Title
Data Science Optimization with Polynomials

Author
Zhong, Suhan

Publication Date
2022

UNIVERSITY OF CALIFORNIA SAN DIEGO

Data Science Optimization with Polynomials

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Mathematics

by

Suhan Zhong

Committee in charge:

Professor Jiawang Nie, Chair
Professor Thomas R. Bewley
Professor Ioana Dumitriu
Professor Philip E. Gill
Professor Behrouz Touri

2022

The dissertation of Suhan Zhong is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2022

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

To begin with, I would like to express my greatest and most sincere thanks to my thesis supervisor, Professor Jiawang Nie. Jiawang is an enthusiastic mathematician, an inclusive teacher, a reliable advisor and an interesting friend. I'm always impressed by his insightful taste in math and good sense of humor. My Ph.D. experience would not be so enjoyable and fulfilled without his comprehensive supervision and constant support. He gave me numerous of invaluable suggestions, which have deeply shaped my life. He is very important to me and I'm so proud of being his student.

There are many thanks to other members of my doctoral committee: Professor Thomas R. Bewley, Professor Ioana Dumitriu, Professor Philip E. Gill and Professor Behrouz Touri. They are all experts in different fields. It is my honor to have them in my doctoral committee.

In addition, I want to thank my coauthors: Dr. Xindong Tang, Prof. Li Wang, Prof. Liu Yang, Prof. Jane J. Ye and Prof. Guangming Zhou. Their welath of knowledge and rich experiences gave me lots of inspirations during the cooperation.

I feel grateful that the Department of Mathematics at UCSD gave me an offer five years ago. It is in this magic place that I met so many great people. Dr. Xindong Tang and Dr. Zi Yang. They also studied under Prof. Nie but a year before me, are my most helpful and reliable seniors. Dr. Jiaqi Liu, my closest fellow. We share so many critical moments from the qualifying exams to the final defense. Prof. Tianyi Zheng, my cycling alliance. Our weekend adventure map ranges from San Diego downtown to Encinitas. My dear friends Zian Wang and Mengying Lan. We have so many joyful dinners and games. And so many other cohorts and colleagues...

I would give most credits to my parents and my family for their endless love and support. No matter where I am, their hearts are always with me, despite the physical distance. It is their love that gives me the brave and confidence to overcome any kind of challenges. In particular, I would like to give special credits to my host family Jane Grupe and Dave Grupe. We knew each other from Solana Beach Host Family Program. Their kindness and love cross two countries and two cultures.

Chapter 3, in full, is a reprint of the material as it appears in *Optimization Methods and Software* 2020. The dissertation author coauthored this paper with Nie, Jiawang and Yang, Liu.

Chapter 4, in full, has been submitted for publication. The dissertation author coauthored this paper with Nie, Jiawang; Yang, Liu and Zhou, Guangming.

Chapter 5, in full, is a reprint of the material as it appears in *SIAM Journal on Optimization* 2021. The dissertation author coauthored this paper with Nie, Jiawang; Wang, Li and Ye, Jane. J.

Chapter 6, in full, has been submitted for publication. The dissertation author coauthored this paper with Nie, Jiawang and Tang, Xindong.

Chapter 7, in full, has been submitted for publication. The dissertation author coauthored this paper with Nie, Jiawang.

VITA

2017	B. S. in Mathematics, Xi'an Jiaotong University
2022	Ph. D. in Mathematics, University of California San Diego

PUBLICATIONS

J. Nie, L. Yang and S. Zhong, "Stochastic Polynomial Optimization", *Optimization Methods and Software*, 35.2 (2020): 329-347.

J. Nie, L. Wang, J. Ye and S. Zhong, "A Lagrange Multiplier Expression Method for Bilevel Polynomial Optimization", *SIAM Journal on Optimization*, 31.3 (2021): 2368–2395.

J. Nie, L. Yang, S. Zhong and G. Zhou, "Distributionally Robust Optimization with Moment Ambiguity Sets", *Preprint*, 2021. `arXiv:2103.12315`

J. Nie, X. Tang and S. Zhong, "Rational Generalized Nash Equilibrium Problems", *Preprint*, 2021. `arXiv:2110.12120`

J. Nie and S. Zhong, "Loss Functions for Finite Sets", *Preprint*, 2021. `arXiv:2112.05927`

ABSTRACT OF THE DISSERTATION

Data Science Optimization with Polynomials

by

Suhan Zhong

Doctor of Philosophy in Mathematics

University of California San Diego, 2022

Professor Jiawang Nie, Chair

Optimization is essential in data science literature. The data science optimization studies all optimization problems that have applications in data science. The polynomial function is broadly used in data science optimization. In data science optimization, we are mostly interested in stochastic optimization, equilibrium games and loss function optimization.

The stochastic optimization studies optimization problems that are constructed with random variables. A classic kind of stochastic optimization is to find the optimizer of a function that is given by the expectation of some random variables. For stochastic optimization with polynomials, we propose an efficient perturbation sample average approximation model. It can be solved globally by Moment-SOS relaxations, and gives a robust approximation of the original problem.

The distributionally robust optimization (DRO) is another kind of stochastic optimization. It assumes the uncertainty is described by an ambiguity set, and aims to optimize the objective function under the worst-case of the ambiguity. For DRO defined with polynomials and under moment ambiguity, we transform it into a linear conic optimization with

moment and psd polynomial cones, and give a semidefinite algorithm to solve it globally.

The bilevel optimization is a kind of challenging optimization problems whose feasible set is constrained by the optimizer set of another optimization problem. For bilevel optimization defined with polynomials, we propose a semidefinite algorithm to solve it globally. Under some general assumptions, the algorithm can either get the global minimizer(s), or detect the nonexistence of them.

The generalized Nash equilibrium problem (GNEP) is formed by a group of mutually parametrized optimization problems. It aims to find a equilibrium such that each objective function cannot be solely further optimized. For GNEPs with rational polynomial functions, we propose a new approach for solving them with a hierarchy of rational optimization problems. Under some general assumptions, we show that the proposed hierarchy can compute a GNE, if it exists, or detect its nonexistence.

Loss functions are essential in data science optimization. We study loss functions for finite sets and propose a kind of efficient sum-of-square (SOS) polynomial loss functions for general finite sets. We show how to compute the SOS loss functions of the lowest degree. In addition, we give a special kind of SOS loss functions such that all their local minimizers are also global minimizers.

Chapter 1

Introduction

Data science is an interdisciplinary area that covers many research topics. The optimization is an important field in data science. This is because many problems in data science are given in form of optimization. For other kind of data science problems, the optimization methods are also often applied. In summary, the optimization theories constitute mathematical foundation of data science. On the other hand, the data science applications enrich research topics in optimization. Therefore, it is interesting to explore the overlapping area of data science and optimization, which we denote as *data science optimization*.

We are mostly interested in three major kinds of data science optimization problems: stochastic optimization, equilibrium optimization and loss function optimization. The stochastic optimization includes the classic model and a new model of distributionally robust optimization. For equilibrium optimization, the bilevel optimization and generalized Nash equilibrium problems are two important problems. In loss function optimization, we study how to construct an efficient kind of loss functions for general finite sets.

1.1 Stochastic optimization

The stochastic optimization is an important class of data science optimization problems. It is usually formulated with random variables and the so-called decision variables. Let $x = (x_1, \ldots, x_n)$ be the decision variable and $\xi = (\xi_1, \ldots, \xi_p)$ be a random vector. A classical stochastic optimization problem is

$$\min_{x \in K} \quad f(x) := \mathbb{E}[F(x, \xi)], \tag{1.1}$$

where $K \subseteq \mathbb{R}^n$ ($\mathbb{R}$ is the real field and $\mathbb{R}^n$ is the n-dimensional Euclidean space), F is a function in (x, ξ), and $\mathbb{E}[F(x, \xi)]$ is the expectation of F with respect to ξ. The coefficients of the objective function f are typically not known explicitly, because the true distribution of ξ is usually not known exactly. Frequently used methods for solving stochastic optimization are often based on sample average approximation (SAA). We refer to [13, 37, 54, 67, 71, 102, 108] for related work on stochastic optimization. The SAA methods use sample averages to approximate the expectation function $f(x)$, transforming the stochastic optimization into deterministic optimization. Many classical SAA methods assume the objective functions are convex and are based on evaluations of gradients or subgradients. They can also be applied to nonconvex problems, however, the global optimality may not be guaranteed. There exists relatively less work on nonconvex stochastic optimization [4, 39, 40]. Generally, nonconvex stochastic optimization problems are computationally challenging, because the deterministic case is already difficult.

We are interested in the stochastic optimization defined with polynomials. The (1.1) is called a stochastic polynomial optimization problem if F is a polynomial in (x, ξ) and K is a semialgebraic set. In this case, the objective function f is also a polynomial. For given samples $\xi^{(1)}, \ldots, \xi^{(N)}$ of ξ, the SAA of (1.1), i.e.,

$$\min_{x \in K} \quad \frac{1}{N} \sum_{k=1}^{N} F(x, \xi^{(k)}) \tag{1.2}$$

is a deterministic polynomial optimization problem, which can be solved globally by Moment-SOS relaxations [59]. SAA methods have good statistical properties: the optimal value and solution set of (1.2) converge to that of (1.1) in probability one as $N \to \infty$ [102]. However, some concerns of SAA methods need to be addressed. First, the solution set of (1.2) may (or may not) be far away from the optimizer set of (1.1), depending on the sampling quality. Second, the SAA (1.2) is only an approximation of (1.1). We do not need to solve it exactly. But we expect to get an approximation of the solution set for the original problem. These concerns require us to construct a more robust approximation of (1.1), which can be solved more efficiently.

The stochastic optimization has broad applications in data science applications. People often use it to study real world data that has an unknown distribution. We refer to books [58, 102] for an overview of stochastic optimization. It is worth to note that the stochastic polynomial optimization plays an important role in financial literature for modeling portfolio investing problems. In the following, we introduce a classical portfolio selection model, which is in form of stochastic polynomial optimization.

Example 1.1. *[72] For a portfolio that consists of n assets, suppose its return is described by the random vector $\xi = (\xi_1, \dots, \xi_n)$. Let $x = (x_1, \dots, x_n)$ be the investing proportion such that each $x_i \geq 0$ and $x_1 + \cdots + x_n = 1$. The classical mean-variance (M-V) portfolio selection model is*

$$\begin{cases} \max & \mathbb{E}[\xi]^T x - \mathbb{E}[(\xi^T x - \mathbb{E}[\xi]^T x)^2] \\ s.t. & x \geq 0,\, x_1 + \cdots + x_n = 1, \end{cases}$$

where $\tau > 0$ is a risk preference parameter. It is clear that the above M-V model is a stochastic polynomial optimization problem.

1.2 Distributionally robust optimization

The distributionally robust optimization (DRO) aims to optimize the objective function under a worst-case random realization, with some given constraints. A typical DRO problem is

$$\begin{cases} \min\limits_{x \in X} & f(x) \\ s.t. & \inf\limits_{\mu \in \mathcal{M}} \mathbb{E}_\mu[h(x,\xi)] \geq 0, \end{cases} \tag{1.3}$$

where $x := (x_1, \dots, x_n)$ is the decision variable constrained in a set $X \subseteq \mathbb{R}^n$ and $\xi := (\xi_1, \dots, \xi_p) \in \mathbb{R}^p$ is the random variable obeying the distribution of a measure $\mu \in \mathcal{M}$. The notation $\mathbb{E}_\mu[h(x,\xi)]$ stands for the expectation of the random function $h(x,\xi)$ with respect to the distribution of ξ. The set $\mathcal{M}$ is called the *ambiguity set*, which is used to describe the uncertainty of the measure μ.

The ambiguity set $\mathcal{M}$ is often moment-based or discrepancy-based. For the moment-based ambiguity, the set $\mathcal{M}$ is usually specified by the first, second moments [25, 44, 110]. Recently, people are also interested in ambiguity set of higher order moment [15, 64], especially in relevant applications with machine learning. For discrepancy-based ambiguity sets, popular examples are the ϕ-divergence ambiguity sets [6, 73] and the Wasserstein ambiguity sets [97]. There are also some other types of ambiguity sets. For instance, [53] assumes $\mathcal{M}$ is given by distributions with SOS polynomial density functions of known degrees. In practice, the ambiguity set $\mathcal{M}$ is usually constructed following the sampling or historic data. For instance, people may know the support of the measure, discrepancy from a reference distribution, or its descriptive statistics from observations. The ambiguity set usually contains a collection of measures satisfying such properties. For the special case that $\mathcal{M}$ only contains the true distribution of the random variable, the distributionally robust optimization is reduced to be the classic stochastic optimization.

We are interested in the moment-based ambiguity sets. For instance, consider

$$\mathcal{M} := \Big\{\mu \in \mathcal{B}(S) : \mathbb{E}_\mu([\xi]_d) \in Y\Big\}.$$

In the above, $\mathcal{M}$ is the set of all Borel measures whose supports and moments, up to a given degree d, are respectively contained in given sets $S \subseteq \mathbb{R}^p$ and $Y \subseteq \mathbb{R}^{\binom{p+d}{d}}$. The $[\xi]_d$ is the monomial vector

$$[\xi]_d := \begin{bmatrix} 1 & \xi_1 & \cdots & \xi_p & (\xi_1)^2 & \xi_1\xi_2 & \cdots & (\xi_p)^d \end{bmatrix}^T.$$

The DRO (1.3) equipped with the above ambiguity set is called the distributionally robust optimization of moment (DROM). When all the defining functions are polynomials, the DROM is an important class of distributionally robust optimization. Here is a concrete DROM problem defined with polynomials.

Example 1.2. *For a univariate random variable $\xi \in \mathbb{R}^1$, consider the DROM*

$$\begin{cases} \min\limits_{x\in\mathbb{R}^4} & -x_1 - 2x_2 - x_3 + 2x_4 \\ s.t. & \inf\limits_{\mu\in\mathcal{M}} \mathbb{E}_\mu[h(x,\xi)] \geq 0, \\ & x \geq 0,\ 1 - e^T x \geq 0, \end{cases}$$

where (supp(μ) denotes the support of μ)

$$\begin{aligned} h(x,\xi) = (x_4 - x_1 - 2)\xi^5 + (x_4 - 1)\xi^4 + (2x_1 + x_2 + x_4 + 1)\xi^3 \\ +(2x_1 - x_2 + x_4 - 1)\xi^2 + (2 - x_2 - x_3)\xi, \end{aligned}$$

$$\mathcal{M} = \left\{ supp(\mu) \subseteq [0,3] : 1 \leq \mathbb{E}_\mu[1] \leq \mathbb{E}_\mu[\xi] \leq \mathbb{E}_\mu[\xi^2] \leq \cdots \leq \mathbb{E}_\mu[\xi^5] \leq 2 \right\}.$$

The optimal value and solution of this optimization problem are given in Example 4.27.

For recent work about distributionally robust optimizatione, we refer to [53, 98, 112, 118, 119, 121]. The DRO has various applications, i.e., [25, 33, 120] in portfolio management, [73, 113] in network design, [9, 110] in inventory problems and [32, 43, 75] in machine learning.

1.3 Bilevel optimization

The bilevel optimization is in form of

$$\begin{cases} \min\limits_{x\in\mathbb{R}^n, y\in\mathbb{R}^p} & F(x,y) \\ \quad s.t. & h(x,y) \geq 0, \\ & y \in S(x), \end{cases}$$

where $S(x)$ is the set of optimizer(s) of the lower level problem

$$\begin{cases} \min\limits_{z\in\mathbb{R}^p} & f(x,z) \\ \ s.t. & z \in Z(x) := \{z : g(x,z) \geq 0\}. \end{cases}$$

In the above, $F(x,y)$ is the upper level objective function and $h(x,y)$, as a tuple of functions, are the upper level constraints. Similarly, the $f(x,z)$ is called the lower level objective function and $g(x,z)$, as a tuple of functions, are the lower level constraints.

Bilevel optimization is a challenging problem. The classical (or the first order) approach is to relax the optimality constraint $y \in S(x)$ by the first order optimality condition for the lower level problem. But solving the resulting single-level problem may not even recover a stationary point of the original bilevel optimization problem if the lower level problem is nonconvex; see [74]. Moreover, even for the case that the lower level optimization is convex, the resulting single-level problem may not be equivalent to the original bilevel optimization problem if local optimality is considered and the lower level multiplier set is not a singleton (see [27]). Another approach to use the value function or semi-infinite programming (SIP) reformulation. For each $y \in Z(x)$, it is easy to see the following equivalence (without any assumptions about the lower level optimization, e.g., convexity)

$$y \in S(x) \Longleftrightarrow f(x,y) - v(x) \leq 0 \Longleftrightarrow f(x,z) - f(x,y) \geq 0 \quad \forall z \in Z(x), \tag{1.4}$$

where $v(x) := \inf_{z\in Z(x)} f(x,z)$ is the so-called value function for the lower level problem. We call any reformulation using the first equivalence in (1.4) the value function reformulation, while those using the second equivalence in (1.4) the semi-infinite programming (SIP) reformulation. Using the value function reformulation results in an intrinsically nonsmooth optimization problem which never satisfies the usual constraint qualification [116]. Despite these difficulties, recent progresses have been made on constraint qualifications and optimality conditions for bilevel optimization problems, where the lower level optimization is not assumed to be convex; see the work [28, 114] and the references therein.

