strict inequality, and then

$$\sum_{i \in J} \lambda_i g_i(\bar{\mathbf{x}}) < \sum_{i \in J} \lambda_i g_i(\hat{\mathbf{x}}),$$

which contradicts the optimality of $\hat{\mathbf{x}}$. Therefore $\hat{\mathbf{x}}$ is weakly efficient to (MOP). □

The reduced ϵ-constraint scalarization (4.9) is motivated by the difficulty caused by the ϵ-constraint approach. When it is applied to (MOP), this method requires $r - 1$ right-hand-side (rhs) values for the ϵ-constraints so that the resulting SOP is feasible. In (4.9), some or all of the $r - 1$ ϵ-constraints are replaced with one constraint for which only one rhs value is needed, and therefore the resulting SOP is referred to as the reduced $\boldsymbol{\epsilon}$-constraint SOP_i, $i = 1, \ldots, r$.

Proposition 4.3.4. *If $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}(\boldsymbol{\lambda}, \boldsymbol{\epsilon})$ is an optimal solution to the reduced ϵ-constraint SOP_i (4.9) for some $\boldsymbol{\lambda} \in \Lambda, \boldsymbol{\epsilon} \in \mathcal{E}$ and some $i \notin J$, then $\hat{\boldsymbol{x}}$ is a weakly efficient solution to (MOP).*

Proof. Let $\hat{\mathbf{x}} = \hat{\mathbf{x}}(\boldsymbol{\lambda}, \boldsymbol{\epsilon})$ be an optimal solution to (4.9) for some $i \notin J$. Therefore, $\hat{\mathbf{x}}$ is feasible to (4.9), that is,

$$\textstyle\sum_{j \in J} \lambda_j g_j(\hat{\mathbf{x}}) \leq \epsilon \tag{4.12}$$

$$g_j(\hat{\mathbf{x}}) \leq \epsilon_j \quad j \notin J,\ j \neq i. \tag{4.13}$$

Assume $\hat{\mathbf{x}} \notin \mathcal{X}_{wE}$. Then there exists a point $\bar{\mathbf{x}} \in \mathcal{X}$ such that

$$g_j(\bar{\mathbf{x}}) < g_j(\hat{\mathbf{x}}) \quad \text{for all} \quad j = 1, \ldots, r. \tag{4.14}$$

Applying $\lambda_j \geq 0$ not all 0, we have $\lambda_j g_j(\bar{\mathbf{x}}) \leq \lambda_j g_j(\hat{\mathbf{x}}), \quad j \in J$ with at least one strict inequality; therefore $\sum_{j \in J} \lambda_j g_j(\bar{\mathbf{x}}) < \sum_{j \in J} \lambda_j g_j(\hat{\mathbf{x}})$. Using (4.12), we obtain

$$\sum_{j \in J} \lambda_j g_j(\bar{\mathbf{x}}) < \epsilon, \tag{4.15}$$

while from (4.13) and (4.14),

$$g_j(\bar{\mathbf{x}}) < \epsilon_j \quad j \notin J,\ j \neq i. \tag{4.16}$$

Since $\bar{\mathbf{x}} \in \mathcal{X}$, (4.15) and (4.16) make $\bar{\mathbf{x}}$ feasible to (4.9). From (4.14), $g_i(\bar{\mathbf{x}}) < g_i(\hat{\mathbf{x}})$, which contradicts the optimality of $\hat{\mathbf{x}}$. Therefore, $\hat{\mathbf{x}}$ is weakly efficient to (MOP). □

The final proposition in this section reveals that the modified hybrid and reduced ϵ-constraint SOPs jointly determine the efficiency of a feasible solution to a strictly convex (MOP).

Proposition 4.3.5. *Let (MOP) be strictly convex. A feasible solution $\hat{\boldsymbol{x}} \in \mathcal{X}$ is efficient to (MOP) if and only if $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}(\boldsymbol{\lambda}, \boldsymbol{\epsilon})$ is an optimal solution to the modified hybrid SOP (4.7) such that $g_j(\hat{\boldsymbol{x}}) = \epsilon_j, j \notin J$, and is also an optimal solution to the reduced ϵ-constraint SOP_i (4.9) such that $\sum_{j \in J} \lambda_j g_j(\hat{\boldsymbol{x}}) = \epsilon$ and $g_j(\hat{\boldsymbol{x}}) = \epsilon_j$, $j \notin J$, $j \neq i$, for some $\boldsymbol{\lambda} \in \Lambda \subseteq \mathbb{R}^p$ and $\boldsymbol{\epsilon} = (\epsilon, \epsilon_{j_1}, \ldots, \epsilon_{j_{r-p}}) \in \mathcal{E} \subseteq \mathbb{R}^{r-p+1}$, where $j_k \notin J$, $j \neq i$ for all $k = 1, \ldots, r-p$ for each $i \notin J$.*

Proof. From Corollary 7 in [68], a solution $\hat{\mathbf{x}} \in \mathcal{X}$ is efficient to (MOP) if and only if $\hat{\mathbf{x}} = \mathbf{x}(\boldsymbol{\lambda})$ is efficient to $\min_{\mathbf{x} \in \mathcal{X}} \left(\sum_{j \in J} \lambda_j g_j(\mathbf{x}), g_{j_1}(\mathbf{x}), \ldots, g_{j_{r-p}}(\mathbf{x}) \right)$ for some $\boldsymbol{\lambda} \in \Lambda \subseteq \mathbb{R}^p$. Theorem 4.1 in [27] then yields the desired result. $\square$

In the next two sections we present algorithms for solving MOQP($\boldsymbol{\theta}$) that is scalarized with the approaches presented in this section.

4.4 Parametric Quadratic Programs with Linear Constraints

In this section we present methods to solve a variation of mpMOQPs in which the quadratic objective functions do not carry parameters and are given as

$$\begin{aligned}
\min_{\mathbf{x}} \ \mathbf{f}(\mathbf{x}; \boldsymbol{\theta}) = [&f_1(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q_1 \mathbf{x} + \mathbf{p}_1^T \mathbf{x}, \ \ldots, \ f_{\tilde{r}}(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q_{\tilde{r}} \mathbf{x} + \mathbf{p}_{\tilde{r}}^T \mathbf{x}, \\
&f_{\tilde{r}+1}(\mathbf{x}; \boldsymbol{\theta}) = \mathbf{p}_{\tilde{r}+1}^T(\boldsymbol{\theta})\mathbf{x}, \ \ldots, \ f_r(\mathbf{x}; \boldsymbol{\theta}) = \mathbf{p}_r^T(\boldsymbol{\theta})\mathbf{x}] \qquad (\text{MOQP}_1(\boldsymbol{\theta})) \\
\text{s.t. } &\mathbf{x} \in \mathcal{X}(\boldsymbol{\theta}) = \{\mathbf{x} \in \mathbb{R}^n : A(\boldsymbol{\theta})\mathbf{x} \leqq \boldsymbol{b}(\boldsymbol{\theta}), \mathbf{x} \geqq \mathbf{0}\} \\
&\boldsymbol{\theta} \in \Theta,
\end{aligned}$$

where $Q_i, i = 1, \ldots, \tilde{r}$, are positive (semi-)definite $n \times n$ matrices, and elements in $\mathbf{p}_i(\boldsymbol{\theta}), i = \tilde{r}+1, \ldots, r$, $A(\boldsymbol{\theta})$, and $\boldsymbol{b}(\boldsymbol{\theta})$ are affine functions of $\boldsymbol{\theta}$. Under these assumptions and Assumption 4.2.1, $\text{MOQP}_1(\boldsymbol{\theta})$ is a convex problem for every $\boldsymbol{\theta} \in \Theta$. This class of mpMOQPs can be reformulated into mpQPs, that is, parametric programs with quadratic objective functions and linear constraints. Scalarizing $\text{MOQP}_1(\boldsymbol{\theta})$ with the weighted (4.4) or modified hybrid (4.7) approaches becomes relevant to this solution approach if the linear objectives are put into the ϵ-constraints that are included in the feasible set $\mathcal{X}(\boldsymbol{\theta})$ while all the quadratic objectives are combined in the weighted sum objective. In effect, we obtain an mpQP that is still challenging to solve due to the parameters present in the quadratic objective function and constraints.

In [68], four state-of-the-art methods to solve mpQPs are compared. Only one of these methods, the mpLCP method [3] based on the linear complementarity problem (LCP) reformulation of the original problem, can solve mpQPs with parameters in general locations. The study in [68] does not include a method that has been developed much earlier in [124] but has only few citations. It solves specially structured single-parametric QPs (spQPs) with a parameter in a general location and is therefore suitable to solve (non-parametric) biobjective quadratic programs (BOQPs) with the weighted-sum SOP. Because this method is also based on the LCP reformulation, it is referred to as the spLCP method. Since both the spLCP method and the mpLCP method have the same mathematical roots, but the efficiency of the former is unknown, it is of interest to compare their performance on solving spQPs that emerge from BOQPs.

Leading to the mpLCP method, below we review the LCP reformulation for $\text{MOQP}_1(\boldsymbol{\theta})$ assuming it has been scalarized with the modified hybrid SOP (4.7). We then focus on the spLCP method, giving its algorithms and presenting their application to a BOQP example. We compare the efficiency of both methods on a collection of BOQP instances.

4.4.1 The mpLCP method

Redefining the set Λ as $\Lambda' = \left\{\boldsymbol{\lambda} \in \mathbb{R}^{\tilde{r}} : \lambda_i \geq 0, i = 1, \ldots, \tilde{r}, \sum_{i=1}^{\tilde{r}-1} \lambda_i \leq 1, \lambda_{\tilde{r}} = 1 - \sum_{i=1}^{\tilde{r}-1} \lambda_i\right\}$ and applying (4.7) to MOQP$_1(\boldsymbol{\theta})$, the resulting mpQP can be written as:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}; \boldsymbol{\lambda}) = \frac{1}{2}\mathbf{x}^T Q(\boldsymbol{\lambda})\mathbf{x} + \mathbf{p}(\boldsymbol{\lambda})^T \mathbf{x} \\ \text{s.t.} \quad & \tilde{A}(\boldsymbol{\theta})\mathbf{x} \leqq \tilde{\boldsymbol{b}}(\boldsymbol{\theta}, \boldsymbol{\epsilon}) \\ & \mathbf{x} \geqq \mathbf{0} \\ & \boldsymbol{\theta} \in \Theta, \boldsymbol{\lambda} \in \Lambda', \boldsymbol{\epsilon} \in \mathcal{E}, \end{aligned} \tag{4.17}$$

where $Q(\boldsymbol{\lambda}) = \sum_{i=1}^{\tilde{r}} \lambda_i Q_i$ and $\mathbf{p}(\boldsymbol{\lambda}) = \sum_{i=1}^{\tilde{r}} \lambda_i \mathbf{p}_i$ for $\boldsymbol{\lambda} \in \Lambda'$, and $\tilde{A} : \Theta \to \in \mathbb{R}^{\tilde{m} \times n}, \tilde{\boldsymbol{b}} : \Theta \to \mathbb{R}^{\tilde{m}}, \tilde{m} = m + r - \tilde{r}$. The linear inequality constraints have been modified to include the $\boldsymbol{\epsilon}$-constraints of the form $\mathbf{p}_i^T(\boldsymbol{\theta})\mathbf{x} \leqq \epsilon_i$ for $i = \tilde{r}+1, \ldots, r$, where $\boldsymbol{\epsilon} \in \mathcal{E} \subseteq \mathbb{R}^{r-\tilde{r}}$ is a new parameter introduced into the model so that the problem remains feasible. When the KKT conditions for optimality are applied to (4.17), the multiparametric LCP (mpLCP) is constructed, which consists in finding a solution $(\mathbf{w}, \mathbf{z}) = (\mathbf{w}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}), \mathbf{z}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}))$ that satisfies the following parametric system:

$$\begin{gathered} \mathbf{w} - M(\boldsymbol{\theta}, \boldsymbol{\lambda})\mathbf{z} = \mathbf{q}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}) \\ \mathbf{w}^T \mathbf{z} = 0 \\ \mathbf{w}, \mathbf{z} \geqq \mathbf{0} \\ \boldsymbol{\theta} \in \Theta, \boldsymbol{\lambda} \in \Lambda', \boldsymbol{\epsilon} \in \mathcal{E} \end{gathered} \tag{4.18}$$

where $M(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \begin{bmatrix} Q(\boldsymbol{\lambda}) & \tilde{A}(\boldsymbol{\theta})^T \\ -\tilde{A}(\boldsymbol{\theta}) & 0 \end{bmatrix} \in \mathbb{R}^{h \times h}$, $\mathbf{q}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}) = \begin{bmatrix} \mathbf{p}(\boldsymbol{\lambda}) \\ \tilde{\mathbf{b}}(\boldsymbol{\theta}, \boldsymbol{\epsilon}) \end{bmatrix} \in \mathbb{R}^h, h = n + \tilde{m}$, and $\mathbf{w} = \begin{bmatrix} \mathbf{r} \\ \mathbf{s} \end{bmatrix}$, $\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix}$, where $\mathbf{s} \geqq \mathbf{0}$ is a slack variable associated with the linear inequality constraints and $\mathbf{u}$ and $\mathbf{r}$ are the dual variables associated with the linear and nonnegativity constraints respectively. Since Q_i, $i = 1, \dots, \tilde{r}$, are assumed to be positive (semi-)definite, so is $Q(\boldsymbol{\lambda})$ for every $\boldsymbol{\lambda} \in \Lambda'$. Consequently, matrix $M(\boldsymbol{\theta}, \boldsymbol{\lambda})$ is also positive (semi-)definite for every $\boldsymbol{\theta} \in \Theta$ and $\boldsymbol{\lambda} \in \Lambda'$, and therefore it is sufficient [32].

Definition 4.4.1. *Let $\boldsymbol{\theta} \in \Theta, \boldsymbol{\lambda} \in \Lambda', \boldsymbol{\epsilon} \in \mathcal{E}$ and B be a basis of the linear system in* (4.18)*.*

1. *The associated solution $(\boldsymbol{w}, \boldsymbol{z}) = (\boldsymbol{w}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}), \boldsymbol{z}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon})) \geqq \mathbf{0}$ to the linear system in* (4.18) *is a basic solution.*

2. *A basis B is complementary if $|\{i, i+h\} \cap B| = 1$ for $i = 1, \dots, h$.*

3. *A complementary basis B is feasible if the associated basic solution is feasible, i.e., $\boldsymbol{w}, \boldsymbol{z} \geqq \mathbf{0}$.*

4. *For a feasible complementary basis (FCB), the associated basic feasible solution $(\boldsymbol{w}, \boldsymbol{z})$ to* (4.18) *is called the basic feasible complementary solution (BFCS).*

The reader is referred to [3] for a state-of-the art study on mpLCPs or to [29, 30, 31, 32, 72] for theory and algorithms for LCPs. The solutions to mpLCP (4.18) are related to the efficient solution to $\text{MOQP}_1(\boldsymbol{\theta})$ in the following proposition.

Proposition 4.4.2. *Let $Q_i, i = 1, \dots, \tilde{r}$, be positive-definite in $MOQP_1(\boldsymbol{\theta})$.*

If $(\boldsymbol{w}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}), \boldsymbol{z}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon})) = \left(\begin{bmatrix} \boldsymbol{r}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}) \\ \boldsymbol{s}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}) \end{bmatrix}, \begin{bmatrix} \boldsymbol{x}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}) \\ \boldsymbol{u}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}) \end{bmatrix} \right)$ *is a BFCS to mpLCP (4.18) for some $\boldsymbol{\theta} \in \Theta, \boldsymbol{\lambda} \in \Lambda', \boldsymbol{\epsilon} \in \mathcal{E}$ then $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon})$ is an efficient solution to $MOQP_1(\boldsymbol{\theta})$.*

Proof. Since $\text{MOQP}_1(\boldsymbol{\theta})$ is a convex problem for every $\boldsymbol{\theta} \in \Theta$, so is mpQP (4.17) for every $\boldsymbol{\theta} \in \Theta, \boldsymbol{\lambda} \in \Lambda', \boldsymbol{\epsilon} \in \mathcal{E}$. Therefore, the KKT necessary optimality conditions for (4.17), which assume the form of mpLCP (4.18), are also sufficient. For some $\boldsymbol{\theta} \in \Theta, \boldsymbol{\lambda} \in \Lambda', \boldsymbol{\epsilon} \in \mathcal{E}$, vector $(\mathbf{w}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}), \mathbf{z}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon})) = \left(\begin{bmatrix} \mathbf{r}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}) \\ \mathbf{s}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}) \end{bmatrix}, \begin{bmatrix} \mathbf{x}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}) \\ \mathbf{u}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}) \end{bmatrix} \right)$ is a BFCS to (4.18) if and only if $\mathbf{x}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon})$ is an optimal solution to (4.17). Since $Q_i, i = 1, \ldots, \tilde{r}$, are positive-definite, the second order sufficient conditions for optimality also hold at $\mathbf{x}$. Then, by Cor. 8(1) in [68], $\mathbf{x} = \mathbf{x}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon})$ is an efficient solution to $\text{MOQP}_1(\boldsymbol{\theta})$. $\square$

The mpLCP method is the first and only method to solve mpLCPs (4.18) with multiple parameters in general locations, i.e., in matrix M or vector $\mathbf{q}$, under the assumption that M is a sufficient matrix for all values of the parameters [3]. The parameter space $\Theta \times \Lambda' \times \mathcal{E}$ is partitioned into possibly nonconvex invariancy regions over which the mpLCP solution $(\mathbf{w}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}), \mathbf{z}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}))$ is computed. The method consists of two phases. In Phase I, an initial invariancy region is found, while in Phase II, the invariancy regions making up the partition are identified. The method performs symbolic computation and provides a closed-form parametric description of the invariancy regions in the form of polynomial inequalities and the associated solutions in the form of rational functions. Using Proposition 4.4.2, these solutions provide the efficient solutions to $\text{MOQP}_1(\boldsymbol{\theta})$.

4.4.2 The spLCP method

The theory for the spLCP method was developed by Väliaho [124] but the proposed algorithm has not been implemented. It is the first ever method to solve spLCPs with a scalar parameter in the matrix M. The spLCP method solves the BOQP of the form

$$\begin{aligned} &\min_{\mathbf{x}} \; [\tfrac{1}{2}\mathbf{x}^T Q_1 \mathbf{x} + \mathbf{p}_1^T \mathbf{x}, \tfrac{1}{2}\mathbf{x}^T Q_2 \mathbf{x} + \mathbf{p}_2^T \mathbf{x}] \\ &\text{s.t. } \mathbf{x} \in \mathcal{X} := \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leqq \boldsymbol{b}, \mathbf{x} \geqq \mathbf{0}\}, \end{aligned} \tag{BOQP}$$

where $Q_i \in \mathbb{R}^{n \times n}, i = 1, 2$, is positive definite, $\mathbf{p}_i \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}$, and $\boldsymbol{b} \in \mathbb{R}^m$. Note that (BOQP) has to fulfill additional assumptions that are given in Assumption 4.4.3 below.

Redefining $\Lambda' = \{\lambda \in \mathbb{R} : \lambda \in [0, 1]\}$ and applying (4.4) to (BOQP), the resulting spQP is

$$\begin{aligned} \min_{\mathbf{x}} \; & f(\mathbf{x}; \lambda) = \tfrac{1}{2}\mathbf{x}^T Q(\lambda)\mathbf{x} + \mathbf{p}(\lambda)^T \mathbf{x} \\ \text{s.t. } & \mathbf{x} \in \mathcal{X}, \end{aligned} \tag{4.19}$$

where $Q(\lambda) = \lambda Q_1 + (1 - \lambda)Q_2$ and $\mathbf{p}(\lambda) = \lambda\mathbf{p}_1 + (1 - \lambda)\mathbf{p}_2$.

Problem (4.19) leads to an spLCP that this method solves. The parameter space Λ' is partitioned into subintervals, say $[\lambda', \lambda'']$, such that $(\mathbf{w}(\lambda), \mathbf{z}(\lambda))$ is a BFCS for the spLCP for $\lambda \in [\lambda', \lambda'']$. The method is based on a pivoting scheme and solving nonparametric LCPs obtained when λ is fixed at some values in the interval $[0, 1]$ starting with 0. Once a BFCS has been obtained for a specific value of λ, the invariancy interval for λ for which this solution remains feasible is found. The method consists of two phases. Phase I is designed to find an initial BFCS. If such a solution is available, Phase II is conducted. In this phase the parameter λ and the associated FCB get updated until $\lambda = 1$. The method outputs a list of λ values in the interval $[0, 1]$ and the associated BFCS. In the subsequent sections we present the spLCP method as a collection of algorithms to complement the theoretical exposition in [124].

4.4.2.1 Reformulation of spLCP

Given (4.19) and the resulting spLCP with $\Lambda' = [0, 1]$, rewrite matrix $Q(\lambda)$ and vector $\mathbf{p}(\lambda)$ as $Q(\lambda) = Q_2 + \lambda(Q_1 - Q_2)$, and $\mathbf{p}(\lambda) = \mathbf{p}_2 + \lambda(\mathbf{p}_1 - \mathbf{p}_2)$. Then matrix $M(\lambda)$ and vector $\mathbf{q}(\lambda)$ can be decomposed as:

$$M(\lambda) = M + \lambda\Delta M, \qquad \mathbf{q}(\lambda) = \mathbf{q} + \lambda\Delta\,\mathbf{q},$$

where $M = \begin{bmatrix} Q_2 & A^T \\ -A & 0 \end{bmatrix} \in \mathbb{R}^{h\times h}, \Delta M = \begin{bmatrix} Q_1 - Q_2 & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{h\times h},\ \mathbf{q} = \begin{bmatrix} \mathbf{p}_2 \\ \mathbf{b} \end{bmatrix} \in \mathbb{R}^h,$ and $\Delta\,\mathbf{q} = \begin{bmatrix} \mathbf{p}_1 - \mathbf{p}_2 \\ \mathbf{0} \end{bmatrix} \in \mathbb{R}^h.$ The spLCP can be written as:

$$\begin{aligned} \mathbf{w} - \big(M + \lambda \Delta M\big)\mathbf{z} &= \mathbf{q} + \lambda\Delta\,\mathbf{q} \\ \mathbf{w}^T\,\mathbf{z} &= 0 \\ \mathbf{w}, \mathbf{z} &\geqq \mathbf{0}. \end{aligned} \tag{4.20}$$

and is referred to as the spLCP$(M(\lambda), \mathbf{q}(\lambda))$ or simply the spLCP if the context is known.

The method is developed under the following assumptions that determine the class of spQPs (4.19) (and also BOQPs) that can be solved:

Assumption 4.4.3. *1. $Q_1 - Q_2$ is positive definite.*

2. $\boldsymbol{p}_1 - \boldsymbol{p}_2$ is in the column space of $Q_1 - Q_2$.

