Noname manuscript No.
(will be inserted by the editor)

On Solving Parametric Multiobjective Quadratic Programs with Parameters in General Locations

Pubudu L.W. Jayasekara · Andrew C. Pangia · Margaret M. Wiecek

the date of receipt and acceptance should be inserted later

Abstract While theoretical studies on parametric multiobjective programs (mpMOPs) have been steadily progressing, the algorithmic development has been comparatively limited despite the fact that parametric optimization can provide a complete parametric description of the efficient set. This paper puts forward the premise that parametrization of the efficient set of nonparametric MOPs can be combined with solving parametric MOPs because the algorithms performing the former can also be used to achieve the latter.

This strategy is realized through (i) development of a generalized scalarization, (ii) a computational study of selected parametric optimization algorithms, and (iii) applications in a real-life context. Several variants of a generalized weighted-sum scalarization allow one to scalarize mpMOPs to match the capabilities of algorithms. Parametric multiobjective quadratic programs are scalarized into parametric quadratic programs (mpQPs) with linear and/or quadratic constraints. In the computational study, three algorithms capable of solving mpQPs are examined on synthetic instances and two of the algorithms are applied to decision-making problems in statistics and portfolio optimization. The real-life context reveals the interplay between the scalarizations and provides additional insight into the obtained parametric solution sets.

Keywords Multiobjective quadratic optimization · parametric optimization · parametric linear complementarity problem · simplex approximation · elastic net · portfolio optimization

Pubudu L.W. Jayasekara
School of Mathematical and Statistical Sciences
Clemson University
Clemson, SC 29634
E-mail: pwijesi@g.clemson.edu

Andrew C. Pangia (Corresponding Author)
School of Mathematical and Statistical Sciences
Clemson University
Clemson, SC 29634
E-mail: apangia@g.clemson.edu

Margaret M. Wiecek
School of Mathematical and Statistical Sciences
Clemson University
Clemson, SC 29634
E-mail: wmalgor@clemson.edu

Conflict of Interest

Funding: The authors gratefully acknowledge support by the United States Office of Naval Research through grant number N00014-16-1-2725.

Conflicts of interest/Competing interests: The authors have no relevant financial or non-financial interests to disclose.

Availability of data and material: Not Applicable

Code availability: Code was devised in MATLAB, available on request.

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 [17]. 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 [49].

Studies on mpMOPs go back to 1979 when Naccache [37] 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 [3,18,29,36,40,41,45]. Parametric linear programs are analyzed in [5,6], while the polyhedral structure of the efficient set for such problems is more recently examined in [20,47]. 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 [13, 34, 51] and applied to mechatronic systems in which the parameter plays the role of time [52]. The theoretical framework for a solution method based on a technique which subdivides the solution and parameter spaces is proposed in [44]. In engineering design, genetic algorithms are tailored to the parametric case to allow for design exploration in the presence of exogenous factors [21, 25].

Independently of parametric multiobjective optimization, parametrization of the efficient set of MOPs can be conducted as a result of treating scalarized MOPs as mpSOPs [50]. This point of view is theoretically examined in [19,23] and used for computational work in [27,28,30,43,46] to obtain different types of parametric descriptions of the efficient set for MOPs arising in portfolio optimization. In [38], 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 2, we formulate the mpMOQP and define solution concepts. The generalized weighted-sum method and related scalarizations for MOPs are developed in Section 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 we provide algorithms for mpQPs with linear constraints, while in Section 5 we present an algorithm for mpQPs with quadratic and linear constraints. In each section, results from computational tests performed on synthetic problems are included. Section 6 contains the applications and the paper is concluded in Section 7.

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:

$$\begin{aligned}\min_{\mathbf{x}}\ \mathbf{f}(\mathbf{x};\boldsymbol{\theta}) = [&f_1(\mathbf{x};\boldsymbol{\theta}) = \frac{1}{2}\mathbf{x}^T Q_1(\boldsymbol{\theta})\mathbf{x} + \mathbf{p}_1^T(\boldsymbol{\theta})\mathbf{x} + c_1(\boldsymbol{\theta}),\ \ldots,\ f_{\tilde{r}}(\mathbf{x};\boldsymbol{\theta}) = \frac{1}{2}\mathbf{x}^T Q_{\tilde{r}}(\boldsymbol{\theta})\mathbf{x} + \mathbf{p}_{\tilde{r}}^T(\boldsymbol{\theta})\mathbf{x} + c_{\tilde{r}}(\boldsymbol{\theta}),\\ &f_{\tilde{r}+1}(\mathbf{x};\boldsymbol{\theta}) = \mathbf{p}_{\tilde{r}+1}^T(\boldsymbol{\theta})\mathbf{x} + c_{\tilde{r}+1}(\boldsymbol{\theta}),\ \ldots,\ f_r(\mathbf{x};\boldsymbol{\theta}) = \mathbf{p}_r^T(\boldsymbol{\theta})\mathbf{x} + c_r(\boldsymbol{\theta})] \qquad (\text{MOQP}(\boldsymbol{\theta}))\\ \text{s.t. }& \mathbf{x} \in \mathcal{X}(\boldsymbol{\theta}) = \{\mathbf{x} \in \mathbb{R}^n : A(\boldsymbol{\theta})\mathbf{x} \leqq \boldsymbol{b}(\boldsymbol{\theta}), \mathbf{x} \geqq \mathbf{0}\}\\ &\boldsymbol{\theta} \in \Theta,\end{aligned}$$

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 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 1 1. Let $\boldsymbol{\theta} \in \Theta$ and $\mathbf{x}^1, \mathbf{x}^2 \in \mathcal{X}(\boldsymbol{\theta})$. Then $\mathbf{f}(\mathbf{x}^1;\boldsymbol{\theta})(<)(\leqq) \leq \mathbf{f}(\mathbf{x}^2;\boldsymbol{\theta})$ if and only if $f_i(\mathbf{x}^1;\boldsymbol{\theta})(<) \leq f_i(\mathbf{x}^2;\boldsymbol{\theta})$ for all $i = 1, \dots, r$, where $\leq$ requires strict inequality for at least one index i, while $\leqq$ allows equality for all i.
2. Let $\mathbf{x}^1, \mathbf{x}^2 \in \mathcal{X}(\Theta)$ and $i \in \{1, \dots, r\}$. Then $f_i(\mathbf{x}^1;\boldsymbol{\theta}) \leq f_i(\mathbf{x}^2;\boldsymbol{\theta})$ if and only if $f_i(\mathbf{x}^1;\boldsymbol{\theta}) \leq f_i(\mathbf{x}^2;\boldsymbol{\theta})$ for all $\boldsymbol{\theta} \in \Theta$.
3. Let $\mathbf{x}^1, \mathbf{x}^2 \in \mathcal{X}(\Theta)$. Then $\mathbf{f}(\mathbf{x}^1;\boldsymbol{\theta})(<)(\leqq) \leq \mathbf{f}(\mathbf{x}^2;\boldsymbol{\theta})$ if and only if $\mathbf{f}(\mathbf{x}^1;\boldsymbol{\theta})(<)(\leqq) \leq \mathbf{f}(\mathbf{x}^2;\boldsymbol{\theta})$ 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 2 A feasible solution $\hat{\mathbf{x}} \in \mathcal{X}(\bar{\boldsymbol{\theta}})$ is called (weakly) efficient to MOQP($\boldsymbol{\theta}$) for $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}} \in \Theta$ if there exists no other solution $\mathbf{x} \in \mathcal{X}(\bar{\boldsymbol{\theta}})$ such that $\mathbf{f}(\mathbf{x};\bar{\boldsymbol{\theta}})(<) \leq \mathbf{f}(\hat{\mathbf{x}};\bar{\boldsymbol{\theta}})$. 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 3 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 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 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.

3 Generalized weighted sum scalarization

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 5 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 5, 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 1 *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 [22], $\hat{\mathbf{x}}$ is an optimal solution to the SOP

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

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

$$\min_{\mathbf{x}\in\mathcal{X}} \; \rho_0 \sum_{i\in J} \lambda_i g_i(\mathbf{x}) + \rho_1 \sum_{i\in J_1} \mu_i^1 g_i(\mathbf{x}) + \cdots + \rho_t \sum_{i\in J_t} \mu_i^t g_i(\mathbf{x}), \tag{2}$$

where

$$\boldsymbol{\rho} := [\rho_0,\ \rho_1,\ \ldots,\ \rho_t] \in \mathcal{P} := \left\{ \boldsymbol{\rho} \in \mathbb{R}^{t+1} : \rho_i \geq 0, i = 1,\ \ldots,\ t+1, \sum_{i=0}^{t} \rho_i = 1 \right\}. \tag{3}$$

If one can find $\boldsymbol{\rho}$, $\boldsymbol{\lambda}$, and $\boldsymbol{\mu}$ such that $\hat{\mathbf{x}}$ is an optimal solution to (2), then, by [22], $\hat{\mathbf{x}}$ will equivalently be weakly efficient to (MOP'). Similar to the proof of Theorem 4 in [30], 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 (3). Then, $\hat{\mathbf{x}}$ is an optimal solution to (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 1, are also weakly efficient to (MOP). Because scalarizations (8) and (9) are new, in the subsequent propositions their relationships with (MOP) are examined.

Corollary 1 *Let* (MOP) *and* (MOP') *be given, and the sets J and J_j, $j = 1, \ldots, t$ be defined as in Def. 5. 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) *[22]:*

$$\begin{aligned} &\min_{\boldsymbol{x} \in X} \sum_{i=1}^{r} \lambda_i g_i(\boldsymbol{x}) \\ &s.t.\ \boldsymbol{\lambda} \in \Lambda \subseteq \mathbb{R}^r. \end{aligned} \tag{4}$$

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) *[24]:*

$$\begin{aligned} &\min_{\boldsymbol{x} \in X} g_i(\boldsymbol{x}) \\ &s.t.\ \ g_j(\boldsymbol{x}) \leq \epsilon_j,\ j = 1, \ldots, r, j \neq i \\ &\quad\ \ \boldsymbol{\epsilon} \in \mathcal{E} \subseteq \mathbb{R}^{r-1}. \end{aligned} \tag{5}$$

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) *[23]:*

$$\begin{aligned} &\min_{\boldsymbol{x} \in X} \sum_{i=1}^{r} \lambda_i g_i(\boldsymbol{x}) \\ &s.t.\ \ g_j(\boldsymbol{x}) \leq \epsilon_j,\ j = 1, \ldots, r \\ &\quad\ \ \boldsymbol{\lambda} \in \Lambda \subseteq \mathbb{R}^r,\ \boldsymbol{\epsilon} \in \mathcal{E} \subseteq \mathbb{R}^r. \end{aligned} \tag{6}$$

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) *[30]:*

$$\begin{aligned} &\min_{\boldsymbol{x} \in X} \sum_{i \in J} \lambda_i g_i(\boldsymbol{x}) \\ &s.t.\ \ g_j(\boldsymbol{x}) \leq \epsilon_j,\ j \notin J \\ &\quad\ \ \boldsymbol{\lambda} \in \Lambda \subseteq \mathbb{R}^p,\ \boldsymbol{\epsilon} \in \mathcal{E} \subseteq \mathbb{R}^{r-p}. \end{aligned} \tag{7}$$

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

$$\begin{aligned} \min_{x \in X} \; & \sum_{i \in J} \lambda_i g_i(\boldsymbol{x}) \\ s.t. \; & \sum_{i \in J_j} \mu_i^j g_i(\boldsymbol{x}) \leq \epsilon_j, \; j = 1, \ldots, t \\ & \boldsymbol{\lambda} \in \Lambda \subseteq \mathbb{R}^{|J|}, \; \boldsymbol{\mu} \in \mathcal{M}, \; \boldsymbol{\epsilon} \in \mathcal{E} \subseteq \mathbb{R}^t. \end{aligned} \tag{8}$$

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, $i \notin J$:*

$$\begin{aligned} \min_{x \in X} \; & g_i(\boldsymbol{x}) \\ s.t. \; & \sum_{j \in J} \lambda_j g_j(\boldsymbol{x}) \leq \epsilon \\ & g_j(\boldsymbol{x}) \leq \epsilon_j, \; j \notin J, \; j \neq i \\ & \boldsymbol{\lambda} \in \Lambda \subseteq \mathbb{R}^p, \; \boldsymbol{\epsilon} \in \mathcal{E} \subseteq \mathbb{R}^{r-p}. \end{aligned} \tag{9}$$

The weighted hybrid SOP (8) allows for weighing all objective functions in the new objective and constraints.

Proposition 2 *Let $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}(\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\epsilon})$ be an optimal solution to the weighted hybrid SOP* (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 (8). Therefore, $\hat{\mathbf{x}}$ is feasible to (8), that is,

$$\textstyle\sum_{i \in J_j} \mu_i^j g_i(\mathbf{x}) \leq \epsilon_j, \; j = 1, \ldots, t. \tag{10}$$

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

$$g_i(\bar{\mathbf{x}}) < g_i(\hat{\mathbf{x}}) \;\text{ for all }\; i = 1, \ldots, r. \tag{11}$$

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 (8) with $\epsilon_j = \sum_{i \in J_j} \mu_i^j g_i(\hat{\mathbf{x}}),\; j = 1, \ldots, t$. From (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 (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 (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 3 *If $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}(\boldsymbol{\lambda}, \boldsymbol{\epsilon})$ is an optimal solution to the reduced ϵ-constraint SOP_i (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 (9) for some $i \notin J$. Therefore, $\hat{\mathbf{x}}$ is feasible to (9), that is,

$$\textstyle\sum_{j \in J} \lambda_j g_j(\hat{\mathbf{x}}) \leq \epsilon \tag{12}$$

$$g_j(\hat{\mathbf{x}}) \leq \epsilon_j \; j \notin J, \; j \neq i. \tag{13}$$

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

$$g_j(\bar{\mathbf{x}}) < g_j(\hat{\mathbf{x}}) \;\text{ for all }\; j = 1, \ldots, r. \tag{14}$$

Applying $\lambda_j \geq 0$ not all 0, we have $\lambda_j g_j(\bar{\mathbf{x}}) \leq \lambda_j g_j(\hat{\mathbf{x}}), \;\; j \in J$ 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 (12), we obtain

$$\sum_{j \in J} \lambda_j g_j(\bar{\mathbf{x}}) < \epsilon, \tag{15}$$

while from (13) and (14),

$$g_j(\bar{\mathbf{x}}) < \epsilon_j \; j \notin J, \; j \neq i. \tag{16}$$

Since $\bar{\mathbf{x}} \in \mathcal{X}$, (15) and (16) make $\bar{\mathbf{x}}$ feasible to (9). From (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 *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 (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 (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 [30], 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 [8] then yields the desired result.

In the next two sections we present algorithms for solving MOQP($\boldsymbol{\theta}$) that is scalarized with the approaches presented in this section.

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

$$\begin{aligned}
\min_{\mathbf{x}} \; & \mathbf{f}(\mathbf{x}; \boldsymbol{\theta}) = [f_1(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q_1 \mathbf{x} + \mathbf{p}_1^T \mathbf{x}, \; \ldots, \; f_{\tilde{r}}(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q_{\tilde{r}} \mathbf{x} + \mathbf{p}_{\tilde{r}}^T \mathbf{x}, \\
& \qquad f_{\tilde{r}+1}(\mathbf{x}; \boldsymbol{\theta}) = \mathbf{p}_{\tilde{r}+1}^T(\boldsymbol{\theta}) \mathbf{x}, \; \ldots, \; f_r(\mathbf{x}; \boldsymbol{\theta}) = \mathbf{p}_r^T(\boldsymbol{\theta}) \mathbf{x}] \\
\text{s.t. } & \mathbf{x} \in \mathcal{X}(\boldsymbol{\theta}) = \{\mathbf{x} \in \mathbb{R}^n : A(\boldsymbol{\theta})\mathbf{x} \leqq \boldsymbol{b}(\boldsymbol{\theta}), \mathbf{x} \geqq \mathbf{0}\} \\
& \boldsymbol{\theta} \in \Theta,
\end{aligned} \tag{MOQP$_1(\boldsymbol{\theta})$}$$

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 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) or modified hybrid (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 [30], four state-of-the-art methods to solve mpQPs are compared. Only one of these methods, the mpLCP method [2] based on the linear complementarity problem (LCP) reformulation of the original problem, can solve mpQPs with parameters in general locations. The study in [30] does not include a method that has been developed much earlier in [48] 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 (nonparametric) 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 (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.