Bilevel optimization is an important class of equilibrium optimization. It has broad applications, e.g., the moral hazard model of the principal-agent problem in economics [74], electricity markets and networks [10], facility location and production problem [11], meta learning and hyper-parameter selection in machine learning [36, 57, 68]. More applications can be found in the monographs [3,26,30,103] and the surveys on bilevel optimization [19,28] and the references therein. Here we briefly introduce the application of bilevel optimization in hyperparameter tuning.

Example 1.3. *[115] Suppose $\{(a_j, b_j) : j \in \Omega\}$, $a_j \in \mathbb{R}^n$, $b_j \in \mathbb{R}$ is a data set with the finite label set Ω. Divide Ω into the nonempty training label set $T \subseteq \Omega$ and the validation label set $V = \Omega \setminus T \neq \emptyset$. The least absolute shrinkage and selection operator (lasso) problem is*

$$\min_{z \in \mathbb{R}^n} \quad \sum_{j \in T} (a_j^T z - b_j)^2 + x\|z\|_1, \tag{1.5}$$

where $\|\cdot\|_1$ is the 1-norm and the penalty parameter $x \in \mathbb{R}^1$ can be regarded as the hyperparameter. The desirable hyperparameter is usually chosen as the minimizer of the validation error function

$$\frac{1}{|V|} \sum_{j \in V} (a_j^T y(x) - b_j)^2,$$

where $y(x)$ is optimal solution of (1.5) with respect to x. Therefore, the hyperparameter problem for the lasso problem can be formulated as the following bilevel optimization

$$\begin{cases} \min\limits_{x \in \mathbb{R}^1, y \in \mathbb{R}^n} & \frac{1}{|V|} \sum\limits_{j \in V} (a_j^T y - b_j)^2 \\ \quad s.t. & x \geq 0, \\ & y \in \operatorname*{argmin}\limits_{z \in \mathbb{R}^n} \sum\limits_{j \in T} (a_j^T z - b_j)^2 + x\|z\|_1, \end{cases}$$

where the symbol argmin denote the set of minimizers.

1.4 Generalized Nash equilibrium problems

The generalized Nash equilibrium problem (GNEP) is to determine a tuple of strategies $u = (u_1, \ldots, u_N)$ such that each u_i minimizes the optimization problem

$$F_i(u_{-i}) : \begin{cases} \min\limits_{x_i \in \mathbb{R}^{n_i}} & f_i(x_i, u_{-i}) \\ \quad s.t. & x_i \in X_i(u_{-i}), \end{cases}$$

for given $u_{-i} := (u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_N)$. In the above, $X_i(u_{-i}) \subseteq \mathbb{R}^{n_i}$ is the feasible set of x_i that is parameterized by the given strategies u_{-i}, which may be empty. The tuple u that satisfies all the above conditions is called a generalized Nash equilibrium (GNE). We use the following example to better explain the concept of GNEs.

Example 1.4. *Consider the 2-player GNEP, where $x_1 = (x_{1,1}, x_{1,2})$ and $x_2 = (x_{2,1}, x_{2,2})$. The first player's optimization problem is*

$$F_1(x_2) : \begin{cases} \min\limits_{x_1 \in \mathbb{R}^2} & (x_{1,1} - x_{1,2})x_{2,1}x_{2,2} - x_1^T x_1 \\ s.t. & 1 - x_{1,1} - x_{1,2} \geq 0, \\ & x_{1,1} \geq 0,\ x_{1,2} \geq 0. \end{cases}$$

the second player's optimization problem is

$$F_2(x_1) : \begin{cases} \min\limits_{x_2 \in \mathbb{R}^2} & 3(x_{2,1} - x_{1,1})^2 + 2(x_{2,2} - x_{1,2})^2 \\ s.t. & 2 - x_{2,1} - x_{2,2} \geq 0, \\ & x_{2,1} \geq 0,\ x_{2,2} \geq 0. \end{cases}$$

The above GNEP has a GNE $x^* = (x_1^*, x_2^*)$ with

$$x_1^* = x_2^* = (0.5, 0).$$

When $x_2 = x_2^*$, the first player's optimization is reduced to be

$$F_1(x_2^*) : \begin{cases} \min\limits_{x_1 \in \mathbb{R}^2} & -x_1^T x_1 \\ s.t. & 1 - x_{1,1} - x_{1,2} \geq 0, \\ & x_{1,1} \geq 0,\ x_{1,2} \geq 0. \end{cases}$$

It has the global minimizer $x_1 = x_1^*$. Similarly, it is easy to verify that $x_2 = x_2^*$ is the global minimizer of $F_2(x_1^*)$. Therefore, x^* is a GNE for this GNEP.

A special case of GNEPs is the *Nash Equilibrium Problems (NEPs)*, which has each feasible set $X_i(x_{-i})$ be independent of x_{-i}. When NEPs are defined by polynomials, a method is given in [87] to solve them. For GNEPs given by convex polynomials, it is studied how to solve them in the recent work [88]. The Karush-Kuhn-Tucker (KKT) conditions are useful for solving GNEPs and NEPs. We refer to [31, 34, 35, 49] for related work.

The GNEPs were originally introduced to model economic problems. They are now widely used in various fields, such as transportation, telecommunications, and machine learning. We refer to [2, 14, 20, 52, 69, 96] for recent applications of GNEPs.

1.5 Loss function optimization

Loss functions are important in data science optimization. Let n, k be positive integers. Suppose S is a set of k distinct points in the n-dimensional real Euclidean space $\mathbb{R}^n$. A function f in $x := (x_1, \ldots, x_n)$ is said to be a loss function for S if the global minimizers of f are precisely the points in S. For convenience, we often select f such that f is nonnegative in $\mathbb{R}^n$ and the minimum value is zero. Mathematically, this is equivalent to that

$$f(x) = 0 \quad \text{if and only if} \quad x \in S.$$

Example 1.5. *Suppose $S = \{y_1, \ldots, u_k\}$ is given explicitly. Then*

$$f(x) = \|x - u_1\|^2 \cdots \|x - u_k\|^2$$

is a straightforward choice of the loss function for S.

In practice, the set S may be given explicitly or be approximated by a large number of sampling points. For the latter case, we need to recover representing points of S and then compute the loss function.

Example 1.6. *Suppose the set $S = \{0\}$ is approximated by a large sampling set $T \subseteq [-\epsilon, \epsilon]$, where $\epsilon > 0$ is a small error bound. Then we would expect to recover a representing set $S^* = \{\eta\}$, $-\epsilon \le \eta \le \epsilon$ of S, and then construct a loss function $f(x) = (x - \eta)^2$ for S^*.*

A frequently used loss function is the class of sum-of-squares polynomials. That is, the loss function f is in the form

$$f = p_1^2 + \cdots + p_m^2,$$

where each p_i is a polynomial in x. Then f is a loss function for S if and only if each $p_i \equiv 0$ on S. For convenience of computation, we prefer that f and each p_i have degrees as low as possible. More preferable is that every local minimizer of f is a global minimizer.

Loss functions are important in data science problems. There are broad applications of loss functions [17, 42, 55, 84, 100, 111]. Selection of loss functions needs to consider different application purposes and data structures. There are various types of loss functions for different applications. We refer to the survey [109] for a comprehensive introduction for all kinds of loss functions in machine learning. Recently, much attention has been paid to the selection and design of loss functions [5, 16, 107]. Most researchers focus on improving the qualitative performance for a specific purpose and a fixed kind of loss functions. We notice few people study the common properties of loss functions, or have interests in constructing a family of loss functions that share the same properties. However, these less popular problems are fundamental for the studies of loss functions. They are also interesting mathematical problems alone that worth more attention.

Chapter 2

Preliminaries

Notation. The symbol $\mathbb{N}$ (resp., $\mathbb{R}, \mathbb{C}$) denotes the set of nonnegative integers (resp., real numbers, complex numbers). The set $\mathbb{N}^n$ (resp., $\mathbb{R}^n$, $\mathbb{C}^n$) is the collection of n-dimensional vector with elements in $\mathbb{N}$ (resp., $\mathbb{R}$, $\mathbb{C}$). The $\mathbb{R}^n_+$ denotes the nonnegative orthant of $\mathbb{R}^n$. For $t \in \mathbb{R}$, $\lceil t \rceil$ denotes the smallest integer that is greater or equal to t. For an integer $n > 0$, $[n] := \{1, \cdots, n\}$. For S_1, $S_2 \subseteq \mathbb{R}^n$, we denote $S_1 + S_2 := \{u + v : u \in S_1, v \in S_2\}$. For a vector $v = (v_1, \ldots, v_n)$, $\|v\|$ denotes the standard Euclidean norm. The symbol $diag[v]$ denotes the n-by-n diagonal matrix with the i-th diagonal entry v_i for all $i \in [n]$. We use $\mathbf{1}$ or e to denote an all-one vector, and e_i to denote the unit vector with all zero entries except the ith entry equaling one. Denote by I_n the n-by-n identity matrix. The superscript ^{T} (resp., $^{\mathtt{H}}$) denotes the operation of matrix transpose (resp., Hermitian). A square matrix A is said to be positive semidefinite or psd (resp., positive definite or pd) if $x^T Ax \geq 0$ (resp., $x^T Ax > 0$) for each nonzero vector x. For two square matrices X, Y of the same dimension, their commutator is

$$[X, Y] \coloneqq XY - YX.$$

That is, X commutes with Y if and only if $[X, Y] = 0$. For a function f that is continuously differentiable in $x = (x_1, \ldots, x_n)$, the ∇f denotes its gradient in x, $\nabla^2 f$ denotes its Hessian, and the $\partial_{x_i} f$ denotes the partial gradient of f in x_i.

Let $\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$. Denote by $\mathbb{F}[x] := \mathbb{F}[x_1, \ldots, x_n]$ the ring of polynomials in $x := (x_1, \ldots, x_n)$ with coefficients in $\mathbb{F}$. For every $d \in \mathbb{N}$, $\mathbb{F}[x]_d$ denotes the subspace of $\mathbb{F}[x]$ which contains all polynomials of degree at most d. For a polynomial $f \in \mathbb{F}[x]$, we use $\deg(f)$ to denote its degree. For a tuple of polynomial $g = (g_1, \ldots, g_m)$, $g_i \in \mathbb{F}[x]$, we use $\deg(g)$ to denote the highest degree of g_i, i.e.,

$$\deg(g) = \max\{\deg(g_1),\, \ldots,\, \deg(g_m)\}.$$

For every $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$, denote the monomial

$$x^\alpha := x_1^{\alpha_1} \cdots x_n^{\alpha_n}.$$

Its total degree is $|\alpha| := \alpha_1 + \cdots + \alpha_n$. For a positive integer k, $[x]_k$ denotes the vector of all monomials of the highest degree k ordered in the graded lexicographic ordering, i.e.,

$$[x]_k := \begin{bmatrix} 1 & x_1 & \cdots & x_n & (x_1)^2 & x_1x_2 & \cdots & (x_n)^k \end{bmatrix}^T.$$

2.1 Basic algebraic geometry

A subset $I \subseteq \mathbb{F}[x]$ is an ideal of $\mathbb{F}[x]$ if $p \cdot q \in I$ for all $p \in I$, $q \in \mathbb{F}[x]$, and $p_1 + p_2 \in I$ for all $p_1, p_2 \in I$. For an ideal I, its *radical* is the set

$$\sqrt{I} := \{f \in \mathbb{F}[x] :\, f^k \in I \text{ for some } k \in \mathbb{N}\}.$$

The set $\sqrt{I}$ is also an ideal and $I \subseteq \sqrt{I}$. The ideal I is said to be radical if $I = \sqrt{I}$. Each ideal I determines the *variety* in $\mathbb{F}^n$ as

$$V_{\mathbb{F}}(I) := \{x \in \mathbb{F}^n :\, p(x) = 0\, (p \in I)\}.$$

For a polynomial tuple $p := (p_1, \ldots, p_m)$, we similarly denote that

$$V_{\mathbb{F}}(p) \;:=\; \{x \in \mathbb{F}^n :\, p(x) = 0\}.$$

In particular, $V(p)$ denotes the real variety of p with the real field omitted. The tuple p generates the ideal

$$\mathrm{Ideal}[p] := p_1 \cdot \mathbb{F}[x] + \cdots + p_m \cdot \mathbb{F}[x].$$

Clearly, $V_{\mathbb{F}}(\mathrm{Ideal}(p)) = V_{\mathbb{F}}(p)$. For a degree $k \geq \deg(p)$, the kth order truncation of Ideal$[p]$ is

$$\mathrm{Ideal}[p]_k = p_1 \cdot \mathbb{F}[x]_{k-\deg(p_1)} + \cdots + p_m \cdot \mathbb{F}[x]_{k-\deg(p_m)}.$$

For a set $S \subseteq \mathbb{C}^n$, its vanishing ideal is

$$I(S) := \{q \in \mathbb{C}[x] :\, q(u) = 0\, (u \in S)\}.$$

If $S = V_{\mathbb{C}}(p)$ for some polynomial tuple p in x, then $\mathrm{Ideal}(p) \subseteq I(S)$ but the equality may not hold. For every $I \subseteq \mathbb{C}[x]$, we have $I(V_{\mathbb{C}}(I)) = \sqrt{I}$. This is Hilbert's Nullstellensatz [22].

For a given ideal $I \subseteq \mathbb{C}[x]$, it determines an equivalence relation $\sim$ on $\mathbb{C}[x]$ such that $p \sim q$ if $p - q \in I$, or equivalently, $p \equiv q \mod I$. Then every $p \in \mathbb{C}[x]$ corresponds to an equivalence class with the module of I, i.e.,

$$[p] = \{q \in \mathbb{C}[x] : q \equiv p \mod I\}.$$

The set of all equivalent classes is the quotient ring

$$\mathbb{C}[x]/I \coloneqq \{[p] : p \in \mathbb{C}[x]\}.$$

2.2 Polynomial optimization

A polynomial $\sigma \in \mathbb{R}[x]$ is said to be a sum-of-squares (SOS) polynomial if

$$\sigma = \sigma_1^2 + \cdots + \sigma_k^2,$$

for some $\sigma_1, \ldots, \sigma_k \in \mathbb{R}[x]$. We denote by $\Sigma[x]$ the set of all SOS polynomials in x, and denote $\Sigma[x]_d := \Sigma[x] \cap \mathbb{R}[x]_d$ for each degree d. In particular, f is said to be SOS-convex [45] if its Hessian matrix $\nabla^2 f(x)$ is SOS, i.e., $\nabla^2 f = A(x)^T A(x)$ for a matrix polynomial $A(x)$. For a set $S \subseteq \mathbb{R}^n$, the symbol $\mathscr{P}(S)$ denotes the set of all polynomials that are nonnegative on S. For a degree d, we denote $\mathscr{P}_d(S) = \mathscr{P}(S) \cap \mathbb{R}[x]_d$. For a tuple of polynomials $q = (q_1, \ldots, q_t)$ in x, we define the *quadratic module* of q by

$$\mathrm{Qmod}[q] \coloneqq \Sigma[x] + q_1 \cdot \Sigma[x] + \cdots + q_t \cdot \Sigma[x].$$

For $k \geq \lceil \deg(q)/2 \rceil$, the k-th order truncation of $\mathrm{Qmod}[q]$ is

$$\mathrm{Qmod}[q]_{2k} := \Sigma[x]_{2k} + q_1 \cdot \Sigma[x]_{2k-deg(q_1)} + \cdots + q_t \cdot \Sigma[x]_{2k-deg(q_t)}.$$

Each polynomial in $\mathrm{Qmod}[q]$ is nonnegative over the basic semi-algebraic set

$$W(q) := \{x \in \mathbb{R}^n : q(x) \geq 0\}.$$

Given real polynomial tuples p and q, if $f \in \mathrm{Ideal}[p] + \mathrm{Qmod}[q]$, then it is easy to see that $f(x) \geq 0$ for all $x \in V(p) \cap W(q)$. To ensure $f \in \mathrm{Ideal}[p] + \mathrm{Qmod}[q]$, we typically need more than $f(x) \geq 0$ for all $x \in V(p) \cap W(q)$. The sum $\mathrm{Ideal}[p] + \mathrm{Qmod}[q]$ is said to be *archimedean* if there exists $b \in \text{Ideal}[p] + \text{Qmod}[q]$ such that $W(b) = \{x \in \mathbb{R}^n : b(x) \geq 0\}$ is a compact set. It is shown that $f \in \text{Ideal}[p] + \text{Qmod}[q]$ if $f > 0$ on $V(p) \cap W(q)$ and $\text{Ideal}[p] + \text{Qmod}[q]$ is archimedean [99]. This conclusion is often referenced as *Putinar's Positivstellensatz*. When f is only nonnegative (but not strictly positive) on $V(p) \cap W(q)$, we still have $f \in \text{Ideal}[p] + \text{Qmod}[q]$ under some generic conditions. This result is shown in [80].