3. $Q_2 - \delta(Q_1 - Q_2)$ is positive semidefinite for some $\delta > 0$.

Based on Assumption 4.4.3(1), rank$(\Delta M) = n$. This assumption also assures that the Cholesky decomposition can be applied to matrix ΔM, which produces auxiliary matrices used in the spLCP method.

4.4.2.2 Full Rank Factorization of Matrix ΔM

The method is equipped with matrix factorization needed to modify spLCP (4.20) into an augmented spLCP. Consider a full rank matrix factorization of matrix ΔM

$$\Delta M = \Psi\Psi^T,$$

where $\Psi \in \mathbb{R}^{h \times n}$ is a lower triangular matrix with positive diagonal elements. Given $\Delta M = \begin{bmatrix} Q_1 - Q_2 & 0 \\ 0 & 0 \end{bmatrix}$, matrix Ψ is composed of two matrices such that $\Psi = \begin{bmatrix} \Psi_1 \\ 0 \end{bmatrix}$, where $\Psi_1 \in \mathbb{R}^{n \times n}$ is a lower triangular matrix and 0 is a zero matrix of size $m \times n$. Then

$$\begin{bmatrix} Q_1 - Q_2 & 0 \\ 0 & 0 \end{bmatrix} = \Delta M = \Psi \Psi^T = \begin{bmatrix} \Psi_1 \\ 0 \end{bmatrix} \begin{bmatrix} \Psi_1^T & 0^T \end{bmatrix} = \begin{bmatrix} \Psi_1 \Psi_1^T & 0 \\ 0 & 0 \end{bmatrix}.$$

Note that only the matrix $Q_1 - Q_2$ in the block $(1,1)$ of ΔM is decomposed, i.e., $Q_1 - Q_2 = \Psi_1 \Psi_1^T$, and the Cholesky decomposition is used to obtain Ψ_1.

In the spLCP method, the parameter λ is updated in every iteration. In this update, further discussed in Section 4.4.2.6, a vector $\boldsymbol{\kappa} \in R^n$, which is computed below, is used. Let

$$\Delta \mathbf{q} \quad = \Psi \boldsymbol{\kappa}, \tag{4.21}$$

or equivalently

$$\begin{bmatrix} \mathbf{p}_1 - \mathbf{p}_2 \\ \mathbf{0} \end{bmatrix} \quad = \begin{bmatrix} \Psi_1 \\ 0 \end{bmatrix} \boldsymbol{\kappa},$$

which is simplified to

$$\mathbf{p}_1 - \mathbf{p}_2 \quad = \Psi_1 \boldsymbol{\kappa}. \tag{4.22}$$

System (4.22) is solved for $\boldsymbol{\kappa}$ using the forward substitution method. Having computed Ψ and $\boldsymbol{\kappa}$, an augmented spLCP is constructed.

4.4.2.3 Initialization and Overview

Let $\lambda \in [0,1]$ be fixed in spLCP (4.20). Given Ψ and $\boldsymbol{\kappa}$ from the factorization, and letting $\mathbf{v} \in \mathbb{R}^h$ and $\mathbf{v}_{\boldsymbol{\kappa}} \in \mathbb{R}^n$, the augmented spLCP is formulated:

$$\begin{aligned}\left(\begin{bmatrix}\mathscr{I}_1 & 0\\ 0 & \mathscr{I}_2\end{bmatrix} - \begin{bmatrix}M(\lambda) & -\Psi\\ \Psi^T & 0\end{bmatrix}\right)\begin{bmatrix}\mathbf{v}_B\\ \mathbf{v}_{\boldsymbol{\kappa}}\end{bmatrix} &= \begin{bmatrix}\mathbf{q}(\lambda)\\ \boldsymbol{\kappa}\end{bmatrix}\\ \mathbf{v}_B &\geqq \mathbf{0}\\ \mathbf{v}_{\boldsymbol{\kappa}} &\quad \text{free,}\end{aligned} \tag{4.23}$$

where $\mathscr{I}_1$ is an $h \times h$ identity matrix and $\mathscr{I}_2$ is an $n \times n$ identity matrix. For the linear system in (4.23), let $\mathbb{E} = \{1, \ldots, h, h+n+1, \ldots, 2h+n\}$, a set $B \subset \mathbb{E}$ be a complementary basis for the linear system in (4.20), i.e., if $|B| = h$, then the set $N = \mathbb{E} \setminus B$ is such that $|N| = h$. Then $\mathbf{v}_B = \mathbf{v}_B(\lambda) = \{v_i(\lambda) : i \in B\}$ and $\mathbf{v}_N = \mathbf{v}_N(\lambda) = \{v_i(\lambda) : i \in N\}$ are the vectors of basic and nonbasic variables respectively in (4.20) and (4.23). Both these vectors consist of the variables in $\mathbf{w} = \mathbf{w}(\lambda)$ and $\mathbf{z} = \mathbf{z}(\lambda)$. The vector $\mathbf{v}_{\boldsymbol{\kappa}}$ consists of the variables that are always basic for (4.23) but are considered dummy with no meaning in the solution to (4.20).

Since the goal is to solve (4.23) for $\mathbf{v}_B(\lambda)$ for a fixed $\lambda \in [0,1]$, for the initial basis $B_0 = \{1, \ldots, h\}$ in (4.20), the entire problem (4.23) is saved in an initial Simplex-type Tableau $T_{B_0}(\lambda)$ as given in Table 4.1. Tableau $T_{B_0}(\lambda)$ has $h+n$ rows and $2(h+n)+1$ columns. Let $\mathbb{E}' = \{1, 2, \ldots, 2h+n\}$ be the set of indices of $2h+n$ columns starting from the second column and ending at the $(2h+n)^{\text{th}}$ column, and let $K = \{2h+n+1, \ldots, 2h+2n\}$ be the set of indices of the last n columns of Tableau 4.1. Note that neither $\mathbb{E}'$ nor K contain the first column, $\mathfrak{g}_{.0}$. At the beginning of Phase I, the right hand side of the linear system in (4.23), $\mathbf{q}(\lambda)$, is stored in the first h rows of column $\mathfrak{g}_{.0}$. The last n rows of $\mathfrak{g}_{.0}$ contain the vector $\boldsymbol{\kappa}$ obtained by solving (4.21). At the end of both phases, the solution vector $\mathbf{v}_B(\lambda)$ is retrieved from the first h rows of column $\mathfrak{g}_{.0}$.

Table 4.1: Initial Tableau $T_{B_0}(\lambda)$

	$\mathfrak{g}_{.0}$	$\mathfrak{g}_{.1},\cdots,\mathfrak{g}_{.h}$	$\mathfrak{g}_{.(h+1)},\cdots,\mathfrak{g}_{.(h+n)}$	$\mathfrak{g}_{.((h+n)+1)},\cdots,\mathfrak{g}_{.(2h+n)}$	$\mathfrak{g}_{.((2h+n)+1)},\cdots,\mathfrak{g}_{.(2h+2n)}$
v_1 $\vdots$ v_h	$\mathbf{q}(\lambda)$	$\mathscr{I}_1$	0	$M(\lambda)$	$-\Psi$
v_{κ_1} $\vdots$ v_{κ_n}	$\boldsymbol{\kappa}$	0	$\mathscr{I}_2$	Ψ^T	0

The pivoting operations in the tableau convert a complementary basis into another complementary basis. Two types of pivotal operations are defined but neither of them guarantees a new FCB.

Definition 4.4.4. *For index $i \in \mathbb{E}'$, the complementary index of i is defined as*

$$\bar{i} = \begin{cases} i+n+h & if \;\; i \leq h \\ i-n-h & otherwise \end{cases}$$

Definition 4.4.5. *1. Replacing a single element of a basis with its complement is called a diagonal pivot.*

2. Replacing two elements of a basis with their complements is called an exchange pivot.

Phase I is executed to find an initial BFCS for spLCP (4.20) for the case when $\lambda = 0$ and $\mathbf{q} \ngeq \mathbf{0}$. When $\lambda = 0$ and $\mathbf{q} \geq \mathbf{0}$, the method starts with Phase II in which a nonparametric LCP is solved for a specific value of the parameter λ that is subsequently updated. The pseudocode for solving spLCP (4.20) is given in Algorithm 2. In the initial Tableau $T_{B_0}(\lambda)$, $\lambda = 0$.

Let $T_{B_0}(\lambda)_{i,0}$ be the element in the i^{th} row and the first column of Tableau $T_{B_0}(\lambda)$. If $T_{B_0}(\lambda)_{i,0} < 0$ for at least one $i \in \{1,\ldots,h\}$, Phase I is executed before Phase II. However, if all those elements are nonnegative, the algorithm begins directly with Phase II.

Algorithm 2 spLCP Method

INPUT: Initial Tableau $T_{B_0}(\lambda)$
if $T_{B_0}(\lambda)_{i,0} < 0$ for at least one $i \in \{1, \ldots, h\}$ **then**
    Goto Phase I (apply Algorithm 3)
Goto Phase II (Algorithm 4)
else
    Goto Phase II (Algorithm 4)
end if
OUTPUT: Solution $(\mathbf{z}(\lambda), \mathbf{w}(\lambda))$ for each λ.

4.4.2.4 Phase I

In Phase I, the Criss-Cross Method (given in Algorithm 3) is used to find an initial BFCS for spLCP (4.20) for the case when $\lambda = 0$ and $q \ngeq \mathbf{0}$. The Criss-Cross Method solves the nonparametric LCP obtained from the spLCP with a fixed λ [38]. Let $\lambda = \bar{\lambda}$ be fixed, B_0 be an initial complementary basis, $\mathbf{v}_{B_0}(\bar{\lambda}) = \mathbf{w}(\bar{\lambda})$ be a vector of basic variables, and $\mathbf{v}_N(\bar{\lambda}) = \mathbf{z}(\bar{\lambda})$ be a vector of nonbasic variables. If $\mathbf{q}(\bar{\lambda}) \geq \mathbf{0}$, i.e., if $T_{B_0}(\bar{\lambda})_{i,0} \geq 0$ for all $i = 1, \ldots, h$, then the algorithm stops immediately and $(\mathbf{w}(\bar{\lambda}) = \mathbf{q}(\bar{\lambda}), \mathbf{z}(\bar{\lambda}) = \mathbf{0})$ is the initial BFCS for spLCP (4.20). Otherwise $\mathbf{q}(\bar{\lambda}) \ngeq \mathbf{0}$ and pivoting is used to obtain a complementary basis B' which is adjacent to the current basis B by replacing at most two elements in B. If the pivot element is negative, the diagonal pivot is executed, and if the pivot element is zero, the exchange pivot is executed. Let $T_B(\bar{\lambda})_{r,\bar{r}}$ be the element in Tableau $T_B(\bar{\lambda})$ associated with index $r \in B$ and its complementary index $\bar{r} \in N$. The element $T_B(\bar{\lambda})_{r,\bar{r}}$ becomes the pivot element if the index $r \in B$ is chosen as the smallest index with a negative entry in the first column in the Tableau $T_B(\bar{\lambda})$, and $\bar{r} \in N$. If there is no such r, i.e., $T_B(\bar{\lambda})_{i,0} \geq 0$ for all $i = 1, \ldots, h$, the algorithm stops; thus, B is a FCB and a BFCS for spLCP (4.20) has been found.

Algorithm 3 Criss-Cross Method

INPUT: $T_{B_0}(\bar{\lambda}), B_0$
if $T_{B_0}(\bar{\lambda})_{i,0} \geq 0$ for all $i = 1, \ldots, h$ **then**
    STOP, a BFCS has been found.
else
    Set $B = B_0$.
    while $T_B(\bar{\lambda})_{i,0} \leq 0$ for at least one $i \in \{1, \ldots, h\}$ **do**
        Let $r = \min\{i \in B : T_B(\bar{\lambda})_{i,0} < 0\}$

        if $T_B(\bar{\lambda})_{r,\bar{r}} < 0$ **then**
            make diagonal pivot, new basis $B' = B \setminus \{r\} \cup \{\bar{r}\}$
        else if $T_B(\bar{\lambda})_{r,\bar{r}} = 0$ **then**
            Let $k = \min\{j : T_B(\bar{\lambda})_{r,j} < 0 \text{ or } T_B(\bar{\lambda})_{j,r} > 0\}$
            if $T_B(\bar{\lambda})_{r,k} \times T_B(\bar{\lambda})_{k,r} < 0$ **then**
                make exchange pivot, new basis $B' = B \setminus \{r, k\} \cup \{\bar{r}, \bar{k}\}$
            else
                STOP. LCP is infeasible
            end if
        else
            STOP. LCP is infeasible
        end if
    end while Set $B = B'$ and $T_B(\bar{\lambda}) = T_{B'}(\bar{\lambda})$
end if
OUTPUT: BFCS to spLCP (4.20) with $\lambda = \bar{\lambda}$

4.4.2.5 Phase II

In Phase II, the parameter λ and basis B are consecutively updated. Let B be a complementary basis in (4.20). For a given λ, let $T_B(\lambda)$ denote the tableau with $h+n+1$ columns consisting of the first (0^{th}) column, h columns associated with the current nonbasic variables, and the last n columns of the tableau in Table 4.1.

For each $i = 1, \dots, h$ and K, define the following auxiliary vector

$$\begin{aligned}\mathcal{B}_{i0} = \Big(& T_B(\lambda)_{i,0}, (-1)^1 T_B(\lambda)_{i,K} T_B(\lambda)_{n,0}, (-1)^2 T_B(\lambda)_{i,K} T_B(\lambda)_{n,K} T_B(\lambda)_{n,0}, \\ & \dots, (-1)^n T_B(\lambda)_{i,K} T_B(\lambda)_{n,K}^{n-1} T_B(\lambda)_{K,0} \Big),\end{aligned} \tag{4.24}$$

where the terms in (4.24) denote the following elements in tableau $T_B(\lambda)$.

- $T_B(\lambda)_{i,0}$ - the element in the i^{th} row and the first (0^{th}) column,
- $T_B(\lambda)_{i,K}$ - the vector of elements in the i^{th} row and all columns associated with the index set K,
- $T_B(\lambda)_{n,0}$ - the vector of elements in the last n rows and the first (0^{th}) column,
- $T_B(\lambda)_{n,K}$ - the matrix of elements in the last n rows and all columns associated with the index set K.

To continue we need the following definition.

Definition 4.4.6. *Let $\boldsymbol{v} \in \mathbb{R}^h$. We say that*

1. *$\boldsymbol{v}$ is lexicographically positive (negative), $\boldsymbol{v} \succ \boldsymbol{0}$ ($\boldsymbol{v} \prec \boldsymbol{0}$), if its first nonzero component is positive (negative).*

2. *$\boldsymbol{v}$ is lexicographically nonnegative (nonpositive), $\boldsymbol{v} \succeq \boldsymbol{0}$ ($\boldsymbol{v} \preceq \boldsymbol{0}$), if $\boldsymbol{v} = \boldsymbol{0}$ or $\boldsymbol{v} \succ \boldsymbol{0}$ ($\boldsymbol{v} = \boldsymbol{0}$ or $\boldsymbol{v} \prec \boldsymbol{0}$).*

Phase II is presented in Algorithm 4 that runs until a BFCS is obtained for $\lambda = 1$. If a vector $\mathcal{B}_{i0} \succ \mathbf{0}$ for some $i = 1, \ldots, h$, the parameter λ is updated using Algorithm 5. Otherwise, the current complementary basis is updated using Algorithm 7.

Algorithm 4 Phase II

INPUT: Current Tableau $T_B(\lambda)$ s.t. $T_B(\lambda)_{i0} \geq \mathbf{0}$ for $i = 1, \ldots, h, B$
while $\lambda \leq 1$ **do**
    if $\mathcal{B}_{i0} \succeq \mathbf{0}$ for all $i = 1, \ldots, h$ **then**
    else
        Update λ (Algorithm 5)
    end if
end while
OUTPUT: Solution $(\mathbf{z}(\lambda), \mathbf{w}(\lambda))$ to spLCP (4.18)

4.4.2.6 Updating Parameter λ

Parameter λ is updated in Algorithm 5. Two types of polynomial equations, $P_{i1}(\tau) = 0$, for $i = 1, \ldots, h$, and $P_2(\tau) = 0$, are solved to obtain an increment τ of the parameter λ, which is at some current value between 0 and 1. Let α_i denote the smallest positive root of the polynomial equation $P_{i1}(\tau) = 0$ for each $i = 1, \ldots, h$. Then the smallest α_i value is recorded as ρ_1. Similarly, the smallest positive root of $P_2(\tau) = 0$ is recorded as ρ_2. If the roots of the polynomial equations are not positive, we set $\rho_1 = \infty$ or $\rho_2 = \infty$ accordingly. Then the smallest value among $1 - \lambda, \rho_1$ and ρ_2 is recorded and denoted as ρ, i.e., $\rho = \min\{1 - \lambda, \rho_1, \rho_2\}$. Based on the value of ρ, one of three strategies is applied to update λ. If $\rho = 1 - \lambda$, the current tableau is feasible for the invariancy interval $[\lambda, 1]$. If the condition $\mathcal{B}_{i0} \succeq \mathbf{0}$ holds for $T_{B_0}(1)$, the algorithm stops and spLCP (4.20) has been solved for $\lambda \in [0, 1]$. Otherwise, the basis has to be updated. If $\rho = \rho_1$, the current tableau is feasible for the invariancy interval $[\lambda, \lambda + \rho_1]$ and the initial tableau is updated using Algorithm 6. If $\rho = \rho_2$, set $\lambda = \lambda + \rho_2$ in the initial Tableau $T_{B_0}(\lambda)$ and the condition $\mathcal{B}_{i0} \succeq \mathbf{0}$ is subsequently checked.

Algorithm 5 Updating λ

INPUT: Current Tableau $T_B(\lambda)$ such that $\mathcal{B}_{i0} \succeq \mathbf{0}$ for all $i = 1, \ldots, h$
for $i = 1, \ldots, h$ **do**
    solve

$$P_{i1}(\tau) = \big(det\big(T_B(\lambda)_{n,K} + \tau^{-1}\mathscr{I}\big)\big)\,\big(T_B(\lambda)_{i,0} - T_B(\lambda)_{i,K}(T_B(\lambda)_{n,K} + \tau^{-1}\mathscr{I})T_B(\lambda)_{n,0}\big) = 0$$

    Let $\alpha_i = \begin{cases} \min\{\tau | P_{i1}(\tau) = 0\} & if\ \ \tau > 0 \\ \infty & o.w. \end{cases}$
    set $\rho_1 = \min\{\alpha_1, \ldots, \alpha_h\}$
end for
Solve $P_2(\tau) = det\big(T_B(\lambda)_{n,K} + \tau^{-1}\mathscr{I}\big) = 0$
set $\rho_2 = \begin{cases} \min\{\tau | P_2(\tau) = 0\} & if\ \ \tau > 0 \\ \infty & o.w. \end{cases}$
Set $\rho = \min\{1 - \lambda, \rho_1, \rho_2\}$
if $\rho = 1 - \lambda$ **then**
    STOP; the Tableau $T_B(\lambda)$ is feasible for $[\lambda, 1]$. Obtain $T_{B_0}(1)$. Go to Phase II (Algorithm 4)
else if $\rho = \rho_1$ **then**
    the tableau $T_B(\lambda)$ is feasible for $[\lambda, \lambda + \rho_1]$. Set $\lambda = \lambda + \rho_1$

    if there exists some $i \in \{1, \ldots, h\}$ for which $T_{B_0}(\lambda)_{i,0} \not\succeq \mathbf{0}$ **then**
        Update the initial Tableau $T_{B_0}(\lambda)$ using Algorithm 6. Go to Phase II (Algorithm 4)
    else
        Go to Phase II (Algorithm 4)
    end if
else
    Set $\lambda = \lambda + \rho_2$ and update the Tableau $T_{B_0}(\lambda)$. Go to Phase II (Algorithm 4)
end if
OUTPUT: new λ

4.4.2.7 Updating the Initial Tableau when $\rho = \rho_1$

Algorithm 6 is designed to provide a new initial tableau when $\rho = \rho_1$ in Algorithm 5. Let $\mathscr{I}$ be an $n \times n$ identity matrix. First, the matrix $-\rho_1^{-1}\mathscr{I}$ is added to the matrix $T_{B_0}(\lambda)_{n,K}$. Then pivotal operations are performed on all variables in the last n rows. In the third step, multiply elements in the last n rows by $-\rho_1^{-1}$ and then the columns $j \in K$ by ρ_1^{-1} of $T_B(\lambda)$. In the last step, the matrix $-\rho_1^{-1}\mathscr{I}$ is added to the matrix $T_{B_0}(\lambda)_{n,K}$. These operations provide an initial tableau for the next iteration with the initial basis B_0. The first column of the starting Tableau $T_{B_0}(\lambda+\rho_1)$ is independent of all the steps in Algorithm 6.

Algorithm 6 Updating the initial tableau

1: INPUT: The initial Tableau $T_{B_0}(\lambda + \rho_1)$
2: Add $-\rho_1^{-1}\mathscr{I}$ to $T_{B_0}(\lambda)_{n,K}$
3: Perform the pivotal operation on all variables in the last n rows.
4: Multiply elements in the last n rows by $-\rho_1^{-1}$ and then columns $j \in K$ by ρ_1^{-1} of $T_{B_0}(\lambda)$.
5: Add $-\rho_1^{-1}\mathscr{I}$ to $T_B(\lambda)_{n,K}$.
6: Go to Phase II (Algorithm 4) with current Tableau $T_{B_0}(\lambda)$.
7: OUTPUT: new Tableau $T_{B_0}(\lambda)$

4.4.2.8 Updating Basis B

Let $T_B(\lambda)$ be the current tableau for a complementary basis B and for a parameter value λ, as defined in Phase II. If $\mathscr{B}_{i0} \prec \mathbf{0}$ for at least one $i \in \{1, \ldots, h\}$ in Algorithm 4, then the basis B is not feasible for system (4.23) for the current value of λ. Then Algorithm 7 is run and the Criss-Cross Method (Algorithm 3) is invoked to find a FCB for the current value of λ.