4.1 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 (7) to $\text{MOQP}_1(\boldsymbol{\theta})$, the resulting mpQP can be written as:

$$\begin{aligned} \min_{\mathbf{x}} \ & f(\mathbf{x};\boldsymbol{\lambda}) = \frac{1}{2}\mathbf{x}^T Q(\boldsymbol{\lambda})\mathbf{x} + \mathbf{p}(\boldsymbol{\lambda})^T\mathbf{x} \\ \text{s.t.} \ & \tilde{A}(\boldsymbol{\theta})\mathbf{x} \leqq \tilde{\boldsymbol{b}}(\boldsymbol{\theta},\boldsymbol{\epsilon}) \\ & \mathbf{x} \geqq \mathbf{0} \\ & \boldsymbol{\theta} \in \Theta, \boldsymbol{\lambda} \in \Lambda', \boldsymbol{\epsilon} \in \mathcal{E}, \end{aligned} \tag{17}$$

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 (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:

$$\begin{gathered} \mathbf{w} - M(\boldsymbol{\theta},\boldsymbol{\lambda})\mathbf{z} = \mathbf{q}(\boldsymbol{\theta},\boldsymbol{\lambda},\boldsymbol{\epsilon}) \\ \mathbf{w}^T\mathbf{z} = 0 \\ \mathbf{w}, \mathbf{z} \geqq \mathbf{0} \\ \boldsymbol{\theta} \in \Theta, \boldsymbol{\lambda} \in \Lambda', \boldsymbol{\epsilon} \in \mathcal{E} \end{gathered} \tag{18}$$

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, $i = 1, \ldots, \tilde{r}$, 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 [12].

Definition 6 Let $\boldsymbol{\theta} \in \Theta, \boldsymbol{\lambda} \in \Lambda', \boldsymbol{\epsilon} \in \mathcal{E}$ and B be a basis of the linear system in (18).

1. The associated solution $(\mathbf{w}, \mathbf{z}) = (\mathbf{w}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon}), \mathbf{z}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon})) \geqq \mathbf{0}$ to the linear system in (18) is a basic solution.
2. A basis B is complementary if $|\{i, i+h\} \cap B| = 1$ for $i = 1, \ldots, h$.
3. A complementary basis B is feasible if the associated basic solution is feasible, i.e., $\mathbf{w}, \mathbf{z} \geqq \mathbf{0}$.
4. For a feasible complementary basis (FCB), the associated basic feasible solution $(\mathbf{w}, \mathbf{z})$ to (18) is called the basic feasible complementary solution (BFCS).

The reader is referred to [2] for a state-of-the art study on mpLCPs or to [9–12, 32] for theory and algorithms for LCPs. The solutions to mpLCP (18) are related to the efficient solution to $\text{MOQP}_1(\boldsymbol{\theta})$ in the following proposition.

Proposition 5 *Let $Q_i, i = 1, \ldots, \tilde{r}$, 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 (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 (17) for every $\boldsymbol{\theta} \in \Theta, \boldsymbol{\lambda} \in \Lambda', \boldsymbol{\epsilon} \in \mathcal{E}$. Therefore, the KKT necessary optimality conditions for (17), which assume the form of mpLCP (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 (18) if and only if $\mathbf{x}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon})$ is an optimal solution to (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 [30], $\mathbf{x} = \mathbf{x}(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\epsilon})$ is an efficient solution to $\text{MOQP}_1(\boldsymbol{\theta})$.

The mpLCP method is the first and only method to solve mpLCPs (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 [2]. 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 5, these solutions provide the efficient solutions to $\text{MOQP}_1(\boldsymbol{\theta})$.

4.2 The spLCP method

The theory for the spLCP method was developed by Väliaho [48] 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 2 below. Redefining $\Lambda' = \{\lambda \in \mathbb{R} : \lambda \in [0, 1]\}$ and applying (4) to (BOQP), the resulting spQP is

$$\begin{aligned} \min_{\mathbf{x}}\ & f(\mathbf{x}; \lambda) = \tfrac{1}{2}\mathbf{x}^T Q(\lambda)\mathbf{x} + \mathbf{p}(\lambda)^T \mathbf{x} \\ \text{s.t. } & \mathbf{x} \in \mathcal{X}, \end{aligned} \tag{19}$$

where $Q(\lambda) = \lambda Q_1 + (1 - \lambda)Q_2$ and $\mathbf{p}(\lambda) = \lambda \mathbf{p}_1 + (1 - \lambda)\mathbf{p}_2$.

Problem (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 [48].

4.2.1 Reformulation of spLCP

Given (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:

$$\begin{aligned} \mathbf{w} - \big(M + \lambda \Delta M\big)\mathbf{z} &= \mathbf{q} + \lambda \Delta\,\mathbf{q} \\ \mathbf{w}^T\,\mathbf{z} &= 0 \\ \mathbf{w}, \mathbf{z} &\geqq \mathbf{0}. \end{aligned} \tag{20}$$

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 (19) (and also BOQPs) that can be solved:

Assumption 2 *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 2(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.

4.2.2 Full Rank Factorization of Matrix ΔM

The method is equipped with matrix factorization needed to modify spLCP (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 4.2.6, a vector $\boldsymbol{\kappa} \in R^n$, which is computed below, is used. Let

$$\Delta\,\mathbf{q} = \Psi\boldsymbol{\kappa}, \tag{21}$$

or equivalently

$$\begin{bmatrix} \mathbf{p}_1 - \mathbf{p}_2 \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \Psi_1 \\ 0 \end{bmatrix} \boldsymbol{\kappa},$$

which is simplified to

$$\mathbf{p}_1 - \mathbf{p}_2 = \Psi_1\boldsymbol{\kappa}. \tag{22}$$

System (22) is solved for $\boldsymbol{\kappa}$ using the forward substitution method. Having computed Ψ and $\boldsymbol{\kappa}$, an augmented spLCP is constructed.

4.2.3 Initialization and Overview

Let $\lambda \in [0, 1]$ be fixed in spLCP (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:

$$\begin{aligned} \left(\begin{bmatrix} \mathscr{I}_1 & 0 \\ 0 & \mathscr{I}_2 \end{bmatrix} - \begin{bmatrix} M(\lambda) & -\Psi \\ \Psi^T & 0 \end{bmatrix} \right) \begin{bmatrix} \mathbf{v}_B \\ \mathbf{v}_{\boldsymbol{\kappa}} \end{bmatrix} &= \begin{bmatrix} \mathbf{q}(\lambda) \\ \boldsymbol{\kappa} \end{bmatrix} \\ \mathbf{v}_B &\geqq \mathbf{0} \\ \mathbf{v}_{\boldsymbol{\kappa}} &\quad \text{free}, \end{aligned} \tag{23}$$

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 (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 (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 (20) and (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 (23) but are considered dummy with no meaning in the solution to (20).

Since the goal is to solve (23) for $\mathbf{v}_B(\lambda)$ for a fixed $\lambda \in [0, 1]$, for the initial basis $B_0 = \{1, \ldots, h\}$ in (20), the entire problem (23) is saved in an initial Simplex-type Tableau $T_{B_0}(\lambda)$ as given in Table 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 1. Note that neither $\mathbb{E}'$ nor K contain the first column, $\mathfrak{g}_{\cdot 0}$. At the beginning of Phase I, the right hand side of the linear system in (23), $\mathbf{q}(\lambda)$, is stored in the first h rows of column $\mathfrak{g}_{\cdot 0}$. The last n rows of $\mathfrak{g}_{\cdot 0}$ contain the vector $\boldsymbol{\kappa}$ obtained by solving (21). At the end of both phases, the solution vector $\mathbf{v}_B(\lambda)$ is retrieved from the first h rows of column $\mathfrak{g}_{\cdot 0}$.

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.

Table 1: Initial Tableau $T_{B_0}(\lambda)$

	$\mathfrak{g}_{\cdot 0}$	$\mathfrak{g}_{\cdot 1}, \cdots, \mathfrak{g}_{\cdot h}$	$\mathfrak{g}_{\cdot (h+1)}, \cdots, \mathfrak{g}_{\cdot (h+n)}$	$\mathfrak{g}_{\cdot ((h+n)+1)}, \cdots, \mathfrak{g}_{\cdot (2h+n)}$	$\mathfrak{g}_{\cdot ((2h+n)+1)}, \cdots, \mathfrak{g}_{\cdot (2h+2n)}$
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

Definition 7 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 8 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 (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 (20) is given in Algorithm 1. 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 1: spLCP Method

```
Input  : Initial Tableau T_{B_0}(λ)
Output: Solution (z(λ), w(λ)) for each λ.
Set λ = 0
if T_{B_0}(λ)_{i,0} < 0 for at least one i ∈ {1, ..., h} then
    Goto Phase I (apply Algorithm 2)
    Goto Phase II (Algorithm 3)
else
    Goto Phase II (Algorithm 3)
```

4.2.4 Phase I

In Phase I, the Criss-Cross Method (given in Algorithm 2) is used to find an initial BFCS for spLCP (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 [14]. 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 (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 (20) has been found.

Algorithm 2: Criss-Cross Method

```
Input  : T_{B_0}(λ̄), B_0
Output: BFCS to spLCP (20) with λ = λ̄
if T_{B_0}(λ̄)_{i,0} ≥ 0 for all i = 1,...,h then
|  STOP, a BFCS has been found.
else
|  Set B = B_0.
|  while T_B(λ̄)_{i,0} ≤ 0 for at least one i ∈ {1,...,h} do
|  |  Let r = min{i ∈ B : T_B(λ̄)_{i,0} < 0}
|  |  if T_B(λ̄)_{r,r̄} < 0 then
|  |  |  make diagonal pivot, new basis B' = B \ {r} ∪ {r̄}
|  |  else if T_B(λ̄)_{r,r̄} = 0 then
|  |  |  Let k = min{j : T_B(λ̄)_{r,j} < 0 or T_B(λ̄)_{j,r} > 0}
|  |  |  if T_B(λ̄)_{r,k} × T_B(λ̄)_{k,r} < 0 then
|  |  |  |  make exchange pivot, new basis B' = B \ {r,k} ∪ {r̄,k̄}
|  |  |  else
|  |  |  |  STOP. LCP is infeasible
|  |  else
|  |  |  STOP. LCP is infeasible
|  Set B = B' and T_B(λ̄) = T_{B'}(λ̄)
```

4.2.5 Phase II

In Phase II, the parameter λ and basis B are consecutively updated. Let B be a complementary basis in (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 1.

For each $i = 1, \ldots, h$ and K, define the following auxiliary vector

$$\mathcal{B}_{i0} = \Big(T_B(\lambda)_{i,0}, (-1)^1 T_B(\lambda)_{i,K} T_B(\lambda)_{n,0}, (-1)^2 T_B(\lambda)_{i,K} T_B(\lambda)_{n,K} T_B(\lambda)_{n,0}, \ldots, (-1)^n T_B(\lambda)_{i,K} T_B(\lambda)_{n,K}^{n-1} T_B(\lambda)_{K,0} \Big), \tag{24}$$

where the terms in (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 9 Let $\boldsymbol{v} \in \mathbb{R}^h$. We say that

1. $\boldsymbol{v}$ is lexicographically positive (negative), $\boldsymbol{v} \succ \mathbf{0}$ ($\boldsymbol{v} \prec \mathbf{0}$), if its first nonzero component is positive (negative).
2. $\boldsymbol{v}$ is lexicographically nonnegative (nonpositive), $\boldsymbol{v} \succeq \mathbf{0}$ ($\boldsymbol{v} \preceq \mathbf{0}$), if $\boldsymbol{v} = \mathbf{0}$ or $\boldsymbol{v} \succ \mathbf{0}$ ($\boldsymbol{v} = \mathbf{0}$ or $\boldsymbol{v} \prec \mathbf{0}$).

Phase II is presented in Algorithm 3 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 4. Otherwise, the current complementary basis is updated using Algorithm 6.

Algorithm 3: Phase II

Input : Current Tableau $T_B(\lambda)$ s.t. $T_B(\lambda)_{i0} \geq \mathbf{0}$ for $i = 1, \ldots, h, B$
Output: Solution $(\mathbf{z}(\lambda), \mathbf{w}(\lambda))$ to spLCP (18)
while $\lambda \leq 1$ **do**
  if $\mathcal{B}_{i0} \succeq \mathbf{0}$ *for all* $i = 1, \ldots, h$ **then**
    Update λ (Algorithm 4)
  else
    Update B (Algorithm 6)

4.2.6 Updating Parameter λ

Parameter λ is updated in Algorithm 4. 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 (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 5. 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.

4.2.7 Updating the Initial Tableau when $\rho = \rho_1$

Algorithm 5 is designed to provide a new initial tableau when $\rho = \rho_1$ in Algorithm 4. 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 5.

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 $\mathcal{B}_{i0} \prec \mathbf{0}$ for at least one $i \in \{1, \ldots, h\}$ in Algorithm 3, then the basis B is not feasible for system (23) for the current value of λ. Then Algorithm 6 is run and the Criss-Cross Method (Algorithm 2) is invoked to find a FCB for the current value of λ.

Algorithm 4: Updating λ

Input : Current Tableau $T_B(\lambda)$ such that $\mathcal{B}_{i0} \succeq \mathbf{0}$ for all $i = 1, \ldots, h$
Output: New λ

1. **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 $\rho_1 = \min\{\alpha_1, \ldots, \alpha_h\}$
2. 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 3)
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 5. Go to Phase II (Algorithm 3)
  else
    Go to Phase II (Algorithm 3)
else
  Set $\lambda = \lambda + \rho_2$ and update the Tableau $T_{B_0}(\lambda)$. Go to Phase II (Algorithm 3)

Algorithm 5: Updating the initial tableau

Input : The initial Tableau $T_{B_0}(\lambda + \rho_1)$
Output: new Tableau $T_{B_0}(\lambda)$

1. Add $-\rho_1^{-1}\mathscr{I}$ to $T_{B_0}(\lambda)_{n,K}$
2. Perform the pivotal operation on all variables in the last n rows.
3. Multiply elements in the last n rows by $-\rho_1^{-1}$ and then columns $j \in K$ by ρ_1^{-1} of $T_{B_0}(\lambda)$.
4. Add $-\rho_1^{-1}\mathscr{I}$ to $T_B(\lambda)_{n,K}$.
5. Go to Phase II (Algorithm 3) with current Tableau $T_{B_0}(\lambda)$.

Algorithm 6: Updating B

Input : Current Tableau $T_{B_0}(\lambda)$ such that $\mathcal{B}_{i0} \prec \mathbf{0}$ for at least one $i \in \{1, \ldots, h\}$
Output: new Tableau $T_B(\lambda)$ such that $\mathcal{B}_{i0} \succeq \mathbf{0}$
while $\mathcal{B}_{i0} \prec \mathbf{0}$ **do**
  Run Criss-Cross Method (Algorithm 2)

4.3 Example

We illustrate the spLCP and mpLCP methods on the following BOQP:

$$\begin{aligned} \min \;& \left[\frac{1}{2}\begin{bmatrix}x_1 & x_2\end{bmatrix}\begin{bmatrix}6 & 0\\0 & 14\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix} + \begin{bmatrix}9 & -5\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}, \frac{1}{2}\begin{bmatrix}x_1 & x_2\end{bmatrix}\begin{bmatrix}2 & 0\\0 & 5\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix} + \begin{bmatrix}-1 & 1\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix}\right] \\ \text{s.t.} \;& \begin{bmatrix}3 & 5\end{bmatrix}\begin{bmatrix}x_1\\x_2\end{bmatrix} \le 15 \\ & x_1, x_2 \ge 0, \end{aligned} \tag{25}$$

and the associated spLCP:

$$\begin{aligned}\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} - \left(\begin{bmatrix} 2 & 0 & 3 \\ 0 & 5 & 5 \\ -3 & -5 & 0 \end{bmatrix} + \lambda \begin{bmatrix} 4 & 0 & 0 \\ 0 & 9 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right) \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} &= \begin{bmatrix} -1 \\ 1 \\ 15 \end{bmatrix} + \lambda \begin{bmatrix} 10 \\ -6 \\ 0 \end{bmatrix} \\ w_i z_i &= 0 \quad i = 1, 2, 3, \\ w_i, z_i &\geq 0 \quad i = 1, 2, 3, \end{aligned} \tag{26}$$

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 2. Four efficient solutions are computed, each of them for a specific value of $\lambda \in [0, 1]$. For comparison, Table 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 2: Efficient solutions to Example (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 3: Efficient solutions to Example (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 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 2. The code of the mpLCP method is available online [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 of the numerical experiments are collected in Table 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 2 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 4, 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 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.