Consider a polynomial optimization problem

$$\left\{ \begin{array}{rl} f_{min} := \min\limits_{x \in \mathbb{R}^n} & f(x) \\ s.t. & p(x) = 0,\ q(x) \geq 0, \end{array} \right. \tag{2.1}$$

where $f \in \mathbb{R}[x]$ and p, q are tuples of polynomials in x. The feasible set of problem (2.1) is $V(p) \cap W(q)$. It is obvious that a scalar $\gamma \leq f_{min}$ if and only if $f - \gamma \geq 0$ on $V(p) \cap W(q)$, which can be ensured by the membership $f - \gamma \in \text{Ideal}[p] + \text{Qmod}[q]$. The kth order SOS relaxation of (2.1) is

$$\left\{ \begin{array}{rl} f_k := \max & \gamma \\ s.t. & f - \gamma \in \text{Ideal}[p]_{2k} + \text{Qmod}[q]_{2k}. \end{array} \right. \tag{2.2}$$

Its dual problem is the kth moment relaxation of(2.1). The asymptotic convergence $f_k \to f_{min}$ as $k \to \infty$ was shown in [59]. Under the archimedeanness and some classical optimality conditions, it holds that $f_k = f_{min}$ for all k big enough, as shown in [80].

2.3 Localizing and moment matrices

For a given dimension n and degree d, denote by $\mathbb{R}^{\mathbb{N}_d^n}$ the space of real vectors that are indexed by $\alpha \in \mathbb{N}_d^n$, i.e.,

$$\mathbb{R}^{\mathbb{N}_d^n} := \{y = (y_\alpha)_{\alpha \in \mathbb{N}_d^n} : y_\alpha \in \mathbb{R}\}.$$

Each vector in $\mathbb{R}^{\mathbb{N}_d^n}$ is called a truncated multi-sequence (tms) of degree d. A tms $y \in \mathbb{R}^{\mathbb{N}_d^n}$ gives the linear functional $\mathscr{L}_y$ acting on $\mathbb{R}[x]_d$ as

$$\mathscr{L}_y\Big(\sum_{\alpha \in \mathbb{N}_d^n} f_\alpha x^\alpha\Big) := \sum_{\alpha \in \mathbb{N}_d^n} f_\alpha y_\alpha.$$

The $\mathscr{L}_y$ is called a Riesz functional. For $f \in \mathbb{R}[x]_d$ and $y \in \mathbb{R}^{\mathbb{N}_d^n}$, we denote

$$\langle f, y \rangle := \mathscr{L}_y(f).$$

For a polynomial $p \in \mathbb{R}[x]_{2d}$, the dth *localizing matrix* of p associated to a tms $y \in \mathbb{R}^{\mathbb{N}^n_{2d}}$, is the symmetric matrix $L_p^{(d)}[y]$ such that

$$vec(a)^T \left(L_p^{(d)}[y] \right) vec(b) = \mathscr{L}_y(pab)$$

for all polynomials $a, b \in \mathbb{R}[x]_t$, with $t = d - \lceil \deg(p)/2 \rceil$. In the above, the $vec(a)$ denotes the coefficient vector of the polynomial a. For instance, when $n = 3$ and $p = x_1x_2 - x_3^3$, for $y \in \mathbb{R}^{\mathbb{N}^3_6}$, we have

$$L_p^{(3)}[y] = \begin{bmatrix} y_{110} - y_{003} & y_{210} - y_{103} & y_{120} - y_{013} & y_{111} - y_{004} \\ y_{210} - y_{103} & y_{310} - y_{203} & y_{220} - y_{113} & y_{211} - y_{104} \\ y_{120} - y_{013} & y_{220} - y_{113} & y_{130} - y_{023} & y_{121} - y_{014} \\ y_{111} - y_{004} & y_{211} - y_{104} & y_{121} - y_{014} & y_{112} - y_{005} \end{bmatrix}.$$

For the special case of constant one polynomial $p = 1$, $L_1^{(d)}[y]$ is reduced to the so-called *moment matrix*

$$M_d[y] \ := \ L_1^{(d)}[y]. \tag{2.3}$$

The columns and rows of $L_p^{(d)}[y]$, as well as $M_d[y]$, are labelled by $\alpha \in \mathbb{N}^n$ with $2|\alpha| + \deg(p) \leq 2d$. For instance, for $n = 3$ and $y \in \mathbb{R}^{\mathbb{N}^3_4}$, we have

$$M_2[y] = \begin{bmatrix} y_{000} & y_{100} & y_{010} & y_{001} & y_{200} & y_{110} & y_{101} & y_{020} & y_{011} & y_{002} \\ y_{100} & y_{200} & y_{110} & y_{101} & y_{300} & y_{210} & y_{201} & y_{120} & y_{111} & y_{102} \\ y_{010} & y_{110} & y_{020} & y_{011} & y_{210} & y_{120} & y_{111} & y_{030} & y_{021} & y_{012} \\ y_{001} & y_{101} & y_{011} & y_{002} & y_{201} & y_{111} & y_{102} & y_{021} & y_{012} & y_{003} \\ y_{200} & y_{300} & y_{210} & y_{201} & y_{400} & y_{310} & y_{301} & y_{220} & y_{211} & y_{202} \\ y_{110} & y_{210} & y_{120} & y_{111} & y_{310} & y_{220} & y_{211} & y_{130} & y_{121} & y_{112} \\ y_{101} & y_{201} & y_{111} & y_{102} & y_{301} & y_{211} & y_{202} & y_{121} & y_{112} & y_{103} \\ y_{020} & y_{120} & y_{030} & y_{021} & y_{220} & y_{130} & y_{121} & y_{040} & y_{031} & y_{022} \\ y_{011} & y_{111} & y_{021} & y_{012} & y_{211} & y_{121} & y_{112} & y_{031} & y_{022} & y_{013} \\ y_{002} & y_{102} & y_{012} & y_{003} & y_{202} & y_{112} & y_{103} & y_{022} & y_{013} & y_{004} \end{bmatrix}.$$

We can use the moment matrix and localizing matrices to describe dual cones of quadratic modules. For a polynomial tuple $q = (q_1, \ldots, q_t)$ and a degree $k \geq \lceil \deg(q)/2 \rceil$, define the tms cone

$$\mathscr{S}[q]_{2k} \coloneqq \left\{ y \in \mathbb{R}^{\mathbb{N}^n_{2k}} : \, M_k[y] \succeq 0, \, L_{q_1}^{(k)}[y] \succeq 0, \ldots, L_{q_t}^{(k)}[y] \succeq 0 \right\}.$$

It can be verified that (see [82]) $\mathscr{S}[q]_{2k}$ is the dual cone of $\text{Qmod}[g]_{2k}$, i.e.,

$$\left(\text{Qmod}[g]_{2k}\right)^* = \mathscr{S}[q]_{2k}.$$

2.4 Truncated moment problems

Let $x = (x_1, \ldots, x_p)$. A tms $y = (y_\alpha) \in \mathbb{R}^{\mathbb{N}_d^n}$ is said to admit a representing measure μ supported in a set $S \subseteq \mathbb{R}^n$ if $y_\alpha = \int x^\alpha \mathtt{d}\mu$ for all $\alpha \in \mathbb{N}_d^n$. Such a measure μ is called an S-representing measure for $y1$. In particular, if $y = 0$ is the zero tms, then it admits the identically zero measure. We refer to [24, 46, 81] for recent work on truncated moment problems.

Denote by $meas(y, S)$ the set of S-measures admitted by y. It gives the moment cone

$$\mathscr{R}_d(S) \coloneqq \{z \in \mathbb{R}^{\mathbb{N}_d^n} \mid meas(y, S) \neq \emptyset\}.$$

The $\mathscr{R}_d(S)$ can also be written as the conic hull

$$\mathscr{R}_d(S) = cone(\{[x]_d : x \in S\}).$$

Recall that $\mathscr{P}_d(S)$ denotes the cone of polynomials in $\mathbb{R}[x]_d$ that are nonnegative on S. It is a closed and convex cone. For all $f \in \mathscr{P}_d(S)$ and $y \in \mathscr{R}_d(S)$, it holds that for every $\mu \in meas(y, S)$,

$$\langle f, y\rangle = \sum_{\alpha \in \mathbb{N}_d^n} f_\alpha y_\alpha = \int f(x)\mathtt{d}\mu \geq 0.$$

This implies that $\mathscr{R}_d(S)^* = \mathscr{P}_d(S)$. When S is compact, we also have $\mathscr{P}_d(S)^* = \mathscr{R}_d(S)$. If S is not compact, then

$$\mathscr{P}_d(S)^* = \overline{\mathscr{R}_d(S)}.$$

We refer to [65, Section 5.2] and [82] for this fact.

Suppose S is the semi-algebraic set determined by a polynomial tuple $g = (g_1, \ldots, g_m)$ in x. For an integer $k \geq \deg(g)/2$, a tms $y \in \mathbb{R}^{\mathbb{N}_{2k}^n}$ admits an S-representing measure μ if $y \in \mathscr{S}[g]_{2k}$ and

$$\operatorname{rank} M_{k-d_0}[y] = \operatorname{rank} M_k[y],$$

where $d_0 = \lceil \deg(g)/2 \rceil$. Moreover, the measure μ is unique and is r-atomic, i.e., $|supp(\mu)| = r$, where $r = \operatorname{rank} M_k[y]$. The above rank condition is called *flat extension* or *flat truncation* [23, 79]. When it holds, the tms y is said to be a *flat* tms. When y is flat, one can obtain the unique representing measure μ for y by computing Schur decompositions and eigenvalues (see [47]).

To obtain a representing measure for a tms $y \in \mathbb{R}^{\mathbb{N}_d^n}$ that is not flat, a semidefinite relaxation method is proposed in [81]. Suppose S is compact and the quadratic module

Qmod$[g]$ is archimedean. Fix a generic polynomial $R \in \Sigma[x]_{2k}$, with $2k > \deg(g)$. Then we solve the moment optimization

$$\begin{cases} \min\limits_{\omega} & \langle R, \omega \rangle \\ s.t. & \omega|_d = y,\, \omega \in \mathscr{S}[g]_{2k}. \end{cases} \tag{2.4}$$

In the above $\omega|_d$ denotes the dth degree truncation of ω, i.e.,

$$\omega|_d \;:=\; (\omega_\alpha)_{|\alpha| \le d}.$$

As k increases, by solving (2.4), one can either get a flat extension of y, or a certificate that y does not have any representing measure. We refer to [81] for more details about solving truncated moment problems.

2.5 Constraint qualifications

Consider the optimization problem

$$\begin{cases} \min\limits_{x \in \mathbb{R}^n} & b(x) \\ s.t. & c_i(x) = 0\,(i \in \mathcal{E}), \\ & c_j(x) \ge 0\,(j \in \mathcal{I}), \end{cases}$$

where b and each c_i, c_j are continuously differentiable, and $\mathcal{E}$, $\mathcal{I}$ are finite index sets. For a feasible point u, denote the active index set of inequalities at u,

$$\mathcal{I}(u) := \{j \in \mathcal{I} : c_j(u) = 0\}.$$

The Karush-Kuhn-Tucker (KKT) condition is said to hold at u if there exist Lagrange multipliers λ_j such that

$$\sum_{j \in \mathcal{E} \cup \mathcal{I}} \lambda_j \nabla c_j(u) = \nabla b(u), \quad \lambda_j \ge 0,\ \lambda_j c_j(u) = 0\,(j \in \mathcal{I}(u)).$$

A feasible point $\bar{x}$ is called a *KKT point* or *critical point* if it satisfies the KKT condition. A local minimizer must be a KKT point if all functions are linear. For nonlinear optimization, certain constraint qualifications are required for KKT points. The *linearly independent constraint qualification* (LICQ) is said to hold at u if the gradient set $\{\nabla c_j(u)\}_{j \in \mathcal{E} \cup \mathcal{I}(u)}$ is linearly independent. The *Mangasarian-Fromovitz constraint qualification* (MFCQ) is said to hold at u if there exists a vector $v \in \mathbb{R}^n$ satisfying

$$\nabla c_i(u)^T v = 0\,(i \in \mathcal{E}), \quad \nabla c_i(u)^T v > 0\,(i \in \mathcal{I}(u)).$$

The MFCQ is equivalent to the following statement

$$\sum_{j \in \mathcal{E} \cup \mathcal{I}(u)} \lambda_j \nabla c_j(u) = 0,\ \lambda_j \geq 0\,(j \in \mathcal{I}(u)) \quad \Longrightarrow \quad \lambda = 0.$$

When the functions $c_i(x)(i \in \mathcal{E})$ are linear and $c_j(x)(j \in \mathcal{I}(u))$ are concave, the *Slater's condition* is said to hold if there exists $\bar{x}$ such that

$$c_i(\bar{x}) = 0(i \in \mathcal{E}), \quad c_i(\bar{x}) > 0(i \in \mathcal{I}).$$

The Slater's condition is equivalent to the MFCQ under the convexity assumption. If the MFCQ holds at a local minimizer $\bar{x}$, then $\bar{x}$ is a KKT point and the set of Lagrange multipliers is compact. If LICQ holds at $\bar{x}$, then the set of Lagrange multipliers is a singleton. We refer to [8] for constraint qualifications in nonlinear programming.

Chapter 3

Stochastic Polynomial Optimization

3.1 Stochastic polynomial optimization problems

Let $x = (x_1, \dots, x_n)$ be the decision variable and $\xi = (\xi_1, \dots, \xi_p)$ be a random vector. A typical stochastic optimization problem is

$$\min_{x \in K} \quad f(x) := \mathbb{E}[F(x, \xi)] \tag{3.1}$$

where $K \subseteq \mathbb{R}^n$, F is a function in (x, ξ), and the symbol $\mathbb{E}$ denotes the expectation of a function in the random vector ξ. The *stochastic polynomial optimization*, which is the stochastic optimization defined with polynomials.

Assume F is a polynomial in x with measurable coefficients in ξ, i.e.,

$$F(x, \xi) := \sum_{\alpha \in \mathbb{R}^{\mathbb{N}^n}} c_\alpha(\xi) x^\alpha,$$

and K is a semialgebraic set given by

$$K := \{x \in \mathbb{R}^n : g_1(x) \geq 0, \dots, g_m(x) \geq 0\}. \tag{3.2}$$

Let $g = (g_1, \dots, g_m)$. We study the stochastic polynomial optimization

$$\begin{cases} \min & f(x) := \mathbb{E}\big[F(x, \xi)\big] \\ s.t. & g(x) \geq 0. \end{cases} \tag{3.3}$$

The $F(x, \xi)$ is a measurable function in ξ for each $x \in K$, so $f(x) := \mathbb{E}[F(x, \xi)]$ is a polynomial in x. The coefficients of f are typically not known explicitly. In practice, they are usually approximated by sample averages of ξ.

3.2 Perturbation sample average approximation

Let $\xi^{(1)}, \ldots, \xi^{(N)}$ be given samples for the random vector ξ. The sample average function of f is

$$f_N(x) := \frac{1}{N}\sum_{k=1}^{N} F(x, \xi^{(k)}).$$

If $F(x, \xi)$ is a polynomial in x, then the f_N is also a polynomial in x. If each sample $\xi^{(k)}$ obeys the same distribution of ξ, then $\mathbb{E}[f_N(x)] = f(x)$. Furthermore, when all $\xi^{(k)}$ are independently identically distributed, the Law of Large Numbers [50] implies that

$$f_N(x) \to f(x) \qquad \text{as } N \to \infty,$$

with probability one and under some regularity conditions.