Algorithm 7 Updating B

1: INPUT: Current Tableau $T_{B_0}(\lambda)$ such that $\mathcal{B}_{i0} \prec \mathbf{0}$ for at least one $i \in \{1, \ldots, h\}$
2: **while** $\mathcal{B}_{i0} \prec \mathbf{0}$ **do**
3: Run Criss-Cross Method (Algorithm 3)
4: **end while**
5: OUTPUT: new Tableau $T_B(\lambda)$ such that $\mathcal{B}_{i0} \succeq \mathbf{0}$

4.4.3 Example

We illustrate the spLCP and mpLCP methods on the following BOQP:

$$\begin{aligned} \min \quad & \left[\frac{1}{2}\begin{bmatrix} x_1 & x_2 \end{bmatrix}\begin{bmatrix} 6 & 0 \\ 0 & 14 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 9 & -5 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \frac{1}{2}\begin{bmatrix} x_1 & x_2 \end{bmatrix}\begin{bmatrix} 2 & 0 \\ 0 & 5 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -1 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}\right] \\ \text{s.t.} \quad & \begin{bmatrix} 3 & 5 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq 15 \\ & x_1, x_2 \geq 0, \end{aligned} \tag{4.25}$$

and the associated spLCP:

$$\begin{aligned} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} - \left(\begin{bmatrix} 2 & 0 & 3 \\ 0 & 5 & 5 \\ -3 & -5 & 0 \end{bmatrix} + \lambda\begin{bmatrix} 4 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 0 \end{bmatrix}\right)\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} &= \begin{bmatrix} -1 \\ 1 \\ 15 \end{bmatrix} + \lambda\begin{bmatrix} 10 \\ -6 \\ 0 \end{bmatrix} \\ w_i z_i &= 0 \qquad i = 1, 2, 3, \\ w_i, z_i &\geq 0 \qquad i = 1, 2, 3, \end{aligned} \tag{4.26}$$

where $\lambda \in [0, 1]$. We have $n = 2, m = 1$, and $h = n + m = 3$. The iterations of Phase I and II of the spLCP method are given in the Appendix while the obtained solutions are given in Table 4.2. Four efficient solutions are computed, each of them for a specific value of $\lambda \in [0, 1]$. For comparison, Table 4.3 shows the solutions obtained by the mpLCP method for the same example. The two methods provide the same invariancy intervals but the solutions are available in different forms. The spLCP method gives the efficient solutions at the end points of the invariancy intervals while the mpLCP method gives the efficient solutions as functions of the parameter λ for each interval.

Table 4.2: Efficient solutions to Example (4.25) obtained with the spLCP method

λ	x_1	x_2
0	1/2	0
1/10	0	0
1/6	0	0
1	0	5/14

Table 4.3: Efficient solutions to Example
(4.25) obtained with the mpLCP method

λ	x_1	x_2
$0 \leq \lambda \leq 1/10$	$\frac{(1-10\lambda)}{(2+4\lambda)}$	0
$1/10 \leq \lambda \leq 1/6$	0	0
$1/6 \leq \lambda \leq 1$	0	$\frac{(-1+6\lambda)}{(5+9\lambda)}$

4.4.4 Numerical Results

To compare the efficiency of the spLCP and mpLCP methods, we implement them in the MATLAB programming language and apply them to randomly generated strictly convex BOQPs satisfying Assumption 4.4.3. The code of the mpLCP method is available online [2]. The tests have been performed on a Lenovo Ideapad FLEX 4 with a 256 GB SSD storage, 6th Generation Intel Core i5-6200U, 2.30GHz, 2401 Mhz, 2 Cores, 4 Logical Processors and 8GB memory.

The results of the numerical experiments are collected in Table 4.4 in which the first column shows the dimension of the decision space, n, and the second column displays the number of instances that have been run for each n. The third column specifies the statistics given in the subsequent columns. The 4th and 5th columns report CPU times for diagonal positive definite matrices, $Q_1, Q_2 \in \mathbb{R}^{n\times n}$ while the 6th and 7th columns report the CPU times for general positive definite matrices, $Q_1, Q_2 \in \mathbb{R}^{n\times n}$. The times in each row are given for the same set of instances that are solved using both methods.

The MATLAB function *rand* randomly generates the matrices Q_i and vectors $\mathbf{p}_i$,

$i = 1, 2$, for the BOQPs. Generating instances that satisfy Assumption 4.4.3 for the spLCP method is time-consuming and the time it takes to generate each instance is not included in the table.

There are two major computational tasks that affect the efficiency of each method: the type of computations being performed, and the way polynomial equations are solved. Because the spLCP method relies on a pivoting scheme with real numbers rather than on symbolic computation as used in the mpLCP method, we expected that the former might be superior to the latter. However, the spLCP method uses the MATLAB function *solve* to solve the polynomial equations in Algorithm 5, while the mpLCP method solves SOPs with linear objective functions and polynomial constraints by means of the MATLAB function *fmincon* with the interior-point algorithm. We suspect that the difference in the effectiveness of these two MATLAB functions may influence each method.

We observe in Table 4.4 that, as the number of decision variables, n, increases for both types of BOQPs, the mpLCP method becomes more efficient relative to the spLCP: its mean, median, and standard deviations times get consistently smaller than the respective times for the spLCP method. The function *fmincon* emerges as a significantly more effective tool than the function *solve* because it offsets the time spent on the symbolic pivoting by the mpLCP method. As a result, given the current MATLAB environment, the mpLCP method remains the state-of-the art method for MOQPs.

4.5 Applications

In this section we apply the scalarizations (4.4) and (4.7) presented in Section 4.3 and the algorithms examined in Sections 4.4 to specific mpMOQPs resulting from applications in statistics and portfolio optimization.

Table 4.4: Statistics for the CPU times for BOQPs solved with the spLCP and mpLCP methods.

n	no. of instances	statistics	diagonal		diagonal+off-diagonal	
			spLCP	mpLCP	spLCP	mpLCP
2	10	mean	10.492	9.239	13.569	10.840
		median	7.793	7.954	12.971	9.800
		std. dev.	6.692	2.618	4.301	3.370
3	10	mean	16.491	9.885	54.782	42.269
		median	15.843	9.708	57.167	39.439
		std. dev.	5.582	3.086	25.947	24.258
4	5	mean	113.167	116.000	163.269	117.658
		median	95.871	110.780	156.310	127.808
		std. dev.	44.842	22.892	74.853	34.983
5	4	mean	216.483	146.758	517.849	199.559
		median	228.215	152.844	491.297	205.487
		std. dev.	84.863	22.144	158.765	105.393

4.5.1 The Elastic Net Problem

We show that the presented methodology can enhance linear regression when the elastic net problem is solved to select regression parameters. Let the data set have n observations with k predictors. Consider the standard linear regression model in which the response $\mathbf{y} \in \mathbb{R}^k$ is predicted by

$$\hat{\mathbf{y}} = \Phi \mathbf{x}, \tag{4.27}$$

where $\Phi = \begin{bmatrix} \boldsymbol{\phi_1} & \boldsymbol{\phi_2} & \dots & \boldsymbol{\phi_n} \end{bmatrix} \in \mathbb{R}^{k \times n}$ is the design matrix with predictors $\boldsymbol{\phi_i} \in \mathbb{R}^k$ and $\mathbf{x} \in \mathbb{R}^n$ is the vector of "parameters" to be estimated. Here the word "parameters" has a different meaning than in parametric optimization. In statistics, the response and the predictors are both known, while the vector of parameters (typically denoted by $\hat{\beta}$) remains unknown and needs to be estimated so that the residual squared error is minimized. In the context of optimization, the unknown parameters become variables to be determined in the process of minimizing the squared error, which is modeled as the QP

$$\min_{\mathbf{x}\in\mathbb{R}^n} \quad f_1(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{p}^T\mathbf{x} + c, \tag{4.28}$$

where $Q = 2\Phi^T\Phi$ is positive-definite, $\mathbf{p} = -2\Phi^T\mathbf{y}$ and $c = \mathbf{y}^T\mathbf{y}$. To avoid confusion, we refer to these parameters $\mathbf{x}$ as coefficients of (4.27). Because model (4.27) performs poorly in both prediction and interpretation when the estimated coefficients are computed from (4.28), this QP has been augmented by two penalty terms: the ridge term imposes an ℓ_2-penalty while the lasso term imposes an ℓ_1-penalty on $\mathbf{x}$ [140]. This leads to the elastic net problem.

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^n} \quad f_{\text{elastic-net}}(\mathbf{x};\alpha,\beta) &= \tfrac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{p}^T\mathbf{x} + c + \tfrac{\alpha}{2}\mathbf{x}^T\mathbf{x} + \beta\textstyle\sum_{i=1}^n |x_i| \\ \alpha,\beta &\geq 0, \end{aligned} \tag{4.29}$$

where α, β play the role of modeling parameters (in agreement with the terminology introduced in Section 4.2). Because these parameters strongly affect the performance of model (4.27), they require tuning.

In [21], the parameter tuning is modeled as a bilevel optimization problem involving minimization of the squared error on the efficient set of the triobjective QP (TOQP)

$$\min_{\mathbf{x}\in\mathbb{R}^n} \quad [f_1(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{p}^T\mathbf{x} + c,\ f_2(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T\mathbf{x},\ f_3(\mathbf{x}) = \textstyle\sum_{i=1}^n |x_i|], \tag{4.30}$$

for which (4.29) is the associated weighted-sum SOP. Since the optimization over the efficient set remains challenging [128], an algorithm is proposed to compute the minimum squared error (minSE) with respect to continuously changing values of β but on a user-selected-grid of fixed values of α. Because of the grid, an optimal solution to the bilevel problem may not be achieved.

The methodology presented in this paper can further facilitate parameter tuning because (4.30) can now be solved for its complete parametric efficient set which then can be used to find α, β yielding the smallest squared error. We demonstrate this application on a simple example with $n = 3, k = 5$, the following data

$$\Phi = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 3 & 2 \\ 1 & 2 & 2 \\ 1 & 4 & 5 \\ 1 & 3.5 & 2.5 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 0.55 \\ 0.623 \\ 0.587 \\ 0.569 \\ 0.758 \end{bmatrix}, \tag{4.31}$$

and by solving (4.30) in three ways.

4.5.1.1 The weighted-sum SOP

Applying the weighted-sum (4.4) to (4.30) yields the following mpQP:

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{R}^3} \quad \lambda_1(\tfrac{1}{2}\mathbf{x}^TQ\mathbf{x} + \mathbf{p}^T\mathbf{x} + c) + \tfrac{\lambda_2}{2}\mathbf{x}^T\mathbf{x} + (1 - \lambda_1 - \lambda_2)\textstyle\sum_{i=1}^n |x_i| \\ &\lambda \in \Lambda', \end{aligned} \tag{4.32}$$

where $\Lambda' = \{\boldsymbol{\lambda} \in \mathbb{R}^2 : \lambda_1, \lambda_2 \geq 0, \lambda_1 + \lambda_2 \leq 1\}$. The problem is reformulated to make all variables nonnegative which at the same time transforms f_3 into a linear function. Having solved (4.30) with the mpLCP method, we obtain a partition of Λ' into 6 invariancy regions (IR) and the optimal solution functions $\hat{x}_i(\lambda_1, \lambda_2)$, $i = 1, 2, 3$ to (4.32) in each region that are all listed in Table 4.6. To return to the original parameters, we substitute

$$\lambda_1 = \frac{1}{1 + \alpha + \beta} \text{ and } \lambda_2 = \frac{\alpha}{1 + \alpha + \beta} \tag{4.33}$$

in Table 4.6 and obtain the invariancy regions and the optimal solutions $\hat{x}_i(\alpha, \beta), i = 1, 2, 3$ for $\alpha, \beta \geq 0$ in Table 4.7. The partitions of both parameter spaces are depicted in Figure 4.1a and 4.1b. For all $\boldsymbol{\lambda} \in \Lambda' \setminus \{(0, 0)\}$, by Cor. 1 in [68], all optimal solutions to (4.32) are efficient to (4.30). In the case of $\boldsymbol{\lambda} = (0, 0)$, note that $f_3(\mathbf{x})$ has a unique minimizer which is also efficient.

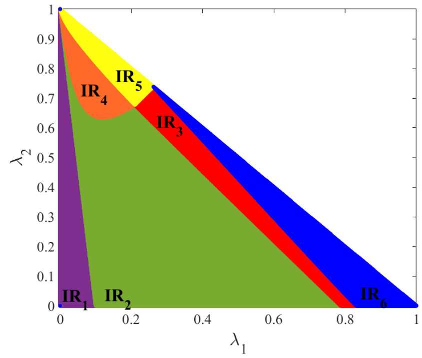
The Pareto set for (4.30) is depicted in Figure 4.2, while Figure 4.3a shows the points $(\alpha, \beta, \hat{f}_1(\alpha, \beta)) = (\alpha, \beta, f_1(\hat{\mathbf{x}}(\alpha, \beta))$ for (4.32), where $f_1(\hat{\mathbf{x}}(\alpha, \beta))$ is the minSE. In all four figures the same colors are consistently used for the same regions and the associated Pareto points. Using the obtained solutions, model (4.27) is available in the parametric form

$$\hat{\mathbf{y}} = \Phi \, \hat{\mathbf{x}}(\alpha, \beta)$$

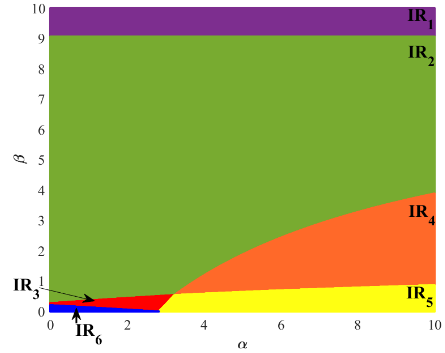
where $\hat{\mathbf{x}}(\alpha, \beta) = \begin{bmatrix} \hat{x}_1(\alpha, \beta) & \hat{x}_2(\alpha, \beta) & \hat{x}_3(\alpha, \beta) \end{bmatrix}^T$ is the efficient solution to (4.30) for $\alpha, \beta \geq 0$.

The solutions $\hat{\mathbf{x}}(\alpha, \beta)$ listed in Table 4.7 are rational functions of parameters α and β and their denominators are polynomial functions of α with a degree of at most 3. For a fixed $\alpha = \bar{\alpha} \geq 0$, all these solutions are linear functions of β and the the optimal squared error $f_1(\hat{\mathbf{x}}(\bar{\alpha}, \beta))$ is a quadratic function of β for $\beta \geq 0$ in $IR_i, i = 2, \ldots, 6$. These observations agree with the results in [21].

The decision maker (DM) can examine the invariancy regions and select values for the tuning parameters to obtain the regression coefficients. While the zero solution obtained in $I\mathcal{R}_1$ for $\beta > 9.07$ is typically ignored in linear regression, the efficient solutions in the other regions can be of interest. In some regions, the components of the solutions are explicitly zero which agrees with the intention of keeping the linear regression model parsimonious [140].



(a) Parameters $\lambda_1, \lambda_2 \geq 0, \lambda_1 + \lambda_2 \leq 1$.



(b) Parameters $\alpha, \beta \in [0, 10]$.

Figure 4.1: Partition of the parameter space into six invariancy regions for elastic net problem (4.30) obtained by the mpLCP method on problem (4.32).

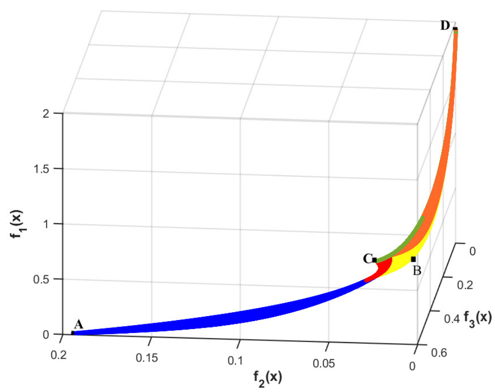


Figure 4.2: Pareto set for elastic net problem (4.30) obtained by the mpLCP method on problem (4.32).

For example, with two nonzero components in the efficient solution, IR_3 bears further investigation. Note that for $\bar{\alpha} = 1$ and $\bar{\beta} = 0.3$, $\hat{\mathbf{x}} = \hat{\mathbf{x}}(1, 0.3) = \begin{bmatrix} 0.028 & 0.181 & 0 \end{bmatrix}^T$ is in IR_3 with the minSE of $f_1(\hat{\mathbf{x}}) = 0.332$. Consider, now, IR_3 with $\bar{\alpha} = 1$. Then $\hat{\mathbf{x}} = \hat{\mathbf{x}}(1, \beta) = \begin{bmatrix} 0.17 - 0.472\beta & 0.126\beta + 0.143 & 0 \end{bmatrix}^T$ for $\beta \in [0.134, 0.359]$ in IR_3 and the minSE is given by the function $f_1(\hat{\mathbf{x}}) = f_1(\hat{\mathbf{x}}(\bar{\alpha}, \beta)) = 0.107\beta^2 + 0.121\beta + 0.064$. Computing $\hat{\beta} = \arg\min_\beta\{f_1(\hat{\mathbf{x}}(\bar{\alpha}, \beta)) : \beta \in [0.134, 0.359]\}$, the DM can find $\hat{\beta} = 0.134$ that yields the minSE of $f_1(\hat{\mathbf{x}}(\bar{\alpha}, \hat{\beta})) = 0.083$.

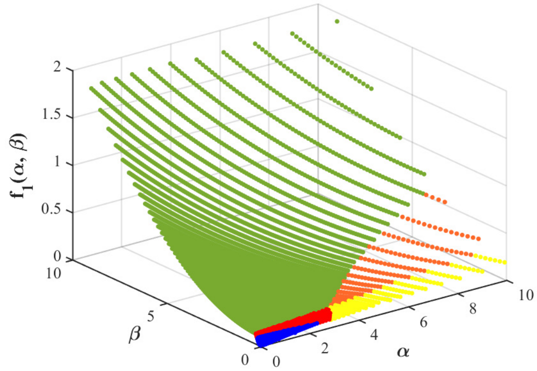


Figure 4.3: minSE computed by the mpLCP method on problem (4.32) .

4.5.2 Portfolio Optimization

Consider the parametric triobjective portfolio optimization problem (TOPOP) with two quadratic objectives being the variances of portfolio return and liquidity and one parametric linear objective being the expected return,

$$
\begin{array}{ll}
\min_{\mathbf{x}\in\mathbb{R}^3} & \left[f_1(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q_1 \mathbf{x}, f_2(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q_2 \mathbf{x}, f_3(\mathbf{x};\theta) = -\boldsymbol{p}_3(\theta)^T \mathbf{x}\right] \\
\text{s.t.} & \mathbf{1}^T\mathbf{x} = 1 \\
& \mathbf{x} \geqq \mathbf{0},
\end{array} \tag{4.34}
$$

and with the following data

$$
Q_1 = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 2 & 0 \\ -1 & 0 & 2.5 \end{bmatrix}, \quad Q_2 = \begin{bmatrix} 3 & -1 & 0 \\ -1 & 4 & 1 \\ 0 & 1 & 3.5 \end{bmatrix}, \quad \boldsymbol{p}_3(\theta) = \begin{bmatrix} -13.5 \\ 20 \\ \theta \end{bmatrix} \text{ for } \theta \in [15, 17].
$$

A nonparametric version of this problem is solved in [68]. The modeling parameter θ represents the uncertain expected return on the capital invested in the third security. Solving this problem requires computing the set of (weakly) efficient solutions $\mathcal{X}_{(w)E} := \{X_{(w)E}(\theta)\}_{\theta\in[15,17]}$ and (weak) Pareto outcomes $\mathcal{Y}_{(w)P} := \{Y_{(w)P}(\theta)\}_{\theta\in[15,17]}$, as given in

Def. 4.2.4 and 4.2.5.

4.5.2.1 The modified-hybrid SOP

Problem (4.34) is reformulated with the modified hybrid method (4.7) that is particularly suited to this application. The two risk functions are combined using a parameter λ that allows the DM to weigh these functions differently, while the uncertain expected return (to be maximized) is bounded from below by another parameter ϵ. Additionally, the equality constraint is reformulated into two inequalities. We obtain an mpQP to type (4.17) with the modeling parameter θ and two parameters λ and ϵ which, despite their scalarization role, fit the context of this application very well:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^3} \quad & \tfrac{1}{2}\mathbf{x}^T Q(\lambda)\mathbf{x} \\ \text{s.t.} \quad & \tilde{A}(\theta)\mathbf{x} \leqq \tilde{\mathbf{b}}(\epsilon) \\ & \mathbf{x} \geqq \mathbf{0} \\ & \theta \in \Theta = [15, 17], \lambda \in \Lambda' = [0, 1], \epsilon \in \mathcal{E} = [-20, 13.5], \end{aligned} \tag{4.35}$$

where $Q(\lambda) = \lambda Q_1 + (1-\lambda)Q_2$, and

$$Q(\lambda) = \begin{bmatrix} 3-2\lambda & \lambda-1 & -\lambda \\ \lambda-1 & 4-2\lambda & 1-\lambda \\ -\lambda & 1-\lambda & 7/2-\lambda \end{bmatrix}, \tilde{A}(\theta) = \begin{bmatrix} 13.5 & -20 & -\theta \\ 1 & 1 & 1 \\ -1 & -1 & -1 \end{bmatrix}, \tilde{\mathbf{b}}(\epsilon) = \begin{bmatrix} \epsilon \\ 1 \\ -1 \end{bmatrix}.$$

The parameter intervals Θ and $\mathcal{E}$ are normalized. Applying $\theta^{nor} = \frac{\theta-\theta^{min}}{\theta^{max}-\theta^{min}} = \frac{\theta-15}{17-15}$ we have $\Theta^{nor} = [0, 1]$. Applying $\epsilon^{nor} = \frac{\epsilon-\epsilon^{min}}{\epsilon^{max}-\epsilon^{min}} = \frac{\epsilon+20}{13.5-(-20)}$, where $\epsilon^{min} = \min\{-\mathbf{p}(\theta)_3^T\mathbf{x} : \mathbf{1}^T\mathbf{x} = 1, \mathbf{x} \geqq \mathbf{0}, \theta \in [15, 17]\} = -20$ and $\epsilon^{max} = \max\{-\mathbf{p}_3(\theta)^T\mathbf{x} : \mathbf{1}^T\mathbf{x} =$

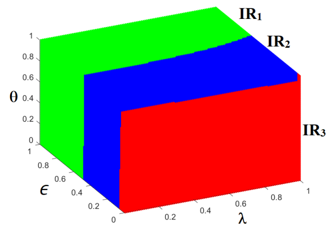


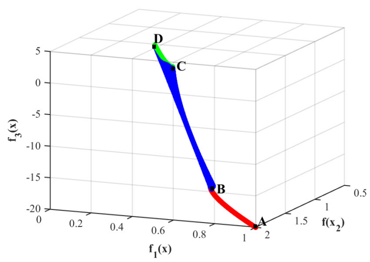
Figure 4.4: Partition of the parameter space for portfolio problem (4.34) obtained with the mpLCP method on problem (4.35).

$1, \mathbf{x} \geqq \mathbf{0}, \theta \in [15, 17]\} = 13.5$, we have $\mathcal{E}^{nor} = [0, 1]$.