Table 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

5 Parametric Quadratic Programs with Quadratic Constraints

An application of the $\boldsymbol{\epsilon}$-constraint formulation (5) causes MOQP($\boldsymbol{\theta}$) to assume the form of a parametric quadratically constrained quadratic program (mpQCQP) formulated as

$$
\begin{aligned}
&\min_{\mathbf{x}} f_i(\mathbf{x};\boldsymbol{\theta}) = \tfrac{1}{2}\mathbf{x}^T Q_i(\boldsymbol{\theta})\mathbf{x} + \mathbf{p}_i^T(\boldsymbol{\theta})\mathbf{x} + c_i(\boldsymbol{\theta}) \\
&\text{s.t. } \mathbf{x} \in \mathcal{X}(\boldsymbol{\theta},\boldsymbol{\epsilon}) = \{\mathbf{x} \in \mathbb{R}^n : \qquad\qquad (\text{QCQP}(\boldsymbol{\theta},\boldsymbol{\epsilon}))\\
&\quad f_j(\mathbf{x};\boldsymbol{\theta},\epsilon_j) = \tfrac{1}{2}\mathbf{x}^T Q_j(\boldsymbol{\theta})\mathbf{x} + \mathbf{p}_j^T(\boldsymbol{\theta})\mathbf{x} + c_j(\boldsymbol{\theta},\epsilon_j) \leq 0 \quad j = 1,\ldots,\tilde{r}, j \neq i,\\
&\quad \tilde{A}(\boldsymbol{\theta})\mathbf{x} \leqq \tilde{\boldsymbol{b}}(\boldsymbol{\theta},\boldsymbol{\epsilon}),\\
&\quad \mathbf{x} \geqq \mathbf{0}\}\\
&\quad \boldsymbol{\theta} \in \Theta, \boldsymbol{\epsilon} \in \mathcal{E} \subseteq \mathbb{R}^{r-1},
\end{aligned}
$$

where $i \in \{1,\ldots,\tilde{r}\}$, $\tilde{A} : \Theta \to \mathbb{R}^{\tilde{m}\times n}, \tilde{\boldsymbol{b}} : \Theta \times \mathcal{E} \to \mathbb{R}^{\tilde{m}}, \tilde{m} = m + r - \tilde{r}$. Recall that $\Theta \subseteq \mathbb{R}^{\kappa}$ is polyhedral as defined in Assumption 1. Scalarizing MOQP($\boldsymbol{\theta}$) into QCQP($\boldsymbol{\theta},\boldsymbol{\epsilon}$), the coefficients $c_j(\boldsymbol{\theta})$ have been redefined into $c_j(\boldsymbol{\theta},\epsilon_j) = c_j(\boldsymbol{\theta}) - \epsilon_j$ to account for the scalarizing parameter ϵ_j, $j = 1,\ldots,\tilde{r}, j \neq i$, and the linear inequality constraints have been modified as in (17) to include the $\boldsymbol{\epsilon}$-constraints of the form $\mathbf{p}_j^T(\boldsymbol{\theta})\mathbf{x} + c_j(\boldsymbol{\theta}) \leq \epsilon_j$ for $j = \tilde{r}+1,\ldots,r$, where $\boldsymbol{\epsilon} \in \mathcal{E} \subseteq \mathbb{R}^{r-1}$ is a new parameter introduced into the model so that the problem remains feasible.

Despite the fact that many algorithms have been developed to solve parametric nonlinear programs with parameters in various locations [42], we believe there is no algorithm that exactly solves mpQPs of type QCQP($\boldsymbol{\theta},\boldsymbol{\epsilon}$). A potential candidate could be the parametric quadratic approximation (mpQA) algorithm [31] which solves parametric nonlinear programs. Making use of quadratic approximation of the objective function and linear approximation of the constraints, this algorithm transforms the original problem into an mpQP that, however, does not allow parameters in the left-hand-side of the linear constraints. Parameters are included in the left-hand-side of the linear constraints in [53] but the functions only contain terms bilinear in $\mathbf{x}$ and parameters. Very recently, an algorithm for a class of mpQCQPs was proposed in [39], but it does not take into account parameters in the quadratic form of $\mathbf{x}$. Given the state-of-the art in parametric optimization, to solve QCQP($\boldsymbol{\theta}, \boldsymbol{\epsilon}$) we choose a parametric approximate simplex (mpAS) method [4] that relies on linear interpolation of all nonlinear functions. Since the numerical effectiveness of this method has not been examined, we intend to investigate the tradeoff between its simplicity and the quality of its obtained optimal solutions.

In addition to Assumption 1 that now holds for $\Theta \times \mathcal{E}$, we make the following assumptions about QCQP($\boldsymbol{\theta}, \boldsymbol{\epsilon}$) that are required by the AS method.

Assumption 3 *1. The dimension of the parameter space $\Theta \subseteq \mathbb{R}^\kappa$ is κ.*
2. The matrices $Q_i(\boldsymbol{\theta}), i = 1, \ldots \tilde{r}$, are positive semi-definite for all $\boldsymbol{\theta} \in \Theta$.
3. The elements in $Q_i, \boldsymbol{p}_i, c_i, i = 1, \ldots, \tilde{r}, \tilde{A}$, and $\tilde{\boldsymbol{b}}$ are convex functions of $\boldsymbol{\theta}$.

Since the last assumption is required for computation of the error between the true value of the objective function and its approximation, it might be relaxed if that error was defined differently. Note that the constraints are all affine with respect to $\boldsymbol{\epsilon}$ due to formulation (5), so there are no assumptions with respect to $\boldsymbol{\epsilon}$ other than feasibility.

5.1 The mpAS method

The mpAS method is given in Algorithm 8. The combined parameter space is redefined as $\Xi := \Theta \times \mathcal{E}$ such that $(\boldsymbol{\theta}, \boldsymbol{\epsilon}) = \boldsymbol{\xi} \in \Xi$. Using processing methods in Multiparametric Toolbox [26], Ξ is first partitioned into a set of simplices Ξ_k, $k = 1, \ldots, s$. Note s is finite because Θ and $\mathcal{E}$ are assumed to be polyhedral. For a given simplex, Ξ_k, let $\boldsymbol{\zeta}_j^k$, $j = 1, \ldots, s_k$, be its vertices. For each $\boldsymbol{\zeta}_j^k$, QCQP($\boldsymbol{\theta}, \boldsymbol{\epsilon}$) is solved to get objective values $f_i(\boldsymbol{\zeta}_j^k)$ and vectors $\mathbf{x}(\zeta_j^k)$. These objective values and vectors are then used to compute linear interpolations of the optimal objective function, $\bar{f}(\boldsymbol{\theta}, \boldsymbol{\epsilon})$, and of the optimal solution function, $\bar{\mathbf{x}}(\boldsymbol{\theta}, \boldsymbol{\epsilon})$ to QCQP($\boldsymbol{\theta}, \boldsymbol{\epsilon}$). Treating $\mathbf{x}$ and $\boldsymbol{\xi}$ as variables, the maximum error between the approximation and the actual objective function is computed. If this error is too large or the split limit has not been reached, Ξ_k is split into smaller simplices and this process is repeated on each new simplex. At termination, the parameter space Ξ has been partitioned into simplices called invariancy regions and the associated approximate optimal objective function has been constructed as a piecewise linear function of the parameters. The approximate optimal solution functions are also available.

The mpAS algorithm proposed in [4] assumes that the program being solved is jointly convex in $\mathbf{x}, \boldsymbol{\theta}, \boldsymbol{\epsilon}$. In problem generation, we settled for mere biconvexity due to the scaling problems from forcing joint convexity. However, this relaxation does not adversely affect the performance of the algorithm.

5.2 Numerical Results

Having not found significant benchmarking of the mpAS algorithm in the literature, we make our own to gauge general effectiveness. We have implemented mpAS in MATLAB building off of code found in [33] and applied it to randomly generated convex QCQPs with one and two parameters. The tests have been run on a Dell Inspiron 3153 with a 128 GB SSD (8 GB free), 8 GB RAM, running an Intel Core i3-6100U 2.30 GHz processor with two cores and four logical processors running Windows 10 Enterprise.

In both the results here and the applications in Section 6, we only split the parameter spaces five times in order to maintain useful visual distinctions in the invariancy regions. Table 5 contains the mean, median,

Algorithm 7: Multiparametric Approximate Simplex Algorithm

Input : The initial polyhedron, $\Xi = \Theta \times \mathcal{E}$ over which to solve QCQP($\boldsymbol{\theta}, \boldsymbol{\epsilon}$), error tolerance, split limit
Output: piecewise function $\mathbf{x}^K(\boldsymbol{\xi})$ for some terminal $K \in \mathbb{N}$, and piecewise optimal function $f_i(\mathbf{x}^K; \boldsymbol{\xi})$
Step 0: Partition parameter space Ξ into a set of simplices Ξ_t, $t = 1, \ldots, s$, and set desirable tolerance and split limit.
Step t: for each $t = 1, \ldots, s$
For each vertex $\boldsymbol{\zeta}^k$ of Ξ_t, compute vectors $\mathbf{x}(\boldsymbol{\zeta}^k)$ and objective values $f_i(\boldsymbol{\zeta}^k)$.
Compute linear interpolations $\bar{f}(\boldsymbol{\xi})$ and $\bar{\mathbf{x}}(\boldsymbol{\xi})$.
Compute error.
if *error within tolerance or split limit reached* **then**
| stop
else
| Split Ξ_i into smaller simplices, which add to the set $\{1, \ldots, s\}$.

and standard deviations of the time and approximation error for a set of single-objective parametric QCQPs which take the form of problem QCQP($\boldsymbol{\theta}, \boldsymbol{\epsilon}$) with $r = 2$, $\tilde{r} = 2$, $m = 1$, $\kappa \in \{1, 2\}$, $\mathcal{E} = \{0\}$. For $\kappa = 2$, the quadratic constraint and objective function are affine in $\boldsymbol{\theta}$. Note that $\epsilon \in \mathcal{E} = \{0\}$, so we henceforth refer only to $\boldsymbol{\theta}$.

Two MATLAB functions are utilized: *randn* randomly generates the coefficient matrices and vectors using a normal distribution, and *fmincon* performs the minimizations at the corresponding steps of Algorithm 7 using the SQP method with the default number of iterations. In addition, Multi-Parametric Toolbox (MPT) [26] is utilized to construct the simplices.

The generation of each instance is extended from [15]. Generating the quadratic constraint, $\mathbf{x}^T Q(\boldsymbol{\theta})\mathbf{x} + \mathbf{p}^T(\boldsymbol{\theta})\mathbf{x} + \boldsymbol{c}(\boldsymbol{\theta}) \leq \boldsymbol{b}(\boldsymbol{\theta})$, such that it is guaranteed to be feasible is as follows. If we assume functions of $\theta \in [0, 1]$ are affine in θ, then

$$\boldsymbol{c}(\theta) := \boldsymbol{c}^1 + \boldsymbol{c}^2\theta$$
$$\mathbf{p}(\theta) := \mathbf{p}^1 + \mathbf{p}^2\theta$$
$$Q(\theta) := Q^1 + Q^2\theta$$

We randomly generate $\mathbf{p}^1, \mathbf{p}^2 \in \mathbb{R}^n$ and $\boldsymbol{c}^1, \boldsymbol{c}^2 \in \mathbb{R}$. To choose the locations of θ in $\mathbf{p}(\theta)$, we generate a 0-1 matrix $R_{\mathbf{p}} \in \mathbb{R}^{n \times n}$ and $r_{\boldsymbol{c}} \in \{0, 1\}$. Then $\mathbf{p}(\theta) := \mathbf{p}^1 + \mathbf{p}^2 R_{\mathbf{p}}\theta$ and $\boldsymbol{c}(\theta) := \boldsymbol{c}^1 + \boldsymbol{c}^2 r_{\boldsymbol{c}}\theta$ are randomly chosen. Guaranteeing the positive semi-definite nature of $Q(\theta)$ proceeds as follows. Let $P(\theta) := P_1 + P_2 R_P \theta$ with $P_1, P_2 \in \mathbb{R}^{n \times n}$ randomly generated in the same manner as for $\mathbf{p}(\theta)$ and $\boldsymbol{c}(\theta)$, with $R_P \in \mathbb{R}^{n \times n}$ being a randomly generated 0-1 matrix like $R_{\mathbf{p}}$. Then $Q(\theta) = P^T(\theta)P(\theta)$ is guaranteed to be positive semi-definite, but is not affine in θ. Therefore for $\kappa = 2$, we choose to guarantee the affine nature by replacing all appearances of θ^2 with a second parameter $\theta_2 \in [0, 1]$. The result is that $Q(\theta_1, \theta_2) = P^T(\theta_1)P(\theta_1)$, with θ_2 replacing θ_1^2, is now PSD and affine in $\boldsymbol{\theta} = (\theta_1, \theta_2)$. To ensure feasibility of the quadratic constraint, we select $\bar{\boldsymbol{b}}$ to be the vector of all ones and define $\boldsymbol{b}(\theta) := \bar{\boldsymbol{b}}^T Q(\boldsymbol{\theta})\bar{\boldsymbol{b}} + \mathbf{p}^T(\boldsymbol{\theta})\bar{\boldsymbol{b}} + \boldsymbol{c}(\boldsymbol{\theta})$. The linear constraint and objective function are generated in the same manner, with the linear constraint using the same chosen $\bar{\boldsymbol{b}}$.

For $\kappa = 2$, poorly scaled instances were sometimes generated when data came from the same distribution (standard normal). To eliminate such issues, the size of the parameter space was altered for each n. We let $\Theta = [0, 1] \times [0, 1]$ for $n = 2$, $\Theta = [0.1, 0.9] \times [0.1, 0.9]$, for $n = 3$, and $\Theta = [0.2, 0.6] \times [0.2, 0.6]$ for $n \in \{4, 5\}$. The downside to this approach is the possibility of losing the convexity as required in Assumption 3.2. However, this did not affect the results.

In Table 5, we observe that, as expected, the time increases both as the number of variables grows and as the number of parameters grows, but more so increasing going from $\kappa = 1$ to $\kappa = 2$. For two parameters ($\kappa = 2$), we observe that some instances extend over large function values; this results in misleadingly large error values. To account for this, we divide the function error by the difference of the largest and smallest minimum objective values. Because we do this after applying the mpAS algorithm with a split limit of five, the errors for objective values which do not change much over the parameter space increase, while the errors for more variant objective values decrease. The error is stable as the number of variables increases, but it increases substantially going from $\kappa = 1$ to $\kappa = 2$, even with the error normalization and the limiting of the $\kappa = 2$ case to being affine in $\boldsymbol{\theta}$. We consider the error values to be small enough to warrant applying mpAS to mpMOQPs to obtain insight into its application context that will be complementary to that obtained with the mpLCP method.

Table 5: Statistics for the CPU times (in seconds) and objective function errors for parametric QCQPs solved with the mpAS method.

n	no. of instances	statistics	$\kappa = 1$		$\kappa = 2$	
			time	error	time	error
2	10	mean	0.319	0.168	0.332	0.122
		median	0.231	0.025	0.393	0.013
		std. dev.	0.238	0.437	0.198	0.224
3	10	mean	0.552	0.041	1.034	0.244
		median	0.439	0.026	0.562	0.174
		std. dev.	0.550	0.042	1.219	0.283
4	10	mean	1.110	0.164	2.785	0.325
		median	0.861	0.055	1.579	0.363
		std. dev.	0.905	0.272	2.839	0.239
5	10	mean	1.271	0.126	1.093	0.355
		median	1.075	0.074	0.585	0.275
		std. dev.	0.791	0.125	1.216	0.325

6 Applications

In this section we apply the scalarizations presented in Section 3 and the algorithms examined in Sections 4 and 5 to specific mpMOQPs resulting from applications in statistics and portfolio optimization.