We propose a perturbation sample average approximation (PSAA) model of (3.3)

$$\begin{cases} \min & f_N(x) + \epsilon \|[x]_{2d}\| \\ s.t. & g(x) \geq 0, \end{cases} \tag{3.4}$$

where $\epsilon > 0$ is a small parameter, $d = \max\{\lceil \deg(f_N)/2 \rceil, \lceil \deg(g)/2 \rceil\}$ and

$$[x]_{2d} = \begin{bmatrix} 1 & x_1 & \cdots & x_n & x_1^2 & x_1x_2 & \cdots & x_n^{2d} \end{bmatrix}^T.$$

In particular, when $\epsilon = 0$, the (3.4) is reduced to be the sample average approximation (SAA) of (3.3),

$$\begin{cases} \min & f_N(x) \\ s.t. & g(x) \geq 0. \end{cases} \tag{3.5}$$

The (3.4)-(3.5) can be solved globally by Lasserre type moment relaxations [59]. If we replace the monomial vector $[x]_{2d}$ by a tms $y \in \mathbb{R}^{\mathbb{N}^n_{2d}}$, the (3.4) is relaxed to the following convex optimization

$$\begin{cases} \min & \langle f_N, y \rangle + \epsilon \|y\| \\ s.t. & M_d[y] \succeq 0,\, L_{g_i}^{(d)}[y] \succeq 0 \, (i \in [m]), \\ & y_0 = 1,\, y \in \mathbb{R}^{\mathbb{N}^n_{2d}}. \end{cases} \tag{3.6}$$

The (3.6) is equivalent to a linear conic optimization problem with a Cartesian product of semidefinite cones and a second order cone. The relaxation (3.6) is said to be *tight* if its optimal value is the same as that of (3.4). The equality constraint $y_0 = 1$ means that the first entry of y is equal to one. The set of all y satisfying linear matrix inequalities in (3.6) is just the cone $\mathscr{S}(g)_{2d}$. The cone $\mathscr{S}(g)_{2d}$ and the truncated quadratic module $\mathrm{Qmod}[g]_{2d}$ are dual to each other. Therefore, the Lagrange function for (3.6) is

$$\begin{aligned}\mathcal{L}(y,q,\gamma) &= \langle f_N, y\rangle + \epsilon\|y\| - \langle q, y\rangle - \gamma(y_0 - 1)\\ &= \langle f_N - q - \gamma, y\rangle + \epsilon\|y\| + \gamma,\end{aligned}$$

for dual variables $q \in \mathrm{Qmod}[g]_{2d}$ and $\gamma \in \mathbb{R}$. The function $\mathcal{L}(y,q,\gamma)$ has a finite minimum value for $y \in \mathbb{R}^{\mathbb{N}^n_{2d}}$ if and only if

$$\|vec(f_N - q - \gamma)\| \le \epsilon,$$

for which case the minimum value is γ. (The $vec(p)$ denotes the coefficient vector of p.) Therefore, the dual optimization problem of (3.6) is

$$\begin{cases} \max & \gamma \\ s.t. & f_N - p - \gamma \in \mathrm{Qmod}[g]_{2d}, \\ & \|vec(p)\| \le \epsilon, p \in \mathbb{R}[x]_{2d}. \end{cases} \tag{3.7}$$

Because the sample average $f_N(x)$ is only an approximation for $f(x)$, it is possible that there is no scalar γ such that $f_N - \gamma \in \mathrm{Qmod}[g]_{2d}$. The perturbation term $\epsilon\|y\|$ in (3.6) motivates us to find the maximum γ such that $f_N - p - \gamma \in \mathrm{Qmod}[g]_{2d}$, for some polynomial p whose coefficient vector has a small norm. This leads to the following algorithm.

Algorithm 3.1. *Generate samples $\xi^{(1)}, \ldots, \xi^{(N)}$, according to the distribution of ξ. Choose a small perturbation parameter $\epsilon > 0$.*

Step 1 *Compute the sample average $f_N = N^{-1}\sum_{k=1}^N F(x, \xi^{(k)})$.*

Step 2 *Solve the semidefinite relaxation problem (3.6). If (3.7) is infeasible, increase the value of ϵ (e.g., let $\epsilon := 2\epsilon$), until (3.6) has a minimizer, which we denote as y^*.*

Step 3 *Let u be the projection of y^* as follows,*

$$u = \pi(y^*) := (y^*_{e_1}, \ldots, y^*_{e_n}).$$

Output u as a candidate minimizer for the sample average optimization with perturbation (3.4), and stop.

For $\epsilon > 0$, the minimizer of the relaxation (3.6) is always unique (if it exists), because its objective is strictly convex.

Theorem 3.2 ([93]). *Assume that u^* is a minimizer of (3.4) and y^* is a minimizer of (3.6). Then, for $\epsilon > 0$, the relaxation (3.6) is tight if and only if rank $M_d[y^*] = 1$. In particular, for the case rank $M_d[y^*] = 1$, the point $u = \pi(y^*)$ is a minimizer of (3.4).*

Proof. Let ϑ_1, ϑ_2 be optimal values of (3.4) and (3.6) respectively. It is clear $\vartheta_1 \geq \vartheta_2$.

Suppose rank$M_d[y^*] = 1$, then for $u = \pi(y^*)$ one can show that $M_d[y^*] = [u]_d([u]_d)^T$. Hence, $y^* = [u]_{2d}$, $\langle f_N, y^*\rangle = f_N(u)$, and each $g_i(u) \geq 0$ (see [47, 79]). So, u is a feasible point of (3.4) and

$$\vartheta_1 \leq f_N(u) + \epsilon\|[u]_{2d}\| = \langle f_N, y^*\rangle + \epsilon\|y^*\| = \vartheta_2.$$

Therefore, $\vartheta_1 = \vartheta_2$, u is a minimizer of (3.4), and the relaxation (3.6) is tight.

Suppose relaxation (3.6) is tight. Then $\vartheta_1 = \vartheta_2$ and $\tilde{y} := [u^*]_{2d}$ is a minimizer of (3.6). This is because $f_N(u^*) = \langle f_N, \tilde{y}\rangle$ and $\|[u^*]_{2d}\| = \|\tilde{y}\|$. For $\epsilon > 0$, the objective of (3.6) is strictly convex, so its minimizer must be unique. Hence, $\tilde{y} = y^*$ and

$$M_d[y^*] \;=\; M_d[\tilde{y}] \;=\; [u^*]_d([u^*]_d)^T.$$

Therefore, rank $M_d[y^*] = $ rank $M_d[\tilde{y}] = 1$. □

When the sample average $f_N(x)$ is unbounded from below on the feasible set K, the moment relaxation (3.6) might still be unbounded from below if $\epsilon > 0$ is small. However, if $\epsilon > 0$ is big, then (3.6) must be feasible and has a minimizer. Indeed, we have the following theorem.

Theorem 3.3 ([93]). *Suppose the feasible set K has nonempty interior. If $\epsilon > 0$ is big, both (3.6) and (3.7) have optimizers and their optimal values are the same.*

Proof. When K has nonempty interior, the quadratic module Qmod$[g]_{2d}$ is a closed cone (see [65, Theorem 3.49]) and the cone $\mathscr{S}(g)_{2d}$ has nonempty interior. For instance, let ν be the Gaussian measure, then the tms

$$\hat{y} \;:=\; \frac{1}{\nu(K)}\int_K [x]_{2d}\mathtt{d}\nu(x)$$

is an interior point of the cone $\mathscr{S}(g)_{2d}$. In other words, $M_d[\hat{y}] \succ 0$ and all $L_{g_i}^{(d)}[\hat{y}] \succ 0$. This is because $\int_K p^2 d\nu > 0$ and $\int_K g_i p^2 d\nu > 0$ for all nonzero polynomials p. Moreover, $\hat{y}_0 = 1$. The convex relaxation (3.6) is strictly feasible (i.e., there is a feasible y such that each matrix in (3.6) is positive definite). When $\epsilon > 0$ is big, the SOS relaxation (3.7) is also strictly feasible. For instance, for the choice

$$\epsilon > \|f_N - [x]_d^T[x]_d\|,\ \hat{p} = f_N - [x]_d^T[x]_d,\ \hat{\gamma} = 0,$$

we have that

$$f_N - \hat{p} - \hat{\gamma} = [x]_d^T[x]_d \in \operatorname{int}\Big(\Sigma[x]_{2d}\Big) \subseteq \operatorname{int}\Big(\mathrm{Qmod}[g]_{2d}\Big).$$

In the above, the symbol "int" denotes the interior of a set. Therefore, for big $\epsilon > 0$, both (3.6) and (3.7) have strictly feasible points. By the strong duality theorem (see [7, 12]), they have the same optimal value and they both achieve the optimal value, i.e., they have optimizers. □

The value of ϵ influences the performance of the PSAA model. In applications, we often choose a small $\epsilon > 0$, because we expect that (3.4) is a good approximation for (3.5). However, when $\epsilon > 0$ is too small, (3.6) might be unbounded from below and has no minimizers. For efficiency, we often anticipate the smallest value of ϵ such that (3.6) is bounded from below and has a minimizer. When K has nonempty interior, the relaxation (3.6) is strictly feasible, i.e., there exists $\hat{y}$ such that all the matrices $M_d[\hat{y}]$ and $L_{g_i}^{(d)}[\hat{y}]$ are positive definite. Therefore, the strong duality holds between (3.6) and (3.7). To ensure that (3.6) is solvable (i.e., it has a minimizer), we need the dual optimization problem (3.7) to be feasible. Consider the optimization problem

$$\left\{\begin{array}{rl} \epsilon^* := \min & \|vec(p)\| \\ s.t. & f_N - p - \gamma \in \mathrm{Qmod}[g]_{2d}, \\ & \gamma \in \mathbb{R},\ p \in \mathbb{R}[x]_{2d}. \end{array}\right. \tag{3.8}$$

The above is a convex optimization problem with semidefinite constraints. In computational practice, we often choose $\epsilon > 0$ in a heuristic way, e.g., $\epsilon = 10^{-2}$. If such ϵ is not enough, we can increase its value until (3.6) performs well.

3.3 Numerical experiments

This section gives numerical experiments of applying Algorithm 3.1 to solve stochastic polynomial optimization. The computation is implemented in `MATLAB` R2018a, in a Laptop with CPU 8th Generation Intel® Core™ i5-8250U and RAM 16 GB. The moment relaxation

(3.6) is solved by the software `GloptiPoly 3` [48], which calls the semidefinite program solver `SeDuMi` [105]. We solve the PSAA model (3.4) by the relaxation (3.6). The (3.6) is said to be *solvable* if it has a minimizer y^*. We denote $u = \pi(y^*)$, if y^* exists. Otherwise, (3.6) is said to be *not solvable*, and we use "n.a." to indicate data not available. The algorithm running time is reported with the unit second. For convenience, the computational accuracy 10^t is denoted as "$e+t$". For the optimization (3.3), we denote its optimal value and minimizer by f_{min} and v^*, respectively. When ξ is approximated by samplings, we use $\bar{\xi} = (\bar{\xi}_1, \ldots, \bar{\xi}_p)$ to denote its sample average. For a point $a \in \mathbb{R}^p$, δ_a denotes the Dirac function supported at a.

Example 3.4. *Consider the stochastic polynomial optimization*

$$\begin{cases} \min\limits_{x\in\mathbb{R}^4} & f(x) = \mathbb{E}[F(x) + H(x,\xi)] \\ s.t. & x_1x_3 + 1 \geq x_2^2 + x_4^2, \\ & x_2x_3 - x_1x_4 + 2 \geq 0, \\ & x_1^3 + x_2^3 + x_3^3 + x_4^3 \leq 8, \\ & x_1,\, x_2,\, x_3,\, x_4 \geq 0, \end{cases} \tag{3.9}$$

where ($\mathcal{N}(\mu, P)$ denotes the normal distribution with the mean μ and the covariance P)

$$\begin{aligned} &G(x) = x_1^2x_2^2 + x_2^2x_3^2 + (1 - x_2x_3)^2 + (3 - x_1x_4)^2 + x_1x_2x_3x_4, \\ &H(x,\xi) = \xi_1x_1x_2^2x_3 + \xi_2x_2^2x_4^2, \\ &\xi \sim \mathcal{N}\left(\begin{bmatrix} -0.41 \\ -2.50 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right). \end{aligned}$$

If $f(x)$ is evaluated exactly,we can get the optimal value and the minimizer of (3.9),

$$f_{min} = 1.0655, \quad v^* = (1.5829,\, 0.6427,\, 0.9316,\, 1.4358).$$

In practice, the sample average usually does not equal the exact expectation. We explore the performance of Algorithm 3.1 for the following sample averages (Note that $\mathbb{E}[\xi] = (-0.41, -2.50)$).

$$(a).\, \bar{\xi} = (-0.42, -2.51), \quad (b).\, \bar{\xi} = (-0.42, 2.50), \quad (c).\, \bar{\xi} = (-0.41, -2.51).$$

The numerical results are reported in Table 3.1. The PSAA model (3.4) performs better than the classical SAA model (3.5) (i.e. $\epsilon = 0$). For each case, (3.4) gives more reliable optimizers; for (a) and (b), solving the relaxation (3.6) costs less computational time.

Table 3.1: Performance of PSAA for Example 3.4

	ϵ	0	10^{-4}	10^{-3}	10^{-2}
(a)	solvable?	yes	yes	yes	yes
	time	0.37	0.29	0.15	0.26
	$\lvert\langle f_N, y^*\rangle - f_N(u)\rvert$	1.81e+05	1.29e-08	5.28e-09	9.53e-09
	$\lvert\langle f_N, y^*\rangle - f_{min}\rvert$	1.81e+05	1.48e-02	1.48e-02	1.47e-02
(b)	ϵ	0	10^{-4}	10^{-3}	10^{-2}
	solvable?	yes	yes	yes	yes
	time	0.23	0.12	0.14	0.12
	$\lvert\langle f_N, y^*\rangle - f_N(u)\rvert$	1.87e+05	1.29e-08	5.61e-09	9.64e-09
	$\lvert\langle f_N, y^*\rangle - f_{min}\rvert$	1.87e+05	6.13e-03	6.12e-03	6.12e-03
(c)	ϵ	0	10^{-4}	10^{-3}	10^{-2}
	solvable?	yes	yes	yes	yes
	time	0.12	0.13	0.10	0.09
	$\lvert\langle f_N, y^*\rangle - f_N(u)\rvert$	5.58e-02	1.29e-08	5.61e-09	9.70e-09
	$\lvert\langle f_N, y^*\rangle - f_{min}\rvert$	1.40e-02	8.56e-03	8.56e-03	8.55e-03

Example 3.5. *Consider the stochastic polynomial optimization*

$$\begin{cases} \min\limits_{x\in\mathbb{R}^3} & f(x) := \mathbb{E}[G(x) + H(x,\xi)] \\ s.t. & 1 - e^T x \geq 0,\ x \geq 0, \end{cases} \tag{3.10}$$

where $e = (1,\ldots,1)^T$, ($Ber(p)$, $Geo(p)$ respectively denote the Bernoullian and geometric distributions with success probability p)

$$\begin{aligned} G(x) &= x_1^4 + x_1x_2x_3 + x_3(1 - x_1^2 - x_2^2), \\ H(x,\xi) &= 2\xi_1 x_2^4 - 4\xi_1 x_1^2 x_2^2 - \xi_2 x_1 x_2, \\ \xi_1 &\sim Ber(0.5), \quad \xi_2 \sim Geo(0.5). \end{aligned}$$

The feasible set K of (3.10) is a simplex, which is closed and compact. The associated quadratic module satisfies the archimedean condition. It implies that the sample average $f_N(x)$ is bounded from below on K and it has a minimizer, for all samples $\xi^{(i)}$. For this example, $\mathbb{E}(\xi_1) = 0.5$ and $\mathbb{E}(\xi_2) = 2$, so

$$f(x) = (x_1^2 - x_2^2)^2 + x_3(1 - x_1^2 - x_2^2) - x_1x_2(2 - x_3).$$

The optimal value and minimizer of (3.3) are

$$f_{min} = -0.5, \quad v^* = (0.5,\, 0.5,\, 0).$$

We consider two cases of sample averages

$$\text{(a). } \bar{\xi} = (0.501, 2), \ \ \epsilon^* \approx 0.001155;$$
$$\text{(b). } \bar{\xi} = (0.5, 2.001), \ \ \epsilon^* \approx 7.5875 \cdot 10^{-9}.$$

In the above, ϵ^* is the minimum value of (3.8). We apply Algorithm 3.1 to solve (3.10).

Table 3.2: Performance of PSAA for Example 3.5.

(a)	ϵ	0	0.0012	0.004	0.008
	solvable?	no	yes	yes	yes
	time	0.12	0.10	0.07	0.09
	$\lvert\langle f_N, y^*\rangle - f_N(u)\rvert$	n.a.	5.31e-03	3.36e-04	1.09e-04
	$\lvert\langle f_N, y^*\rangle - f_{min}\rvert$	n.a.	5.44e-03	4.61e-04	2.34e-04
(b)	ϵ	0	10^{-4}	10^{-3}	10^{-2}
	solvable?	yes	yes	yes	yes
	time	0.08	0.08	0.07	0.06
	$\lvert\langle f_N, y^*\rangle - f_N(u)\rvert$	5.03e-09	1.84e-09	1.61e-09	1.86e-09
	$\lvert\langle f_N, y^*\rangle - f_{min}\rvert$	2.50e-04	2.50e-04	2.50e-04	2.50e-04

The numerical results are reported in Table 3.2. The PSAA model (3.4) performs very well for both cases. Compared with the classical SAA model (3.5) (i.e., $\epsilon = 0$), it has quite clear advantages for case (a). It successfully returned a good minimizer, while (3.5) is unbounded from below and does not retrun a minimizer.