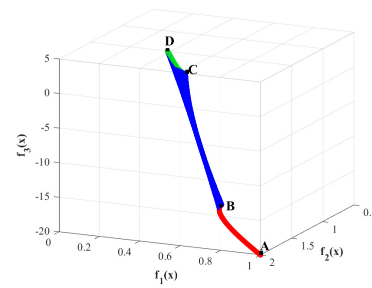
The mpLCP method Problem (4.35) is solved in the parameter space $\Theta^{nor} \times \Lambda' \times \mathcal{E}^{nor} = [0, 1]^3$ with the mpLCP method. At optimality, the parameter space is partitioned into three invariancy regions, IR_i, $i = 1, 2, 3$, that are depicted in Figure 4.4 and listed in Table 4.8 along with their respective optimal solution functions $\hat{x}_i(\theta, \lambda, \epsilon), i = 1, 2, 3$ that, by Proposition 4.4.2, are efficient to (4.35). We have

$$\mathcal{X}_E = \{\hat{x}_i(\theta, \lambda, \epsilon), i = 1, 2, 3 \text{ for } \theta \in [15, 17], \lambda \in [0, 1], \epsilon \in [-20, 13.5]\}.$$

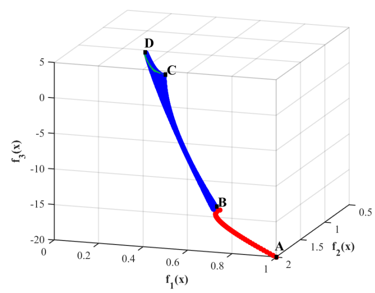
Note that the efficient solutions in IR_1 do not depend on λ and the efficient solutions in IR_3 do not depend on θ and ϵ. However, in IR_2, the efficient solutions depend on all three parameters. To show the evolution of the parametric Pareto sets $\mathcal{Y}_P(\theta)$ in the objective space w.r.t. θ, three Pareto sets for selected values of θ are depicted in Figure 4.5 and the coordinates of four points are reported in Table 4.5 for comparison. Note that point A in IR_3 remains Pareto for each value of θ.



(a) $\theta = 15$



(b) $\theta = 16$



(c) $\theta = 17$

Figure 4.5: Pareto sets $\mathcal{Y}_P(15), \mathcal{Y}_P(16), \mathcal{Y}_P(17) \subset \mathcal{Y}_P$ for portfolio problem (4.34) obtained with the mpLCP method on problem (4.35).

Table 4.5: Coordinates of four Pareto points in Figure 4.5 for portfolio problem (4.34)

θ	15			16			17		
	$f_1(\mathbf{x})$	$f_2(\mathbf{x})$	$f_3(\mathbf{x})$	$f_1(\mathbf{x})$	$f_2(\mathbf{x})$	$f_3(\mathbf{x})$	$f_1(\mathbf{x})$	$f_2(\mathbf{x})$	$f_3(\mathbf{x})$
A	1.000	2.000	-20.000	1.000	2.000	-20.000	1.000	2.000	-20.000
B	0.576	1.276	-18.320	0.556	1.232	-18.320	0.537	1.188	-18.320
C	0.173	0.530	-3.455	0.173	0.530	-3.664	0.173	0.530	-3.873
D	0.120	0.650	0.360	0.120	0.650	0.040	0.120	0.648	-0.280

A similar analysis to that in [68] can be carried out for $\theta^{nor} \in [0, 1]$. Assume that (i) the vectors $\boldsymbol{p}_3(\theta)$ and $\mathbf{1}$ are linearly independent for all $\theta \in \Theta$ and (ii), at optimality of problem (4.35), the inequality ϵ-constraint is active and all the nonnegativity constraints are inactive for a subset of the parameter space. Then the optimal objective value function in

(4.35), being the minimum weighted risk (MWR), $\hat{\sigma}(\theta, \lambda, \epsilon)$, can be obtained in this subset of the parameter space [68]

$$\hat{\sigma}(\theta, \lambda, \epsilon) = \frac{1}{2} \begin{bmatrix} \epsilon & 1 \end{bmatrix} \Psi(\theta, \lambda)^{-1} \begin{bmatrix} \epsilon \\ 1 \end{bmatrix}, \tag{4.36}$$

where

$$\Psi(\theta, \lambda) = \begin{bmatrix} \boldsymbol{p}_3(\theta)^T Q(\lambda)^{-1} \boldsymbol{p}_3(\theta) & -\boldsymbol{p}_3(\theta)^T Q(\lambda)^{-1} \mathbf{1} \\ -\boldsymbol{p}_3(\theta)^T Q(\lambda)^{-1} \mathbf{1} & \mathbf{1}^T Q(\lambda)^{-1} \mathbf{1} \end{bmatrix}.$$

For $\theta \in \Theta, \lambda \in \Lambda'$ the matrix $\Psi(\theta, \lambda)$ is PD as is its inverse; therefore, $\hat{\sigma}(\theta, \lambda, \epsilon)$ is a strictly convex quadratic function of ϵ.

Since the assumptions (i) and (ii) above hold in IR_2, we further examine the obtained MWR function and the associated efficient solution functions in this region. From (4.36), the MWR function is available for normalized parameters,

$$\hat{\sigma}(\theta^{nor}, \lambda, \epsilon^{nor}) = \frac{1}{2} \begin{bmatrix} 33.5\epsilon^{nor} - 20 & 1 \end{bmatrix} \Psi(\theta^{nor}, \lambda)^{-1} \begin{bmatrix} 33.5\epsilon^{nor} - 20 \\ 1 \end{bmatrix}. \tag{4.37}$$

Letting $\lambda = 0.5$ and assuming a desired gain of 9, i.e., $\epsilon = -9$ and $\epsilon^{nor} = 0.328$, this function becomes

$$\hat{\sigma}(\theta^{nor}, 0.5, 0.328) = (368(\theta^{nor})^2 - 120\theta^{nor} + 68{,}435)/(8(96(\theta^{nor})^2 + 860\theta^{nor} + 19{,}695)),$$

which is decreasing for $\theta^{nor} \in [0, 1]$. Thus, the lowest MWR, $\hat{\sigma}(1, 0.5, 0.328) = 0.416$, is achieved at $\theta^{nor} = 1$. The associated efficient portfolio

$$\hat{\mathbf{x}}(\theta^{nor}, 0.5, 0.328) = \begin{bmatrix} \frac{112(\theta^{nor})^2 + 1{,}220.056\theta^{nor} + 10{,}991.736}{192(\theta^{nor})^2 + 1{,}720\theta^{nor} + 39{,}390} \\ \frac{80(\theta^{nor})^2 - 319.922\theta^{nor} + 15{,}373.866}{192(\theta^{nor})^2 + 1{,}720\theta^{nor} + 39{,}390} \\ \frac{819.866\theta^{nor} + 13{,}024.398}{192(\theta^{nor})^2 + 1{,}720\theta^{nor} + 39{,}390} \end{bmatrix}, \quad \theta^{nor} \in [0, 1],$$

is equal to $\hat{\mathbf{x}}(1, 0.5, 0.328) = \begin{bmatrix} 0.298 & 0.367 & 0.335 \end{bmatrix}^T$ for $\theta^{nor} = 1$.

We now analyze the MWR with respect to the bound ϵ on the negative expected return that is minimized.

Corollary 4.5.0.1. *Let the matrices Q_1, Q_2 be positive definite and the vectors $\boldsymbol{p}_3(\theta)$ and $\mathbf{1}$ be linearly independent for all $\theta \in \Theta$ for TOPOP (4.34). At optimality of problem (4.35), let the inequality ϵ-constraint be active and all the nonnegativity constraints be inactive for a subset of the parameter space $\Theta \times \Lambda' \times \mathcal{E}$. Then at the value of the expected return $\hat{\epsilon} = \arg\min_{\epsilon \in \mathcal{E}'} \hat{\sigma}(\theta, \lambda, \epsilon)$, the lowest MWR w.r.t. ϵ, $\hat{\sigma}(\theta, \lambda, \hat{\epsilon})$, is independent of the parameter θ.*

Proof. Solving $\frac{\partial \hat{\sigma}(\theta,\lambda,\epsilon)}{\partial \epsilon} = 0$ for ϵ, we obtain $\hat{\epsilon} = \hat{\epsilon}(\theta, \lambda) = \frac{-\boldsymbol{p}_3(\theta)^T Q(\lambda)^{-1}\mathbf{1}}{\mathbf{1}^T Q(\lambda)^{-1}\mathbf{1}}$. Substituting $\hat{\epsilon}$ into (4.36) we get $\hat{\sigma}(\theta, \lambda, \hat{\epsilon}) = \frac{1}{2\mathbf{1}^T Q(\lambda)^{-1}\mathbf{1}}$, which is independent of θ. □

We can illustrate Corollary 4.5.0.1 on IR_2 because this region satisfies the required assumptions. The lowest MWR, $\hat{\sigma}(\theta, \lambda, \hat{\epsilon}) = \frac{1}{2\mathbf{1}^T Q(\lambda)^{-1}\mathbf{1}} = \frac{-(6\lambda^3+15\lambda^2-86\lambda+71)}{2(6\lambda^2+36\lambda-67)}$, can be calculated by the DM for any $\lambda \in [0, 1]$. For example, if $\lambda = 0.5$, then $\hat{\sigma}(\theta, 0.5, \hat{\epsilon}) = 0.342$. The corresponding bound, $\hat{\epsilon}$, can be calculated for a specific value of $\theta \in [15, 17]$. For example, given $\lambda = 0.5$ and choosing $\theta = 17$, we have $\hat{\epsilon}(17, 0.5) = -3.342$ or $\hat{\epsilon}^{nor} = 0.497$, which is the upper end value for ϵ^{nor} in IR_2.

4.6 Conclusion

This paper appears to present the first numerical study on solving parametric MOPs. We showed that the state-of-the art parametric optimization algorithms allow computation of efficient sets for convex mpMOQPs in which parameters model unknown or uncertain quantities. mpMOQPs are solved by scalarization which introduces additional parameters when transforming the original problem into mpQPs. Because the efficient set for mpMOQPs is a parametrized collection of the efficient sets, the former can be computed by the algorithms that have been designed for mpQPs as long as they are able to handle multiple parameters and work well on mpQPs of different types. To offer flexibility with mpQPs, we proposed a generalized weighted-sum scalarization that reduces to six SOPs of the weighted-sum/epsilon-constraint type that can be applied depending on the real-life context and available solver.

We compared two LCP-based methods for solving spQPs with linear constraints: the mpLCP method, a recently developed method to solve mpQPs with linear constraints, and the spLCP method, a method proposed in the 1990s and never implemented. We anticipated that the latter would be more efficient than the former due to its special features but discovered otherwise. The mpLCP method, which turned out to be superior to three other methods in [68], is also superior to the spLCP method in the current MATLAB environment. Consequently, we conclude that the mpLCP method determines the state-of-the-art in solving mpMOQPs with linear constraints. Using the elastic net problem in statistics and portfolio optimization, we showed that matching the two methods with specific scalarizations provides mutually complementary insight into the real-life context and therefore supports decision making.

Future research can go in different directions. The spLCP method should not be ignored because software advances in solving systems of polynomial equations might make it attractive in the future. More generally, since parametric multiobjective optimization requires methodologies that are customized at the stage of SOP reformulation and the stage of algorithmic development, it is advisable to design algorithms for specific classes of mpSOPs and utilize them for mpMOPs.

Supporting Information

A Solving the BOQP in Example (4.25) with the spLCP method

Below we present the iterations of the spLCP method when solving the BOQP in (4.25).

Initialization: The full rank factorization of matrix ΔM yields $\Psi = \begin{bmatrix} 2 & 0 \\ 0 & 3 \\ 0 & 0 \end{bmatrix}$ and $\boldsymbol{\kappa} = \begin{bmatrix} 5 \\ -2 \end{bmatrix}$.

Let $B_0 = \{1, 2, 3\}$; for the sake of clarity, we write $B_0 = \{w_1, w_2, w_3\}$. The initial Tableau $T_{B_0}(\lambda)$ without the columns associated with the basic variables and columns $h+1$ to $h+n$ is given below.

Initial Tableau $T_{B_0}(\lambda)$

	$\mathfrak{g}_{.0}$	$\mathfrak{g}_{.6}$	$\mathfrak{g}_{.7}$	$\mathfrak{g}_{.8}$	$\mathfrak{g}_{.9}$	$\mathfrak{g}_{.10}$
w_1	-1+10λ	-2-4λ	0	-3	2	0
w_2	1-6λ	0	-5-9λ	-5	0	3
w_3	15	3	5	0	0	0
v_{κ_1}	5	-2	0	0	0	0
v_{κ_2}	-2	0	-3	0	0	0

Iteration 1: Set $\lambda = 0$ in the initial tableau and obtain $T_{B_0}(0)$.

Tableau $T_{B_0}(0)$

	$\mathfrak{g}_{.0}$	$\mathfrak{g}_{.6}$	$\mathfrak{g}_{.7}$	$\mathfrak{g}_{.8}$	$\mathfrak{g}_{.9}$	$\mathfrak{g}_{.10}$
w_1	-1	$\boxed{-2}$	0	-3	2	0
w_2	1	0	-5	-5	0	3
w_3	15	3	5	0	0	0
v_{κ_1}	5	-2	0	0	0	0
v_{κ_2}	-2	0	-3	0	0	0

Since $(T_{B_0}(0))_{1,0} = -1$, we have that $(T_{B_0}(0))_{.0} \ngeq \mathbf{0}$. Thus we apply Phase I. Let $r = \min\{i : (T_{B_0}(0))_{i,0} < 0, i = 1,2,3\} = 1$. Then $(T_{B_0}(0))_{r,\bar{r}} = (T_{B_0}(0))_{1,6} = -2 < 0$, and a diagonal pivot is performed.

Tableau $T_{B_1}(0)$

	$\mathfrak{g}_{.0}$	$\mathfrak{g}_{.1}$	$\mathfrak{g}_{.7}$	$\mathfrak{g}_{.8}$	$\mathfrak{g}_{.9}$	$\mathfrak{g}_{.10}$
z_1	1/2	-1/2	0	3/2	-1	0
w_2	1	0	-5	-5	0	3
w_3	27/2	3/2	5	-9/2	3	0
v_{κ_1}	6	-1	0	3	-2	0
v_{κ_2}	-2	0	-3	0	0	0

The obtained complementary basis $B_1 = \{z_1, w_2, w_3\}$ is feasible for $\lambda = 0$ since $(T_{B_1}(0))_{i,0} > 0$ for $i = 1,2,3$. Hence Phase I is terminated. A BFCS is obtained for spLCP (4.26) for $\lambda = 0$

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 27/2 \end{bmatrix} \qquad \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 0 \\ 0 \end{bmatrix}, \tag{4.38}$$

which yields an efficient solution to BOQP (4.25)

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 0 \end{bmatrix}.$$

Phase II is initiated. Since $(T_{B_1}(0))_{i,0} > 0$, the condition $\mathscr{B}_{i0} \succ \mathbf{0}$ holds for all $i = 1, 2, 3$. Algorithm 5 is used to find an invariancy interval for λ, i.e., the increment τ above 0, in which (4.38) remains the BFCS. To do so, the polynomial equations, $P_{i1}(\tau) = 0$ for each $i = 1, 2, 3$ and $P_2(\tau) = 0$, are solved for their positive roots:

$$P_{i1}(\tau) = \det\big(-T_B(\lambda)_{KK} + \tau^{-1}\mathscr{I}\big)\big(T_B(\lambda)_{i0} - (-T_B(\lambda)_{iK})(T_B(\lambda)_{KK} + \tau^{-1}\mathscr{I})^{-1}T_B(\lambda)_{K0}\big) = 0.$$

For $i = 1$, $$\det\left(\begin{bmatrix} 2+\tau^{-1} & 0 \\ 0 & \tau^{-1} \end{bmatrix}\right)\left(1/2 - \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} 2+\tau^{-1} & 0 \\ 0 & \tau^{-1} \end{bmatrix}^{-1}\begin{bmatrix} 6 \\ -2 \end{bmatrix}\right) = 0$$

$$\left(\frac{2\tau+1}{\tau^2}\right)\left(\frac{1-10\tau}{4\tau+2}\right) = 0$$

$$\alpha_1 = \tau = 1/10.$$

For $i = 2$, $$\det\left(\begin{bmatrix} 2+\tau^{-1} & 0 \\ 0 & \tau^{-1} \end{bmatrix}\right)\left(1 - \begin{bmatrix} 0 & -3 \end{bmatrix}\begin{bmatrix} 2+\tau^{-1} & 0 \\ 0 & \tau^{-1} \end{bmatrix}^{-1}\begin{bmatrix} 6 \\ -2 \end{bmatrix}\right) = 0$$

$$\left(\frac{2\tau+1}{\tau^2}\right)(1-6\tau) = 0$$

$$\alpha_2 = \tau = 1/6.$$

For $i = 3$,
$$\det\left(\begin{bmatrix} 2+\tau^{-1} & 0 \\ 0 & \tau^{-1} \end{bmatrix}\right)\left(27/2 - \begin{bmatrix} -3 & 0 \end{bmatrix}\begin{bmatrix} 2+\tau^{-1} & 0 \\ 0 & \tau^{-1} \end{bmatrix}^{-1}\begin{bmatrix} 6 \\ -2 \end{bmatrix}\right) = 0$$
$$\left(\frac{2\tau+1}{\tau^2}\right)(2\tau+3) = 0$$
$$\alpha_3 = \tau = \infty.$$

Then $\rho_1 = \min\{\alpha_1, \alpha_2, \alpha_3\} = \min\{1/10, 1/6, \infty\} = 1/10$. Equation $P_2(\tau) = 0$ is solved:

$$P_2(\tau) = \det\big(- T_B(\lambda)_{KK} + \tau^{-1}\mathscr{I}\big) = 0$$

$$\det\left(\begin{bmatrix} 2+\tau^{-1} & 0 \\ 0 & \tau^{-1} \end{bmatrix}\right) = 0$$
$$\left(\frac{2\tau+1}{\tau^2}\right) = 0$$

Since the roots of equation $P_2(\tau) = 0$ are not positive, set $\rho_2 = \tau = \infty$. Find $\rho = \min\{1-\lambda, \rho_1, \rho_2\} = \min\{1, 1/10, \infty\} = \rho_1 = 1/10$, and the invariancy interval for λ for the current BFCS (4.38) is $[0, 1/10]$. The starting tableau for the next iteration is constructed.

<u>Iteration 2:</u> Set $\lambda = 1/10$ in the initial tableau and obtain $T_{B_0}(1/10)$.

Tableau $T_{B_0}(1/10)$

	$\mathfrak{g}_{.0}$	$\mathfrak{g}_{.6}$	$\mathfrak{g}_{.7}$	$\mathfrak{g}_{.8}$	$\mathfrak{g}_{.9}$	$\mathfrak{g}_{.10}$
w_1	0	-12/5	0	-3	2	0
w_2	2/5	0	-59/10	-5	0	3
w_3	15	3	5	0	0	0
v_{κ_1}	5	-2	0	0	0	0
v_{κ_2}	-2	0	-3	0	0	0

The resulting $(T_{B_0}(1/10))_{i0} \geq 0$ for $i = 1, 2, 3$, i.e., the current complementary basis $B_2 = B_0 = \{w_1, w_2, w_3\}$ is feasible for $\lambda = 1/10$. A new BFCS is obtained for spLCP (4.26) for $\lambda = 1/10$:

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 2/5 \\ 15 \end{bmatrix} \qquad \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{4.39}$$

which yields an efficient solution to BOQP (4.25)

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Since $(T_{B_0}(1/10))_{i,0} > 0$, the condition $\mathscr{B}_{i0} \succ \mathbf{0}$ holds for all $i = 1, 2, 3$. Algorithm 5 is used to find an invariancy interval for λ, i.e., the increment τ above 1/10, in which (4.39) remains the BFCS. To do so, polynomial equations, $P_{i1}(\tau) = 0$ for each $i = 1, 2, 3$, and $P_2(\tau) = 0$, are solved for their positive roots:

$$P_{i1}(\tau) = \det\big(-T_B(\lambda)_{KK} + \tau^{-1}\mathscr{I}\big)\big(T_B(\lambda)_{i0} - (-T_B(\lambda)_{iK})(T_B(\lambda)_{KK} + \tau^{-1}\mathscr{I})^{-1}T_B(\lambda)_{K0}\big) = 0.$$

$$\text{For } i=1, \qquad \det\left(\begin{bmatrix}\tau^{-1} & 0\\ 0 & \tau^{-1}\end{bmatrix}\right)\left(0-\begin{bmatrix}-2 & 0\end{bmatrix}\begin{bmatrix}\tau^{-1} & 0\\ 0 & \tau^{-1}\end{bmatrix}^{-1}\begin{bmatrix}5\\ -2\end{bmatrix}\right)=0$$

$$\left(\frac{1}{\tau^2}\right)(10\tau)=0$$

$$\alpha_1=\tau=\infty.$$

$$\text{For } i=2, \qquad \det\left(\begin{bmatrix}\tau^{-1} & 0\\ 0 & \tau^{-1}\end{bmatrix}\right)\left(2/5-\begin{bmatrix}0 & -3\end{bmatrix}\begin{bmatrix}\tau^{-1} & 0\\ 0 & \tau^{-1}\end{bmatrix}^{-1}\begin{bmatrix}5\\ -2\end{bmatrix}\right)=0$$

$$\left(\frac{1}{\tau^2}\right)(2/5-6\tau)=0$$

$$\alpha_2=\tau=1/15.$$

$$\text{For } i=3, \qquad \det\left(\begin{bmatrix}\tau^{-1} & 0\\ 0 & \tau^{-1}\end{bmatrix}\right)\left(15-\begin{bmatrix}0 & 0\end{bmatrix}\begin{bmatrix}\tau^{-1} & 0\\ 0 & \tau^{-1}\end{bmatrix}^{-1}\begin{bmatrix}5\\ -2\end{bmatrix}\right)=0$$

$$\left(\frac{15}{\tau^2}\right)=0$$

$$\alpha_3=\tau=\infty.$$

Then $\rho_1=\min\{\alpha_1,\alpha_2,\alpha_3\}=\min\{\infty,1/15,\infty\}=1/15$. Equation $P_2(\tau)=0$ is solved.

$$P_2(\tau)=\det\big(-T_B(\lambda)_{KK}+\tau^{-1}\mathscr{I}\big)=0$$

$$\det\left(\begin{bmatrix}\tau^{-1} & 0\\ 0 & \tau^{-1}\end{bmatrix}\right) = 0$$
$$\left(\frac{1}{\tau^2}\right) = 0$$

Set $\rho_2 = \tau = \infty$. Find $\rho = \min\{1-\lambda, \rho_1, \rho_2\} = \min\{9/10, 1/15, \infty\} = \rho_1 = 1/15$, and the invariancy interval for λ for the current BFCS (4.39) is $[1/10, (1/10)+(1/15)] = [1/10, 1/6]$. The starting tableau for the next iteration is constructed.

<u>Iteration 3:</u> Set $\lambda = 1/6$ in the initial tableau and obtain $T_{B_0}(1/6)$.