6.1 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

$$\hat{\mathbf{y}} = \Phi \mathbf{x}, \tag{27}$$

where $\Phi = \begin{bmatrix} \boldsymbol{\phi_1} \; \boldsymbol{\phi_2} \ldots \boldsymbol{\phi_n} \end{bmatrix} \in \mathbb{R}^{k \times n}$ 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

$$\min_{\mathbf{x} \in \mathbb{R}^n} \; f_1(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T Q \mathbf{x} + \mathbf{p}^T \mathbf{x} + c, \tag{28}$$

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 (27). Because model (27) performs poorly in both prediction and interpretation when the estimated coefficients are computed from (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 $\mathbf{x}$ [54]. This leads to the elastic net problem.

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{R}^n}\ f_{\text{elastic-net}}(\mathbf{x};\alpha,\beta) = \tfrac{1}{2}\mathbf{x}^TQ\mathbf{x} + \mathbf{p}^T\mathbf{x} + c + \tfrac{\alpha}{2}\mathbf{x}^T\mathbf{x} + \beta\textstyle\sum_{i=1}^n |x_i| \\ &\qquad\qquad \alpha,\beta \geq 0, \end{aligned} \tag{29}$$

where α, β play the role of modeling parameters (in agreement with the terminology introduced in Section 2). Because these parameters strongly affect the performance of model (27), they require tuning.

In [7], 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)

$$\min_{\mathbf{x}\in\mathbb{R}^n}\ [f_1(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^TQ\mathbf{x} + \mathbf{p}^T\mathbf{x} + c,\ f_2(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T\mathbf{x},\ f_3(\mathbf{x}) = \textstyle\sum_{i=1}^n |x_i|], \tag{30}$$

for which (29) is the associated weighted-sum SOP. Since the optimization over the efficient set remains challenging [50], 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 (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

$$\varPhi = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 3 & 2 \\ 1 & 2 & 2 \\ 1 & 4 & 5 \\ 1 & 3.5 & 2.5 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 0.55 \\ 0.623 \\ 0.587 \\ 0.569 \\ 0.758 \end{bmatrix}, \tag{31}$$

and by solving (30) in three ways.

6.1.1 The weighted-sum SOP

Applying the weighted-sum (4) to (30) yields the following mpQP:

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{R}^3}\ \lambda_1(\tfrac{1}{2}\mathbf{x}^TQ\mathbf{x} + \mathbf{p}^T\mathbf{x} + c) + \tfrac{\lambda_2}{2}\mathbf{x}^T\mathbf{x} + (1-\lambda_1-\lambda_2)\textstyle\sum_{i=1}^n |x_i| \\ &\lambda \in \varLambda', \end{aligned} \tag{32}$$

where $\varLambda' = \{\boldsymbol{\lambda} \in \mathbb{R}^2 : \lambda_1, \lambda_2 \geq 0, \lambda_1 + \lambda_2 \leq 1\}$. The problem is reformulated to make all variables nonnegative which at the same time transforms f_3 into a linear function. Having solved (30) with the mpLCP method, we obtain a partition of $\varLambda'$ into 6 invariancy regions (IR) and the optimal solution functions $\hat{x}_i(\lambda_1, \lambda_2)$, $i = 1, 2, 3$ to (32) in each region that are all listed in Table 7. To return to the original parameters, we substitute

$$\lambda_1 = \frac{1}{1+\alpha+\beta} \quad \text{and} \quad \lambda_2 = \frac{\alpha}{1+\alpha+\beta} \tag{33}$$

in Table 7 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 8. The partitions of both parameter spaces are depicted in Figure 1a and 1b. For all $\boldsymbol{\lambda} \in \varLambda' \setminus \{(0,0)\}$, by Cor. 1 in [30], all optimal solutions to (32) are efficient to (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 (30) is depicted in Figure 2, while Figure 3a shows the points $(\alpha, \beta, \hat{f}_1(\alpha, \beta)) = (\alpha, \beta, f_1(\hat{\mathbf{x}}(\alpha, \beta))$ for (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 (27) is available in the parametric form

$$\hat{\mathbf{y}} = \Phi\, \hat{\mathbf{x}}(\alpha,\beta)$$

where $\hat{\mathbf{x}}(\alpha,\beta) = \left[\hat{x}_1(\alpha,\beta)\ \hat{x}_2(\alpha,\beta)\ \hat{x}_3(\alpha,\beta)\right]^T$ is the efficient solution to (30) for $\alpha, \beta \geq 0$.

The solutions $\hat{\mathbf{x}}(\alpha,\beta)$ listed in Table 8 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 [7].

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 [54].

For example, with two nonzero components in the efficient solution, IR_3 bears further investigation. Note that for $\bar{\alpha} = 1$ and $\bar{\beta} = 0.3$, $\hat{\mathbf{x}} = \hat{\mathbf{x}}(1, 0.3) = \left[0.028\ 0.181\ 0\right]^T$ is in IR_3 with the minSE of $f_1(\hat{\mathbf{x}}) = 0.332$. Consider, now, IR_3 with $\bar{\alpha} = 1$. Then $\hat{\mathbf{x}} = \hat{\mathbf{x}}(1,\beta) = \left[0.17 - 0.472\beta\ 0.126\beta + 0.143\ 0\right]^T$ 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 $f_1(\hat{\mathbf{x}}(\bar{\alpha},\hat{\beta})) = 0.083$.

6.1.2 The $\boldsymbol{\epsilon}$-constraint SOP

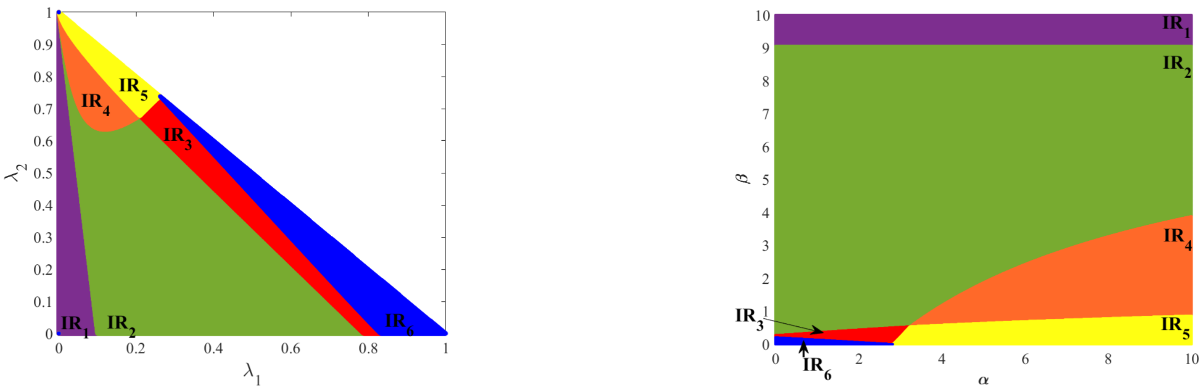
The $\boldsymbol{\epsilon}$-constraint formulation may be more attractive to the DM than the weighted-sum (32) because the squared error, as the primary criterion, is directly minimized while the values of the secondary criteria are controlled by the scalarization parameters ϵ_2 and ϵ_3.

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^3}\ & \tfrac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{p}^T\mathbf{x} + c \\ \text{s.t.}\ & \tfrac{1}{2}\mathbf{x}^T\mathbf{x} \leq \epsilon_2 \\ & \sum_{i=1}^{3} |x_i| \leq \epsilon_3 \\ & \boldsymbol{\epsilon} \in \mathcal{E}. \end{aligned} \tag{34}$$

By computing $\mathbf{x}^1 = \mathrm{argmin}_{\mathbf{x}}\, f_1(\mathbf{x})$, we establish $\mathcal{E} = [0, f_2(\mathbf{x}^1) = 0.186] \times [0, f_3(\mathbf{x}^1) = 0.601]$. We solve (34) with the mpAS method and the obtained approximate optimal solutions, $\tilde{\mathbf{x}}(\boldsymbol{\epsilon})$, by Prop. 4.3 in [16], are approximate weakly efficient to (30). In Figure 3b, the approximate minSE, $\tilde{f}_1(\epsilon_2,\epsilon_3)$, is depicted, along with the Pareto points $(0.194, 0.605, 0.012)$, $(0.016, 0.203, 0.207)$, $(0.038, 0.195, 0.116)$, and $(0, 0, 1.933)$, labeled A, B, C, and D, respectively. Placing these Pareto points also in Figure 2 allows one to see that the approximation works well. Note that point A is beyond the unconstrained minimum of $f_1(\mathbf{x})$ and thus is not located on the weak Pareto set computed by applying the mpAS method to (34). The optimal solutions are available for every region in the partition.

Consider the region containing $\bar{\boldsymbol{\epsilon}} = (0.119, 0.506)$,

$$IR = \left\{ \boldsymbol{\epsilon} \in \mathcal{E} : \begin{bmatrix} 0.853 & -0.502 \\ -0.709 & 0.655 \\ -0.940 & 0.334 \end{bmatrix} \begin{bmatrix} \epsilon_2 \\ \epsilon_3 \end{bmatrix} \leq \begin{bmatrix} -0.144 \\ 0.262 \\ 0.072 \end{bmatrix} \right\}.$$



(a) Parameters $\lambda_1, \lambda_2 \geq 0, \lambda_1 + \lambda_2 \leq 1$.

(b) Parameters $\alpha, \beta \in [0, 10]$.

Fig. 1: Partition of the parameter space into six invariancy regions for elastic net problem (30) obtained by the mpLCP method on problem (32).

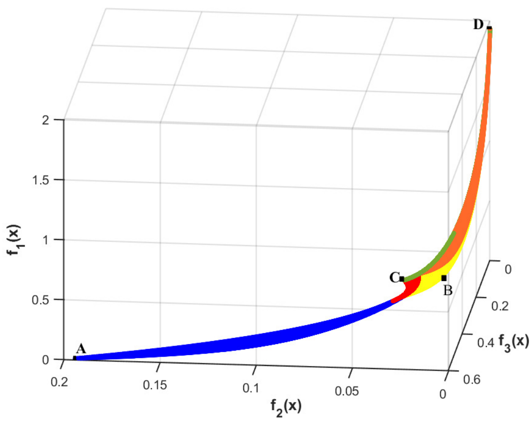


Fig. 2: Pareto set for elastic net problem (30) obtained by the mpLCP method on problem (32).

Then the optimal solution on this region is

$$\tilde{\mathbf{x}}(\boldsymbol{\epsilon}) = \begin{bmatrix} 1.539\epsilon_2 + 0.267\epsilon_3 - 0.041 \\ -1.002\epsilon_2 + 0.320\epsilon_3 + 0.123 \\ 0.537\epsilon_2 - 0.413\epsilon_3 + 0.083 \end{bmatrix}$$

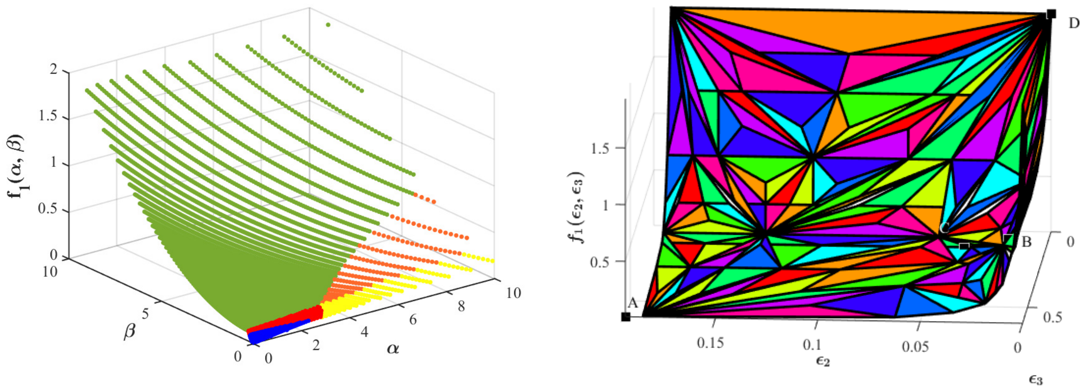
with the approximate minSE function $\tilde{f}_1(\boldsymbol{\epsilon}) = 0.188\epsilon_2 - 0.249\epsilon_3 + 0.126$. Note that $f_2(\bar{\boldsymbol{\epsilon}}) = 0.094$ and $f_3(\bar{\boldsymbol{\epsilon}}) = 0.456$, which result in the $\boldsymbol{\epsilon}$-constraints being approximately active.

According to [35], if the $\boldsymbol{\epsilon}$-constraints are active at optimality of (34),

$$\frac{\partial \hat{f}_1(\boldsymbol{\epsilon})}{\partial \epsilon_2} = -\frac{\lambda_2}{\lambda_1}, \qquad \frac{\partial \hat{f}_1(\boldsymbol{\epsilon})}{\partial \epsilon_3} = -\frac{\lambda_3}{\lambda_1} = -\frac{(1 - \lambda_1 - \lambda_2)}{\lambda_1},$$

where (λ_1, λ_2) is established in (32). Using (33) and the solutions $\tilde{\mathbf{x}}(\boldsymbol{\epsilon})$, we can approximate values of the coefficients α and β

$$\tilde{\alpha}(\boldsymbol{\epsilon}) = -\frac{\partial f_1(\tilde{\mathbf{x}}(\boldsymbol{\epsilon}))}{\partial \epsilon_2}, \quad \tilde{\beta}(\boldsymbol{\epsilon}) = -\frac{\partial f_1(\tilde{\mathbf{x}}(\boldsymbol{\epsilon}))}{\partial \epsilon_3}$$



(a) minSE computed by the mpLCP method on problem (32) .

(b) Approximate minSE computed by the mpAS method on problem (34).

Fig. 3: The minSE for elastic net problem (30)

and calculate

$$f_1(\tilde{\mathbf{x}}) = 0.968\epsilon_2^2 - 0.812\epsilon_2\epsilon_3 + 0.129\epsilon_2 + 0.805\epsilon_3^2 - 0.817\epsilon_3 + 0.245.$$

Then

$$\tilde{\alpha}(\boldsymbol{\epsilon}) = -\left(2 \cdot 0.968\epsilon_2 - 0.812\epsilon_3 + 0.129\right),$$

and

$$\tilde{\beta}(\boldsymbol{\epsilon}) = -\left(-0.812\epsilon_2 + 2 \cdot 0.805\epsilon_3 - 0.817\right).$$

Hence

$$\tilde{\alpha}(\bar{\boldsymbol{\epsilon}}) = 0.052, \quad \tilde{\beta}(\bar{\boldsymbol{\epsilon}}) = 0.099. \tag{35}$$

Observe in Figure 3b that α and β are the negatives of the slopes of the minSE in Figure 3a. The solution associated with parameters α and β in (35) is in IR_6 of Table 8.

6.1.3 The reduced $\boldsymbol{\epsilon}$-constraint SOP

Alternatively, the two secondary criteria may be combined into one constraint in the reduced $\boldsymbol{\epsilon}$-constraint formulation for (30), which results in the following mpQCQP

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^3} \ & \tfrac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{p}^T\mathbf{x} + c \\ \text{s.t. } & \tfrac{1}{2}\lambda\mathbf{x}^T\mathbf{x} + (1-\lambda)\sum_{i=1}^{3}|x_i| \le \epsilon \\ & \lambda \in [0,1], \epsilon \in \mathcal{E}. \end{aligned} \tag{36}$$

Problem (36) is presented in [54] and referred to as the naive elastic net problem, while the function $\frac{\lambda}{2}\lambda\mathbf{x}^T\mathbf{x} + (1-\lambda)\sum_{i=1}^{3}|x_i|$ is called the elastic net penalty. The authors of [54] solve this problem only for fixed values of the parameters λ and ϵ, while the approach presented here obtains optimal solutions for the entire parameter space.

Using $\mathbf{x}^1$, the unconstrained minimizer of f_1, we select $\mathcal{E} = [0, 0.6]$, as $\epsilon > 0.6$ results in f_1 attaining its global minimum, which is not of interest. Solving (36) with the mpAS method, we obtain the optimal solutions that, by Prop. 3, are weakly efficient to (30). The approximate minSE as a function of λ and ϵ is depicted in Figure 4. Note that ϵ constrains the λ-weighing of f_2 and f_3.