Example 3.6. *Consider the unconstrained stochastic optimization*

$$\min_{x \in \mathbb{R}^4} \mathbb{E}[G(x) + H(x, \xi)] \tag{3.11}$$

where ($\mathcal{P}(\lambda)$ denotes the Posisson distribution with parameter $\lambda > 0$)

$$\xi \sim \mathcal{P}(2), \ \ G(x) = (x_3 - x_4)^4 + (x_1 + x_2)^4 + x_1^2 + x_2^2 + x_3^2 + x_4^2,$$
$$H(x, \xi) = \xi - (\xi^2 - 2\xi)(x_1 - x_4) - 2(\xi - 1)(x_3 - x_4)^2(x_1 + x_2)^2.$$

Evaluating the expectation, we get $\mathbb{E}(\xi) = 2, \mathbb{E}(\xi^2) = 6$ and

$$f(x) = \left((x_3 - x_4)^2 - (x_1 + x_2)^2\right)^2 + (x_1 - 1)^2 + (1 + x_4)^2 + x_2^2 + x_3^2.$$

The optimal value and minimizer of (3.3) are

$$f_{min} = 0, \quad v^* = (1,\, 0,\, 0,\, -1).$$

For convenience, denote the sample averages

$$s_1 := \bar{\xi} = \frac{1}{N}\sum_{k=1}^{N} \xi^{(k)}, \quad s_2 := \frac{1}{N}\sum_{k=1}^{N} (\xi^{(k)})^2.$$

We make samples of different sizes and compute ϵ^* in (3.8) for each case.

Table 3.3: The values of ϵ^* for Example 3.6.

	(a)	(b)	(c)	(d)
N	500	1000	5000	10000
s_1	2.11	1.96	2.01	2.02
s_2	6.43	5.71	6.13	6.07
ϵ^*	0.807543	3.3618e-10	0.073413	0.146826

We focus on cases (c) and (d). By applying Algortihm 3.1, the numerical results are

Table 3.4: Performance of PSAA for Example 3.6.

	(c)			(d)		
ϵ	0	ϵ^*	0.1	0	ϵ^*	0.2
solvable?	no	yes	yes	no	yes	yes
time	0.23	0.20	0.11	0.10	0.16	0.07
$\lvert\langle f_N, y^*\rangle - f_N(u)\rvert$	n.a.	5.30e+01	2.02e-01	n.a.	6.47e+01	2.28e-01
$\lvert\langle f_N, y\rangle - f_{min}\rvert$	n.a.	5.39e+01	3.90e-01	n.a.	6.49e+01	2.96e-01
$\lVert u - v^*\rVert$	n.a.	3.09e-02	1.28e-01	n.a.	5.86e-02	2.73e-01
u	n.a.	$\begin{bmatrix} 1.0216 \\ 0.0035 \\ 0.0035 \\ -1.0216 \end{bmatrix}$	$\begin{bmatrix} 0.9102 \\ 0.0071 \\ 0.0071 \\ -0.9102 \end{bmatrix}$	n.a.	$\begin{bmatrix} 0.9591 \\ 0.0059 \\ 0.0059 \\ -0.9591 \end{bmatrix}$	$\begin{bmatrix} 0.8070 \\ 0.0085 \\ 0.0085 \\ -0.8070 \end{bmatrix}$

reported in Table 3.4. The perturbation term in the PSAA model (3.4) makes a big difference for computing reliable minimizers. The PSAA model returned minimizers that are close to the optimizer of (3.3), while the classical SAA model (i.e., $\epsilon = 0$) is unbounded from below lland fails to return a minimizer.

Acknowledgments. This Chapter, in full, is a reprint of the material as it appears in *Optimization Methods and Software* 2020. The dissertation author coauthored this paper with Nie, Jiawang and Yang, Liu.

Chapter 4

Distributionally Robust Optimization

4.1 DRO of moments and polynomials

The distributionally robust optimization (DRO) is

$$\begin{cases} \min\limits_{x \in X} & f(x) \\ s.t. & \inf\limits_{\mu \in \mathcal{M}} \mathbb{E}_\mu[h(x,\xi)] \geq 0, \end{cases} \tag{4.1}$$

where $x := (x_1, \ldots, x_n)$ is the decision variable constrained in a set $X \subseteq \mathbb{R}^n$ and $\xi := (\xi_1, \ldots, \xi_p) \in \mathbb{R}^p$ is the random variable obeying the distribution of a measure $\mu \in \mathcal{M}$. The notation $\mathbb{E}_\mu[h(x,\xi)]$ stands for the expectation of the random function $h(x,\xi)$ with respect to the distribution of ξ. The set $\mathcal{M}$ is called the *ambiguity set*, which is used to describe the uncertainty of the measure μ.

We are interested in the moment-based ambiguity sets. In this case, $\mathcal{M}$ contains Borel measures that are constrained by moment bounds. For $S \subseteq \mathbb{R}^p$, let $\mathcal{B}(S)$ denote the set of Borel measures supported in S. Let Y be a nonempty subset of $\mathbb{R}^{\mathbb{N}^p_d}$. We assume the ambiguity set is given as

$$\mathcal{M} := \Big\{\mu \in \mathcal{B}(S) : \mathbb{E}_\mu([\xi]_d) \in Y\Big\}, \tag{4.2}$$

where $[\xi]_d$ is the monomial vector

$$[\xi]_d := \begin{bmatrix} 1 & \xi_1 & \cdots & \xi_p & (\xi_1)^2 & \xi_1\xi_2 & \cdots & (\xi_p)^d \end{bmatrix}^T.$$

The optimization (4.1) equipped with the above ambiguity set is called the *distributionally robust optimization of moment* (DROM). Moreover, if all the defining functions are polynomials, we say (4.1) is the DROM with polynomials.

We focus on DROM with polynomials. For the DRO (4.1), assume $f \in \mathbb{R}[x]$ and $h(x,\xi)$ is a polynomial in ξ whose coefficients are linear in x, i.e.,

$$h(x,\xi) = (Ax+b)^T[\xi]_d, \quad A \in \mathbb{R}^{\binom{p+d}{d}\times n},\ b \in \mathbb{R}^{\binom{p+d}{d}}. \tag{4.3}$$

Suppose $\mathcal{M}$ is given as in (4.2), where S is a semialgebraic set

$$S = \{\xi \in \mathbb{R}^p : g_1(\xi) \geq 0, \ldots, g_{m_1}(\xi) \geq 0\}. \tag{4.4}$$

The $g \coloneqq (g_1, \ldots, g_{m_1})$ is a given tuple of polynomials in ξ. The Y is the constraining set for moments of μ up to a degree d. The set Y is not necessarily closed or convex. The closure of its conic hull is denoted as $\overline{cone(Y)}$. In computation, it is often a Cartesian product of linear, second-order or semidefinite cones. For instance, if ξ is a univariate random variable, $d = 4$, Y is the hypercube $[0,1]^5$ and $S = [a_1, a_2]$, then $cone(Y)$ is the nonnegative orthant. The constraining set X for x is assumed to be the set

$$X \coloneqq \{x \in \mathbb{R}^n \mid c_1(x) \geq 0, \ldots, c_{m_2}(x) \geq 0\}, \tag{4.5}$$

for a tuple $c = (c_1, \ldots, c_{m_2})$ of polynomials in x.

In addition, we remark that the distributionally robust min-max optimization

$$\min_{x\in X} \max_{\mu\in\mathcal{M}} \mathbb{E}_\mu[F(x,\xi)] \tag{4.6}$$

is a special case of the distributionally robust optimization in the form (4.1). Assume each $\mu \in \mathcal{M}$ is a probability measure (i.e., $\mathbb{E}_\mu[1] = 1$), then the min-max optimization (4.6) is equivalent to

$$\begin{cases} \min\limits_{(x,x_0)\in X\times\mathbb{R}} & x_0 \\ s.t. & \inf\limits_{\mu\in\mathcal{M}} \mathbb{E}_\mu[x_0 - F(x,\xi)] \geq 0. \end{cases} \tag{4.7}$$

This is a distributionally robust optimization problem in the form (4.1).

4.2 Moment optimization reformulation

In this section, we transform the DROM into a polynomial optimization with moment conic conditions. Let $\mathscr{R}_d(S)$ denote the moment cone of all degree-d tms' that admit S-measures. Then for every measure in $\mathcal{M}$, its truncated moment sequence of degree d must be contained in the intersection $\mathscr{R}_d(S) \cap cone(Y)$. In other words, we have the equivalence relations

$$\begin{aligned}\inf_{\mu \in \mathcal{M}} \mathbb{E}_\mu[h(x,\xi)] \geq 0 \quad &\Longleftrightarrow \quad \langle Ax+b, \mathbb{E}_\mu[\xi]_d\rangle \geq 0,\ \forall \mu \in \mathcal{M}, \\ &\Longleftrightarrow \quad (Ax+b)^T y \geq 0,\ \forall\, y \in \mathscr{R}_d(S) \cap cone(Y).\end{aligned}$$

The intersection of $\mathscr{R}_d(S)$ and $cone(Y)$ gives a cone

$$K \;=\; \mathscr{R}_d(S) \cap cone(Y). \tag{4.8}$$

Let K^* denote the dual cone of K. The problem (4.1) can be equivalently reformulated as

$$\begin{cases} \min\limits_{x \in X} & f(x) \\ s.t. & Ax+b \in K^*. \end{cases} \tag{4.9}$$

Then we characterize the moment cone K and its dual cone K^*. Observe that

$$\mathscr{R}_d(S)^* = \mathscr{P}_d(S), \quad \mathscr{P}_d(S)^* = \overline{\mathscr{R}_d(S)},$$
$$\Big(\mathscr{P}_d(S) + Y^*\Big)^* \;=\; \overline{\mathscr{R}_d(S)} \cap \overline{cone(Y)}.$$

In the above, Y^* is the dual cone of Y and $\mathscr{P}_d(S)$ is the set of polynomials with the highest degree d that are nonnegative on S. When both $\mathscr{R}_d(S)$ and $cone(Y)$ are closed, we have

$$\overline{\mathscr{R}_d(S) \cap cone(Y)} \quad = \quad \overline{\mathscr{R}_d(S)} \cap \overline{cone(Y)}. \tag{4.10}$$

Suppose (4.10) holds and the sum $\mathscr{P}_d(S) + Y^*$ is a closed cone, then we have

$$K^* \;=\; \mathscr{P}_d(S) + Y^*. \tag{4.11}$$

The (4.11) holds for most applications. We refer to [7, Proposition B.2.7] for a sufficient condition. Assume (4.11) is true, then (4.1) is equivalent to

$$\begin{cases} \min\limits_{x \in \mathbb{R}^n} & f(x) \\ s.t. & c_1(x) \geq 0, \ldots, c_{m_2}(x) \geq 0, \\ & h(x,\xi) \in \mathscr{P}_d(S) + Y^*, \end{cases} \tag{4.12}$$

where $c = (c_1, \ldots, c_{m_1})$ determines X as in (4.5). The membership constraint in (4.12) means that $h(x,\xi)$, as a polynomial in ξ, is the sum of a polynomial in $\mathscr{P}_d(S)$ and a polynomial in Y^*. When $f, c_1, \ldots, c_{m_2}$ are all linear functions, (4.12) is a linear conic optimization problem. When f and every c_i are polynomials, we can apply Moment-SOS relaxations to solve it.

Denote the degree

$$d_1 := \max\{\deg(f)/2, \lceil \deg(c)/2 \rceil\}, \tag{4.13}$$

and the projection map $\pi : \mathbb{R}^{\mathbb{N}^n_{2d_1}} \to \mathbb{R}^n$ such that

$$\pi(w) := (w_{e_1}, \ldots, w_{e_n}), \quad w \in \mathbb{R}^{\mathbb{N}^n_{2d_1}}. \tag{4.14}$$

The optimization (4.12) can be relaxed to

$$\begin{cases} \min\limits_{(x,w)} & \langle f, w \rangle \\ s.t. & M_{d_1}[w] \succeq 0,\, L_{c_i}^{(d_1)}[w] \succeq 0\,(i \in [m_2]), \\ & h(x,\xi) \in \mathscr{P}_d(S) + Y^*, \\ & w_0 = 1, x = \pi(w),\, w \in \mathbb{R}^{\mathbb{N}^n_{2d_1}}. \end{cases} \tag{4.15}$$

The dual optimization of (4.15) is

$$\begin{cases} \max\limits_{(\gamma,y)} & \gamma - \langle b, y \rangle \\ s.t. & f(x) - y^T A x - \gamma \in \mathrm{Qmod}[c]_{2d_1}, \\ & \gamma \in \mathbb{R},\, y \in \overline{K}. \end{cases} \tag{4.16}$$

The relaxation (4.15) is said to be *tight* if it has the same optimal value as (4.12) does. Under the SOS-convexity assumption, the relaxation (4.15) is equivalent to (4.12). This is the following result.

Theorem 4.1 ([94]). *For the DROM (4.1), assume the polynomials $f, -c_1, \ldots, -c_{m_2}$ are SOS-convex. Then, the optimization problems (4.15) and (4.12) are equivalent in the following sense: they have the same optimal value, and w^* is a minimizer of (4.15) if and only if $x^* := \pi(w^*)$ is a minimizer of (4.12).*

Proof. Let w be a feasible point for (4.15) and $x = \pi(w)$, then $Ax + b \in K^*$. Since $f, -c_1, \ldots, -c_{m_2}$ are SOS-convex, by the Jensen's inequality (see [60]), we have the following inequalities

$$f(x) = f(\pi(w)) \leq \langle f, w \rangle,$$

$$c_i(x) = c_i(\pi(w)) \geq \langle c_i, w \rangle,\, i = 1, \ldots, m_2.$$

The $(1,1)$-entry of $L_{c_i}^{(d_1)}[w]$ is $\langle c_i, w \rangle$, so $L_{c_i}^{(d_1)}[w] \succeq 0$ implies that $\langle c_i, w \rangle \geq 0$. This means that $x = \pi(w) \in X$ for every w that is feasible for (4.15). Let $f_0,\ f_1$ denote the optimal values of (4.12), (4.15) respectively. The $f_0 \geq f_1$ since (4.15) is a relaxation of (4.12). For every $\epsilon > 0$, there exists a feasible w such that $\langle f, w \rangle \leq f_1 + \epsilon$, which implies that $f(\pi(w)) \leq f_1 + \epsilon$. Hence $f_0 \leq f_1 + \epsilon$ for every $\epsilon > 0$. Therefore, $f_0 = f_1$, i.e., (4.15) and (4.12) have the same optimal value.

If w^* is a minimizer of (4.15), we also have $x^* = \pi(w^*) \in X$ and

$$f(x^*) = f(\pi(w^*)) \leq \langle f, w^* \rangle.$$

Since (4.15) is a relaxation of (4.12), they must have the same optimal value, and x^* is a minimizer of (4.12). Conversely, if x^* is a minimizer of (4.12), then $w^* := [x^*]_{2d_1}$ is feasible for (4.15) and $f(x^*) = \langle f, w^* \rangle$. So w^* must also be a minimizer of (4.15), since (4.15) and (4.12) have the same optimal value. $\square$

4.3 A Moment-SOS relaxation method

The S is a semialgebraic set given as in (4.4). For every integer $k \geq d/2$, it holds the nesting containment

$$\mathrm{Qmod}[g]_{2k} \cap \mathbb{R}[\xi]_d \subseteq \mathrm{Qmod}[g]_{2k+2} \cap \mathbb{R}[\xi]_d \subseteq \cdots \subseteq \mathscr{P}_d(S).$$

We thus consider the following restriction of (4.15):

$$\begin{cases} \min\limits_{(x,w)} & \langle f, w \rangle \\ s.t. & M_{d_1}[w] \succeq 0,\; L_{c_i}^{(d_1)}[w] \succeq 0\,(i \in [m_2]), \\ & h(x,\xi) \in \mathrm{Qmod}[g]_{2k} + Y^*, \\ & w_0 = 1, x = \pi(w), w \in \mathbb{R}^{\mathbb{N}^n_{2d_1}}, \end{cases} \tag{4.17}$$

where d_1 is given as in (4.13). The dual optimization of (4.17) is

$$\begin{cases} \max\limits_{(\gamma,y,z)} & \gamma - \langle b, y \rangle \\ s.t. & f(x) - y^T A x - \gamma \in \mathrm{Qmod}[c]_{2d_1}, \\ & \gamma \in \mathbb{R},\; z \in \mathscr{S}[g]_{2k},\; y \in \overline{cone(Y)},\; y = z|_d. \end{cases} \tag{4.18}$$

We would like to remark that $\mathrm{Qmod}[g]$ is a quadratic module in the polynomial ring $\mathbb{R}[\xi]$, and $\mathrm{Qmod}[c]$ is a quadratic module in the polynomial $\mathbb{R}[x]$. The notation $z|_d$ denotes the degree-d truncation of z. The optimization (4.18) is a relaxation of (4.16), since it has a bigger feasible set. There exist both quadratic module and moment constraints in (4.18). The primal-dual pair (4.17)-(4.18) can be solved as semidefinite programs. The following is a basic property about the above optimization.