Tableau $T_{B_0}(1/6)$

	$\mathfrak{g}_{.0}$	$\mathfrak{g}_{.6}$	$\mathfrak{g}_{.7}$	$\mathfrak{g}_{.8}$	$\mathfrak{g}_{.9}$	$\mathfrak{g}_{.10}$
w_1	4/6	-16/6	0	-3	2	0
w_2	0	0	-39/6	-5	0	3
w_3	15	3	5	0	0	0
v_{κ_1}	5	-2	0	0	0	0
v_{κ_2}	-2	0	-3	0	0	0

The resulting $(T_{B_0}(1/6))_{i0} \geq 0$ for $i = 1, 2, 3$, i.e., the current complementary basis $B_3 = B_0 = \{w_1, w_2, w_3\}$ is feasible for $\lambda = 1/6$. A BFCS is obtained for spLCP (4.26) for $\lambda = 1/6$:

$$\begin{bmatrix} w_1\\ w_2\\ w_3\end{bmatrix} = \begin{bmatrix} 4/6\\ 0\\ 15\end{bmatrix} \qquad \begin{bmatrix} z_1\\ z_2\\ z_3\end{bmatrix} = \begin{bmatrix} 0\\ 0\\ 0\end{bmatrix}, \tag{4.40}$$

which yields an efficient solution to BOQP (4.25)

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Since $(T_{B_0}(1/6))_{i,0} > 0$, the condition $\mathscr{B}_{i0} \succ \mathbf{0}$ holds for all $i = 1, 2, 3$ Algorithm 5 is used to find an invariancy interval, i.e., the increment τ from above 1/6, in which (4.40) remains the BFCS. To do so, polynomial equations, $P_{i1}(\tau) = 0$ for each $i = 1, 2, 3$ and $P_2(\tau) = 0$, are solved for their positive roots.

$$P_{i1}(\tau) = \det\big(- T_B(\lambda)_{KK} + \tau^{-1}\mathscr{I}\big)\big(T_B(\lambda)_{i0} - (-T_B(\lambda)_{iK})(T_B(\lambda)_{KK} + \tau^{-1}\mathscr{I})^{-1} T_B(\lambda)_{K0}\big) = 0.$$

$$\text{For } i = 1, \qquad \det\left(\begin{bmatrix} \tau^{-1} & 0 \\ 0 & \tau^{-1} \end{bmatrix}\right)\left(4/6 - \begin{bmatrix} -2 & 0 \end{bmatrix}\begin{bmatrix} \tau^{-1} & 0 \\ 0 & \tau^{-1} \end{bmatrix}^{-1}\begin{bmatrix} 5 \\ -2 \end{bmatrix}\right) = 0$$

$$\left(\frac{1}{\tau^2}\right)(4/6 + 10\tau) = 0$$

$$\alpha_1 = \tau = \infty.$$

$$\text{For } i = 2, \qquad \det\left(\begin{bmatrix} \tau^{-1} & 0 \\ 0 & \tau^{-1} \end{bmatrix}\right)\left(0 - \begin{bmatrix} 0 & -3 \end{bmatrix}\begin{bmatrix} \tau^{-1} & 0 \\ 0 & \tau^{-1} \end{bmatrix}^{-1}\begin{bmatrix} 5 \\ -2 \end{bmatrix}\right) = 0$$

$$\left(\frac{1}{\tau^2}\right)(-6\tau) = 0$$

$$\alpha_2 = \tau = \infty.$$

For $i = 3$, $$\det\left(\begin{bmatrix} \tau^{-1} & 0 \\ 0 & \tau^{-1} \end{bmatrix}\right)\left(15 - \begin{bmatrix} 0 & 0 \end{bmatrix}\begin{bmatrix} \tau^{-1} & 0 \\ 0 & \tau^{-1} \end{bmatrix}^{-1}\begin{bmatrix} 5 \\ -2 \end{bmatrix}\right) = 0$$

$$\left(\frac{15}{\tau^2}\right) = 0$$

$$\alpha_3 = \tau = \infty$$

Then $\rho_1 = \infty$. Equation $P_2(\tau) = 0$ is solved.

$$P_2(\tau) = \det\big(- T_B(\lambda)_{KK} + \tau^{-1}\mathscr{I}\big) = 0$$

$$\det\left(\begin{bmatrix} \tau^{-1} & 0 \\ 0 & \tau^{-1} \end{bmatrix}\right) = 0$$

$$\left(\frac{1}{\tau^2}\right) = 0$$

Set $\rho_2 = \tau = \infty$. Find $\rho = \min\{1 - \lambda, \rho_1, \rho_2\} = \min\{5/6, \infty, \infty\} = 1 - \lambda = 5/6$, and the invariancy interval for λ for the current BFCS (4.40) is $[1/6, 1]$. The starting tableau for the next iteration is constructed.

<u>Iteration 4:</u> Set $\lambda = 1$ in the initial tableau and obtain $T_{B_0}(1)$:

Tableau $T_{B_0}(1)$

	$\mathfrak{g}_{.0}$	$\mathfrak{g}_{.6}$	$\mathfrak{g}_{.7}$	$\mathfrak{g}_{.8}$	$\mathfrak{g}_{.9}$	$\mathfrak{g}_{.10}$
w_1	9	-6	0	-3	2	0
w_2	-5	0	$\boxed{-14}$	-5	0	3
w_3	15	3	5	0	0	0
v_{κ_1}	5	-2	0	0	0	0
v_{κ_2}	-2	0	-3	0	0	0

The resulting $(T_{B_0}(1))_{2,0} = -5 < 0$ implying $\mathcal{B}_{i0} \prec \mathbf{0}$. Therefore the current basis $B_0 = \{w_1, w_2, w_3\}$ is updated using Algorithm 7. Let $r = \min\{i : (T_{B_0}(1))_{i,0} < 0, i = 1, 2, 3\} = 2$. Then $(T_{B_0}(1))_{r,\bar{r}} = (T_{B_0}(0))_{2,7} = -14 < 0$, and a diagonal pivot is performed.

Tableau $T_{B_4}(1)$

	$\mathfrak{g}_{.0}$	$\mathfrak{g}_{.6}$	$\mathfrak{g}_{.2}$	$\mathfrak{g}_{.8}$	$\mathfrak{g}_{.9}$	$\mathfrak{g}_{.10}$
w_1	9	-6	0	-3	2	0
z_2	5/14	0	-1/14	5/14	0	-3/14
w_3	170/14	3	5	-25/14	0	15/14
v_{κ_1}	5	-2	0	0	0	0
v_{κ_2}	-11/14	0	-3	15/14	0	-9/14

The resulting $(T_{B_2}(1))_{i0} \geq 0$ for $i = 1, 2, 3$, i.e., the obtained complementary basis $B_4 = \{w_1, z_2, w_3\}$ is feasible for $\lambda = 1$. A new BFCS is obtained for spLCP (4.26) for $\lambda = 1$

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 9 \\ 0 \\ 170/14 \end{bmatrix} \qquad \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 5/14 \\ 0 \end{bmatrix}, \tag{4.41}$$

which yields an efficient solution to BOQP (4.25)

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 5/14 \end{bmatrix}.$$

Since the entire parameter space has been examined, the spLCP method terminates.

B Efficient Solutions to Examples

Table 4.6: Invariancy regions ($I\mathcal{R}$) and efficient solution functions for TOQP (4.30) with data (4.31) solved as (4.32) with the mpLCP method.

$$I\mathcal{R}_1 = \left\{ \boldsymbol{\lambda} \in \Lambda' : \begin{array}{l} 100 - 100\lambda_2 - 409\lambda_1 \geq 0 \\ 100 - 100\lambda_2 - 1007\lambda_1 \geq 0 \\ 100 - 100\lambda_2 - 871\lambda_1 \geq 0 \\ 209\lambda_1 - 100\lambda_2 + 100 \geq 0 \\ 807\lambda_1 - 100\lambda_2 + 100 \geq 0 \\ 671\lambda_1 - 100\lambda_2 + 100 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\boldsymbol{\lambda}) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 0 \\ x_2 = 0 \\ x_3 = 0 \end{array} \right\} \text{ for } \boldsymbol{\lambda} \in I\mathcal{R}_1$$

$$I\mathcal{R}_2 = \left\{ \boldsymbol{\lambda} \in \Lambda' : \begin{array}{l} -15,623\lambda_1^2 - 13,936\lambda_1\lambda_2 + 12,300\lambda_1 - 400\lambda_2^2 + 400\lambda_2 \geq 0 \\ 1,007\lambda_1 + 100\lambda_2 - 100 \geq 0 \\ 900\lambda_1 + 200\lambda_2 - 2,642\lambda_1\lambda_2 + 3,245\lambda_1^2 - 200\lambda_2^2 \geq 0 \\ 209\lambda_1 - 100\lambda_2 - 29,203\lambda_1^2 + 2,900\lambda_1\lambda_2 - 2,900\lambda_1 \geq 0 \\ 1 - \lambda_2 - \lambda_1 \geq 0 \\ -15,858\lambda_1\lambda_2 - 200\lambda_2 - 17,200\lambda_1 + 21,345\lambda_1^2 + 200\lambda_2^2 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\boldsymbol{\lambda}) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 0 \\ x_2 = \frac{1,007\lambda_1 + 100\lambda_2 - 100}{4,525\lambda_1 + 100\lambda_2} \\ x_3 = 0 \end{array} \right\} \text{ for } \boldsymbol{\lambda} \in I\mathcal{R}_2$$

$$I\mathcal{R}_3 = \left\{ \boldsymbol{\lambda} \in \Lambda' : \begin{array}{l} 13,936\lambda_1\lambda_2 - 400\lambda_2 - 12,300\lambda_1 + 15,623\lambda_1^2 + 400\lambda_2^2 \geq 0 \\ 1,900\lambda_1 - 200\lambda_2 + 114\lambda_1\lambda_2 - 1,791\lambda_1^2 + 200\lambda_2^2 \geq 0 \\ -1,200\lambda_1\lambda_2 - 2,284\lambda_1\lambda_2^2 + 2,020\lambda_1^2\lambda_2 + 7,500\lambda_1^2 - 6,423\lambda_1^3 + 400\lambda_2^2 - 400\lambda_2^3 \geq 0 \\ 1 - \lambda_2 - \lambda_1 \geq 0 \\ -38,716\lambda_1\lambda_2^2 + 41,400\lambda_1\lambda_2 - 55,020\lambda_1^2\lambda_2 + 5,300\lambda_1^2 - 6,377\lambda_1^3 + 400\lambda_2^2 - 400\lambda_2^3 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\boldsymbol{\lambda}) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{13,936\lambda_1\lambda_2 - 400\lambda_2 - 12,300\lambda_1 + 15,623\lambda_1^2 + 400\lambda_2^2}{20,100\lambda_1\lambda_2 + 6,400\lambda_1^2 + 400\lambda_2^2} \\ x_2 = \frac{1,900\lambda_1 - 200\lambda_2 + 114\lambda_1\lambda_2 - 1,791\lambda_1^2 + 200\lambda_2^2}{10,050\lambda_1\lambda_2 + 3,200\lambda_1^2 + 200\lambda_2^2} \\ x_3 = 0 \end{array} \right\} \text{ for } \boldsymbol{\lambda} \in I\mathcal{R}_3$$

$$I\mathcal{R}_4 = \left\{ \boldsymbol{\lambda} \in \Lambda' : \begin{array}{l} -25,036\lambda_1\lambda_2^2 + 23,400\lambda_1\lambda_2 - 82,622\lambda_1^2\lambda_2 + 44,700\lambda_1^2 - 51,879\lambda_1^3 + 400\lambda_2^2 - 400\lambda_2^3 \geq 0 \\ 100\lambda_1 - 200\lambda_2 + 1,914\lambda_1\lambda_2 + 10,077\lambda_1^2 + 200\lambda_2^2 \geq 0 \\ -900\lambda_1 - 200\lambda_2 + 2,642\lambda_1\lambda_2 - 3,245\lambda_1^2 + 200\lambda_2^2 \geq 0 \\ -44,164\lambda_1\lambda_2^2 + 45,000\lambda_1\lambda_2 - 114,378\lambda_1^2\lambda_2 + 83,900\lambda_1^2 - 76,721\lambda_1^3 + 400\lambda_2^2 - 400\lambda_2^3 \geq 0 \\ 1 - \lambda_2 - \lambda_1 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\boldsymbol{\lambda}) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 0 \\ x_2 = \frac{100\lambda_1 - 200\lambda_2 + 1,914\lambda_1\lambda_2 + 10,077\lambda_1^2 + 200\lambda_2^2}{17,100\lambda_1\lambda_2 + 32,150\lambda_1^2 + 500\lambda_2^2} \\ x_3 = \frac{-900\lambda_1 - 200\lambda_2 + 2642\lambda_1\lambda_2 - 3,245\lambda_1^2 + 200\lambda_2^2}{17,100\lambda_1\lambda_2 + 32,150\lambda_1^2 + 500\lambda_2^2} \end{array} \right\} \text{ for } \boldsymbol{\lambda} \in I\mathcal{R}_4$$

$$I\mathcal{R}_5 = \left\{ \boldsymbol{\lambda} \in \Lambda' : \begin{array}{l} 25,036\lambda_1\lambda_2^2 - 23,400\lambda_1\lambda_2 + 82,622\lambda_1^2\lambda_2 - 44,700\lambda_1^2 + 51,879\lambda_1^3 - 400\lambda_2^2 + 400\lambda_2^3 \geq 0 \\ 4,000\lambda_1\lambda_2 + 28\lambda_1\lambda_2^2 - 4,128\lambda_1^2\lambda_2 + 20,700\lambda_1^2 - 18,603\lambda_1^3 - 400\lambda_2^2 + 400\lambda_2^3 \geq 0 \\ 1,200\lambda_1\lambda_2 + 2,284\lambda_1\lambda_2^2 - 2,020\lambda_1^2\lambda_2 - 7,500\lambda_1^2 + 6,423\lambda_1^3 - 400\lambda_2^2 + 400\lambda_2^3 \geq 0 \\ 1 - \lambda_2 - \lambda_1 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\boldsymbol{\lambda}) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{25,036\lambda_1\lambda_2^2 - 23,400\lambda_1\lambda_2 + 82,622\lambda_1^2\lambda_2 - 44,700\lambda_1^2 + 51,879\lambda_1^3 - 400\lambda_2^2 + 400\lambda_2^3}{36,200\lambda_1\lambda_2^2 + 88,700\lambda_1^2\lambda_2 + 17,100\lambda_1^3 + 400\lambda_2^3} \\ x_2 = \frac{4,000\lambda_1\lambda_2 + 28\lambda_1\lambda_2^2 - 4,128\lambda_1^2\lambda_2 + 20,700\lambda_1^2 - 18,603\lambda_1^3 - 400\lambda_2^2 + 400\lambda_2^3}{36,200\lambda_1\lambda_2^2 + 88,700\lambda_1^2\lambda_2 + 17,100\lambda_1^3 + 400\lambda_2^3} \\ x_3 = \frac{1,200\lambda_1\lambda_2 + 2,284\lambda_1\lambda_2^2 - 2,020\lambda_1^2\lambda_2 - 7,500\lambda_1^2 + 6,423\lambda_1^3 - 400\lambda_2^2 + 400\lambda_2^3}{36,200\lambda_1\lambda_2^2 + 88,700\lambda_1^2\lambda_2 + 17,100\lambda_1^3 + 400\lambda_2^3} \end{array} \right\} \text{ for } \boldsymbol{\lambda} \in I\mathcal{R}_5$$

$$I\mathcal{R}_6 = \left\{ \boldsymbol{\lambda} \in \Lambda' : \begin{array}{l} 35,036\lambda_1\lambda_2^2 - 33,400\lambda_1\lambda_2 + 72,422\lambda_1^2\lambda_2 - 24,500\lambda_1^2 + 31,679\lambda_1^3 - 400\lambda_2^2 + 400\lambda_2^3 \geq 0 \\ 32,628\lambda_1\lambda_2^2 - 28,600\lambda_1\lambda_2 + 46,472\lambda_1^2\lambda_2 + 2,700\lambda_1^2 - 603\lambda_1^3 - 400\lambda_2^2 + 400\lambda_2^3 \geq 0 \\ 38,716\lambda_1\lambda_2^2 - 41,400\lambda_1\lambda_2 + 55,020\lambda_1^2\lambda_2 - 5,300\lambda_1^2 + 6,377\lambda_1^3 - 400\lambda_2^2 + 400\lambda_2^3 \geq 0 \\ 1 - \lambda_2 - \lambda_1 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\boldsymbol{\lambda}) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{35,036\lambda_1\lambda_2^2 - 33,400\lambda_1\lambda_2 + 72,422\lambda_1^2\lambda_2 - 24,500\lambda_1^2 + 31,679\lambda_1^3 - 400\lambda_2^2 + 400\lambda_2^3}{36,200\lambda_1\lambda_2^2 + 88,700\lambda_1^2\lambda_2 + 17,100\lambda_1^3 + 400\lambda_2^3} \\ x_2 = \frac{32,628\lambda_1\lambda_2^2 - 28,600\lambda_1\lambda_2 + 46,472\lambda_1^2\lambda_2 + 2,700\lambda_1^2 - 603\lambda_1^3 - 400\lambda_2^2 + 400\lambda_2^3}{36,200\lambda_1\lambda_2^2 + 88,700\lambda_1^2\lambda_2 + 17,100\lambda_1^3 + 400\lambda_2^3} \\ x_3 = \frac{-38,716\lambda_1\lambda_2^2 + 41,400\lambda_1\lambda_2 - 55,020\lambda_1^2\lambda_2 + 5,300\lambda_1^2 - 6,377\lambda_1^3 + 400\lambda_2^2 - 400\lambda_2^3}{36,200\lambda_1\lambda_2^2 + 88,700\lambda_1^2\lambda_2 + 17,100\lambda_1^3 + 400\lambda_2^3} \end{array} \right\} \text{ for } \boldsymbol{\lambda} \in I\mathcal{R}_6$$

Table 4.7: Invariancy regions ($I\mathcal{R}$) and efficient solution functions for TOQP (4.30) with data (4.31).

$$I\mathcal{R}_1 = \left\{ \alpha, \beta \geq 0 : \begin{array}{c} 100\beta - 309 \geq 0 \\ 100\beta - 907 \geq 0 \\ 100\beta - 771 \geq 0 \\ 100\beta + 309 \geq 0 \\ 100\beta + 907 \geq 0 \\ 100\beta + 771 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\alpha, \beta) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{c} x_1 = 0 \\ x_2 = 0 \\ x_3 = 0 \end{array} \right\} \text{ for } (\alpha, \beta) \in I\mathcal{R}_1$$

$$I\mathcal{R}_2 = \left\{ \alpha, \beta \geq 0 : \begin{array}{l} -1,236\alpha + 12,300\beta + 400\alpha\beta - 3,323 \geq 0 \\ -100\beta + 907 \geq 0 \\ 900\beta - 1,542\alpha + 200\alpha\beta + 4,145 \geq 0 \\ 1,236\alpha + 23,900\beta + 400\alpha\beta + 3,323 \geq 0 \\ 1,542\alpha + 17,200\beta + 200\alpha\beta - 4,145 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\alpha, \beta) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 0 \\ x_2 = \frac{-100\beta + 907}{100\alpha + 4,525} \\ x_3 = 0 \end{array} \right\} \text{ for } (\alpha, \beta) \in I\mathcal{R}_2$$

$$I\mathcal{R}_3 = \left\{ \alpha, \beta \geq 0 : \begin{array}{l} 1,236\alpha - 12,300\beta - 400\alpha\beta + 3,323 \geq 0 \\ 1,814\alpha + 1,900\beta - 200\alpha\beta + 109 \geq 0 \\ 8,320\alpha + 7,500\beta - 1,200\alpha\beta + 400\alpha^2\beta - 3,084\alpha^2 + 1,077 \geq 0 \\ 5,300\beta - 8,320\alpha + 41,400\alpha\beta + 400\alpha^2\beta + 3,084\alpha^2 - 1,077 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\alpha, \beta) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{1,236\alpha - 12,300\beta - 400\alpha\beta + 3,323}{400\alpha^2 + 20,100\alpha + 6,400} \\ x_2 = \frac{1,814\alpha + 1,900\beta - 200\alpha\beta + 109}{200\alpha^2 + 10,050\alpha + 3,200} \\ x_3 = 0 \end{array} \right\} \text{ for } (\alpha, \beta) \in I\mathcal{R}_3$$

$$
I\mathcal{R}_4 = \left\{ \alpha,\beta \geq 0 : \begin{array}{l} -14,522\alpha + 44,700\beta + 23,400\alpha\beta + 400\alpha^2\beta - 1,236\alpha^2 - 7,179 \geq 0 \\ 1,814\alpha + 100\beta - 200\alpha\beta + 10,177 \geq 0 \\ -900\beta + 1,542\alpha - 200\alpha\beta - 4,145 \geq 0 \\ 14,522\alpha + 83,900\beta + 45,000\alpha\beta + 400\alpha^2\beta + 1,236\alpha^2 + 7,179 \geq 0 \end{array} \right\}
$$

$$
\hat{\mathbf{x}}(\alpha,\beta) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 0 \\ x_2 = \frac{1,814\alpha+100\beta-200\alpha\beta+10,177}{200\alpha^2+17,100\alpha+32,150} \\ x_3 = \frac{-900\beta+1,542\alpha-200\alpha\beta-4,145}{200\alpha^2+17,100\alpha+32,150} \end{array} \right\} \text{ for } (\alpha,\beta) \in I\mathcal{R}_4
$$

$$
I\mathcal{R}_5 = \left\{ \alpha,\beta \geq 0 : \begin{array}{l} 14,522\alpha - 44,700\beta - 23,400\alpha\beta - 400\alpha^2\beta + 1,236\alpha^2 + 7,179 \geq 0 \\ 20,572\alpha + 20,700\beta + 4,000\alpha\beta - 400\alpha^2\beta + 3,628\alpha^2 + 2,097 \geq 0 \\ -8,320\alpha - 7,500\beta + 1,200\alpha\beta - 400\alpha^2\beta + 3,084\alpha^2 - 1,077 \geq 0 \end{array} \right\}
$$

$$
\hat{\mathbf{x}}(\alpha,\beta) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{14,522\alpha-44,700\beta-23,400\alpha\beta-400\alpha^2\beta+1,236\alpha^2+7,179}{400\alpha^3+36,200\alpha^2+88,700\alpha+17,100} \\ x_2 = \frac{20,572\alpha+20,700\beta+4,000\alpha\beta-400\alpha^2\beta+3,628\alpha^2+2,097}{400\alpha^3+36,200\alpha^2+88,700\alpha+17,100} \\ x_3 = \frac{-8,320\alpha-7,500\beta+1,200\alpha\beta-400\alpha^2\beta+3,084\alpha^2-1,077}{400\alpha^3+36,200\alpha^2+88,700\alpha+17,100} \end{array} \right\} \text{ for } (\alpha,\beta) \in I\mathcal{R}_5
$$

$$
I\mathcal{R}_6 = \left\{ \alpha,\beta \geq 0 : \begin{array}{l} 14,522\alpha - 24,500\beta - 33,400\alpha\beta - 400\alpha^2\beta + 1,236\alpha^2 + 7,179 \geq 0 \\ 20,572\alpha + 2,700\beta - 28,600\alpha\beta - 400\alpha^2\beta + 3,628\alpha^2 + 2,097 \geq 0 \\ -5,300\beta + 8,320\alpha - 41,400\alpha\beta - 400\alpha^2\beta - 3,084\alpha^2 + 1,077 \geq 0 \end{array} \right\}
$$

$$
\hat{\mathbf{x}}(\alpha,\beta) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{14,522\alpha-24,500\beta-33400\alpha\beta-400\alpha^2\beta+1236\alpha^2+7179}{400\alpha^3+36,200\alpha^2+88,700\alpha+17,100} \\ x_2 = \frac{20,572\alpha+2,700\beta-28,600\alpha\beta-400\alpha^2\beta+3,628\alpha^2+2,097}{400\alpha^3+36,200\alpha^2+88,700\alpha+17,100} \\ x_3 = \frac{5,300\beta-8,320\alpha+41,400\alpha\beta+400\alpha^2\beta+3,084\alpha^2-1,077}{400\alpha^3+36,200\alpha^2+88,700\alpha+17,100} \end{array} \right\} \text{ for } (\alpha,\beta) \in I\mathcal{R}_6
$$