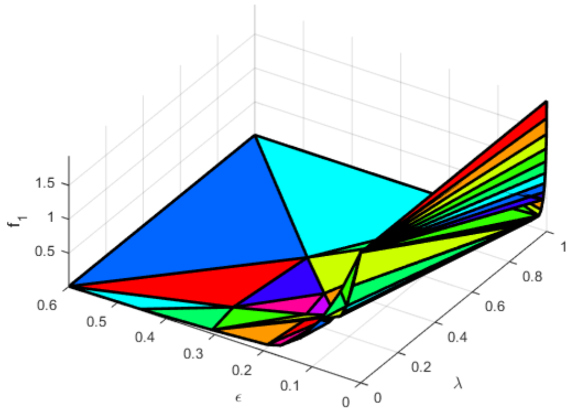


Fig. 4: Approximate minSE for elastic net problem (30) obtained by the mpAS method on problem (36).

As an example, consider the region containing $\bar{\lambda} = 0.955$, $\bar{\epsilon} = 0.087$,

$$IR = \left\{ (\lambda, \epsilon) : \begin{bmatrix} 0.707 & 0 \\ -0.105 & 0.994 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \epsilon \end{bmatrix} \leq \begin{bmatrix} 0.707 \\ 0.0158 \\ -0.452 \end{bmatrix} \right\}.$$

Then the optimal solution on this region is

$$\tilde{\mathbf{x}}(\lambda, \epsilon) = \begin{bmatrix} 0.704\lambda + 2.235\epsilon - 0.682 \\ -0.096\lambda + 0.203\epsilon + 0.243 \\ -0.134\lambda - 0.892\epsilon + 0.169 \end{bmatrix}$$

with the approximate minSE function $\tilde{f}_1(\lambda, \epsilon) = -0.092\lambda - 0.880\epsilon + 0.215$. Using the reduced-$\epsilon$-constraint formulation, the DM can set $\bar{\lambda} = 0.955$ and $\bar{\epsilon} = 0.087$ to obtain the coefficients for fitting a curve to the data being $\tilde{\mathbf{x}}(\bar{\lambda}, \bar{\epsilon}) = (0.185, 0.169, -0.037)$. Based on how the approximated minSE value, $\tilde{f}_1(\bar{\lambda}, \bar{\epsilon}) = 0.04$, compares to other minSE values for other values of λ and ϵ, the DM can decide whether the obtained coefficients are appropriate.

6.2 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,

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^3} \ & \left[f_1(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T Q_1 \mathbf{x}, f_2(\mathbf{x}) = \tfrac{1}{2}\mathbf{x}^T Q_2 \mathbf{x}, f_3(\mathbf{x}; \theta) = -\boldsymbol{p}_3(\theta)^T \mathbf{x} \right] \\ \text{s.t.} \ & \mathbf{1}^T \mathbf{x} = 1 \\ & \mathbf{x} \geqq \mathbf{0}, \end{aligned} \tag{37}$$

and with the following data

$$Q_1 = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 2 & 0 \\ -1 & 0 & 2.5 \end{bmatrix}, \qquad Q_2 = \begin{bmatrix} 3 & -1 & 0 \\ -1 & 4 & 1 \\ 0 & 1 & 3.5 \end{bmatrix}, \qquad \boldsymbol{p}_3(\theta) = \begin{bmatrix} -13.5 \\ 20 \\ \theta \end{bmatrix} \text{ for } \theta \in [15, 17].$$

A nonparametric version of this problem is solved in [30]. 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. 3 and 4.

6.2.1 The modified-hybrid SOP

Problem (37) is reformulated with the modified hybrid method (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 (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:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^3} \quad & \tfrac{1}{2}\mathbf{x}^T Q(\lambda)\mathbf{x} \\ \text{s.t.} \quad & \tilde{A}(\theta)\mathbf{x} \leqq \tilde{\mathbf{b}}(\epsilon) \\ & \mathbf{x} \geqq \mathbf{0} \\ & \theta\in\Theta=[15,17], \lambda\in\Lambda'=[0,1], \epsilon\in\mathcal{E}=[-20,13.5], \end{aligned} \tag{38}$$

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 $\epsilon^{max} = \max\{-\mathbf{p}_3(\theta)^T\mathbf{x} : \mathbf{1}^T\mathbf{x}=1, \mathbf{x}\geqq\mathbf{0}, \theta\in[15,17]\} = 13.5$, we have $\mathcal{E}^{nor} = [0,1]$.

The mpLCP method Problem (38) 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 5 and listed in Table 9 along with their respective optimal solution functions $\hat{x}_i(\theta,\lambda,\epsilon), i=1,2,3$ that, by Proposition 5, are efficient to (38). 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 6 and the coordinates of four points are reported in Table 6 for comparison. Note that point A in IR_3 remains Pareto for each value of θ.

Table 6: Coordinates of four Pareto points in Figure 6 for portfolio problem (37)

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

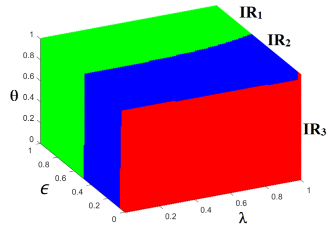
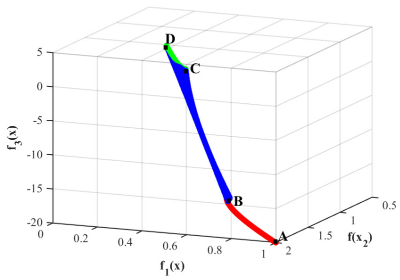
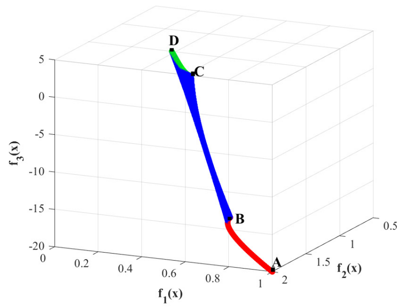


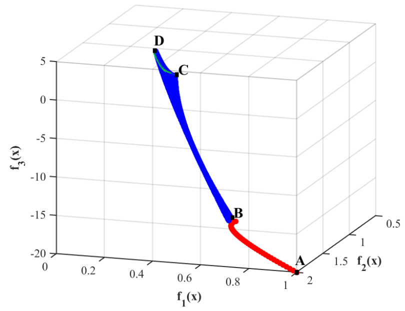
Fig. 5: Partition of the parameter space for portfolio problem (37) obtained with the mpLCP method on problem (38).



(a) $\theta = 15$



(b) $\theta = 16$



(c) $\theta = 17$

Fig. 6: Pareto sets $\mathcal{Y}_P(15), \mathcal{Y}_P(16), \mathcal{Y}_P(17) \subset \mathcal{Y}_P$ for portfolio problem (37) obtained with the mpLCP method on problem (38).

A similar analysis to that in [30] 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 (38), 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 (38), being the minimum weighted risk (MWR), $\hat{\sigma}(\theta, \lambda, \epsilon)$, can be obtained in this subset of the parameter space [30]

$$\hat{\sigma}(\theta, \lambda, \epsilon) = \tfrac{1}{2} \begin{bmatrix} \epsilon & 1 \end{bmatrix} \Psi(\theta, \lambda)^{-1} \begin{bmatrix} \epsilon \\ 1 \end{bmatrix}, \tag{39}$$

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 (39), the MWR function is available for normalized parameters,

$$\hat{\sigma}(\theta^{nor}, \lambda, \epsilon^{nor}) = \frac{1}{2} \begin{bmatrix} 33.5\epsilon^{nor} - 20 & 1 \end{bmatrix} \Psi(\theta^{nor}, \lambda)^{-1} \begin{bmatrix} 33.5\epsilon^{nor} - 20 \\ 1 \end{bmatrix}. \tag{40}$$

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.

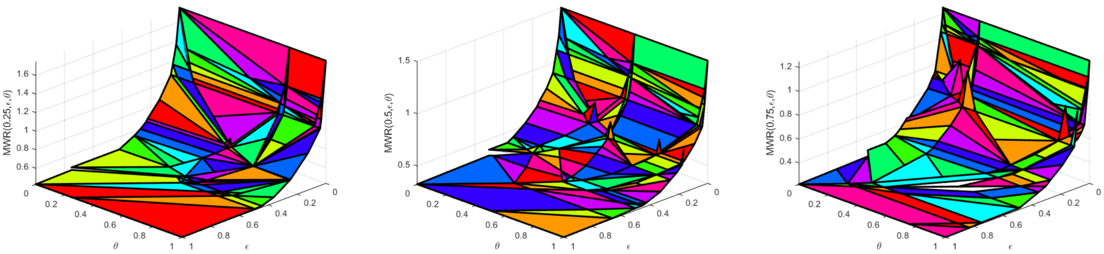
Corollary 2 *Let the matrices Q_1, Q_2 be positive definite and the vectors $\boldsymbol{p}_3(\theta)$ and $\mathbf{1}$ be linearly independent for all $\theta \in \Theta$ for TOPOP (37). At optimality of problem (38), let the inequality ϵ-constraint be active and all the nonnegativity constraints be 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 (39) 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 2 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.

The mpAS method We also solve (38) with the mpAS method to compare its performance against the mpLCP method. Since the mpAS method does not require equality constraints to be relaxed, we solve the following problem:

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^3} \; & \frac{1}{2}\mathbf{x}^T Q(\lambda)\mathbf{x} \\ \text{s.t. } & 13.5x_1 - 20x_2 - \theta x_3 \leq \epsilon \\ & \mathbf{1}^T\mathbf{x} = 1 \\ & \mathbf{x} \geqq \mathbf{0} \\ & \theta \in \Theta, \ \lambda \in [0, 1], \ \epsilon \in \mathcal{E}. \end{aligned} \tag{41}$$



(a) $\lambda = 0.25$. (b) $\lambda = 0.5$. (c) $\lambda = 0.75$.

Fig. 7: Approximate MWR obtained with the mpAS method on problem (38).

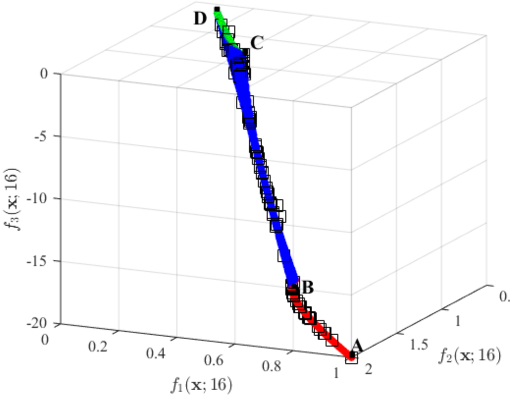
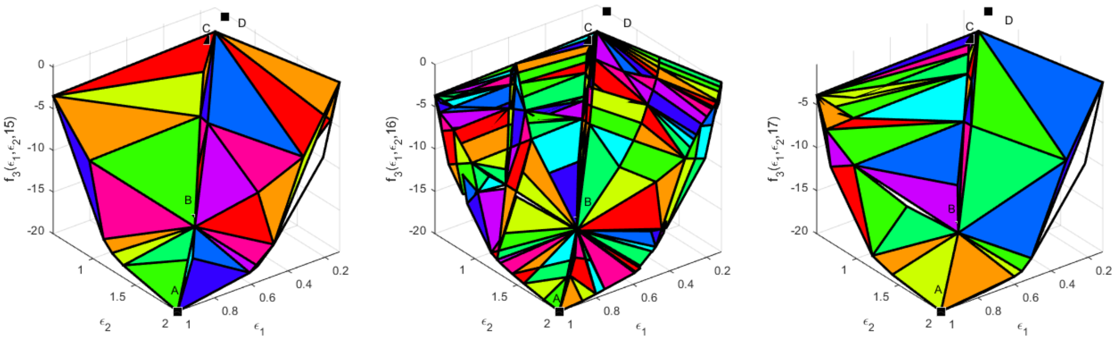


Fig. 8: Approximate Pareto points for (37) and $\theta = 16$ computed by the mpAS method on problem (41).

where $Q(\lambda)$, Θ, and $\mathcal{E}$ are defined in (38), with Θ and $\mathcal{E}$ normalized.

To further demonstrate Corollary 2, $\bar{\lambda} \in \{0.25, 0.5, 0.75\}$ is taken and the MWR is plotted against ϵ^{nor} and θ^{nor} in Figure 7. Notice that the values of the MWR do not change over $\theta^{nor} \in [0, 1]$.

Due to the fact that the approximations of $\mathbf{x}$ are in terms of λ, ϵ, and θ, to readily compare the solution of problem (38) to the solution of problem (41), we consider the vertices, $(\bar{\lambda}, \bar{\epsilon})$, of the invariancy regions containing $\bar{\theta} = 16$. We then graph the return $f_3(\mathbf{x}(\bar{\lambda}, \bar{\epsilon}, 16))$ against the risks $f_1(\mathbf{x}(\bar{\lambda}, \bar{\epsilon}, 16))$ and $f_2(\mathbf{x}(\bar{\lambda}, \bar{\epsilon}, 16))$, and overlay the points on Figure 6b. The results are in Figure 8. Note that, while the mpLCP method provides a complete description of the Pareto set, the mpAS method is capable of providing a good approximations on its own.



(a) $\theta = 15$. (b) $\theta = 16$. (c) $\theta = 17$.

Fig. 9: Approximate weak Pareto sets for (37) obtained by the mpAS method on problem (42).

6.2.2 The $\boldsymbol{\epsilon}$*-constraint SOP*

The $\boldsymbol{\epsilon}$-constraint formulation allows us to treat the return maximization as the primary objective while the values of each risk are controlled parametrically.

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^3} \ & -\boldsymbol{p}_3(\theta)^T\mathbf{x} \\ \text{s.t.} \ & \tfrac{1}{2}\mathbf{x}^T Q_1 \mathbf{x} \leq \epsilon_1 \\ & \tfrac{1}{2}\mathbf{x}^T Q_2 \mathbf{x} \leq \epsilon_2 \\ & \mathbf{1}^T\mathbf{x} = 1 \\ & \mathbf{x} \geqq \mathbf{0} \\ & \theta \in \Theta,\ \boldsymbol{\epsilon} \in \mathcal{E}. \end{aligned} \tag{42}$$

Note that $\mathcal{E} = [\epsilon_1^{\min}, \epsilon_2^{\min}] \times [\epsilon_1^{\max}, \epsilon_2^{\max}]$, where $\epsilon_1^{\min}$, $\epsilon_1^{\max}$, $\epsilon_2^{\min}$, and $\epsilon_2^{\max}$, are computed by individually optimizing each criterion in problem (37) using $\theta = 16$. Then $\epsilon_1^{\min} = 0.120$ and $\epsilon_2^{\min} = 0.530$, while $\epsilon_1^{\max} = \max\{f_1(\hat{x}_2), f_1(\hat{x}_3)\} = 1$, $\epsilon_2^{\max} = \max\{f_2(\hat{x}_1), f_2(\hat{x}_3)\} = 2$, where $\hat{x}_i$ is the minimizer of f_i.

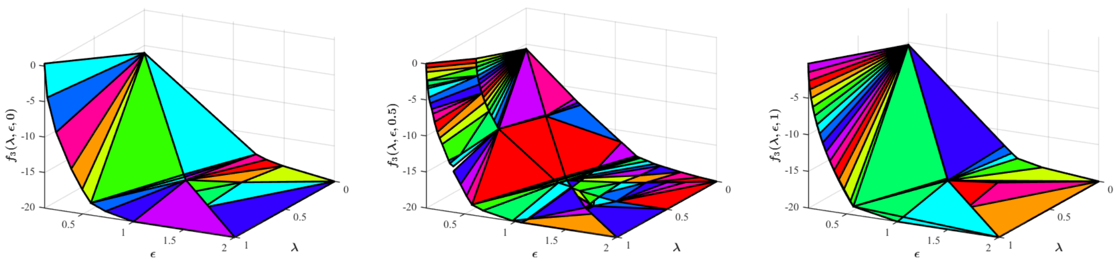
The graphs in Figure 9 depict the weak Pareto sets for problem (37) [24]. Note that ϵ_1 constrains f_1 and ϵ_2 constrains f_2. The figures have been rotated to show all four points from Table 6, and the narrow invariancy regions caused by numerical discrepancies are included in Figures 9a and 9c for completion. Note that points A, B, and C from Table 6 are included in the obtained approximate solutions, while D, also a Pareto point, is not. Nevertheless, this example demonstrates that the mpAS method provides relevant information and can be applied if necessary.