Theorem 4.2 ([94])**.** *Assume (4.10) holds. Suppose* (γ^*, y^*, z^*) *is an optimizer of (4.18) for the relaxation order* k*. Then* (γ^*, y^*) *is a maximizer of (4.16) if and only if it holds that* $y^* \in \overline{\mathscr{R}_d(S)}$*.*

Proof. If (γ^*, y^*) is a maximizer of (4.16), then it is clear that $y^* \in \overline{\mathscr{R}_d(S)}$. Conversely, if $y^* \in \overline{\mathscr{R}_d(S)}$, then (γ^*, y^*) is feasible for (4.16), since (4.10) holds. Since (4.18) is a relaxation of (4.16), we know (γ^*, y^*) must also be a maximizer of (4.16). □

If $\mathscr{R}_d(S)$ is a closed cone, then we only need to check $y^* \in \mathscr{R}_d(S)$ in the above. Interestingly, when S is compact, the moment cone $\mathscr{R}_d(S)$ is closed [61,65,82]. As introduced in the Section 2.4, the membership $y^* \in \mathscr{R}_d(S)$ can be checked by solving a truncated moment problem. This can be done by solving the optimization (2.4) for a generically selected objective. Once (γ^*, y^*) is confirmed to be a maximizer of (4.16), we show how to get a minimizer for (4.1). This is shown as follows.

Theorem 4.3 ([94])**.** *Assume (4.10) holds. For a relaxation order* k*, suppose* (x^*, w^*) *is a minimizer of (4.17) and* (γ^*, y^*, z^*) *is a maximizer of (4.18) such that* $y^* \in \overline{\mathscr{R}_d(S)}$*. Assume there is no duality gap between (4.17) and (4.18). If* $x^* \in X$ *and* $f(x^*) = \langle f, w^* \rangle$*, then* x^* *is a minimizer of (4.12). If in addition (4.11) holds, then* x^* *is also a minimizer of (4.1).*

Proof. Let f_1, f_2 be optimal values of the optimization problems (4.15) and (4.16) respectively. Then, by the weak duality, it holds that

$$f_1 \geq f_2.$$

The membership $y^* \in \overline{\mathscr{R}_d(S)}$ implies that (γ^*, y^*) is a maximizer of (4.16), so $f_2 = \gamma^* - b^T y^*$. Assume there is no duality gap between the primal-dual pair (4.17)-(4.18). Then

$$\langle f, w^* \rangle = \gamma^* - b^T y^* = f_2.$$

The constraint $h(x^*, \xi) \in \text{Qmod}[g]_{2k} + Y^*$ implies that $h(x^*, \xi) \in \mathscr{P}_d(S) + Y^*$. Suppose $x^* \in X$ is a feasible point of (4.12) and $f(x^*) = \langle f, w^* \rangle$. The optimal value of (4.12) is greater than or equal to that of (4.15), hence

$$f_1 \geq f_2 = \langle f, w^* \rangle = f(x^*) \geq f_1.$$

So $f(x^*) = f_1$. This implies that x^* is a minimizer of (4.12). Moreover, if in addition K^* can be expressed as in (4.11), the optimization (4.1) is equivalent to (4.12). So x^* is also a minimizer of (4.1). $\square$

In the above theorem, the assumptions that $x^* \in X$ and $f(x^*) = \langle f, w^* \rangle$ must hold if $f, -c_1, \ldots, -c_{m_2}$ are SOS-convex polynomials. We have the following theorem.

Theorem 4.4. *Assume (4.10) holds. For a relaxation order k, suppose (x^*, w^*) is a minimizer of (4.17) and (γ^*, y^*, z^*) is a maximizer of (4.18) such that $y^* \in \overline{\mathscr{R}_d(S)}$. Assume there is no duality gap between (4.17) and (4.18). If $f, -c_1, \ldots, -c_{m_2}$ are SOS-convex polynomials, then $x^* := \pi(w^*)$ is a minimizer of (4.12). If in addition (4.11) holds, then x^* is also a minimizer of (4.1).*

Proof. Since f and $-c_1, \ldots, -c_{m_2}$ are SOS-convex polynomials, by the Jensen's inequality (see [60]), it holds that

$$f(x^*) = f(\pi(w^*)) \le \langle f, w^* \rangle,$$

$$c_i(x^*) = c_i(\pi(w^*)) \ge \langle c_i, w^* \rangle,\ i = 1, \ldots, m_2.$$

Similarly, the constraint $L_{c_i}^{(d_1)}[w^*] \succeq 0$ implies that $\langle c_i, w^* \rangle \ge 0$. So $x^* \in X$ is a feasible point of (4.12). As in the proof of Theorem 4.3, we can similarly show that

$$f_1 \ge f_2 = \langle f, w^* \rangle \ge f(x^*) \ge f_1,$$

so $f(x^*) = \langle f, w^* \rangle$. The conclusions follow from Theorem 4.3. $\square$

Then we give an algorithm for solving DROM with polynomials.

Algorithm 4.5. *For the given DROM (4.1), do the following:*

Step 0 *Get a computational representation for $\overline{cone(Y)}$ and the dual cone Y^*. Initialize*

$$d_0 := \lceil \deg(g)/2 \rceil, \quad t_0 := \lceil d/2 \rceil, \quad k := \lceil d/2 \rceil, \quad l := t_0 + 1.$$

Choose a generic polynomial $R \in \Sigma[\xi]_{2t_0+2}$.

Step 1 *Solve (4.17) for a minimizer (x^*, w^*) and solve (4.18) for a maximizer (γ^*, y^*, z^*).*

Step 2 Solve the moment optimization

$$\begin{cases} \min\limits_{\omega} & \langle R, \omega \rangle \\ s.t. & \omega|_d = y^*,\ \omega \in \mathscr{S}[g]_{2\ell},\ \omega \in \mathbb{R}^{\mathbb{N}^p_{2\ell}}. \end{cases} \tag{4.19}$$

If (4.19) is infeasible, then y^* admits no S-measure, update $k := k+1$ and go back to Step 1. Otherwise, solve (4.19) for a minimizer ω^* and go to Step 3.

Step 3 Check whether or not there exists an integer $s \in [\max(d_0, t_0), \ell]$ such that

$$\mathit{rank}\, M_{s-d_0}[\omega^*] \;=\; \mathit{rank}\, M_s[\omega^*].$$

If such s does not exist, update $\ell := \ell + 1$ and go to Step 2. If such s exists, then $y^* = \int [\xi]_d \mathsf{d}\mu$ for the measure

$$\mu \;=\; \theta_1 \delta_{u_1} + \cdots + \theta_r \delta_{u_r},$$

where $r = \mathit{rank}\, M_s[\omega^*]$, and δ_{u_i} denotes the Dirac measure supported at u_i.

Remark. In Step 3, a measure $\mu^* \in \mathcal{M}$ that achieves the worst case expectation constraint can be recovered as a multiple of μ, up to scaling.

The convergent properties of Algorithm 4.5 can be proved using techniques and conclusions from [81, 82]. First, we consider the relatively simple but still interesting case that ξ is a univariate random variable (i.e., $p = 1$) and the support set $S = [a_1, a_2]$ is an interval.

Theorem 4.6 ([94]). *Suppose the random variable ξ is univariate and the set $S = [a_1, a_2]$, for scalars $a_1 < a_2$, is an interval with the constraint $g(\xi) := (\xi - a_1)(a_2 - \xi) \geq 0$. If (γ^*, y^*, z^*) is a maximizer of (4.18) for $k = \lceil d/2 \rceil$, then we must have $z^* \in \mathscr{R}_{2k}(S)$ and hence $y^* \in \mathscr{R}_d(S)$.*

Proof. In the relaxation (4.18), the tms z has the even degree $2k$. We label the entries of z as $z = (z_0, z_1, \ldots, z_{2k})$. The condition $z \in \mathscr{S}[g]_{2k}$ implies that

$$M_k[z] \succeq 0, \quad L_g^{(k)}[z] \succeq 0. \tag{4.20}$$

Since $g = (\xi - a_1)(a_2 - \xi)$, one can verify that $L_g^{(k)}[z] \succeq 0$ is equivalent to

$$(a_1 + a_2) \begin{bmatrix} z_1 & z_2 & \cdots & z_k \\ z_2 & z_3 & \cdots & z_{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ z_k & z_{k+1} & \cdots & z_{2k-1} \end{bmatrix} \succeq a_1 a_2 \begin{bmatrix} z_0 & z_1 & \cdots & z_{k-1} \\ z_1 & z_2 & \cdots & z_k \\ \vdots & \vdots & \ddots & \vdots \\ z_{k-1} & z_k & \cdots & z_{2k-2} \end{bmatrix} +$$

$$\begin{bmatrix} z_2 & z_3 & \cdots & z_k \\ z_3 & z_4 & \cdots & z_{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ z_k & z_{k+1} & \cdots & z_{2k} \end{bmatrix}.$$

As shown in [23, 56], the (4.20) are sufficient and necessary conditions for $z \in \mathscr{R}_{2k}(S)$. So, if (γ^*, y^*, z^*) is a maximizer of (4.18), then $M_k[z^*] \succeq 0$ and $L_g^{(k)}[z^*] \succeq 0$. Hence, we have $z^* \in \mathscr{R}_{2k}(S)$ and hence $y^* = z^*|_d \in \mathscr{R}_d(S)$. □

Second, we prove the asymptotic convergence of Algorithm 4.5 when the random variable ξ is multi-variate. It requires that the quadratic module Qmod$[g]$ is archimedean and (4.15) has interior points.

Theorem 4.7 ([94]). *Assume that Qmod$[g]$ is archimedean and there exists a point $\hat{x} \in X$ such that $h(\hat{x}, \xi) = a_1(\xi) + a_2(\xi)$ with $a_1 > 0$ on S and $a_2 \in Y^*$. Suppose $(\gamma^{(k)}, y^{(k)}, z^{(k)})$ is an optimal triple of (4.18) with the relaxation order k. Then, the sequence $\{y^{(k)}\}_{k=1}^{\infty}$ is bounded and every accumulation point of $\{y^{(k)}\}_{k=1}^{\infty}$ belongs to the cone $\mathscr{R}_d(S)$. Therefore, every accumulation point of $\{(\gamma^{(k)}, y^{(k)})\}_{k=1}^{\infty}$ is a maximizer of (4.16).*

Proof. For every (γ, y, z) that is feasible for (4.18) and for $\hat{w} := [\hat{x}]_{2d_1}$, it holds that

$$\langle f, \hat{w} \rangle - \big(\gamma - \langle b, y \rangle\big) = \langle f - y^T A x - \gamma, \hat{w} \rangle + (A\hat{x} + b)^T y \geq (A\hat{x} + b)^T y. \tag{4.21}$$

There exists $\epsilon > 0$ such that $a_1(\xi) - \epsilon \in \text{Qmod}[g]_{2k_0}$, for some $k_0 \in \mathbb{N}$, since Qmod$[g]$ is archimedean. Noting $a_2 \in Y^*$, one can see that

$$(A\hat{x} + b)^T y = \langle h(\hat{x}, \xi), y \rangle = \langle a_1(\xi), y \rangle + \langle a_2(\xi), y \rangle \geq \langle a_1(\xi), y \rangle.$$

For all $k \geq k_0$, it holds that

$$\langle a_1(\xi), y \rangle = \langle a_1(\xi) - \epsilon, y \rangle + \epsilon \langle 1, y \rangle \geq \epsilon \langle 1, y \rangle = \epsilon y_0.$$

(Note $\langle 1, y \rangle = y_0$.) Let f_2 be the optimal value of (4.16), then

$$\gamma^{(k)} - \langle b, y^{(k)} \rangle \geq f_2,$$

because $(\gamma^{(k)}, y^{(k)}, z^{(k)})$ is an optimizer of (4.18), and (4.18) is a relaxation of the maximization (4.16). So (4.21) implies that

$$(A\hat{x} + b)^T y^{(k)} \leq \langle f, \hat{w} \rangle - f_2.$$

Hence, we can get that

$$(y^{(k)})_0 \leq \frac{1}{\epsilon}(\langle f, \hat{w}\rangle - f_2).$$

The sequence $\{(y^{(k)})_0\}_{k=1}^{\infty}$ is bounded.

Since Qmod$[g]$ is archimedean, there exists $N > 0$ such that $N - \|\xi\|^2 \in \text{Qmod}[g]_{2k_1}$ for some $k_1 \geq k_0$. For all $k \geq k_1$, the membership $z^{(k)} \in \mathscr{S}[g]_{2k}$ implies that

$$N \cdot (z^{(k)})_0 - \big((z^{(k)})_{2e_1} + \cdots + (z^{(k)})_{2e_p}\big) \geq 0.$$

Note that $y^{(k)} = z^{(k)}|_d$, hence $(y^{(k)})_0 = (z^{(k)})_0$. Since $z^{(k)} \in \mathscr{S}[g]_{2k}$ and the sequence $\{(z^{(k)})_0\}_{k=1}^{\infty}$ is bounded, one can further show that the set

$$\{z^{(k)}|_d : z \in \mathscr{S}[g]_{2k}\}_{k=1}^{\infty}$$

is bounded. We refer to [82, Theorem 4.3] for more details about the proof. Therefore, the sequence $\{y^{(k)}\}_{k=1}^{\infty}$ is bounded. Since Qmod$[g]$ is archimedean, we also have

$$\mathscr{R}_d(S) = \bigcap_{k=1}^{\infty} S_k, \quad \text{where} \quad S_k := \{z|_d :\, z \in \mathscr{S}[g]_{2k}\}.$$

This is shown in Proposition 3.3 of [82]. So, if $\hat{y}$ is an accumulation point of $\{y^{(k)}\}_{k=1}^{\infty}$, then we must have $\hat{y} \in \mathscr{R}_d(S)$. Similarly, if $(\hat{\gamma}, \hat{y}, \hat{z})$ is an accumulation point of $\{(\gamma^{(k)}, y^{(k)}, z^{(k)})\}_{k=1}^{\infty}$, then $\hat{y} \in \mathscr{R}_d(S)$. As in the proof of Theorem 4.2, one can similarly show that $(\hat{\gamma}, \hat{y})$ is a maximizer of (4.16). □

Last, we prove that Algorithm 4.5 will terminate within finitely many steps under certain assumptions.

Theorem 4.8 ([94]). *Assume Qmod$[g]$ is archimedean and there is no duality gap between (4.15) and (4.16). Suppose (x^*, w^*) is a minimizer of (4.15) and (γ^*, y^*) is a maximizer of (4.16) satisfying:*

(i) There exists $k_1 \in \mathbb{N}$ such that $h(x^*, \xi) = h_1(\xi) + h_2(\xi)$, with $h_1 \in$ Qmod$[g]_{2k_1}$ and $h_2 \in Y^*$.

(ii) The polynomial optimization problem in ξ

$$\begin{cases} \min\limits_{\xi \in \mathbb{R}^p} & h_1(\xi) \\ s.t. & g_1(\xi) \geq 0, \ldots, g_{m_1}(\xi) \geq 0 \end{cases} \tag{4.22}$$

has finitely many critical points u such that $h_1(u) = 0$.

Then, when k is large enough, for every optimizer $(\gamma^{(k)}, y^{(k)}, z^{(k)})$ of (4.18), we must have $y^{(k)} \in \mathscr{R}_d(S)$.

Proof. Since there is no duality gap between (4.15) and (4.16),

$$0 = \langle f, w^* \rangle - \big(\gamma^* - \langle b, y^* \rangle\big) = \langle f - (y^*)^T Ax - \gamma^*, w^* \rangle + (Ax^* + b)^T y^*.$$

Due to the feasibility constraints, we further have

$$\langle f(x) - (y^*)^T Ax - \gamma^*, w^* \rangle = 0, \quad (Ax^* + b)^T y^* = 0.$$

Therefore, it holds that

$$(Ax^* + b)^T y^* = \langle h(x^*, \xi), y^* \rangle = \langle h_1(\xi), y^* \rangle + \langle h_2(\xi), y^* \rangle = 0.$$

The conic membership $y^* \in \overline{K}$ implies that

$$\langle h_1(\xi), y^* \rangle = \langle h_2(\xi), y^* \rangle = 0.$$

We consider the polynomial optimization problem (4.22) in the variable ξ. For each order $k \geq k_1$, the kth order Moment-SOS relaxation pair for solving (4.22) is

$$\begin{cases} \min & \langle h_1(\xi), z \rangle \\ s.t. & z \in \mathscr{S}[g]_{2k}, z_0 = 1, \end{cases} \tag{4.23}$$

$$\begin{cases} \nu_k := & \max \quad \gamma \\ & s.t. \quad h_1(\xi) - \gamma \in \mathrm{Qmod}[g]_{2k}. \end{cases} \tag{4.24}$$

The archimedeanness of $\mathrm{Qmod}[g]$ implies that S is compact, so

$$\overline{\mathscr{R}_d(S)} = \mathscr{R}_d(S).$$

The membership $y^* \in \overline{K}$ implies that $y^* \in \mathscr{R}_d(S)$. Since

$$\langle h_1(\xi), y^* \rangle = 0,$$

the polynomial $h_1(\xi)$ vanishes on the support of each S-representing measure for y^*, so the optimal value of (4.22) is zero. By the given assumption, the sequence $\{\nu_k\}$ has finite convergence to the optimal value 0 and the relaxation (4.24) achieves its optimal value for all $k \geq k_1$. The optimization (4.22) has only finitely many critical points that are global optimizers. So, Assumption 2.1 of [81] for the optimization (4.22) is satisfied. Moreover, the given assumption also implies that (x^*, w^*) is an optimizer of (4.17) and (γ^*, y^*, z^*) is an optimizer of (4.18) for all $k \geq k_1$. Suppose $(x^{(k)}, w^{(k)})$ is an arbitrary optimizer of (4.17) and $(\gamma^{(k)}, y^{(k)}, z^{(k)})$ is an arbitrary optimizer of (4.18), for the relaxation order k.