Table 4.8: Invariancy region ($\mathcal{IR}$) and efficient solution functions for Example (4.34) scalarized with the modified hybrid method ($\theta = \theta^{nor}, \epsilon = \epsilon^{nor}$, and the parameter space $\Omega = \Theta^{nor} \times \Lambda' \times \mathcal{E}^{nor}$)

$$\mathcal{IR}_1 = \left\{ (\theta,\lambda,\epsilon) \in \Omega : \begin{array}{l} -20\lambda - 737\epsilon - 24\theta + 134\lambda\epsilon + 8\lambda\theta + 60 \geq 0 \\ -740\lambda - 44,019\epsilon - 808\theta + 4,958\lambda\epsilon + 56\lambda\theta - 2,144\epsilon\theta + 96\lambda\theta^2 - 160\theta^2 \\ \qquad +1,608\lambda\epsilon\theta + 3,020 \geq 0 \\ 67\epsilon + 4\theta - 10 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\theta,\lambda,\epsilon) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 0 \\ x_2 = \frac{67\epsilon+4\theta-10}{4\theta-10} \\ x_3 = \frac{-67\epsilon}{4\theta-10} \end{array} \right\} \text{ for } (\theta,\lambda,\epsilon) \in \mathcal{IR}_1$$

$$\mathcal{IR}_2 = \left\{ (\theta,\lambda,\epsilon) \in \Omega : \begin{array}{l} 1,139\lambda + 4,489\epsilon + 56\theta - 2,412\lambda\epsilon - 402\lambda^2\epsilon - 24\lambda^2\theta \\ \qquad +60\lambda^2 - 2,217 \geq 0 \\ -31,021\epsilon + 21,562\lambda - 1552\theta - 96\lambda^2\theta^2 + 20,167\lambda\epsilon + 4,512\lambda\theta \\ \qquad -3,752\epsilon\theta + 12,060\lambda^2\epsilon + 384\lambda\theta^2 - 1,776\lambda^2\theta - 3,938\lambda^2 \\ \qquad -352\theta^2 + 1,608\lambda^2\epsilon\theta - 27,554 \geq 0 \\ -740\lambda - 44,019\epsilon - 808\theta + 4,958\lambda\epsilon + 56\lambda\theta - 2,144\epsilon\theta + 96\lambda\theta^2 \\ \qquad -160\theta^2 + 1,608\lambda\epsilon\theta + 3,020 \geq 0 \\ 20,167\epsilon - 2,080\lambda + 640\theta + 13,936\lambda\epsilon + 592\lambda\theta - 2680\epsilon\theta + 96\lambda\theta^2 \\ \qquad -128\theta^2 + 1,608\lambda\epsilon\theta - 23,245 \geq 0 \\ 37\lambda + 178\epsilon - 20\theta - 141\lambda\epsilon + 12\lambda\theta + 36\epsilon\theta - 24\lambda\epsilon\theta - 151 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\theta,\lambda,\epsilon) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{-740\lambda-44{,}019\epsilon-808\theta+4{,}958\lambda\epsilon+56\lambda\theta-2{,}144\epsilon\theta+96\lambda\theta^2-160\theta^2+1{,}608\lambda\epsilon\theta+3{,}020}{2{,}138\lambda-2{,}848\theta+2{,}256\lambda\theta+192\lambda\theta^2-288\theta^2-40{,}459} \\ x_2 = \frac{20{,}167\epsilon-2{,}080\lambda+640\theta+13{,}936\lambda\epsilon+592\lambda\theta-2{,}680\epsilon\theta+96\lambda\theta^2-128\theta^2+1{,}608\lambda\epsilon\theta-23{,}245}{2{,}138\lambda-2{,}848\theta+2{,}256\lambda\theta+192\lambda\theta^2-288\theta^2-40{,}459} \\ x_3 = \frac{4{,}958\lambda+23{,}852\epsilon-2{,}680\theta-18{,}894\lambda\epsilon+1{,}608\lambda\theta+4{,}824\epsilon\theta-3{,}216\lambda\epsilon\theta-20{,}234}{2{,}138\lambda-2{,}848\theta+2{,}256\lambda\theta+192\lambda\theta^2-288\theta^2-40{,}459} \end{array} \right\} \text{ for } (\theta,\lambda,\epsilon) \in \mathcal{IR}_2$$

$$\mathcal{IR}_3 = \left\{ (\theta,\lambda,\epsilon) \in \Omega : \begin{array}{l} -1,139\lambda - 4,489\epsilon - 56\theta + 2,412\lambda\epsilon + 402\lambda^2\epsilon + 24\lambda^2\theta - 60\lambda^2 + 2,217 \geq 0 \\ -15\lambda^2 + 86\lambda - 6\lambda^3 - 71 \geq 0 \\ 17\lambda - 31 \geq 0 \\ 19\lambda - 22 \geq 0 \\ 3\lambda^2 - 7 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\theta,\lambda,\epsilon) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{17\lambda-31}{36\lambda+6\lambda^2-67} \\ x_2 = \frac{19\lambda-22}{36\lambda+6\lambda^2-67} \\ x_3 = \frac{6\lambda^2-14}{36\lambda+6\lambda^2-67} \end{array} \right\} \text{ for } (\theta,\lambda,\epsilon) \in \mathcal{IR}_3$$

Chapter 5

Conclusions and Future Research

This chapter first provides a summary of our contributions to the field of Multiobjective Optimization (Operations Research) and then discusses directions for future research.

5.1 Summary of Contributions

In this dissertation, we have presented our contributions in different areas of multiobjective programs (MOPs) by introducing methodology which can be used to solve specific classes of multiobjective and parametric quadratic programs. Particularly, the branch and bound (BB) algorithm to solve the biobjective mixed integer quadratic programs is the first-ever method to solve this class of problems. Also, we have studied and compared the current state-of-the-art algorithms for solving multiobjective quadratic programs (MOQPs). We have implemented some of the proposed methods and conducted computational experiments. In addition to the generated synthetic instances, the discussed algorithms have been applied to the decision-making problems in statistics and portfolio optimization.

5.1.1 Theoretical/Methodological Contributions

In Chapter 2, we have proposed a BB algorithm for solving biobjective mixed integer quadratic programs (BOMIQPs). Two new modules have been introduced based on the structure of the Pareto set of BOMIQPs: (1) a fathoming rule module to eliminate node problems that do not contain Pareto points to the BOMIQP, (2) a set dominance module to examine the mutual location of two Pareto sets and determine the nondominated set. In addition, we have proposed a branching scheme that integrates with the mpLCP method, the node problem solver. Both of these modules, the branching scheme, and the node problem solver are combined into a BB algorithm for computing efficient solutions and Pareto points to BOMIQP with linear constraints. According to our knowledge, an algorithm for solving BOMIQPs does not exist in the literature. Note that solving MOQPs is crucial to our work for developing a BB algorithm.

The proposed set dominance module can be used to make the dominance decision between two strictly convex Pareto sets. The end points of the Pareto sets are the only inputs for this module. These Pareto sets are coming from biobjective programs with the same objective functions and different feasible sets. Four subprocedures are designed to make the dominance decision by comparing the mutual locations of the end points and the associated ideal points of the Pareto sets.

In Chapter 3, we have reviewed some well-established scalarizations in multiobjective programming from the perspective of parametric optimization and have proposed a modified hybrid scalarization suitable for a class of specially structured (strictly) convex MOPs. In addition, we have reviewed and compared the algorithms for solving (strictly) convex MOQPs. As a result, we have concluded that the mpLCP method [3, 2] emerges as the most universal approach to solve MOQPs. This method has been used as the primary method to solve MOQPs and parametric quadratic programs (mpQPs) in our work.

In Chapter 4, we have discussed how to use the mpLCP method [3, 2] and the spLCP method [124] to solve the parametric programs with quadratic objective functions and linear constraints. These methods provide a complete parametric description of the solution to the mpQPs. The mpQPs were obtained by scalarizing parametric multiobjective quadratic programs. The weighted-sum or modified hybrid scalarizations were used in this solution approach.

5.1.2 Implementation/Computational Contributions

In Chapter 2 we have introduced the first-ever BB algorithm designed for solving BOMIQPs with linear constraints. A variation of this algorithm is implemented in *MATLAB* programming language and tested with randomly generated BOMIQP instances. Due to the limitations of the current implementation of the mpLCP method and limitations of *MATLAB* environment, the five modules have been executed independently in this rudimentary experiment. Also, the mpLCP method is sensitive to adding branching constraints. Therefore, small-sized instances with at most two binary/ integer variables out of two or three total variables, and at most two original linear constraints have been solved. Note that all the integer variables are bounded. This preliminary implementation of the BB algorithm has provided the complete set of efficient solutions in the associated invariancy interval as functions of the weight parameter λ.

In Chapter 4, we have discussed how to use the spLCP method to solve BOQPs with linear constraints. This method is originally designed to solve a class of single parametric linear complementarity problems. According to our knowledge, this method has never been implemented. We have implemented the spLCP method in *MATLAB* programming language as a two-phase method to solve the BOQPs. Phase I is designed to find an initial basic feasible solution and once such a solution is available, Phase II is initialized. Phase

II is designed to update the parameter λ, the weight of the weighted sum scalarization of a BOQP, and finds the associated solution for the current value of λ. We have expected that the spLCP method would be more efficient than the mpLCP method for solving BOQPs. A computational experiment has been performed to compare the efficiency of the spLCP and mpLCP methods with randomly generated strictly convex BOQPs. However, due to the limitation of some inbuilt functions such as polynomial equation solvers in the current *MATLAB* environment, we were not able to gain the expected advantage. That is, the mpLCP method remains the state-of-the-art method for solving BOQPs. Another disadvantage of the spLCP method is that only the BOQPs that satisfy the special assumptions can be solved.

5.2 Future Research

We have proposed a BB algorithm for solving BOMIQPs. This class of problems appears in many real-world applications and a method to solve such a problem is in demand. The proposed algorithm is combined with five modules and finds the complete set of efficient solutions and the Pareto set. This work immediately opens up several avenues for continued research. The proposed branching module depends on the solution functions obtained by the mpLCP method. Therefore, a method to solve single variable fractional programs will enhance the performance of the branching module.

Two practical bound-based fathoming rules have been proposed and future research will be devoted to improving the current rules and designing new strong fathoming rules. Both the fathoming rule and the set dominance modules are performed in the objective space. Therefore, there is a potential of combining these two modules for small-sized problems. However, a broad numerical experiment will be needed for this modification.

In the proposed set dominance module, a system of polynomial equations is solved to find the intersection points between two Pareto sets. A more robust commercially available solver can be used to improve this module. In general, the set dominance module can be generalized to make the dominance decision between two (strictly) convex Pareto sets.

The implementation of the mpLCP method is sensitive to adding more constraints in the branching step. Therefore, a more stable node problem solver with lower complexity is suggested to use in the future. However, it seems that in order to develop a truly efficient solver for BOQPs, one must either propose a new solver without using commercial packages or wait until the tools of commercial software are incorporated with the current solver. Unfortunately, neither of these options is very promising in the near future.

The CPLEX optimization software can be used instead of GUROBI to obtain the initial efficient set and the associated initial Pareto set to BOMIQP. Free versions of both of these solvers are available for students and researchers.

In addition, computational tests will be performed on BOMIQP instances that are adapted from the literature. We have not solved any convex biobjective (pure) binary/ integer quadratic programs as it is not our primary goal. However, our algorithm contains all tools to solve problems in that class and can be modified accordingly.

In general, our algorithm relies on solving four types of optimization problems and systems of polynomial equations. With our study on BOMIQPs, we understood that, although there is a good potential of developing an efficient branch and bound algorithms for BOMIQP, these algorithms are quite difficult and complex to implement efficiently using the current commercial software packages.

Appendices

Appendix A Fathoming Rules: The Case with One Nadir Point

In this section we discuss fathoming rules by considering only two attainable points $\mathbf{y}^i, \mathbf{y}^j \in \bar{\mathcal{Y}}_a^s$ and the associated nadir point $\mathbf{y}^{i,j} = \left(y_1^{i,j} = y_1^j, y_2^{i,j} = y_2^i\right) \in \mathcal{Y}^{s\mathcal{N}}$. We analyze the location of these points with respect to the set $\{\tilde{\mathbf{y}}^{sI}\}^{\geqq}$, since we assume that the points $\tilde{\mathbf{y}}^{s1}$, $\tilde{\mathbf{y}}^{s2}$, and $\tilde{\mathbf{y}}^{sI}$ have already been obtained. Consider the following three locations of $\mathbf{y}^i, \mathbf{y}^j$ and $\mathbf{y}^{i,j}$ with respect to the set $\{\tilde{\mathbf{y}}^{sI}\}^{\geqq}$. We discuss how to make a fathoming decision based on these locations for a single nadir point case.

Location 1: $\mathbf{y}^i \in \{\tilde{\mathbf{y}}^{sI}\}^{\leqq}$ or $\mathbf{y}^j \in \{\tilde{\mathbf{y}}^{sI}\}^{\leqq}$.

This implies the $\tilde{\mathcal{Y}}_P^s \subset (\{\mathbf{y}^i\}^{\geqq} \cup \{\mathbf{y}^j\}^{\geqq})$ and hence node s can be fathomed. Figure 1 depicts this case.

Location 2: $\mathbf{y}^{i,j} \in T^s$.

If the nadir point is in the set T^s, a fathoming decision is not immediate. We need to further investigate this case by considering the locations of $\mathbf{y}^i$ and $\mathbf{y}^j$ with respect to T^s and the location of $\mathbf{y}^{i,j}$ with respect to $\tilde{\mathcal{Y}}_P^s$. We consider the following three cases and make a fathoming decision accordingly.

The first case addresses the instance when all points $\mathbf{y}^i, \mathbf{y}^j$ and $\mathbf{y}^{i,j}$ are in T^s.

Location 2 Case 1: $\mathbf{y}^i$ or $\mathbf{y}^j \in T^s$.

If $\mathbf{y}^i$ or $\mathbf{y}^j \in T^s$, then there exists a point $\tilde{\mathbf{y}}^s \in \tilde{\mathcal{Y}}_P^s$ such that $\tilde{\mathbf{y}}^s \notin (\{\mathbf{y}^i\}^{\geqq} \cup \{\mathbf{y}^j\}^{\geqq})$. Hence, node s can not be fathomed as depicted in Figures 2a and 2b.

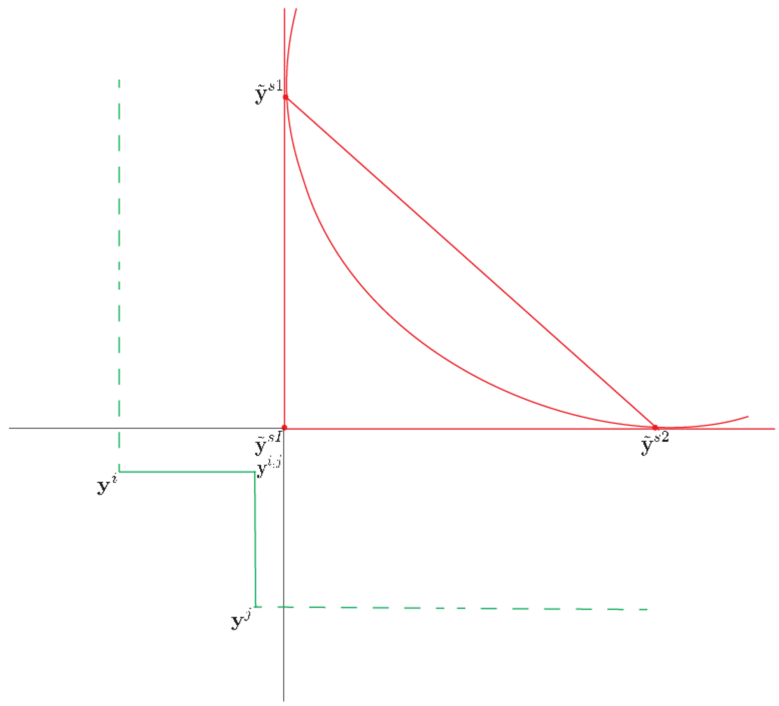


Figure 1: Location 1 - node s can be fathomed

The second case depends only on the location of the nadir point $\mathbf{y}^{i,j}$ with respect to the set $\tilde{\mathcal{Y}}_P^s$ and does not depend on the locations of $\mathbf{y}^i$ and $\mathbf{y}^j$.

Location 2 Case 2: $\mathbf{y}^{i,j} \in \tilde{\mathcal{Y}}_P^s + \mathbb{R}_>$.

If $\mathbf{y}^{i,j} \in \tilde{\mathcal{Y}}_P^s + \mathbb{R}_>$, then there exists a point $\tilde{\mathbf{y}}^s \in \tilde{\mathcal{Y}}_P^s$ such that $\tilde{\mathbf{y}}^s \notin (\{\mathbf{y}^i\}^{\geqq} \cup \{\mathbf{y}^j\}^{\geqq})$. Hence, regardless of the locations of the points $\mathbf{y}^i$ and $\mathbf{y}^j$, node s can not be fathomed as depicted in Figures 3a and 3b.

In the third case we assume $\mathbf{y}^i$ and $\mathbf{y}^j \notin T^s$ for $\mathbf{y}^{i,j} \in T^s$. Then $\mathbf{y}^i$ must be in $\bar{\mathcal{Y}}_a^s \cap C^{sW}$ and $\mathbf{y}^j$ must be in $\bar{\mathcal{Y}}_a^s \cap C^{sS}$.

Location 2 Case 3: $\mathbf{y}^{i,j} \in \tilde{\mathcal{Y}}_P^s - \mathbb{R}^2_{\geqq}$, and $\mathbf{y}^i$ and $\mathbf{y}^j \notin T^s$.

Here, $\tilde{\mathcal{Y}}_P^s \subset (\{\mathbf{y}^i\}^{\geqq} \cup \{\mathbf{y}^j\}^{\geqq})$ and hence node s can be fathomed. Figures 4a and 4b depict this case.

The complete set $\tilde{\mathcal{Y}}_P^s$ is needed to make a fathoming decision for the second and the third cases of Location 2. However, this complete set $\tilde{\mathcal{Y}}_P^s$ is not available and hence making a fathoming decision is not straightforward as in the other cases we discussed. We solve an achievement function problem to overcome this issue. That is, we can use the achievement function problem to make a fathoming decision in cases 2 and 3 under Location 2, without computing the entire Pareto set $\tilde{\mathcal{Y}}_P^s$. While this section is concerned so far with the one nadir point case, the following achievement function problem can be used for any number of nadir points in T^s.

Making a Fathoming Decision with the Solution of the Achievement Function Problem

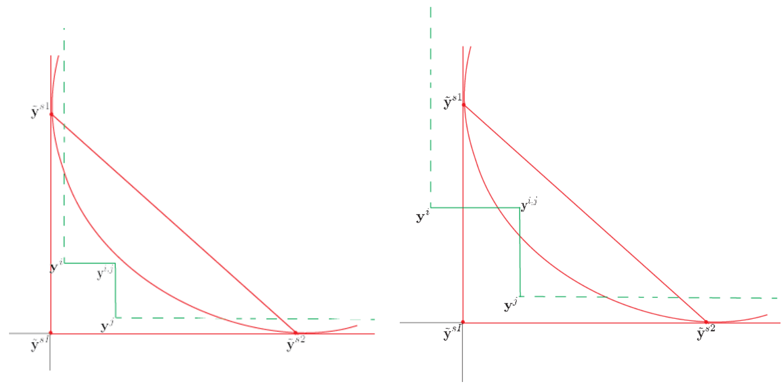
Proposition A.1. *Let $\boldsymbol{y}^i \in \bar{\mathcal{Y}}_a^s \cap C^{sW}$ and $\boldsymbol{y}^j \in \bar{\mathcal{Y}}_a^s \cap C^{sS}$, the implied nadir point $\boldsymbol{y}^{i,j}$ be in T^s, $\hat{\boldsymbol{x}}^s$ be an optimal solution to* (2.10) *and $\hat{\boldsymbol{y}}^s = (f_1(\hat{\boldsymbol{x}}^s), f_2(\hat{\boldsymbol{x}}^s))$. Then node s can be fathomed if $\hat{\boldsymbol{y}}^s \in \{\boldsymbol{y}^{i,j}\}^{\geq}$.*

Proof. Assume $\hat{\mathbf{y}^s} \in \{\mathbf{y}^{i,j}\}^{\geq}$. Since $\mathbf{y}^i \in (\bar{\mathcal{Y}}_a^s \cap C^{sW})$ and $\mathbf{y}^j \in (\bar{\mathcal{Y}}_a^s \cap C^{sS})$ we have $\tilde{\mathcal{Y}}_P^s \subset \big(\{\mathbf{y}^i\}^{\geqq} \cup \{\mathbf{y}^j\}^{\geqq} \cup \{\mathbf{y}^{i,j}\}^{\geq}\big)$. Also $\{\mathbf{y}^{i,j}\}^{\geq} \subset \big(\{\mathbf{y}^i\}^{\geqq} \cup \{\mathbf{y}^j\}^{\geqq}\big)$ and then $\tilde{\mathcal{Y}}_P^s \subset \big(\{\mathbf{y}^i\}^{\geqq} \cup \{\mathbf{y}^j\}^{\geqq}\big)$. Therefore, node s can be fathomed based on (2.21). $\square$

Note that examining $\hat{\mathbf{y}^s} \in \{\mathbf{y}^{i,j}\}^{\geq}$ is equivalent to checking the condition $\hat{y}_1^s \geq y_1^{i,j}$ and $\hat{y}_2^s \geq y_2^{i,j}$. Otherwise, $\hat{\mathbf{y}^s} \notin \{\mathbf{y}^{i,j}\}^{\geq}$, and node s can not be fathomed.