By holding $\bar{\theta}$ constant, the DM can readily see how limiting the risk with various values $\bar{\boldsymbol{\epsilon}}$ affects the return. Suppose that the return is $\bar{\theta} = 17$, and the DM wishes to limit the risks with $\bar{\boldsymbol{\epsilon}} = (0.517, 1.445)$. Then the encapsulating invariancy region is

$$IR = \left\{ (\epsilon_1, \epsilon_2, \theta^{nor}) : \begin{bmatrix} 0.774 & 0.155 & -0.463 \\ -0.569 & -0.340 & -0 \\ -0.822 & 0.292 & 0.312 \\ 0.6 & 0 & 0.528 \end{bmatrix} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \theta^{nor} \end{bmatrix} \leq \begin{bmatrix} 0.403 \\ -0.749 \\ 0.379 \\ 0.600 \end{bmatrix} \right\}.$$

The optimal solution on this region,

$$\tilde{\mathbf{x}}(\epsilon_1, \epsilon_2, \theta^{nor}) = \begin{bmatrix} -0.789\epsilon_1 - 0.011\epsilon_2 - 0.059\theta^{nor} + 0.489 \\ 0.731\epsilon_1 - 0.011\epsilon_2 + 0.016\theta^{nor} + 0.212 \\ 0.05\epsilon_1 + 0.0219\epsilon_2 + 0.043\theta^{nor} + 0.299 \end{bmatrix}$$



(a) $\theta = 15$. (b) $\theta = 16$. (c) $\theta = 17$.

Fig. 10: Approximate optimal return obtained by the mpAS method on problem (43).

has the approximate negative return function $\tilde{f}_3(\boldsymbol{\epsilon}, \theta^{nor}) = -26.173\epsilon_1 - 0.290\epsilon_2 - 2.545\theta - 2.069$. Then the amounts to be invested are found by the mp-AS to be $\tilde{\mathbf{x}}(0.517, 1.445, 1) = (0.036, 0.583, 0.382)$ yielding a negative return $\tilde{f}_3(0.517, 1.445, 1) = -17.590$, which is a gain of 17.590.

6.2.3 The reduced $\boldsymbol{\epsilon}$-constraint SOP

In the reduced $\boldsymbol{\epsilon}$-constraint formulation, the return is still maximized while the parametrized weighted-sum of both risk functions is parametrically controlled.

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^3} \quad & -\boldsymbol{p}_3(\theta)^T\mathbf{x} \\ \text{s.t.} \quad & \tfrac{1}{2}\lambda\mathbf{x}^T Q_1\mathbf{x} + \tfrac{1}{2}(1-\lambda)\mathbf{x}^T Q_2\mathbf{x} \le \epsilon \\ & \mathbf{1}^T\mathbf{x} = 1 \\ & \mathbf{x} \geqq \mathbf{0} \\ & \theta \in \Theta, \lambda \in [0,1], \epsilon \in \mathcal{E}. \end{aligned} \tag{43}$$

Note that solving the $\boldsymbol{\epsilon}$-constraint formulation gives the weak Pareto set as seen in Figure 9, but the weak Pareto set is not readily available from solving the reduced-$\boldsymbol{\epsilon}$-constraint formulation. As a result, we instead provide Figure 10. We note that as we fix $\bar{\theta}$, adjusting the weighting of the risks (the value of λ) affects the return given a total risk bound $\bar{\epsilon}$, as is expected. Likewise notice that Figure 10 demonstrates that, for fixed $\bar{\lambda}$, decreasing the total allowable risk, ϵ, will result in a lower return.

Consider the region containing $\bar{\lambda} = 0.613$, $\bar{\epsilon} = 0.639$, $\bar{\theta} = 0.5$ corresponding to a return value of 16. Then the encapsulating invariancy region is

$$IR = \left\{ (\lambda, \epsilon, \theta^{nor}) : \begin{bmatrix} 0 & 0.469 & -0.881 \\ 0.415 & -0.903 & -0.014 \\ -0.707 & 0 & 0.707 \\ 0.373 & 0.590 & 0.009 \end{bmatrix} \begin{bmatrix} \lambda \\ \epsilon \\ \theta^{nor} \end{bmatrix} \le \begin{bmatrix} 0.056 \\ -0.108 \\ 0 \\ 0.716 \end{bmatrix} \right\}.$$

The optimal solution on this region,

$$\tilde{\mathbf{x}}(\lambda, \epsilon, \theta^{nor}) = \begin{bmatrix} -0.179\lambda - 0.283\epsilon - 0.004\theta + 0.343 \\ 0.245\lambda + 0.184\epsilon - 0.040\theta + 0.310 \\ 0.005\lambda + 0.212\epsilon + 0.046\theta + 0.210 \end{bmatrix}$$

has the approximate negative return function $\tilde{f}_3(\lambda, \epsilon, \theta^{nor}) = -7.336\lambda - 10.738\epsilon - 0.733\theta^{nor} - 4.725$. Then $\tilde{\mathbf{x}}(\bar{\lambda}, \bar{\epsilon}, \bar{\theta}) = (0.050, 0.558, 0.372)$ yields $\tilde{f}_3(\bar{\lambda}, \bar{\epsilon}, \bar{\theta}) = -16.451$, which is a gain of 16.451.

7 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 [30], 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. Additionally, we experimented with applying the mpAS method to single objective mpQCQPs. Despite trading accuracy for solution speed, it worked accurately enough that we applied it to mpMOQPs. The mpAS method cannot find the true Pareto sets as well as the mpLCP method but is capable of approximating weak Pareto sets in totality and providing tradeoff information that is different from that offered by the mpLCP method. In the absence of an exact method, the mpAS method turns out to be an effective solution tool as the only method to approximately solve mpQCQPs resulting from scalarizing mpMOQPs. 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. There is a need to develop algorithms for mpQCQPs since they emerge from useful epsilon-constraint-type scalarizations of mpMOQPs. The mpAS method, that relies on joint convexity, could be extended biconvexity to allow for broader applicability. Dealing with nonconvex problems is also important. Dropping the convexity assumption and making use of available convex relaxations for nonconvex QPs may allow application of the presented algorithms to nonconvex MOQPs and, consequently, nonconvex mpMOQPs. 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.

References

1. Adelgren, N.: `https://github.com/Nadelgren/mpLCP_solver` (2019). Accessed March 11, 2021
2. Adelgren, N.: Advancing Parametric Optimization: Theory and Solution Methodology for Multiparametric Linear Complementarity Problems with Parameters in General Locations. SpringerBriefs on Optimization Series (2020)
3. Bednarczuk, E.: Continuity of minimal points with applications to parametric multiple objective optimization. *European Journal of Operational Research* **157**, 59–67 (2004)
4. Bemporad, A., Filippi, C.: An algorithm for approximate multiparametric convex programming. Computational Optimization and Applications **35**(1), 87–108 (2006)
5. Benson, H.: Multiple-objective linear programming with parametric criteria coefficients. Management Science **31**, 461–474 (1985)
6. Bitran, G.: Linear multi-objective programs with interval coefficients. Management Science **26**, 694–706 (1980)
7. Bonnel, H., Schneider, C.: 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)

8. Chankong, V., Haimes, Y.: Multiobjective Decision Making: Theory and Methodology. North-Holland Series in System Science and Engineering. North Holland (1983). URL https://books.google.com/books?id=ZIg-AQAAIAAJ
9. Cottle, R.: The principal pivoting method revisited. *Mathematical Programming* **48**, 369 – 385 (1990). DOI https://doi.org/10.1007/BF01582264
10. Cottle, R., Guu, S.: Two characterizations of sufficient matrices. Linear Algebra and its Applications **170**, 65 – 74 (1992). DOI https://doi.org/10.1016/0024-3795(92)90410-C. URL http://www.sciencedirect.com/science/article/pii/002437959290410C
11. Cottle, R., Pang, J., Stone, R.: The Linear Complementarity Problem, 1st edn. Society for Industrial and Applied Mathematics (2009). DOI 10.1137/1.9780898719000. URL https://epubs.siam.org/doi/abs/10.1137/1.9780898719000
12. Cottle, R., Pang, J., Venkateswaran, V.: Sufficient matrices and the linear complementarity problem. Linear Algebra and its Applications **114-115**, 231 – 249 (1989). DOI https://doi.org/10.1016/0024-3795(89)90463-1. URL http://www.sciencedirect.com/science/article/pii/0024379589904631
13. Dellnitz, M., Witting, K.: Computation of robust Pareto points. International Journal of Computing Science and Mathematics **2**(3), 243–266 (2009)
14. Den Hertog, D., Roos, C., Terlaky, T.: 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). URL https://doi.org/10.1007/978-3-642-77489-8_18
15. Diamond S. Agrawal, A., Murray, R.: CVXPY. https://www.cvxpy.org/examples/basic/quadratic_program.html (2020). Accessed March 3, 2021
16. Ehrgott, M.: Multicriteria Optimization. Springer (2005)
17. Ehrgott, M., Greco, S., Figueira, J.: Multiple Criteria Decision Analysis: State of the Art Surveys, 2nd edn. International Series in Operations Research and Management Science. Springer (2016). URL https://books.google.com/books?id=xQRhAQAACAAJ
18. El-Banna, A.: A study on parametric multiobjective programming problems without differentiability. Comput. Math. Appl. **26**(12), 87–92 (1993)
19. Enkhbat, R., Guddat, J., Chinchuluun, A.: Parametric multiobjective optimization. In: A. Chinchuluun, P. Pardalos, A. Migdalas, L. Pitsoulis (eds.) Pareto Optimality, Game Theory and Equilibria, *Springer Optimization and Its Applications*, vol. 17, pp. 529–538. Springer (2008)
20. Fang, Y., Yang, X.: Smooth representations of optimal solution sets of piecewise linear parametric multiobjective programs. In: Variational Analysis and Generalized Differentiation in Optimization and Control, *Springer Optim. Appl.*, vol. 47, pp. 163–176. Springer, New York (2010)
21. Galvan, E., Malak, R., Hartl, D., Baur, J.: 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)
22. Geoffrion, A.: Propr efficiency and the theory of vector maximization. Journal of Mathematical Analysis and Applications **22**(3), 618 – 630 (1968). DOI https://doi.org/10.1016/0022-247X(68)90201-1. URL http://www.sciencedirect.com/science/article/pii/0022247X68902011
23. Guddat, J., Vasquez, F., Tammer, K., Wendler, K.: Multiobjective and Stochastic Optimization Based on Parametric Optimization, *Mathematical Research*, vol. 26. Akademie-Verlag, Berlin (1985)
24. Haimes, Y., Lasdon, L., Wismer, D.: 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). DOI 10.1109/TSMC.1971.4308298
25. Hartl, D., Galvan, E., Malak, R., Baur, J.: Parameterized design optimization of a magnetohydrodynamic liquid metal active cooling concept. J. Mech. Des. **138**, 031402 (11 pages) (2016)
26. Herceg, M., Kvasnica, M., Jones, C., Morari, M.: Multi-parametric toolbox 3.0. In: 2013 European Control Conference (ECC), pp. 502–510. IEEE (2013)
27. Hirschberger, M., Qi, Y., Steuer, R.: Large-scale MV efficient frontier computation via a procedure of parametric quadratic programming. *European Journal of Operational Research* **204**(3), 581–588 (2010). URL https://ideas.repec.org/a/eee/ejores/v204y2010i3p581-588.html
28. Hirschberger, M., Steuer, R., Utz, S., Wimmer, W., Qi, Y.: Computing the nondominated surface in tri-criterion portfolio selection. Operational Research **61**, 169–183 (2013)
29. Huy, N., Mordukhovich, B., Yao, J.: Coderivatives of frontier and solution maps in parametric multiobjective optimization. Taiwanese J. Math. **12**(8), 2083–2111 (2008)
30. Jayasekara, P., Adelgren, N., Wiecek, M.: On convex multiobjective programs with application to portfolio optimization. Journal of Multi-Criteria Decision Analysis **27**(3-4), 189–202 (2019)
31. Johansen, T.: On multi-parametric nonlinear programming and explicit nonlinear model predictive control. In: Proceedings of the 41st IEEE Conference on Decision and Control, vol. 3, pp. 2768–2773. IEEE (2002)
32. Klafszky, E., Terlaky, T.: Some generalizations of the criss-cross method for quadratic programming. Optimization **24**(1-2), 127–139 (1992). DOI 10.1080/02331939208843783. URL https://doi.org/10.1080/02331939208843783
33. Leverenz, J.: Network target coordination for multiparametric programming. PhD dissertation, Clemson University (2016)
34. Leverenz, J., Xu, M., Wiecek, M.: Multiparametric optimization for multidisciplinary engineering design. Structural and Multidisciplinary Optimization **54**(4), 1–16 (2016)
35. Li, D., Yang, J., Biswal, M.: Quantitative parametric connections between methods for generating noninferior solutions in multiobjective optimization. *European Journal of Operational Research* **117**, 84 – 99 (1999)
36. Lucchetti, R., Miglierina, E.: Stability for convex vector optimization problems. Optimization **53**(5–6), 517–528 (2004)
37. Naccache, P.: Stability in multicriteria optimization. J. Math. Anal. Appl. **68**(2), 441–453 (1979)
38. Oberdiecka, R., Pistikopoulos, N.: Multiobjective optimization with convex quadratic cost functions: A multiparametric programming approach. Computers and Chemical Engineering **85**, 36–39 (2016)

39. Pappas, I., Diangelakis, N., Pistikopoulos, E.: The exact solution of multiparametric quadratically constrained quadratic programming problems. Journal of Global Optimization pp. 1–27 (2020)
40. Penot, J., Sterna-Karwat, A.: Parametrized multicriteria optimization: Continuity and closedness of optimal multifunctions. Journal of Mathematical Analysis and Applications **120**(1), 150–168 (1986)
41. Penot, J., Sterna-Karwat, A.: Parametrized multicriteria optimization: Order continuity of the marginal multifunctions. Journal of Mathematical Analysis and Applications **144**(1), 1–15 (1989)
42. Pistikopoulos, E., Diangelakis, N., Oberdieck, R.: *Multi-parametric Optimization and Control*. Operations Research and Management Science. Wiley (2021)
43. Romanko, O., Ghaffari-Hadigheh, A., Terlaky, T.: Multiobjective optimization via parametric optimization: Models, algorithms, and applications. In: T. Terlaky, F. Curtis (eds.) Modeling and Optimization: Theory and Applications, pp. 77–119. Springer New York (2012)
44. Ruetsch, G.: Using interval techniques to solve a parametric multi-objective optimization problem. United States Patent No. 7.664,622 B2 (2010). URL `https://patents.google.com/patent/US7664622B2/en`
45. Sawaragi, Y., Nakayama, H., Tanino, T.: Theory of Multiobjective Optimization. Academic Press (1985)
46. Steuer, R., Qi, Y., Hirschberger, M.: Comparative issues in large-scale mean-variance efficient frontier computation. Decision Support Systems **51**, 250–255 (2011)
47. Thuan, L., Luc, D.: On sensitivity in linear multiobjective programming. J. Optim. Theory Appl. **107**(3), 615–626 (2000)
48. Väliaho, H.: A procedure for the one-parametric linear complementarity problem. Optimization **29**(3), 235–256 (1994). DOI 10.1080/02331939408843953. URL `https://doi.org/10.1080/02331939408843953`
49. Wiecek, M., Dranichak, G.: Robust multiobjective optimization for decision making under uncertainty and conflict. In: J. Smith (ed.) Optimization Challenges in Complex, Networked, and Risky Systems, Tutorials in Operations Research, pp. 84–114. INFORMS (2016)
50. Wiecek, M., Ehrgott, M., Engau, A.: Continuous multiobjective programming. In: S. Greco, M. Ehrgott, J. Figueira (eds.) Multiple Criteria Decision Analysis: State of the Art Surveys, pp. 738–815. Springer, 2nd edition (2016)
51. Witting, K., Ober-Blöbaum, S., Dellnitz, M.: A variational approach to define robustness for parametric multiobjective optimization problems. J. Global Optim. **57**(2), 331–345 (2013)
52. Witting, K., Schulz, B., Dellnitz, M., Böcker, J., Fröhleke, N.: A new approach for online multiobjective optimization of mechatronic systems. International Journal of Software Tools and Technology Transfer **10**, 223–231 (2008)
53. Wittmann-Hohlbein, M., Pistikopoulos, E.: On the global solution of multi-parametric mixed integer linear programming problems. Journal of Global Optimization **57**(1), 51–73 (2013)
54. Zhou, H., Hastie, T.: Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society B **67**(2), 301–320 (2005)