When $(z^{(k)})_0 = 0$, we have $vec(1)^T M_k[z^{(k)}]vec(1) = 0$. Since $M_k[z^{(k)}] \succeq 0$,

$$M_k[z^{(k)}]vec(1) = 0.$$

Consequently, we further have $M_k[z^{(k)}]vec(\xi^\alpha) = 0$ for all $|\alpha| \leq k-1$ (see Lemma 5.7 of [65]). Then, for each power $\alpha = \beta + \eta$ with $|\beta|, |\eta| \leq k-1$, one can get $(z^{(k)})_\alpha = vec(\xi^\beta)^T M_k[z^{(k)}]vec(\xi^\eta) = 0$. This means that $z^{(k)}|_{2k-2}$ is the zero vector and hence $y^{(k)} \in \mathscr{R}_d(S)$.

For the case $(z^{(k)})_0 > 0$, let $\hat{z} := z^{(k)}/(z^{(k)})_0$. The given assumption implies that (x^*, w^*) is also a minimizer of (4.17) and (γ^*, y^*, z^*) is optimal for (4.18), for all $k \geq k_1$. So there is no duality gap between (4.17) and (4.18). Since $(\gamma^{(k)}, y^{(k)}, z^{(k)})$ is optimal for (4.18), so $\langle h_1(\xi), z^{(k)} \rangle = 0$ and hence $\hat{z}$ is a minimizer of (4.23) for all $k \geq k_1$. By Theorem 2.2 of [81], the minimizer $z^{(k)}$ must have a flat truncation $z^{(k)}|_{2t}$ for some t, when k is sufficiently big. This means that the truncation $z^{(k)}|_{2t}$, as well as $y^{(k)}$, has a representing measure supported in S. Therefore, we have $y^{(k)} \in \mathscr{R}_d(S)$. $\square$

4.4 Numerical experiments

In this section, we give numerical experiments for Algorithm 4.5 to solve distributionally robust optimization problems. The computation is implemented in `MATLAB` R2018a, in a Laptop with CPU 8th Generation Intel® Core™ i5-8250U and RAM 16 GB. The software `GloptiPoly3` [48], `Yalmip` [70] and `SeDuMi` [105] are used for the implementation. For neatness of presentation, we only display four decimal digits.

To apply implement Algorithm 4.5, we need a computational representation for the cone $\overline{cone(Y)}$. In the following, we give some frequently appearing cases.

- If $Y = \{y : Ty + u \geq 0\}$ is a nonempty polyhedron, given by some matrix T and vector u, then

$$\overline{cone(Y)} \;=\; \{y : Ty + su \geq 0,\, s \in \mathbb{R}_+\}. \tag{4.25}$$

  It is also a polyhedron and is closed.

- Consider that $Y = \{y : \mathcal{A}(y) + B \succeq 0\}$ is given by a linear matrix inequality, for a homogeneous linear symmetric matrix valued function $\mathcal{A}$ and a symmetric matrix B. If Y is nonempty and bounded, then

$$\overline{cone(Y)} = \{y : \mathcal{A}(y) + sB \succeq 0,\ s \in \mathbb{R}_+\}. \tag{4.26}$$

When Y is unbounded, the $cone(Y)$ may not be closed and its closure $\overline{cone(Y)}$ may be tricky. We refer to the work [78] for such cases. When Y is given by second order conic conditions, we can do similar things for obtaining $\overline{cone(Y)}$.

Example 4.9. *Consider the DROM problem*

$$\begin{cases} \min\limits_{x\in\mathbb{R}^4} & f(x) = -x_1 - 2x_2 - x_3 + 2x_4 \\ s.t. & \inf\limits_{\mu\in\mathcal{M}} \mathbb{E}_\mu[h(x,\xi)] \geq 0, \\ & x \geq 0,\ 1 - e^T x \geq 0, \end{cases} \tag{4.27}$$

where (the random variable ξ is univariate, i.e, $p = 1$)

$$h(x,\xi) = (x_4 - x_1 - 2)\xi^5 + (x_4 - 1)\xi^4 + (2x_1 + x_2 + x_4 + 1)\xi^3 \\ +(2x_1 - x_2 + x_4 - 1)\xi^2 + (2 - x_2 - x_3)\xi,$$

$$S = [0,3], \quad g = 3\xi - \xi^2,$$

$$Y = \left\{ y = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_5 \end{bmatrix} \in \mathbb{R}^6 \;\middle|\; \begin{array}{r} 1 \leq y_0 \leq y_1 \leq y_2 \leq \\ y_3 \leq y_4 \leq y_5 \leq 2 \end{array} \right\}.$$

The $\overline{cone(Y)}$ is given as in (4.25). The objective f and constraints c_1, c_2 are all linear. We start with $k = 3$, and the Algorithm 4.5 terminates in the initial loop. The optimal value F^* and the optimizer x^* for (4.12) are respectively

$$F^* \approx -0.0326, \quad x^* \approx (0.6775, 0.0000, 0.0000, 0.3225).$$

The optimizer for (4.18) is

$$y^* \approx (0.9355, 0.9355, 0.9517, 1.0163, 1.2260, 1.8710).$$

The measure μ for achieving $y^* = \int [\xi]_5 \mathtt{d}\mu$ is supported at the points

$$u_1 \approx 0.9913, \quad u_2 \approx 3.0000.$$

By a proper scaling, we get the measure $\mu^* = 0.9957\delta_{u_1} + 0.0043\delta_{u_2}$ that achieves the worst case expectation constraint.

Example 4.10. *Consider the DROM problem*

$$\begin{cases} \min\limits_{x\in\mathbb{R}^3} & f(x) = (x_1 - x_3 + x_1x_3)^2 + (2x_2 + 2x_1x_2 - x_3^2)^2 \\ s.t. & \inf\limits_{\mu\in\mathcal{M}} \mathbb{E}_\mu[h(x,\xi)] \geq 0, \\ & c_1(x) = 1 - x_1^2 - x_2^2 - x_3^2 \geq 0, \\ & c_2(x) = 3x_3 - x_1^2 - 2x_2^4 \geq 0, \end{cases} \tag{4.28}$$

where (the random variable ξ is bivariate, i.e, $p = 2$)

$$h(x,\xi) = (1 - x_3)\xi_1^2\xi_2^2 + (x_1 - x_2 + x_3 - 1)\xi_1\xi_2^2 + \\ (x_1 + x_2 + x_3 + 1)\xi_2^2 + (x_1 - x_3)\xi_1^2 - \xi_2,$$

$$S = \{\xi \in \mathbb{R}^2 : 1 - \xi^T\xi \geq 0\}, \quad g := 1 - \xi^T\xi,$$

$$Y = \left\{ y \in \mathbb{R}^{\mathbb{N}_4^2} \,\middle|\, \begin{array}{c} y_{00} = 1, \ 0.1 \leq y_\alpha \leq 1 \, (0 < |\alpha| \leq 4) \\ \begin{pmatrix} y_{20} & y_{11} & y_{30} & y_{12} \\ y_{11} & y_{02} & y_{21} & y_{03} \\ y_{30} & y_{21} & y_{40} & y_{22} \\ y_{12} & y_{03} & y_{22} & y_{04} \end{pmatrix} \preceq 2I_4 \end{array} \right\}.$$

The $\overline{cone(Y)}$ is given as in (4.26). One can verify that f and all $-c_i$ are SOS-convex. We start with $k = 2$, and Algorithm 4.5 terminates in the initial loop. The optimal value F^* and optimizer x^* of (4.12) are respectively

$$F^* \approx 0.0160, \quad x^* \approx (0.4060, 0.0800, 0.4706).$$

The optimizer for (4.18) is

$$y^* \approx (0.3180, 0.2750, 0.1411, 0.2436, 0.1137, 0.0744, 0.2199, 0.0950, \\ 0.0552, 0.0460, 0.2011, 0.0819, 0.0426, 0.0318, 0.0318).$$

The measure μ for achieving $y^* = \int [\xi]_4 \mathtt{d}\mu$ is supported at the points

$$u_1 \approx (0.6325, 0.7745), \quad u_2 \approx (0.9434, 0.3317).$$

By a proper scaling, we get the measure $\mu^* = 0.2527\delta_{u_1} + 0.7473\delta_{u_2}$ that achieves the worst case expectation constraint.

Example 4.11. *Consider the DROM problem*

$$\begin{cases} \min\limits_{x\in\mathbb{R}^3} & f(x) = x_1^4 - x_1x_2x_3 + x_3^3 + 3x_1x_3 + x_2^2 \\ s.t. & \inf\limits_{\mu\in\mathcal{M}} \mathbb{E}_\mu[h(x,\xi)] \geq 0, \\ & c_1(x) = x_1x_2 - 0.25 \geq 0, \\ & c_2(x) = 6 - x_1^2 - 4x_1x_2 - x_2^2 - x_3^2 \geq 0 \end{cases} \tag{4.29}$$

where (the random variable ξ is bivariate, i.e, $p = 2$)

$$\begin{aligned} h(x,\xi) = (2 - x_1 + x_2)\xi_2^4 + (x_1 + x_3 + 1)\xi_1\xi_2^2 + (2 - x_1 + 2x_2)\xi_2^3 \\ +(x_1 + 2x_2 + x_3 + 2)\xi_1^2 + (3x_2 - x_1)\xi_2^2, \end{aligned}$$

$$S = \{\xi \in \mathbb{R}^2 | 1 \leq \xi^T\xi \leq 4\}, \quad g = (\xi^T\xi - 1,\, 4 - \xi^T\xi),$$

$$Y = \left\{ y \in \mathbb{R}^{\mathbb{N}_4^2} \,\middle|\, y_{00} = 1, \sum_{|\alpha|\geq 1} y_\alpha^2 = 36 \right\}.$$

The set Y is not convex. Its convex hull is $\|y\| \leq \sqrt{37}$ with $y_{00} = 1$. Hence,

$$\overline{cone(Y)} = \left\{ y \in \mathbb{R}^{\mathbb{N}_4^2} \,\middle|\, \|y\|_2 \leq \sqrt{37} y_{00} \right\}.$$

The functions f and $-c_1, -c_2$ are not convex. We start with $k = 2$. The optimizers for (4.17) and (4.18) are respectively

$$\begin{aligned} w^* \approx (&1.0000, 0.6790, 0.3682, -2.0984, 0.4611, 0.2500, -1.4249, 0.1356, -0.7726, \\ &4.4034, 0.3131, 0.1698, -0.9675, 0.0920, -0.5246, 2.9900, 0.0499, -0.2845, \\ &1.6212, -9.2402, 0.2126, 0.1153, -0.6569, 0.0625, -0.3562, 2.0302, 0.0339, \\ &-0.1932, 1.1008, -6.2742, 0.0184, -0.1047, 0.5969, -3.4021, 19.3898), \\ y^* \approx (&1.2272, 0.2992, -1.1902, 0.0730, -0.2902, 1.1543, 0.0178, -0.0708, 0.2814, \\ &-1.1194, 0.0043, -0.0173, 0.0686, -0.2729, 1.0857). \end{aligned}$$

The optimal value is $F^* \approx -12.6420$ *for both of them. The measure for achieving* $y^* = \int [\xi]_4 \mathtt{d}\mu$ *is* $\mu = 1.2272\delta_u$*, with* $u \approx (0.2438, -0.9698) \in S$*. So* $\mu^* = \delta_u$*. For the point*

$$x^* = \pi(w^*) \approx (0.6790, 0.3682, -2.0984),$$

one can verify that x^* *is feasible for (4.29), since*

$$c_1(x^*) \approx -1.6654 \cdot 10^{-9},\ c_2(x^*) \approx 5.6235 \cdot 10^{-8},\ F^* - f(x^*) \approx -7.7271 \cdot 10^{-8}.$$

By Theorem 4.3, we know x^* *is the optimizer for (4.29).*

Example 4.12 (Portfolio selection [25,53])**.** *Consider that there exist* n *risky assets that can be chosen by the investor in the financial market. The uncertain loss* r_i *of each asset can be described by the random risk variable* ξ *which admits a probability measure supported in* $S = [0,1]^p$*. Assume the moments of* $\mu \in \mathcal{M}$ *are constrained in the set*

$$Y = \left\{ y \in \mathbb{R}^{\mathbb{N}_3^3} \,|\, y_{000} = 1,\ 0.1 \le y_\alpha \le 1,\ |\alpha| \ge 1 \right\}.$$

The cone $\overline{\mathrm{cone}(Y)}$ *can be given as in (4.25). Minimizing the portfolio loss over the ambiguity set* $\mathcal{M}$ *is equivalent to solving the following min-max optimization problem*

$$\min_{x \in \Delta_3} \max_{\mu \in \mathcal{M}} \mathbb{E}_\mu \left[x_1 r_1(\xi) + x_2 r_2(\xi) + x_3 r_3(\xi) \right], \tag{4.30}$$

for the simplex $\Delta_n \coloneqq \left\{ x \in \mathbb{R}^3 \,\middle|\, e^T x = 1,\ x \ge 0 \right\}$*. The functions* $r_i(\xi)$ *are*

$$\begin{cases} r_1(\xi) = -1 + \xi_1 + \xi_1\xi_2 - \xi_1\xi_3 - 2\xi_1^3, \\ r_2(\xi) = -1 - \xi_1\xi_2 + \xi_2^2 - \xi_2\xi_3 + \xi_2^3, \\ r_3(\xi) = -1 + \xi_2\xi_3 - \xi_3^2 - \xi_3^3. \end{cases} \tag{4.31}$$

Then (4.30) can be equivalently reformulated as

$$\begin{cases} \min\limits_{(x_0, x) \in \mathbb{R} \times \mathbb{R}^3} & x_0 \\ \quad s.t. & \inf\limits_{\mu \in \mathcal{M}} \mathbb{E}_\mu \left[x_0 - \big(x_1 r_1(\xi) + x_2 r_2(\xi) + x_3 r_3(\xi) \big) \right] \ge 0, \\ & x \ge 0,\ e^T x = 1. \end{cases} \tag{4.32}$$

Applying Algorithm 4.5 to solve (4.32), we get the optimal value F^* *and the optimizer* (x_0^*, x^*) *in the initial loop* $k = 2$*:*

$$F^* \approx -1.0136, \quad (x_0^*, x^*) \approx (-1.0136, 0.1492, 0.3501, 0.5007).$$

The optimizer for (4.18) is

$$y^* \approx (1.0000, 0.6077, 0.4440, 0.3725, 0.3864, 0.3347, 0.2530, 0.4440, 0.2666, 0.1803,\\ 0.2560, 0.2523, 0.1771, 0.3347, 0.2010, 0.1306, 0.4440, 0.2666, 0.1601, 0.1000).$$

The measure for achieving $y^* = \int [\xi]_4 \mathtt{d}\mu$ *is*

$$\mu \;=\; 0.5560\delta_{u_1} + 0.4440\delta_{u_2},$$

with the following two points in S*:*

$$u_1 \approx (0.4911, -0.0000, 0.1905), \quad u_2 \approx (0.7538, 1.0000, 0.6005).$$

Since μ *belongs to* $\mathcal{M}$*, it is also the measure that achieves the worst case expectation constraint. Therefore, the optimizer for (4.30) is* x^* *and the optimal value is* -0.9792*.*

Example 4.13 (Newsvendor problem [110]). *Consider that there is a newsvendor trade product with an uncertain daily demand. Assume the demand quantity* $D(\xi)$ *is affected by a random variable* $\xi \in \mathbb{R}^2$ *such that*

$$D(\xi) \;=\; 2 - \xi_1 + \xi_2 - \xi_1^2 + 2\xi_2^2 + \xi_1^4.$$

In each day, the newsvendor orders x *units of the product at the wholesale price* P_1*, sells the product with quantity* $\min\{x, D(\xi)\}$ *at the retail price* P_2 *and clears the unsold stock at the salvage price* P_0*. Assume that* $P_0 < P_1 < P_2$*, then the newsvendor's daily loss is given as*

$$l(x,\xi) \;\coloneqq\; (P_1 - P_2)x + (P_2 - P_0) \cdot \max\{x - D(\xi), 0\}.$$

Clearly, the newsboy will earn the most if he can buy the greatest order quantity that is guaranteed to be sold out. Suppose ξ *admits a probability measure supported in* S *and has its true distribution contained in the ambiguity set* $\mathcal{M}$*. Then the best order decision for the newsvendor product can be obtained from the following DROM problem*

$$\begin{cases} \min\limits_{x\in\mathbb{R}} & (P_1 - P_2)x \\ s.t. & \inf\limits_{\mu\in\mathcal{M}} \mathbb{E}_\mu[D(\xi) - x] \geq 0, \\ & x \geq 0. \end{cases} \tag{4.33}$$

Suppose $P_0 = 0.25, P_1 = 0.5, P_2 = 1,$ *and*

$$S = [0,5]^2, \quad Y = \left\{ y \in \mathbb{R}^{\mathbb{N}_4^2} \;\middle|\; \begin{array}{c} y_{00} = 1,\ 1 \leq y_{01} \leq y_{02} \leq 4 \\ 2^i \leq y_{i0} \leq 4^i,\ i = 1,2,3,4 \end{array} \right\}.$$

The cone $\overline{cone(Y)}$ *can be given as in (4.25). Applying Algorithm 4.5 to solve (4.33), we get the optimal value* F *and the optimizer* x^* *respectively*

$$F^* \approx -7.5000, \quad x^* \approx 15.0000.$$

The optimizer of (4.18) is

$$\begin{aligned} y^* \approx\ & (0.5000, 1.0000, 0.5000, 2.0000, 1.0000, 0.5000, 4.0000, 2.0000, \\ & 1.0000, 0.5000, 8.0000, 4.0000, 2.0000, 1.0000, 0.5000). \end{aligned}$$

The measure for achieving $y^* = \int [\xi]_4 \mathtt{d}\mu$ *is* $\mu = 0.5\delta_u$ *with* $u = (2.0000, 1.0000) \in S$. *So* $\mu^* = \delta_u$ *achieves the worst case expectation constraint.*

Acknowledgments. This Chapter, in full, has been submitted for publication. The dissertation author coauthored this paper with Nie, Jiawang; Yang, Liu and Zhou, Guangming.