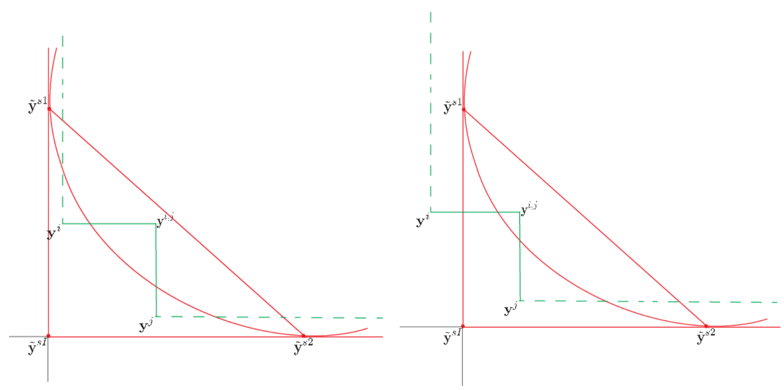
Location 3: All points $\mathbf{y}^i, \mathbf{y}^j$ and $\mathbf{y}^{i,j}$ are not in either Location 1 or Location 2.

The Location 1 and case 3 under Location 2 are the only cases that node s can be fathomed.



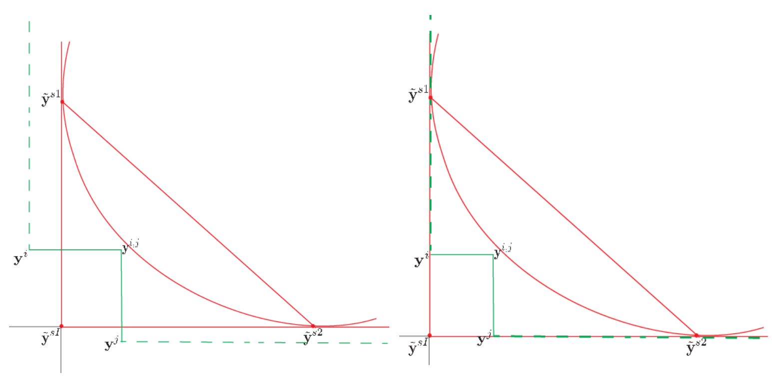
(a) (b)

Figure 2: Two instances of Location 2 Case 1 - node s can not be fathomed



(a) (b)

Figure 3: Two instances of Location 2 Case 2 - node s can not be fathomed



(a) (b)

Figure 4: Two instances of Location 2 Case 3 - node s can be fathomed

If the attainable points or the associated nadir point are not in these locations, then node s can not be fathomed.

We can summarize all of the above cases as follows:

1. If $\mathbf{y}^i \in \{\tilde{\mathbf{y}}^{sI}\}^{\leqq}$ or $\mathbf{y}^j \in \{\tilde{\mathbf{y}}^{sI}\}^{\leqq}$, then node s can be fathomed.

2. Else if $\mathbf{y}^{i,j} \in T^s$ and the associated $\mathbf{y}^i \in (\bar{\mathcal{Y}}_a^s \cap C^{sW})$ and $\mathbf{y}^j \in (\bar{\mathcal{Y}}_a^s \cap C^{sS})$, then we solve achievement function problem (2.10) and obtain a solution $\hat{\mathbf{y}}^s = (\hat{y}_1^s, \hat{y}_2^s) \in \tilde{\mathcal{Y}}_P^s$,

   (a) if $\hat{y}_1^s \geq y_1^{i,j}$ and $\hat{y}_2^s \geq y_2^{i,j}$, then node s can be fathomed,

   (b) else node s can not be fathomed.

3. Else node s can not be fathomed.

Based on this summary, we propose a fathoming procedure with one nadir point. A flowchart of this procedure is given bellow. Inputs to this procedure are $\mathbf{y}^i, \mathbf{y}^j, \tilde{\mathbf{y}}^{s1}$ and $\tilde{\mathbf{y}}^{s2}$. Using the inputs, we compute $\mathbf{y}^{i,j}$ and $\tilde{\mathbf{y}}^{sI}$. First, we check the locations of $\mathbf{y}^i$ and $\mathbf{y}^j$. If at least one of these points is in $\{\tilde{\mathbf{y}}^{sI}\}^{\leqq}$, then node s can be fathomed. Otherwise, we check whether the nadir point is not in T^s. If this holds, node s can not be fathomed. If the nadir point is in T^s, then we check the locations of the associated nondominated points. If at least one of these points is also in T^s, then node s can not be fathomed. Else, we solve the achievement function problem with the nadir point $\mathbf{y}^{i,j}$ and fathom node s if $\hat{y}_1^s > y_1^{i,j}$ and $\hat{y}_2^s > y_2^{i,j}$, where $\hat{\mathbf{y}}^s = (\hat{y}_1^s, \hat{y}_2^s)$ is the image of an optimal solution to the achievement function problem. If the above condition does not hold, then node s can not be fathomed.

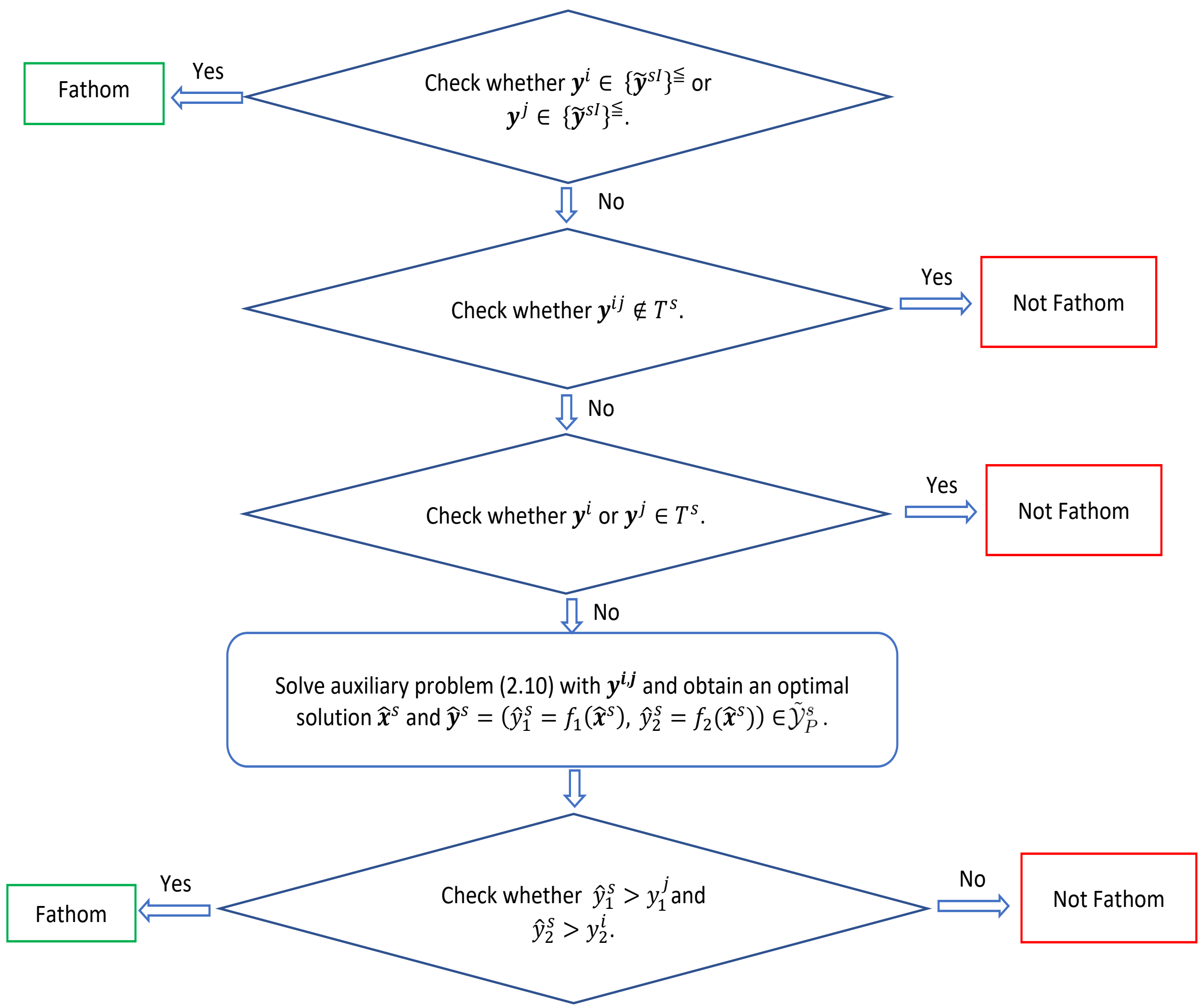
Flowchart - One nadir point

Input: Points $\boldsymbol{y}^i = (y_1^i, y_2^i), \boldsymbol{y}^j = (y_1^j, y_2^j) \in \bar{\mathcal{Y}}_a^s$

$\widetilde{\boldsymbol{y}}^{s1}, \widetilde{\boldsymbol{y}}^{s2} \in \tilde{\mathcal{Y}}_P^s$

Obtain: Nadir point $\boldsymbol{y}^{i,j} = \left(y_1^j, y_2^i\right)$

Ideal point $\widetilde{\boldsymbol{y}}^{sI}$, Set T^s



Appendix B Example

To illustrate the behavior of BOMIQPs, we present an example in this section. Note that the proposed BB algorithm is not used to solve this example.

Consider the following biobjective mixed binary quadratic program:

$$
\begin{aligned}
\min \quad & [f_1(\mathbf{x}) = (x_1-2)^2 + (x_2-2)^2 - 3x_1x_3 + 5x_2x_4 + 5x_2x_5, \\
& \qquad f_2(\mathbf{x}) = x_1^2 + (x_2-3)^2 + 3x_3 - 5x_2x_4] \\
\text{s.t.} \quad & \begin{array}{lll}
x_2 & \leq 2.25 & \\
x_1 + x_2 & \leq 2.75 & \\
2x_1 + x_2 & \leq 3.75 & \\
x_1, x_2 & \geq 0 & \\
x_3 + x_4 + x_5 & \leq 1 & (*) \\
x_3, x_4, x_5 & \in \{0,1\} & (**)
\end{array}
\end{aligned} \tag{1}
$$

Notice that in this example, x_1, x_2 are continuous variables, x_3, x_4, x_5 are binary variables, and the first four constraints contain only continuous variables. We define the set:

$$
\begin{aligned}
\mathcal{X}' = \{x_1, x_2 : \quad & x_2 \leq 2.25 \\
& x_1 + x_2 \leq 2.75 \\
& 2x_1 + x_2 \leq 3.75 \\
& x_1, x_2 \geq 0\}.
\end{aligned} \tag{2}
$$

This feasible set $\mathcal{X}'$ is illustrated in Figure 6.

In addition, the last two constraints contain only integer variables for which there are the following four possibilities,

(i) $(x_3, x_4, x_5) = (0, 0, 0)$

(ii) $(x_3, x_4, x_5) = (1, 0, 0)$

(iii) $(x_3, x_4, x_5) = (0, 1, 0)$

(iv) $(x_3, x_4, x_5) = (0, 0, 1)$.

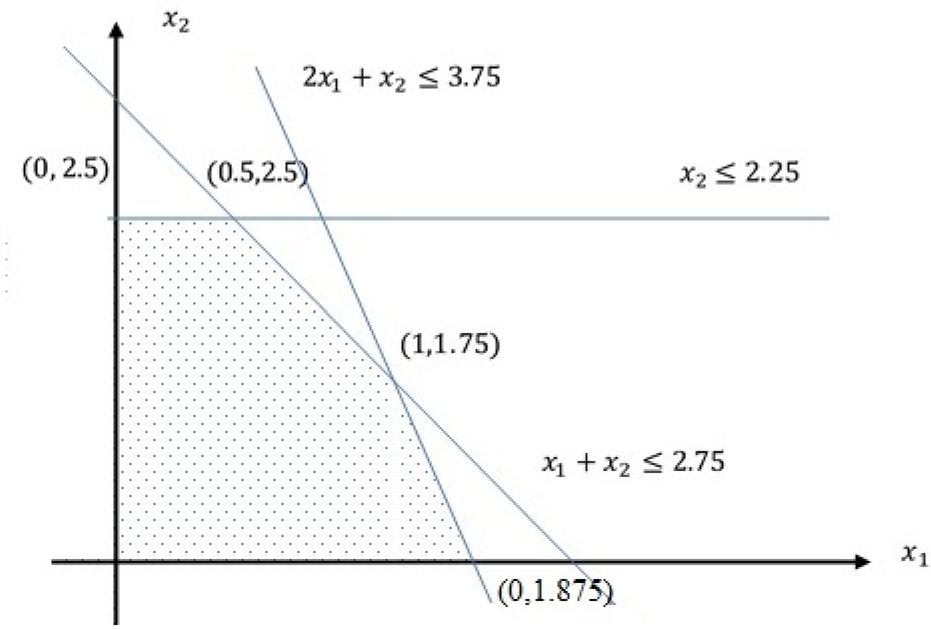


Figure 6: The decision space of example (1) with continuous variables x_1 and x_2

For each of the above cases, we solve the associated BOQP using the mpLCP method [3]. Each BOQP contains the continuous variables x_1 and x_2 only. In the solution, $\lambda \in [0, 1]$ is the weight parameter of the weighted sum problem associated with each BOQP.

(i) $(x_3, x_4, x_5) = (0, 0, 0)$

BOQP:

$$\begin{array}{ll} \min & [f_1(\mathbf{x}) = (x_1-2)^2 + (x_2-2)^2, \quad f_2(\mathbf{x}) = x_1^2 + (x_2-3)^2] \\ \text{s.t.} & \qquad x_1, x_2 \in \mathcal{X}' \end{array} \tag{3}$$

(ii) $(x_3, x_4, x_5) = (1, 0, 0)$

BOQP:

$$\begin{array}{ll} \min & [f_1(\mathbf{x}) = (x_1-2)^2 + (x_2-2)^2 - 3x_1, \quad f_2(\mathbf{x}) = x_1^2 + (x_2-3)^2] \\ \text{s.t.} & \qquad x_1, x_2 \in \mathcal{X}' \end{array} \tag{4}$$

(iii) $(x_3, x_4, x_5) = (0, 1, 0)$

BOQP:

$$\begin{array}{ll} \min & [f_1(\mathbf{x}) = (x_1-2)^2 + (x_2-2)^2 + 5x_2, \quad f_2(\mathbf{x}) = x_1^2 + (x_2-3)^2 - 5x_2] \\ \text{s.t.} & \qquad x_1, x_2 \in \mathcal{X}' \end{array} \tag{5}$$

(iv) $(x_3, x_4, x_5) = (0, 0, 1)$

BOQP:

$$\begin{array}{ll} \min & [f_1(\mathbf{x}) = (x_1-2)^2 + (x_2-2)^2 + 5x_2, \quad f_2(\mathbf{x}) = x_1^2 + (x_2-3)^2] \\ \text{s.t.} & \qquad x_1, x_2 \in \mathcal{X}' \end{array} \tag{6}$$

The solutions of these four BOQPs are given in Table 1, while Figure 7 depicts the objective space of the four problems.

The Pareto set $\mathcal{Y}_P$ of the biobjective mixed binary quadratic problem can be found geometrically by taking the lower envelope of the solution curves of the four BOQPs.

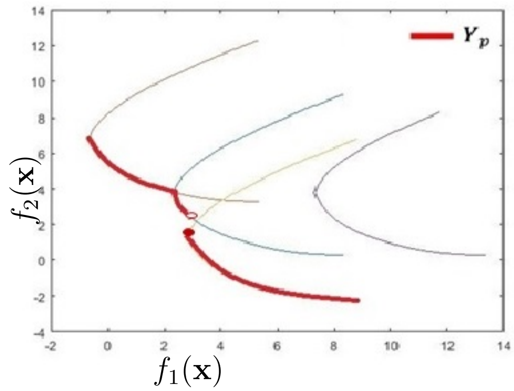


Figure 7: The objective space of Example (1)

Based on Example (1), we observe that the Pareto set of BOMIQPs is a union of curves in $\mathbb{R}^2$. Each curve being a subset of the Pareto set of the associated BOQP may be open or closed, or neither open nor closed, and may be reduced to a point. The union curve may be nonconvex and discontinuous which makes the Pareto set disconnected [74].

Case 1	Regions			
	$0 \leq \lambda \leq 1/8$	$1/8 \leq \lambda \leq 17/24$	$17/24 \leq \lambda \leq 7/8$	$7/8 \leq \lambda \leq 1$
x_1	4λ	$1/2$	$-13/8 + 3\lambda$	1
x_2	$9/4$	$9/4$	$35/8 - 3\lambda$	$7/4$

Case 2	Regions				
	$0 \leq \lambda \leq 1/14$	$1/14 \leq \lambda \leq 17/36$	$17/36 \leq \lambda \leq 7/12$	$7/12 \leq \lambda \leq 19/22$	$19/22 \leq \lambda \leq 1$
x_1	7λ	$1/2$	$-13/8 + 9\lambda/2$	1	$-9/10 + 11\lambda/5$
x_2	$9/4$	$9/4$	$35/8 - 9\lambda/2$	$7/4$	$111/20 - 22\lambda/5$

Case 3	Regions					
	$0 \leq \lambda \leq 1/8$	$1/8 \leq \lambda \leq 37/56$	$37/56 \leq \lambda \leq 41/56$	$41/56 \leq \lambda \leq 13/16$	$13/16 \leq \lambda \leq 191/192$	$191/192 \leq \lambda \leq 1$
x_1	4λ	$1/2$	$-33/8 + 7\lambda$	1	$-29/10 + 24\lambda/5$	$15/8$
x_2	$9/4$	$9/4$	$55/8 - 7\lambda$	$7/4$	$191/20 - 48\lambda/5$	0

Case 4	Regions					
	$0 \leq \lambda \leq 1/8$	$1/8 \leq \lambda \leq 17/44$	$17/44 \leq \lambda \leq 21/44$	$21/44 \leq \lambda \leq 19/36$	$19/36 \leq \lambda \leq 37/44$	$37/44 \leq \lambda \leq 1$
x_1	4λ	$1/2$	$-13/8 + 11\lambda/2$	1	$-9/10 + 18\lambda/5$	$15/8$
x_2	$9/4$	$9/4$	$35/8 - 11\lambda/2$	$7/4$	$111/20 - 36\lambda/5$	0

Table 1: The parametric solutions of the four BOQPs associated with Example (1).

Bibliography

[1] Z. Abraham. *Multiobjective Regression with Application to the Climate Domain*. PhD thesis, Michigan State University, East Lansing, MI, 2013.

[2] N. Adelgren. `https://github.com/Nadelgren/mpLCP_solver`, 2019. Accessed Novenber 3, 2021.

[3] N. Adelgren. *Advancing Parametric Optimization On Multiparametric Linear Complementarity Problems with Parameters in General Locations*. SpringerBriefs in Optimization. Springer International Publishing, 2021.

[4] N. Adelgren, B. Belotti, and A. Gupte. Efficient storage of pareto points in biobjective mixed integer programming. *CoRR*, abs/1411.6538, 2014.

[5] N. Adelgren and A. Gupte. Branch-and-bound for biobjective mixed-integer linear programming. `https://doi.org/10.1287/ijoc.2021.1092`, 2021. Accessed Novenber 3, 2021.

[6] M.J. Alves and J. Costa. Graphical exploration of the weight space in three-objective mixed integer linear programs. *European Journal of Operational Research*, 248(1):72–83, 2016.

[7] K.P. Anagnostopoulos and G. Mamanis. The mean-variance cardinality constrained portfolio optimization problem: An experimental evaluation of five multiobjective evolutionary algorithms. *Expert Systems with Applications*, 38(11):14208 – 14217, 2011.

[8] B. Bank, J. Guddat, D. Klatte, B. Kummer, and K. Tamer. *Non-Linear Parametric Optimization*. Springer Basel AG, 1983.

[9] T.Q. Bao and B.S. Mordukhovich. Set-valued optimization in welfare economics. In S. Kusuoka and T. Maruyama, editors, *Advances in Mathematical Economics*, volume 13. Springer, Tokyo, 2010.

[10] P.D. Barba. *Multiobjective Shape Design in Electricity and Magnetism*. Springer, New York, 2010.

[11] P. Baroso, T. Coudert, E. Villeneuve, and L. Geneste. Multi-objective optimization and risk assessment in system engineering project planning by ant colony algorithm. In *2014 IEEE International Conference on Industrial Engineering and Engineering Management*, pages 438–442, 2014.

[12] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty. *Nonlinear Programming.* John Wiley & Sons, 2006.

[13] E. Bednarczuk. Continuity of minimal points with applications to parametric multiple objective optimization. *European Journal of Operational Research*, 157(1):59–67, 2004.

[14] P. Belotti, B. Soylu, and M.M. Wiecek. Fathoming rules for biobjective mixed integer linear programs: Review and extensions. *Discrete Optimization*, 22:341 – 363, 2016.

[15] A. Bemporad and D.M. De la Peña. Multiobjective model predictive control. *Automatica*, 45(12):2823 – 2830, 2009.

[16] H.P. Benson. Multiple-objective linear programming with parametric criteria coefficients. *Management Science*, 31:461–474, 1985.

[17] G.R. Bitran. Linear multi-objective programs with interval coefficients. *Management Science*, 26:694–706, 1980.

[18] G.R. Bitran and J.M. Rivera. A combined approach to solve binary multicriteria problems. *Naval Research Logistics Quarterly*, 29(2):181–201.

[19] N. Boland, H. Charkhgard, and M. Savelsbergh. A criterion space search algorithm for biobjective mixed integer programming: The triangle splitting method. *INFORMS Journal on Computing*, 27(4):597–618, 2015.

[20] N. Boland, H. Charkhgard, and M. Savelsbergh. The l-shape search method for triobjective integer programming. *Mathematical Programming Computation*, 8(1):217–251, 2016.

[21] H. Bonnel and C. Schneider. Post-Pareto analysis and a new algorithm for the optimal parameter tuning of the elastic net. *Journal of Optimization Theory and Applications*, 183:993–1027, 2019.

[22] S. Boyd, E. Busseti, S. Diamond, R.N. Kahn, P. Nystrup, and J. Speth. Multi-period trading via convex optimization. *Foundations and Trends in Optimization*, 3(1):1–76, 2017.