A Appendix

A.1 Solving the BOQP in Example (25) with the spLCP method

Below we present the iterations of the spLCP method when solving the BOQP in (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 $T_{B_0}(\lambda)$

	$\boldsymbol{\mathfrak{g}}_{.0}$	$\boldsymbol{\mathfrak{g}}_{.6}$	$\boldsymbol{\mathfrak{g}}_{.7}$	$\boldsymbol{\mathfrak{g}}_{.8}$	$\boldsymbol{\mathfrak{g}}_{.9}$	$\boldsymbol{\mathfrak{g}}_{.10}$
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 $(T_{B_0}(0))_{.0} \not\geq \mathbf{0}$. 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 (26) for $\lambda = 0$

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 27/2 \end{bmatrix} \qquad \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 0 \\ 0 \end{bmatrix}, \tag{44}$$

which yields an efficient solution to BOQP (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 4 is used to find an invariancy interval for λ, i.e., the increment τ above 0, in which (44) 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:

$$P_{i1}(\tau) = \det\big(-T_B(\lambda)_{KK} + \tau^{-1}\mathscr{I}\big)\big(T_B(\lambda)_{i0} - (-T_B(\lambda)_{iK})(T_B(\lambda)_{KK} + \tau^{-1}\mathscr{I})^{-1}T_B(\lambda)_{K0}\big) = 0.$$

$$\text{For } i = 1, \qquad \det\left(\begin{bmatrix} 2+\tau^{-1} & 0 \\ 0 & \tau^{-1} \end{bmatrix}\right)\left(1/2 - \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} 2+\tau^{-1} & 0 \\ 0 & \tau^{-1} \end{bmatrix}^{-1}\begin{bmatrix} 6 \\ -2 \end{bmatrix}\right) = 0$$

$$\left(\frac{2\tau+1}{\tau^2}\right)\left(\frac{1-10\tau}{4\tau+2}\right) = 0$$

$$\alpha_1 = \tau = 1/10.$$

$$\text{For } i = 2, \qquad \det\left(\begin{bmatrix} 2+\tau^{-1} & 0 \\ 0 & \tau^{-1} \end{bmatrix}\right)\left(1 - \begin{bmatrix} 0 & -3 \end{bmatrix}\begin{bmatrix} 2+\tau^{-1} & 0 \\ 0 & \tau^{-1} \end{bmatrix}^{-1}\begin{bmatrix} 6 \\ -2 \end{bmatrix}\right) = 0$$

$$\left(\frac{2\tau+1}{\tau^2}\right)(1-6\tau) = 0$$

$$\alpha_2 = \tau = 1/6.$$

$$\text{For } i = 3, \qquad \det\left(\begin{bmatrix} 2+\tau^{-1} & 0 \\ 0 & \tau^{-1} \end{bmatrix}\right)\left(27/2 - \begin{bmatrix} -3 & 0 \end{bmatrix}\begin{bmatrix} 2+\tau^{-1} & 0 \\ 0 & \tau^{-1} \end{bmatrix}^{-1}\begin{bmatrix} 6 \\ -2 \end{bmatrix}\right) = 0$$

$$\left(\frac{2\tau+1}{\tau^2}\right)(2\tau+3) = 0$$

$$\alpha_3 = \tau = \infty.$$

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 (44) 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
$v_{\kappa 1}$	5	-2	0	0	0	0
$v_{\kappa 2}$	-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 (26) for $\lambda = 1/10$:

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 2/5 \\ 15 \end{bmatrix} \qquad \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{45}$$

which yields an efficient solution to BOQP (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 4 is used to find an invariancy interval for λ, i.e., the increment τ above 1/10, in which (45) 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:

$$P_{i1}(\tau) = \det\big(-T_B(\lambda)_{KK} + \tau^{-1}\mathscr{I}\big)\big(T_B(\lambda)_{i0} - (-T_B(\lambda)_{iK})(T_B(\lambda)_{KK} + \tau^{-1}\mathscr{I})^{-1}T_B(\lambda)_{K0}\big) = 0.$$

For $i = 1$,
$$\det\left(\begin{bmatrix} \tau^{-1} & 0 \\ 0 & \tau^{-1} \end{bmatrix}\right)\left(0 - \begin{bmatrix} -2 & 0 \end{bmatrix}\begin{bmatrix} \tau^{-1} & 0 \\ 0 & \tau^{-1} \end{bmatrix}^{-1}\begin{bmatrix} 5 \\ -2 \end{bmatrix}\right) = 0$$
$$\left(\frac{1}{\tau^2}\right)(10\tau) = 0$$
$$\alpha_1 = \tau = \infty.$$

For $i = 2$,
$$\det\left(\begin{bmatrix} \tau^{-1} & 0 \\ 0 & \tau^{-1} \end{bmatrix}\right)\left(2/5 - \begin{bmatrix} 0 & -3 \end{bmatrix}\begin{bmatrix} \tau^{-1} & 0 \\ 0 & \tau^{-1} \end{bmatrix}^{-1}\begin{bmatrix} 5 \\ -2 \end{bmatrix}\right) = 0$$
$$\left(\frac{1}{\tau^2}\right)(2/5 - 6\tau) = 0$$
$$\alpha_2 = \tau = 1/15.$$

$$\text{For } i=3, \qquad \det\left(\begin{bmatrix}\tau^{-1} & 0\\ 0 & \tau^{-1}\end{bmatrix}\right)\left(15-\begin{bmatrix}0 & 0\end{bmatrix}\begin{bmatrix}\tau^{-1} & 0\\ 0 & \tau^{-1}\end{bmatrix}^{-1}\begin{bmatrix}5\\ -2\end{bmatrix}\right)=0$$

$$\left(\frac{15}{\tau^2}\right)=0$$

$$\alpha_3=\tau=\infty.$$

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 (45) is $[1/10, (1/10) + (1/15)] = [1/10, 1/6]$. The starting tableau for the next iteration is constructed.

Iteration 3: 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 (26) for $\lambda = 1/6$:

$$\begin{bmatrix}w_1\\ w_2\\ w_3\end{bmatrix}=\begin{bmatrix}4/6\\ 0\\ 15\end{bmatrix} \qquad \begin{bmatrix}z_1\\ z_2\\ z_3\end{bmatrix}=\begin{bmatrix}0\\ 0\\ 0\end{bmatrix}, \tag{46}$$

which yields an efficient solution to BOQP (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 4 is used to find an invariancy interval, i.e., the increment τ from above 1/6, in which (46) 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.

$$P_{i1}(\tau) = \det\big(-T_B(\lambda)_{KK} + \tau^{-1}\mathscr{I}\big)\big(T_B(\lambda)_{i0} - (-T_B(\lambda)_{iK})(T_B(\lambda)_{KK} + \tau^{-1}\mathscr{I})^{-1}T_B(\lambda)_{K0}\big) = 0.$$

$$\text{For } i=1, \qquad \det\left(\begin{bmatrix}\tau^{-1} & 0\\ 0 & \tau^{-1}\end{bmatrix}\right)\left(4/6-\begin{bmatrix}-2 & 0\end{bmatrix}\begin{bmatrix}\tau^{-1} & 0\\ 0 & \tau^{-1}\end{bmatrix}^{-1}\begin{bmatrix}5\\ -2\end{bmatrix}\right)=0$$

$$\left(\frac{1}{\tau^2}\right)(4/6+10\tau)=0$$

$$\alpha_1=\tau=\infty.$$

$$\text{For } i = 2, \qquad \det\left(\begin{bmatrix}\tau^{-1} & 0\\ 0 & \tau^{-1}\end{bmatrix}\right)\left(0 - \begin{bmatrix}0 & -3\end{bmatrix}\begin{bmatrix}\tau^{-1} & 0\\ 0 & \tau^{-1}\end{bmatrix}^{-1}\begin{bmatrix}5\\ -2\end{bmatrix}\right) = 0$$
$$\left(\frac{1}{\tau^2}\right)(-6\tau) = 0$$
$$\alpha_2 = \tau = \infty.$$

$$\text{For } i = 3, \qquad \det\left(\begin{bmatrix}\tau^{-1} & 0\\ 0 & \tau^{-1}\end{bmatrix}\right)\left(15 - \begin{bmatrix}0 & 0\end{bmatrix}\begin{bmatrix}\tau^{-1} & 0\\ 0 & \tau^{-1}\end{bmatrix}^{-1}\begin{bmatrix}5\\ -2\end{bmatrix}\right) = 0$$
$$\left(\frac{15}{\tau^2}\right) = 0$$
$$\alpha_3 = \tau = \infty$$

Then $\rho_1 = \infty$. 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 (46) 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 $\mathscr{B}_{i0} \prec \mathbf{0}$. Therefore the current basis $B_0 = \{w_1, w_2, w_3\}$ is updated using Algorithm 6. 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 (26) for $\lambda = 1$

$$\begin{bmatrix}w_1\\ w_2\\ w_3\end{bmatrix} = \begin{bmatrix}9\\ 0\\ 170/14\end{bmatrix} \qquad \begin{bmatrix}z_1\\ z_2\\ z_3\end{bmatrix} = \begin{bmatrix}0\\ 5/14\\ 0\end{bmatrix}, \tag{47}$$

which yields an efficient solution to BOQP (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.

A.2 Solution to the Elastic Net Problem in Example (31)

Table 7: Invariancy regions ($I\mathcal{R}$) and efficient solution functions for TOQP (30) with data (31) solved as (32) with the mpLCP method.

$$I\mathcal{R}_1 = \left\{ \boldsymbol{\lambda} \in \Lambda' : \begin{array}{l} 100 - 100\lambda_2 - 409\lambda_1 \geq 0 \\ 100 - 100\lambda_2 - 1007\lambda_1 \geq 0 \\ 100 - 100\lambda_2 - 871\lambda_1 \geq 0 \\ 209\lambda_1 - 100\lambda_2 + 100 \geq 0 \\ 807\lambda_1 - 100\lambda_2 + 100 \geq 0 \\ 671\lambda_1 - 100\lambda_2 + 100 \geq 0 \end{array} \right\}$$

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

$$I\mathcal{R}_2 = \left\{ \boldsymbol{\lambda} \in \Lambda' : \begin{array}{l} -15,623\lambda_1^2 - 13,936\lambda_1\lambda_2 + 12,300\lambda_1 - 400\lambda_2^2 + 400\lambda_2 \geq 0 \\ 1,007\lambda_1 + 100\lambda_2 - 100 \geq 0 \\ 900\lambda_1 + 200\lambda_2 - 2,642\lambda_1\lambda_2 + 3,245\lambda_1^2 - 200\lambda_2^2 \geq 0 \\ 209\lambda_1 - 100\lambda_2 - 29,203\lambda_1^2 + 2,900\lambda_1\lambda_2 - 2,900\lambda_1 \geq 0 \\ 1 - \lambda_2 - \lambda_1 \geq 0 \\ -15,858\lambda_1\lambda_2 - 200\lambda_2 - 17,200\lambda_1 + 21,345\lambda_1^2 + 200\lambda_2^2 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\boldsymbol{\lambda}) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 0 \\ x_2 = \frac{1,007\lambda_1 + 100\lambda_2 - 100}{4,525\lambda_1 + 100\lambda_2} \\ x_3 = 0 \end{array} \right\} \text{ for } \boldsymbol{\lambda} \in I\mathcal{R}_2$$

$$I\mathcal{R}_3 = \left\{ \boldsymbol{\lambda} \in \Lambda' : \begin{array}{l} 13,936\lambda_1\lambda_2 - 400\lambda_2 - 12,300\lambda_1 + 15,623\lambda_1^2 + 400\lambda_2^2 \geq 0 \\ 1,900\lambda_1 - 200\lambda_2 + 114\lambda_1\lambda_2 - 1,791\lambda_1^2 + 200\lambda_2^2 \geq 0 \\ -1,200\lambda_1\lambda_2 - 2,284\lambda_1\lambda_2^2 + 2,020\lambda_1^2\lambda_2 + 7,500\lambda_1^2 - 6,423\lambda_1^3 + 400\lambda_2^2 - 400\lambda_2^3 \geq 0 \\ 1 - \lambda_2 - \lambda_1 \geq 0 \\ -38,716\lambda_1\lambda_2^2 + 41,400\lambda_1\lambda_2 - 55,020\lambda_1^2\lambda_2 + 5,300\lambda_1^2 - 6,377\lambda_1^3 + 400\lambda_2^2 - 400\lambda_2^3 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\boldsymbol{\lambda}) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{13,936\lambda_1\lambda_2 - 400\lambda_2 - 12,300\lambda_1 + 15,623\lambda_1^2 + 400\lambda_2^2}{20,100\lambda_1\lambda_2 + 6,400\lambda_1^2 + 400\lambda_2^2} \\ x_2 = \frac{1,900\lambda_1 - 200\lambda_2 + 114\lambda_1\lambda_2 - 1,791\lambda_1^2 + 200\lambda_2^2}{10,050\lambda_1\lambda_2 + 3,200\lambda_1^2 + 200\lambda_2^2} \\ x_3 = 0 \end{array} \right\} \text{ for } \boldsymbol{\lambda} \in I\mathcal{R}_3$$

$$I\mathcal{R}_4 = \left\{ \boldsymbol{\lambda} \in \Lambda' : \begin{array}{l} -25,036\lambda_1\lambda_2^2 + 23,400\lambda_1\lambda_2 - 82,622\lambda_1^2\lambda_2 + 44,700\lambda_1^2 - 51,879\lambda_1^3 + 400\lambda_2^2 - 400\lambda_2^3 \geq 0 \\ 100\lambda_1 - 200\lambda_2 + 1,914\lambda_1\lambda_2 + 10,077\lambda_1^2 + 200\lambda_2^2 \geq 0 \\ -900\lambda_1 - 200\lambda_2 + 2,642\lambda_1\lambda_2 - 3,245\lambda_1^2 + 200\lambda_2^2 \geq 0 \\ -44,164\lambda_1\lambda_2^2 + 45,000\lambda_1\lambda_2 - 114,378\lambda_1^2\lambda_2 + 83,900\lambda_1^2 - 76,721\lambda_1^3 + 400\lambda_2^2 - 400\lambda_2^3 \geq 0 \\ 1 - \lambda_2 - \lambda_1 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\boldsymbol{\lambda}) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 0 \\ x_2 = \frac{100\lambda_1 - 200\lambda_2 + 1{,}914\lambda_1\lambda_2 + 10{,}077\lambda_1^2 + 200\lambda_2^2}{17{,}100\lambda_1\lambda_2 + 32{,}150\lambda_1^2 + 500\lambda_2^2} \\ x_3 = \frac{-900\lambda_1 - 200\lambda_2 + 2642\lambda_1\lambda_2 - 3{,}245\lambda_1^2 + 200\lambda_2^2}{17{,}100\lambda_1\lambda_2 + 32{,}150\lambda_1^2 + 500\lambda_2^2} \end{array} \right\} \text{ for } \boldsymbol{\lambda} \in I\mathcal{R}_4$$