[23] C. Buchheim, M. De Santis, F. Rinaldi, and L. Trieu. A frank-wolfe based branch-and-bound algorithm for mean-risk optimization. *Journal Global Optimization*, 70:625–644, 2018.

[24] R.S. Burachik, C.Y. Kaya, and M.M. Rizvi. A new scalarization technique to approximate pareto fronts of problems with disconnected feasible sets. *Journal of Optimization Theory and Applications*, 162:428—446, 2014.

[25] R.S. Burachik, C.Y. Kaya, and M.M. Rizvi. Algorithms for generating pareto fronts of multi-objective integer and mixed-integer programming problems. `https://arxiv.org/abs/1903.07041`, 2021. Accessed Novenber 3, 2021.

[26] V. Cacchiani and C. D'Ambrosio. A branch-and-bound based heuristic algorithm for convex multi-objective MINLPs. *European Journal of Operational Research*, 260(3):920 – 933, 2017.

[27] V. Chankong and Y.Y. Haimes. *Multiobjective Decision Making: Theory and Methodology.* North-Holland Series in System Science and Engineering. North Holland, 1983.

[28] H. Corley. A new scalar equivalence for pareto optimization. *IEEE Transactions on Automatic Control*, 25(4):829–830, 1980.

[29] R.W. Cottle. The principal pivoting method revisited. *Mathematical Programming*, 48:369 – 385, 1990.

[30] R.W. Cottle and S.M. Guu. Two characterizations of sufficient matrices. *Linear Algebra and its Applications*, 170:65 – 74, 1992.

[31] R.W. Cottle, J.S. Pang, and R. Stone. *The Linear Complementarity Problem.* Society for Industrial and Applied Mathematics, 1st edition, 2009.

[32] R.W. Cottle, J.S. Pang, and V. Venkateswaran. Sufficient matrices and the linear complementarity problem. *Linear Algebra and its Applications*, 114-115:231 – 249, 1989.

[33] K. Dächert and K. Klamroth. A linear bound on the number of scalarizations needed to solve discrete tricriteria optimization problems. *Journal of Global Optimization*, 16(1):643–676, 2015.

[34] M. De Santis and G. Eichfelder. A decision space algorithm for multiobjective convex quadratic integer optimization. *Computers and Operations Research*, 134:53–96, 2021.

[35] M. De Santis, G. Eichfelder, J. Niebling, and S. Rocktäschel. Solving multiobjective mixed integer convex optimization problems. *SIAM Journal on Optimization*, 30(4):3122–3145, 2020.

[36] M. Dellnitz and K. Witting. Computation of robust Pareto points. *International Journal of Computing Science and Mathematics*, 2(3):243–266, 2009.

[37] C. Delort and O. Spanjaard. Using bound sets in multiobjective optimization: Application to the biobjective binary knapsack problem. *SEA 2010: Experimental Algorithms*, 6049:253–265, 2010.

[38] D. Den Hertog, C. Roos, and T. Terlaky. The linear complementary problem, sufficient matrices and the criss-cross method. *Combinatorial Optimization. NATO ASI Series (Series F: Computer and Systems Sciences)*, 82:253–257, 1992.

[39] E. Diessel. An adaptive patch approximation algorithm for bicriteria convex mixed-integer problems. `http://www.optimization-online.org/DB_HTML/2020/07/7935`, 2021. Accessed Novenber 3, 2021.

[40] W. Dinkelbach. On nonlinear fractional programming. *Management Science*, 13(7):492–498, 1967.

[41] V. Dua, N.A. Bozinis, and E.N. Pistikopoulos. A multiparametric programming approach for mixed-integer quadratic engineering problems. *Computers and Chemical Engineering*, 26(4):715–733, 2002.

[42] M. Ehrgott. *Multicriteria Optimization*. Springer, 2005.

[43] M. Ehrgott and X. Gandibleux. Bound sets for biobjective combinatorial optimization problems. *Computers & Operations Research*, 34(9):2674 – 2694, 2007.

[44] M. Ehrgott, S. Greco, and J.R. Figueira. *Multiple Criteria Decision Analysis: State of the Art Surveys*. International Series in Operations Research and Management Science. Springer, 2nd edition, 2016.

[45] M. Ehrgott, K. Klamroth, and C. Schwehm. An mcdm approach to portfolio optimization. *European Journal of Operational Research*, 155(3):752–770, 2004. Traffic and Transportation Systems Analysis.

[46] M. Ehrgott and M.M. Wiecek. Multiobjective programming. In M. Ehrgott, S. Greco, and J.R. Figueira, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages 667–722. Springer, 2005.

[47] G. Eichfelder and L. Warnow. A hybrid patch decomposition approach to compute an enclosure for multi-objective mixed-integer convex optimization problems. `http://www.optimization-online.org/DB_HTML/2021/08/85411`, 2021. Accessed Novenber 3, 2021.

[48] G. Eichfelder and L. Warnow. On implementation details and numerical experiments for the hypad algorithm to solve multi-objective mixed-integer convex optimization problems. `http://www.optimization-online.org/DB_HTML/2021/08/8538`, 2021. Accessed Novenber 3, 2021.

[49] A.Z.H. El-Banna. A study on parametric multiobjective programming problems without differentiability. *Comput. Math. Appl.*, 26(12):87–92, 1993.

[50] R. Enkhbat, J. Guddat, and A. Chinchuluun. Parametric multiobjective optimization. In A. Chinchuluun, P. Pardalos, A. Migdalas, and L. Pitsoulis, editors, *Pareto Optimality, Game Theory and Equilibria*, volume 17 of *Springer Optimization and Its Applications*, pages 529–538. Springer, 2008.

[51] Y.P. Fang and X.Q. Yang. Smooth representations of optimal solution sets of piecewise linear parametric multiobjective programs. In *Variational Analysis and Generalized Differentiation in Optimization and Control*, volume 47 of *Springer Optim. Appl.*, pages 163–176. Springer, New York, 2010.

[52] A. Fiacco. *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming*, volume 165 of *Mathematics in Science and Engineering*. Academic Press, 1983.

[53] A. Fiacco and J. Kyparisis. Convexity and concavity properties of the optimal value function in parametric nonlinear programming. *Journal of Optimization Theory and Applications*, 48(1):95–126, 1986.

[54] N. Forget, K. Klamroth, S.L. Gadegaard, A. Przybylski, and L.R. Nielsen. Branch-and-bound and objective branching with three objectives. `http://www.optimization-online.org/DB_HTML/2020/12/8158.html`, 2020. Preprint.

[55] E. Galvan, R.J. Malak, D.J. Hartl, and J.W. Baur. Performance assessment of a multi-objective parametric optimization algorithm with application to a multi-physical engineering system. *Structural and Multidisciplinary Optimization*, 58:489–509, 2018.

[56] A.M. Geoffrion. Propr efficiency and the theory of vector maximization. *Journal of Mathematical Analysis and Applications*, 22(3):618 – 630, 1968.

[57] C.J. Goh and X.Q. Yang. Analytic efficient solution set for multi-criteria quadratic programs. *European Journal of Operational Research*, 92(1):166–181, 1996.

[58] J. Guddat, F.G. Vasquez, K. Tammer, and K. Wendler. *Multiobjective and Stochastic Optimization Based on Parametric Optimization*, volume 26 of *Mathematical Research*. Akademie-Verlag, Berlin, 1985.

[59] Y.Y. Haimes, L. Lasdon, and D Wismer. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-1(3):296–297, 1971.

[60] P.R. Halmos. *Naive Set Theory*. Undergraduate Texts in Mathematics. Springer New York, 1974.

[61] M.P. Hansen and A. Jaszkiewicz. Evaluating the quality of approximations to the non-dominated set. *IMM Technical Report IMM-REP-1998-7*, 1998.

[62] D.J. Hartl, E. Galvan, R.J. Malak, and J.W. Baur. Parameterized design optimization of a magnetohydrodynamic liquid metal active cooling concept. *J. Mech. Des.*, 138:031402 (11 pages), 2016.

[63] M. Hirschberger, Y. Qi, and R.E. Steuer. Large-scale MV efficient frontier computation via a procedure of parametric quadratic programming. *European Journal of Operational Research*, 204(3):581–588, 2010.

[64] M. Hirschberger, R.E. Steuer, S. Utz, W. Wimmer, and Y. Qi. Computing the non-dominated surface in tri-criterion portfolio selection. *Operational Research*, 61:169–183, 2013.

[65] M. Hirschberger, R.E. Steuer, S. Utz, W. Wimmer, and Y. Qi. Computing the non-dominated surface in tri-criterion portfolio selection. *Operational Research*, 61:169–183, 2013.

[66] B.F. Hobbs and P. Meier. *Energy Decisions and the Environment: A Guide to the Use of Multicriteria Methods*. International Series in Operations Research & Management Science. Springer US, 2012.

[67] N.Q. Huy, B.S. Mordukhovich, and J.C. Yao. Coderivatives of frontier and solution maps in parametric multiobjective optimization. *Taiwanese J. Math.*, 12(8):2083–2111, 2008.

[68] P.L.W. Jayasekara, N. Adelgren, and M.M. Wiecek. On convex multiobjective programs with application to portfolio optimization. *Journal of Multi-Criteria Decision Analysis*, 27(3-4):189–202, 2019.

[69] P.L.W. Jayasekara, A.C. Pangia, and M.M. Wiecek. On solving parametric multiobjective quadratic programs with parameters in general locations, 2021. Manuscript submitted for publication.

[70] P. Juszczuk, I. Kaliszewski, and J. Miroforidis. Trade-off guided search for approximate Pareto optimal portfolios. In T. Trzaskalik, editor, *Multiple Criteria Decision Analysis*, volume 12, pages 49–59. The University of Economics in Katowice, 2017.

[71] G. Kiziltan and E. Yucao**u**glu. An algorithm for multiobjective zero-one linear programming. *Management Science*, 29(12):1444–1453, 1983.

[72] E. Klafszky and T. Terlaky. Some generalizations of the criss-cross method for quadratic programming. *Optimization*, 24(1-2):127–139, 1992.

[73] K. Klamroth and M.M. Wiecek. A bi-objective median location problem with a line barrier. *Operations Research*, 50(4):670–679.

[74] J. Leverenz. *Network Target Coordination for Multiparametric Programming*. PhD thesis, Clemson University, 2016.

[75] J. Leverenz, M. Xu, and M.M. Wiecek. Multiparametric optimization for multidisciplinary engineering design. *Structural and Multidisciplinary Optimization*, 54(4):1–16, 2016.

[76] D. Li, J.B. Yang, and M.P. Biswal. Quantitative parametric connections between methods for generating noninferior solutions in multiobjective optimization. *European Journal of Operational Research*, 117:84 – 99, 1999.

[77] J. Li and J. Xu. Multi-objective portfolio selection model with fuzzy random returns and a compromise approach-based genetic algorithm. *Information Sciences*, 220:507 – 521, 2013.

[78] J. G. Lin. Multiple-objective problems: Pareto-optimal solutions by method of proper equality constraints. *IEEE Transactions on Automatic Control*, 21(5):641–650, 1976.

[79] J.G. Lin. Proper inequality constraints and maximization of index vectors. *Journal of Optimization Theory and Applications*, 21(4):505–521, 1977.

[80] C. Lu, Z. Deng, J. Zhou, and X. Guo. A sensitive-eigenvector based global algorithm for quadratically constrained quadratic programming. *Journal of Global Optimization*, 73(2):371–388, 2019.

[81] D.T. Luc. *Multiobjective Linear Programming*. Springer Books. Springer, 2016.

[82] R.E. Lucchetti and E. Miglierina. Stability for convex vector optimization problems. *Optimization*, 53(5–6):517–528, 2004.

[83] S.J. Mardle, S. Pascoe, and M. Tamiz. An investigation of genetic algorithms for the optimization of multi-objective fisheries bioeconomic models. *International Transactions in Operational Research*, 7(1):33–49, 2000.

[84] H. Markowitz. Portfilio selection. *The Journal of Finance*, 7(1):77–91, 1952.

[85] B. Martin, A. Goldsztejn, L. Granvilliers, and C. Jermann. Constraint propagation using dominance in interval branch and bound for nonlinear biobjective optimization. *European Journal of Operational Research*, 260(3):934 – 948, 2017.

[86] G. Mavrotas and D. Diakoulaki. A branch and bound algorithm for mixed zero-one multiple objective linear programming. *European Journal of Operational Research*, 107(3):530 – 541, 1998.

[87] R.C. Merton. An analytic derivation of the efficient portfolio frontier. *The Journal of Financial and Quantitative Analysis*, 7(4):1851–1872, 1972.

[88] K. Metaxiotis and K. Liagkouras. Multiobjective evolutionary algorithms for portfolio management: A comprehensive literature review. *Expert Syst. Appl.*, 39(14):11685–11698, 2012.

[89] A. Mostashari. *Collaborative Modeling and Decision-Making for Complex Energy Systems*. World Scientific, 2011.

[90] P.H. Naccache. Stability in multicriteria optimization. *J. Math. Anal. Appl.*, 68(2):441–453, 1979.

[91] J. Niebling and G. Eichfelder. A branch and bound-based algorithm for nonconvex multiobjective optimization. *SIAM Journal on Optimization*, 29(1):794–821, 2019.

[92] R. Oberdiecka and N. Pistikopoulos. Multiobjective optimization with convex quadratic cost functions: A multiparametric programming approach. *Computers and Chemical Engineering*, 85:36–39, 2016.

[93] S. N. Parragh and F. Tricoire. Branch-and-bound for bi-objective integer programming. *INFORMS Journal on Computing*, 31(4):805–822, 2019.

[94] J.P. Penot and A. Sterna-Karwat. Parametrized multicriteria optimization: Continuity and closedness of optimal multifunctions. *Journal of Mathematical Analysis and Applications*, 120(1):150–168, 1986.

[95] J.P. Penot and A. Sterna-Karwat. Parametrized multicriteria optimization: Order continuity of the marginal multifunctions. *Journal of Mathematical Analysis and Applications*, 144(1):1–15, 1989.

[96] T. Perini, N. Boland, D. Pecin, and M. Savelsbergh. A criterion space method for biobjective mixed integer programming: The boxed line method. *INFORMS Journal on Computing*, 32(1):16–39, 2020.

[97] A.D. Pia, S.S. Dey, and M. Molinaro. Mixed-integer quadratic programming is in NP. *Mathematical Programming*, pages 225–240, 2017.

[98] A. Pizzo, A. Zappone, and L. Sanguinetti. Solving fractional polynomial problems by polynomial optimization theory. *IEEE Signal Processing Letters*, 25(10):1540–1544, 2018.

[99] A. Przybylski and X. Gandibleux. Multi-objective branch and bound. *European Journal of Operational Research*, 260(3):856 – 872, 2017.

[100] Y. Qi. On the criterion vectors of lines of portfolio selection with multiple quadratic and multiple linear objectives. *Central European Journal of Operations Research*, 25(1):145–158, 2017.

[101] Y. Qi, R.E. Steuer, and M. Wimmer. An analytical derivation of the efficient surface in portfolio selection with three criteria. *Annals of Operations Research*, 251(1):161–177, 2017.

[102] G.P. Rangaiah. *Multi-Objective Optimization Techniques and Applications in Chemical Engineering*. World Scientific, 2008.

[103] S.A.B. Rasmi, A. Fattahi, and M. Türkay. SASS: slicing with adaptive steps search method for finding the non-dominated points of tri-objective mixed-integer linear programming problems. *Annals of Operations Research*, 296(1):841–876, 2021.

[104] S.A.B. Rasmi and M. Türkay. GoNDEF: an exact method to generate all non-dominated points of multi-objective mixed-integer linear programs. *Optimization and Engineering*, 20(1):89–117, 2019.

[105] S. Rocktäschel. *A Branch-and-Bound Algorithm for Multiobjective Mixed-integer Convex Optimization*. BestMasters. Springer Spektrum, 2020.

[106] O. Romanko. *Parametric and Multiobjective Optimization with Applications in Finance*. PhD thesis, McMaster University, 2010.

[107] O. Romanko, A. Ghaffari-Hadigheh, and T. Terlaky. Multiobjective optimization via parametric optimization: Models, algorithms, and applications. In T. Terlaky and F.E. Curtis, editors, *Modeling and Optimization: Theory and Applications*, pages 77–119. Springer New York, 2012.

[108] G.R. Ruetsch. Using interval techniques to solve a parametric multi-objective optimization problem. *United States Patent No. 7,664,622 B2*, 2010.

[109] F. Salas-Molina, D. Pla-Santamaria, and J.A. Rodríguez-Aguilar. Empowering cash managers through compromise programming. In H. Masri, B. Pèrez-Gladish, and C. Zopounidis, editors, *Financial Decision Aid Using Multiple Criteria*, volume 12, pages 149–173. Springer, 2018.

[110] Y. Sawaragi, H. Nakayama, and T. Tanino. *Theory of Multiobjective Optimization*. Academic Press, 1985.

[111] B. Sawik. *MultiObjective Portfolio Optimization by Mixed Integer Programming*. PhD thesis, AGH University of Science and Technology, Krakow, Poland, 2011.

[112] S. Sharma and G. P. Rangaiah. *Multi-Objective Optimization Applications in Chemical Engineering*. John Wiley and Sons, Ltd, 2013.

[113] K. Smimou. International portfolio choice and political instability risk: A multi-objective approach. *European Journal of Operational Research*, 234(2):546 – 560, 2014.

[114] F. Sourd and O. Spanjaard. A multiobjective branch-and-bound framework: Application to the biobjective spanning tree problem. *INFORMS Journal on Computing*, 20(3):472–484, 2008.

[115] J. Spronk, R.E. Steuer, and C. Zopounidis. *Multicriteria decision aid/analysis in finance*. 2016.

[116] R.E. Steuer. *Multiple Criteria Optimization: Theory, Computation, and Application*. Wiley, 1986.

[117] R.E. Steuer, Y. Qi, and M. Hirschberger. Comparative issues in large-scale mean-variance efficient frontier computation. *Decision Support Systems*, 51:250–255, 2011.

[118] T. Stidsen, K.A. Andersen, and B. Dammann. A branch and bound algorithm for a class of biobjective mixed integer programs. *Management Science*, 60(4):1009–1032, 2014.

[119] J. Stillwell. Complex numbers and curves. In *Mathematics and Its History*, pages 295–313. Springer-Verlag New York, 2010.

[120] L.V. Thuan and D.T. Luc. On sensitivity in linear multiobjective programming. *J. Optim. Theory Appl.*, 107(3):615–626, 2000.

[121] L. Trieu. *Continuous optimization methods for convex mixed-integer nonlinear programming*. PhD thesis, Technischen Universität Dortmund, Dortmund, Germany, 2015.

[122] M.G. Tsionas. Multi-objective optimization using statistical models. *European Journal of Operational Research*, 276(1):364–378, 2019.

[123] S. Utz, M. Wimmer, M. Hirschberger, and R.E. Steuer. Tri-criterion inverse portfolio optimization with application to socially responsible mutual funds. *European Journal of Operational Research*, 234:491–498, 2014.

[124] H. Väliaho. A procedure for the one-parametric linear complementarity problem. *Optimization*, 29(3):235–256, 1994.

[125] T. Vincent, F. Seipp, S. Ruzika, A. Przybylski, and X. Gandibleux. Multiple objective branch and bound for mixed 0-1 linear programming: Corrections and improvements for the biobjective case. *Computers & Operations Research*, 40(1):498–509, 2013.

[126] R.E. Wendell and D.N. Lee. Efficiency in multiple objective optimization problems. *Mathematical Programming*, 12(1):406–414, 1977.

[127] M.M. Wiecek and G. Dranichak. Robust multiobjective optimization for decision making under uncertainty and conflict. In J.C. Smith, editor, *Optimization Challenges in Complex, Networked, and Risky Systems*, Tutorials in Operations Research, pages 84–114. INFORMS, 2016.

[128] M.M. Wiecek, M. Ehrgott, and A. Engau. Continuous multiobjective programming. In S. Greco, M. Ehrgott, and J.R. Figueira, editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages 738–815. Springer, 2nd edition, 2016.

[129] P. R. Wiecha, A. Arbouet, C. Girard, A. Lecestre, G. Larrieu, and V. Paillard. Evolutionary multi-objective optimization of colour pixels based on dielectric nanoantennas. *Nature Nanotechnology*, 12:163–169, 201.

[130] A. P. Wierzbicki. On the completeness and constructiveness of parametric characterizations to vector optimization problems. *Operations-Research-Spektrum*, 8:73–87, 1986.

[131] A.P. Wierzbicki. A methodological approach to comparing parametric characterizations of efficient solutions. In G. Fandel, M. Grauer, A. Kurzhanski, and A.P. Wierzbicki, editors, *Large-Scale Modelling and Interactive Decision Analysis*, volume 273, pages 27–45. Springer Berlin Heidelberg, 1986.

[132] K. Witting, S. Ober-Blöbaum, and M. Dellnitz. A variational approach to define robustness for parametric multiobjective optimization problems. *J. Global Optim.*, 57(2):331–345, 2013.

[133] K. Witting, B. Schulz, M. Dellnitz, J. Böcker, and N. Fröhleke. A new approach for online multiobjective optimization of mechatronic systems. *International Journal of Software Tools and Technology Transfer*, 10:223–231, 2008.

[134] P. Xidonas and G. Mavrotas. Multiobjective portfolio optimization with non-convex policy constraints: Evidence from the Eurostoxx 50. *The European Journal of Finance*, 20(11):957–977, 2012.

[135] P. Xidonas and G. Mavrotas. Multiobjective portfolio optimization with non-convex policy constraints: Evidence from the Eurostoxx 50. *The European Journal of Finance*, 20(11):957–977, 2012.

[136] P. Xidonas and G. Mavrotas. Comparative issues between linear and non-linear risk measures for non-convex portfolio optimization: evidence from the S& P 500. *Quantitative Finance*, 14(7):1229–1242, 2014.

[137] P. Xidonas, G. Mavrotas, C. Hassapis, and C. Zopounidis. Robust multiobjective portfolio optimization: A minimax regret approach. *European Journal of Operational Research*, 262(1):299 – 305, 2017.

[138] P. Xidonas, G. Mavrotas, T. Krintas, J. Psarras, and C. Zopounidis. *Multicriteria Portfolio Management*. Springer Optimization and Its Applications. Springer New York, 2012.

[139] M. Zarghami and F. Szidarovszky. *Multicriteria Analysis: Applications to Water and Environment Management*. Springer Berlin Heidelberg, 2011.

[140] H. Zhou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society B*, 67(2):301–320, 2005.

[141] E. Zitzler, L. Thiele, and J. Bader. On set-based multiobjective optimization. *Trans. Evol. Comp*, 14(1):58–79, 2010.

[142] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, and V.G. da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.

[143] C. Zopounidis, D.K. Despotis, and I. Kamaratou. Profilio selection using the Adelais multiobjective linear programming system. *Computational Economics*, 11(3):189–204, 1998.