$$I\mathcal{R}_5 = \left\{ \boldsymbol{\lambda} \in \Lambda' : \begin{array}{l} 25,036\lambda_1\lambda_2^2 - 23,400\lambda_1\lambda_2 + 82,622\lambda_1^2\lambda_2 - 44,700\lambda_1^2 + 51,879\lambda_1^3 - 400\lambda_2^2 + 400\lambda_2^3 \geq 0 \\ 4,000\lambda_1\lambda_2 + 28\lambda_1\lambda_2^2 - 4,128\lambda_1^2\lambda_2 + 20,700\lambda_1^2 - 18,603\lambda_1^3 - 400\lambda_2^2 + 400\lambda_2^3 \geq 0 \\ 1,200\lambda_1\lambda_2 + 2,284\lambda_1\lambda_2^2 - 2,020\lambda_1^2\lambda_2 - 7,500\lambda_1^2 + 6,423\lambda_1^3 - 400\lambda_2^2 + 400\lambda_2^3 \geq 0 \\ 1 - \lambda_2 - \lambda_1 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\boldsymbol{\lambda}) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{25{,}036\lambda_1\lambda_2^2 - 23{,}400\lambda_1\lambda_2 + 82{,}622\lambda_1^2\lambda_2 - 44{,}700\lambda_1^2 + 51{,}879\lambda_1^3 - 400\lambda_2^2 + 400\lambda_2^3}{36{,}200\lambda_1\lambda_2^2 + 88{,}700\lambda_1^2\lambda_2 + 17{,}100\lambda_1^3 + 400\lambda_2^3} \\ x_2 = \frac{4{,}000\lambda_1\lambda_2 + 28\lambda_1\lambda_2^2 - 4{,}128\lambda_1^2\lambda_2 + 20{,}700\lambda_1^2 - 18{,}603\lambda_1^3 - 400\lambda_2^2 + 400\lambda_2^3}{36{,}200\lambda_1\lambda_2^2 + 88{,}700\lambda_1^2\lambda_2 + 17{,}100\lambda_1^3 + 400\lambda_2^3} \\ x_3 = \frac{1{,}200\lambda_1\lambda_2 + 2{,}284\lambda_1\lambda_2^2 - 2{,}020\lambda_1^2\lambda_2 - 7{,}500\lambda_1^2 + 6{,}423\lambda_1^3 - 400\lambda_2^2 + 400\lambda_2^3}{36{,}200\lambda_1\lambda_2^2 + 88{,}700\lambda_1^2\lambda_2 + 17{,}100\lambda_1^3 + 400\lambda_2^3} \end{array} \right\} \text{ for } \boldsymbol{\lambda} \in I\mathcal{R}_5$$

$$I\mathcal{R}_6 = \left\{ \boldsymbol{\lambda} \in \Lambda' : \begin{array}{l} 35,036\lambda_1\lambda_2^2 - 33,400\lambda_1\lambda_2 + 72,422\lambda_1^2\lambda_2 - 24,500\lambda_1^2 + 31,679\lambda_1^3 - 400\lambda_2^2 + 400\lambda_2^3 \geq 0 \\ 32,628\lambda_1\lambda_2^2 - 28,600\lambda_1\lambda_2 + 46,472\lambda_1^2\lambda_2 + 2,700\lambda_1^2 - 603\lambda_1^3 - 400\lambda_2^2 + 400\lambda_2^3 \geq 0 \\ 38,716\lambda_1\lambda_2^2 - 41,400\lambda_1\lambda_2 + 55,020\lambda_1^2\lambda_2 - 5,300\lambda_1^2 + 6,377\lambda_1^3 - 400\lambda_2^2 + 400\lambda_2^3 \geq 0 \\ 1 - \lambda_2 - \lambda_1 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\boldsymbol{\lambda}) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{35{,}036\lambda_1\lambda_2^2 - 33{,}400\lambda_1\lambda_2 + 72{,}422\lambda_1^2\lambda_2 - 24{,}500\lambda_1^2 + 31{,}679\lambda_1^3 - 400\lambda_2^2 + 400\lambda_2^3}{36{,}200\lambda_1\lambda_2^2 + 88{,}700\lambda_1^2\lambda_2 + 17{,}100\lambda_1^3 + 400\lambda_2^3} \\ x_2 = \frac{32{,}628\lambda_1\lambda_2^2 - 28{,}600\lambda_1\lambda_2 + 46{,}472\lambda_1^2\lambda_2 + 2{,}700\lambda_1^2 - 603\lambda_1^3 - 400\lambda_2^2 + 400\lambda_2^3}{36{,}200\lambda_1\lambda_2^2 + 88{,}700\lambda_1^2\lambda_2 + 17{,}100\lambda_1^3 + 400\lambda_2^3} \\ x_3 = \frac{-38{,}716\lambda_1\lambda_2^2 + 41{,}400\lambda_1\lambda_2 - 55{,}020\lambda_1^2\lambda_2 + 5{,}300\lambda_1^2 - 6{,}377\lambda_1^3 + 400\lambda_2^2 - 400\lambda_2^3}{36{,}200\lambda_1\lambda_2^2 + 88{,}700\lambda_1^2\lambda_2 + 17{,}100\lambda_1^3 + 400\lambda_2^3} \end{array} \right\} \text{ for } \boldsymbol{\lambda} \in I\mathcal{R}_6$$

A.3 Solution to the Elastic Net Problem in Example (31) (with α and β)

Table 8: Invariancy regions ($I\mathcal{R}$) and efficient solution functions for TOQP (30) with data (31).

$$I\mathcal{R}_1 = \left\{ \alpha,\beta \geq 0 : \begin{array}{l} 100\beta - 309 \geq 0 \\ 100\beta - 907 \geq 0 \\ 100\beta - 771 \geq 0 \\ 100\beta + 309 \geq 0 \\ 100\beta + 907 \geq 0 \\ 100\beta + 771 \geq 0 \end{array} \right\}$$

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

$$I\mathcal{R}_2 = \left\{ \alpha,\beta \geq 0 : \begin{array}{l} -1,236\alpha + 12,300\beta + 400\alpha\beta - 3,323 \geq 0 \\ -100\beta + 907 \geq 0 \\ 900\beta - 1,542\alpha + 200\alpha\beta + 4,145 \geq 0 \\ 1,236\alpha + 23,900\beta + 400\alpha\beta + 3,323 \geq 0 \\ 1,542\alpha + 17,200\beta + 200\alpha\beta - 4,145 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\alpha,\beta) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 0 \\ x_2 = \frac{-100\beta+907}{100\alpha+4,525} \\ x_3 = 0 \end{array} \right\} \text{ for } (\alpha,\beta) \in I\mathcal{R}_2$$

$$I\mathcal{R}_3 = \left\{ \alpha,\beta \geq 0 : \begin{array}{l} 1,236\alpha - 12,300\beta - 400\alpha\beta + 3,323 \geq 0 \\ 1,814\alpha + 1,900\beta - 200\alpha\beta + 109 \geq 0 \\ 8,320\alpha + 7,500\beta - 1,200\alpha\beta + 400\alpha^2\beta - 3,084\alpha^2 + 1,077 \geq 0 \\ 5,300\beta - 8,320\alpha + 41,400\alpha\beta + 400\alpha^2\beta + 3,084\alpha^2 - 1,077 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\alpha,\beta) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{1,236\alpha-12,300\beta-400\alpha\beta+3,323}{400\alpha^2+20,100\alpha+6,400} \\ x_2 = \frac{1,814\alpha+1,900\beta-200\alpha\beta+109}{200\alpha^2+10,050\alpha+3,200} \\ x_3 = 0 \end{array} \right\} \text{ for } (\alpha,\beta) \in I\mathcal{R}_3$$

$$I\mathcal{R}_4 = \left\{ \alpha,\beta \geq 0 : \begin{array}{l} -14,522\alpha + 44,700\beta + 23,400\alpha\beta + 400\alpha^2\beta - 1,236\alpha^2 - 7,179 \geq 0 \\ 1,814\alpha + 100\beta - 200\alpha\beta + 10,177 \geq 0 \\ -900\beta + 1,542\alpha - 200\alpha\beta - 4,145 \geq 0 \\ 14,522\alpha + 83,900\beta + 45,000\alpha\beta + 400\alpha^2\beta + 1,236\alpha^2 + 7,179 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\alpha,\beta) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 0 \\ x_2 = \frac{1,814\alpha+100\beta-200\alpha\beta+10,177}{200\alpha^2+17,100\alpha+32,150} \\ x_3 = \frac{-900\beta+1,542\alpha-200\alpha\beta-4,145}{200\alpha^2+17,100\alpha+32,150} \end{array} \right\} \text{ for } (\alpha,\beta) \in I\mathcal{R}_4$$

$$I\mathcal{R}_5 = \left\{ \alpha,\beta \geq 0 : \begin{array}{c} 14,522\alpha - 44,700\beta - 23,400\alpha\beta - 400\alpha^2\beta + 1,236\alpha^2 + 7,179 \geq 0 \\ 20,572\alpha + 20,700\beta + 4,000\alpha\beta - 400\alpha^2\beta + 3,628\alpha^2 + 2,097 \geq 0 \\ -8,320\alpha - 7,500\beta + 1,200\alpha\beta - 400\alpha^2\beta + 3,084\alpha^2 - 1,077 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\alpha,\beta) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{c} x_1 = \frac{14,522\alpha - 44,700\beta - 23,400\alpha\beta - 400\alpha^2\beta + 1,236\alpha^2 + 7,179}{400\alpha^3 + 36,200\alpha^2 + 88,700\alpha + 17,100} \\ x_2 = \frac{20,572\alpha + 20,700\beta + 4,000\alpha\beta - 400\alpha^2\beta + 3,628\alpha^2 + 2,097}{400\alpha^3 + 36,200\alpha^2 + 88,700\alpha + 17,100} \\ x_3 = \frac{-8,320\alpha - 7,500\beta + 1,200\alpha\beta - 400\alpha^2\beta + 3,084\alpha^2 - 1,077}{400\alpha^3 + 36,200\alpha^2 + 88,700\alpha + 17,100} \end{array} \right\} \text{ for } (\alpha,\beta) \in I\mathcal{R}_5$$

$$I\mathcal{R}_6 = \left\{ \alpha,\beta \geq 0 : \begin{array}{c} 14,522\alpha - 24,500\beta - 33,400\alpha\beta - 400\alpha^2\beta + 1,236\alpha^2 + 7,179 \geq 0 \\ 20,572\alpha + 2,700\beta - 28,600\alpha\beta - 400\alpha^2\beta + 3,628\alpha^2 + 2,097 \geq 0 \\ -5,300\beta + 8,320\alpha - 41,400\alpha\beta - 400\alpha^2\beta - 3,084\alpha^2 + 1,077 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\alpha,\beta) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{c} x_1 = \frac{14,522\alpha - 24,500\beta - 33400\alpha\beta - 400\alpha^2\beta + 1236\alpha^2 + 7179}{400\alpha^3 + 36,200\alpha^2 + 88,700\alpha + 17,100} \\ x_2 = \frac{20,572\alpha + 2,700\beta - 28,600\alpha\beta - 400\alpha^2\beta + 3,628\alpha^2 + 2,097}{400\alpha^3 + 36,200\alpha^2 + 88,700\alpha + 17,100} \\ x_3 = \frac{5,300\beta - 8,320\alpha + 41,400\alpha\beta + 400\alpha^2\beta + 3,084\alpha^2 - 1,077}{400\alpha^3 + 36,200\alpha^2 + 88,700\alpha + 17,100} \end{array} \right\} \text{ for } (\alpha,\beta) \in I\mathcal{R}_6$$

A.4 Solution to the Portfolio Problem in Example (37)

Table 9: Invariancy region ($I\mathcal{R}$) and efficient solution functions for Example (37) scalarized with the modified hybrid method ($\theta = \theta^{nor}, \epsilon = \epsilon^{nor}$, and the parameter space $\Omega = \Theta^{nor} \times \Lambda' \times \mathcal{E}^{nor}$)

$$I\mathcal{R}_1 = \left\{ (\theta,\lambda,\epsilon) \in \Omega : \begin{array}{l} -20\lambda - 737\epsilon - 24\theta + 134\lambda\epsilon + 8\lambda\theta + 60 \geq 0 \\ -740\lambda - 44,019\epsilon - 808\theta + 4,958\lambda\epsilon + 56\lambda\theta - 2,144\epsilon\theta + 96\lambda\theta^2 - 160\theta^2 \\ \qquad +1,608\lambda\epsilon\theta + 3,020 \geq 0 \\ 67\epsilon + 4\theta - 10 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\theta,\lambda,\epsilon) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = 0 \\ x_2 = \frac{67\epsilon+4\theta-10}{4\theta-10} \\ x_3 = \frac{-67\epsilon}{4\theta-10} \end{array} \right\} \text{ for } (\theta,\lambda,\epsilon) \in I\mathcal{R}_1$$

$$I\mathcal{R}_2 = \left\{ (\theta,\lambda,\epsilon) \in \Omega : \begin{array}{l} 1,139\lambda + 4,489\epsilon + 56\theta - 2,412\lambda\epsilon - 402\lambda^2\epsilon - 24\lambda^2\theta \\ \qquad +60\lambda^2 - 2,217 \geq 0 \\ -31,021\epsilon + 21,562\lambda - 1552\theta - 96\lambda^2\theta^2 + 20,167\lambda\epsilon + 4,512\lambda\theta \\ \qquad -3,752\epsilon\theta + 12,060\lambda^2\epsilon + 384\lambda\theta^2 - 1,776\lambda^2\theta - 3,938\lambda^2 \\ \qquad -352\theta^2 + 1,608\lambda^2\epsilon\theta - 27,554 \geq 0 \\ -740\lambda - 44,019\epsilon - 808\theta + 4,958\lambda\epsilon + 56\lambda\theta - 2,144\epsilon\theta + 96\lambda\theta^2 \\ \qquad -160\theta^2 + 1,608\lambda\epsilon\theta + 3,020 \geq 0 \\ 20,167\epsilon - 2,080\lambda + 640\theta + 13,936\lambda\epsilon + 592\lambda\theta - 2680\epsilon\theta + 96\lambda\theta^2 \\ \qquad -128\theta^2 + 1,608\lambda\epsilon\theta - 23,245 \geq 0 \\ 37\lambda + 178\epsilon - 20\theta - 141\lambda\epsilon + 12\lambda\theta + 36\epsilon\theta - 24\lambda\epsilon\theta - 151 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\theta,\lambda,\epsilon) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{-740\lambda-44,019\epsilon-808\theta+4,958\lambda\epsilon+56\lambda\theta-2,144\epsilon\theta+96\lambda\theta^2-160\theta^2+1,608\lambda\epsilon\theta+3,020}{2,138\lambda-2,848\theta+2,256\lambda\theta+192\lambda\theta^2-288\theta^2-40,459} \\ x_2 = \frac{20,167\epsilon-2,080\lambda+640\theta+13,936\lambda\epsilon+592\lambda\theta-2,680\epsilon\theta+96\lambda\theta^2-128\theta^2+1,608\lambda\epsilon\theta-23,245}{2,138\lambda-2,848\theta+2,256\lambda\theta+192\lambda\theta^2-288\theta^2-40,459} \\ x_3 = \frac{4,958\lambda+23,852\epsilon-2,680\theta-18,894\lambda\epsilon+1,608\lambda\theta+4,824\epsilon\theta-3,216\lambda\epsilon\theta-20,234}{2,138\lambda-2,848\theta+2,256\lambda\theta+192\lambda\theta^2-288\theta^2-40,459} \end{array} \right\} \text{ for } (\theta,\lambda,\epsilon) \in I\mathcal{R}_2$$

$$I\mathcal{R}_3 = \left\{ (\theta,\lambda,\epsilon) \in \Omega : \begin{array}{l} -1,139\lambda - 4,489\epsilon - 56\theta + 2,412\lambda\epsilon + 402\lambda^2\epsilon + 24\lambda^2\theta - 60\lambda^2 + 2,217 \geq 0 \\ -15\lambda^2 + 86\lambda - 6\lambda^3 - 71 \geq 0 \\ 17\lambda - 31 \geq 0 \\ 19\lambda - 22 \geq 0 \\ 3\lambda^2 - 7 \geq 0 \end{array} \right\}$$

$$\hat{\mathbf{x}}(\theta,\lambda,\epsilon) = \left\{ \mathbf{x} \in \mathbb{R}^3 : \begin{array}{l} x_1 = \frac{17\lambda-31}{36\lambda+6\lambda^2-67} \\ x_2 = \frac{19\lambda-22}{36\lambda+6\lambda^2-67} \\ x_3 = \frac{6\lambda^2-14}{36\lambda+6\lambda^2-67} \end{array} \right\} \text{ for } (\theta,\lambda,\epsilon) \in I\mathcal{R}_3$$