Chapter 5

Bilevel Polynomial Optimziation

5.1 Bilevel optimization problems

The bilevel optimization problem is

$$\left\{\begin{array}{ll} \min\limits_{x\in\mathbb{R}^n, y\in\mathbb{R}^p} & F(x,y) \\ s.t. & h_i(x,y)=0\,(i\in\mathcal{E}_1), \\ & h_j(x,y)\geq 0\,(j\in\mathcal{I}_1), \\ & y\in S(x), \end{array}\right. \tag{5.1}$$

where $S(x)$ is the set of optimizer(s) of the lower level problem

$$(P_x) \qquad \left\{\begin{array}{ll} \min\limits_{z\in\mathbb{R}^p} & f(x,z) \\ s.t. & g_i(x,z)=0\,(i\in\mathcal{E}_2), \\ & g_j(x,z)\geq 0\,(j\in\mathcal{I}_2). \end{array}\right.$$

In the above, $F(x,y)$ is the upper level objective function and $h_i(x,y), h_j(x,y)$ are the upper level constraints. Similarly, $f(x,z)$ is the lower level objective function and $g_i(x,z), g_j(x,z)$ are the lower level constraints. The $\mathcal{E}_1, \mathcal{I}_1, \mathcal{E}_2, \mathcal{I}_2$ are finite index sets (some or all of them are possibly empty). For convenience, we denote by $\mathcal{F}$ the feasible set of (5.1) and denote the lower feasible set

$$Z(x) := \left\{ z\in\mathbb{R}^p \;\middle|\; \begin{array}{l} g_i(x,z)=0\ (i\in\mathcal{E}_2), \\ g_j(x,z)\geq 0\,(j\in\mathcal{I}_2) \end{array} \right\}. \tag{5.2}$$

The (5.1) is called a *simple* bilevel optimization problem (SBOP) if $Z(x)\equiv Z$ is independent of x. When $Z(x)$ depends on x, the (5.1) is called a *general* bilevel optimization problem

(GBOP). The (5.1) is called a bilevel polynomial optimization problem if all the defining functions are polynomials. We study the bilevel polynomial optimization.

Assume (5.1) is a bilevel polynomial optimization problem. Denote the broad feasible set

$$\mathcal{U} := \left\{ (x,y) \in \mathbb{R}^n \times \mathbb{R}^p \;\middle|\; \begin{array}{l} h_i(x,y) = 0\,(i \in \mathcal{E}_1),\; g_i(x,y) = 0\,(i \in \mathcal{E}_2), \\ h_j(x,y) \ge 0\,(j \in \mathcal{I}_1),\; g_j(x,y) \ge 0\,(j \in \mathcal{I}_2) \end{array} \right\}. \tag{5.3}$$

For convenience, we assume $S(x)$ is nonempty for all feasible x.

5.2 An algorithm for bilevel polynomial optimization

In this section, we propose a framework for solving the bilevel polynomial optimization (5.1). It is based on solving a sequence of polynomial optimization relaxations, with the usage of KKT conditions and Lagrange multiplier expressions.

For each $y \in S(x)$, we assume the KKT conditions hold

$$\left\{ \begin{array}{l} \nabla_z f(x,y) - \sum\limits_{j \in \mathcal{E}_2 \cup \mathcal{I}_2} \lambda_j \nabla_z g_j(x,y) = 0, \\ \lambda_j \ge 0,\; \lambda_j g_j(x,y) = 0\,(j \in \mathcal{I}_2), \end{array} \right. \tag{5.4}$$

where the λ_j's are Lagrange multipliers. This can be guaranteed if f and all g_j are linear, or by imposing the LICQ/MFCQ (see Section 2.5). For convenience, assume $[m_2] := \mathcal{E}_2 \cup \mathcal{I}_2$ and the lower constraining polynomial tuple is

$$g \;:=\; (g_1(x,z), \dots, g_{m_2}(x,z)).$$

Then the KKT condition (5.4) implies that

$$\underbrace{\begin{bmatrix} \nabla_z g_1(x,y) & \nabla_z g_2(x,y) & \cdots & \nabla_z g_{m_2}(x,y) \\ g_1(x,y) & 0 & \cdots & 0 \\ 0 & g_2(x,y) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & g_{m_2}(x,y) \end{bmatrix}}_{G(x,y)} \underbrace{\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_{m_2} \end{bmatrix}}_{\lambda} = \underbrace{\begin{bmatrix} \nabla_z f(x,y) \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\hat{f}(x,y)}. \tag{5.5}$$

Suppose there is a polynomial matrix $W(x,y)$ such that

$$W(x,y)G(x,y) \;=\; \mathrm{diag}[d(x,y)],$$

$$d(x,y) \;:=\; (d_1(x,y),\ldots,d_{m_2}(x,y)).$$

Then we can get rational expressions of Lagrange multipliers from

$$\mathrm{diag}[d(x,y)]\lambda = W(x,y)G(x,y)\lambda = W(x,y)\hat{f}(x,y). \tag{5.6}$$

The above equation is the same as

$$d_j(x,y)\lambda_j \;=\; \big(W(x,y)\hat{f}(x,y)\big)_j, \tag{5.7}$$

where the subscript j denotes the jth entry. The polynomial $\phi_j(x,y)$ in (5.8) is then $\big(W(x,y)\hat{f}(x,y)\big)_j$. Then we make the following assumption.

Assumption 5.1. *Suppose the KKT condition (5.5) holds for every minimizer of (5.1), there exist polynomials* $d_1(x,y),\ldots,d_{m_2}(x,y)\geq 0$ *on* $\mathcal{U}$ *and there are non-identically zero polynomials* $\phi_1(x,y),\ldots,\phi_{m_2}(x,y)$ *such that*

$$\lambda_j d_j(x,y) \;=\; \phi_j(x,y), \quad j=1,\ldots,m_2 \tag{5.8}$$

for all KKT points (x,y) *as in (5.5).*

Such rational functions $\phi_j(x,y)/d_j(x,y)$ is called the *Lagrange multiplier expression* (LME). The concept of LME is proposed in [85]. Under Assumption 5.1, let $D(x,y)$ be the least common multiple of $d_1(x,y),\ldots,d_{m_2}(x,y)$ and $D_j(x,y)$ be the quotient polynomial $D(x,y)/d_j(x,y)$. Then the set of KKT points of (5.1) is contained in

$$\mathcal{K} := \left\{ (x,y) \;\middle|\; \begin{array}{c} D(x,y)\nabla_z f(x,y) = \sum\limits_{j=1}^{m_2} D_j(x,y)\phi_j(x,y)\nabla_z g_j(x,y), \\ \phi_j(x,y)\geq 0,\; \phi_j(x,y)g_j(x,y)=0\,(j\in\mathcal{I}_2) \end{array} \right\}. \tag{5.9}$$

Indeed, the equivalence holds when $d(x,y)$ is positive on $\mathcal{U}$. If $d_j(\hat{x},\hat{y})=0$ for some j and $(\hat{x},\hat{y})\in\mathcal{U}$ then $D(\hat{x},\hat{y})=0$ and hence the equations in (5.9) are automatically satisfied. The set $\mathcal{K}$ may contain extra points other than the solutions of the bilevel optimization (5.1). To preclude these points in computation, we make another assumption.

Assumption 5.2. *For every pair* $(\hat{x},\hat{y})\in\mathcal{U}\cap\mathcal{K}$ *and for every* $\hat{z}\in S(\hat{x})$*, there exists a polynomial tuple* $q(x,y) := (q_1(x,y),\ldots,q_p(x,y))$ *such that*

$$q(\hat{x},\hat{y}) = \hat{z}, \quad q(x,y)\in Z(x) \quad \forall\,(x,y)\in\mathcal{U}. \tag{5.10}$$

We call the above function $q(x, y)$ a *polynomial extension* of the point $\hat{z}$ at $(\hat{x}, \hat{y})$. They have explicit expressions for some common lower constraints.

Then we propose an algorithm to solve the bilevel polynomial optimization (5.1), under Assumptions 5.1-5.2.

Algorithm 5.3. *For the bilevel polynomial optimization (5.1), do the following:*

Step 0 *Find rational expressions for Lagrange multipliers as in (5.8), for Assumption 5.1. Let* $\mathcal{U}_0 := \mathcal{U} \cap \mathcal{K}$, *where* $\mathcal{K}$ *is the set in (5.9). Let* $k := 0$.

Step 1 *Apply the Moment-SOS hierarchy to solve the polynomial optimization*

$$(P_k) \quad \begin{cases} F_k^* := \min\limits_{x \in \mathbb{R}^n, y \in \mathbb{R}^p} & F(x, y) \\ \qquad\qquad s.t. & (x, y) \in \mathcal{U}_k. \end{cases} \tag{5.11}$$

If (P_k) *is infeasible, then either (5.1) has no optimizers, or none of its optimizers satisfy the KKT condition for the lower level optimization. If it is feasible and has a minimizer, solve it for a minimizer* $(x^{(k)}, y^{(k)})$.

Step 2 *Apply the Moment-SOS hierarchy to solve the lower level optimization*

$$(Q_k) \quad \begin{cases} v_k^* := \min\limits_{z \in \mathbb{R}^p} & f(x^{(k)}, z) - f(x^{(k)}, y^{(k)}) \\ \qquad\quad s.t. & z \in Z(x^{(k)}),\ (x^{(k)}, z) \in \mathcal{K}. \end{cases} \tag{5.12}$$

for an optimizer $z^{(k)}$. *If the optimal value* $v_k^* = 0$, *then* $(x^{(k)}, y^{(k)})$ *is an optimizer for (5.1) and stop. Otherwise, go to the next step.*

Step 3 *Construct* $q^{(k)}(x, y)$, *a polynomial extension of the vector* $z^{(k)}$, *such that*

$$q^{(k)}(x^{(k)}, y^{(k)}) = z^{(k)}, \quad q^{(k)}(x, y) \in Z(x) \quad \forall (x, y) \in \mathcal{U}.$$

Update the set $\mathcal{U}_{k+1}$ *as*

$$\mathcal{U}_{k+1} := \{(x, y) \in \mathcal{U}_k | f(x, q^{(k)}(x, y)) - f(x, y) \geq 0\}.$$

Let $k := k + 1$ *and go to Step 1.*

In Algorithm 5.3, the polynomial optimization problems $(P_k), (Q_k)$ can be solved globally by Moment-SOS relaxations.

We study the convergence of Algorithm 5.3. First, we show that if the problem (P_x) is convex for each x, then Algorithm 5.3 will find a global optimizer of the bilevel optimization (5.1) in the initial loop.

Proposition 5.4 ([92]). *Suppose Assumptions 5.1-5.2 hold and all $d_j(x,y) > 0$ on $\mathcal{U}$. For every given x, assume that $f(x,z)$ is convex in z, $g_i(x,z)$ is linear in z for $i \in \mathcal{E}_2$, and $g_j(x,z)$ is concave in z for $j \in \mathcal{I}_2$. Assume that the Slater's condition holds for $Z(x)$ for all feasible x. Then, the bilevel optimization (5.1) is equivalent to (P_0) and Algorithm 5.3 terminates at the loop $k = 0$.*

Proof. Under the given assumptions, $y \in S(x)$ if and only if y is a KKT point for problem (P_x), which is then equivalent to $(x,y) \in \mathcal{K}$, since all $d_j(x,y) > 0$ on $\mathcal{U}$. Then, the feasible set of (5.1) is equivalent to $\mathcal{U} \cap \mathcal{K}$. This implies that (5.1) is equivalent to (P_0) and Algorithm 5.3 terminates at the initial loop $k = 0$. $\square$

Second, if Algorithm 5.3 terminates at some loop k, we can show that it produces a global optimizer for the bilevel optimization (5.1).

Proposition 5.5 ([92]). *Suppose Assumptions 5.1-5.2 hold. If Algorithm 5.3 terminates at the loop k, then the point $(x^{(k)}, y^{(k)})$ is a global optimizer of (5.1).*

Proof. By Assumption 5.1, the KKT condition (5.4) holds at each $(x,y) \in \mathcal{F}$ and hence $\mathcal{F} \subseteq \mathcal{U}_0 := \mathcal{U} \cap \mathcal{K}$. By the construction of $q^{(k)}(x,y)$ as required for Assumptions 5.2, we have shown $\mathcal{F} \subseteq \mathcal{U}_k$ for each k. Then $F_k^* \le F^*$ for all k, where F^* denotes the optimal value of (5.1). According to the stopping rule, if Algorithm 5.3 terminates at the kth loop, then $y^{(k)} \in S(x^{(k)})$. This means $(x^{(k)}, y^{(k)}) \in \mathcal{F}$. Consequently $F_k^* = F(x^{(k)}, y^{(k)}) \ge F^*$. Hence $(x^{(k)}, y^{(k)})$ is a global optimizer of (5.1). $\square$

Last, we study the asymptotic convergence of Algorithm 5.3. To prove the convergence, we need to assume that the value function

$$v(x) := \inf_{z \in Z(x)} f(x,z) \tag{5.13}$$

is continuous at an accumulation point x^*. This is the case under the so-called *restricted inf-compactness* (RIC) condition (see e.g., [41, Definition 3.13]) and either $Z(x)$ is independent of x or the MFCQ holds at some $\bar{z} \in Z(x^*)$; see [38, Lemma 3.2] for the upper semicontinuity and [18, page 246] for the lower semicontinuity.

Theorem 5.6 ([92]). *For Algorithm 5.3, we assume the following:*

(a) All optimization problems (P_k) and (Q_k) have global minimizers.

(b) *The Algorithm 5.3 does not terminate at any loop, so it produces the infinite sequence* $\{(x^{(k)}, y^{(k)}, z^{(k)})\}_{k=0}^{\infty}$.

(c) *Suppose* (x^*, y^*, z^*) *is an accumulation point of* $\{(x^{(k)}, y^{(k)}, z^{(k)})\}_{k=0}^{\infty}$ *and the value function* $v(x)$ *is continuous at* x^*.

(d) *The polynomial functions* $q^{(k)}(x, y)$ *converge to* $q^{(k)}(x^*, y^*)$ *uniformly for* $k \in \mathbb{N}$ *as* $(x, y) \to (x^*, y^*)$.

Then, (x^*, y^*) *is a global minimizer for the bilevel optimization (5.1).*

Proof. Since (x^*, y^*) is an accumulation point of the sequence $\{(x^{(k)}, y^{(k)})\}_{k=0}^{\infty}$, there is a subsequence $\{k_\ell\}$ such that $k_\ell \to \infty$ and

$$(x^{k_\ell}, y^{k_\ell}, z^{k_\ell}) \to (x^*, y^*, z^*).$$

Since each $z^{(k_\ell)} \in Z(x^{(k_\ell)})$, we can see that $z^* \in Z(x^*)$. The feasible set of (P_{k_ℓ}) contains that of (5.1), so

$$F(x^*, y^*) = \lim_{\ell \to \infty} F(x^{(k_\ell)}, y^{(k_\ell)}) \le F^*,$$

where F^* is the optimal value of the bilevel optimization (5.1). (The polynomial $F(x, y)$ is a continuous function.) To prove $F(x^*, y^*) \ge F^*$, we show that (x^*, y^*) is feasible for problem (5.1). Define the functions

$$H(x, y, z) := f(x, z) - f(x, y), \quad \phi(x, y) := \inf_{z \in Z(x)} H(x, y, z). \tag{5.14}$$

Observe that $\phi(x, y) = v(x) - f(x, y) \le 0$ for all $(x, y) \in \mathcal{U}$ and $\phi(x^*, y^*) = 0$ if and only if (x^*, y^*) is feasible for (5.1). Since $v(x)$ is continuous at x^*, we have $\phi(x^*, y^*) \le 0$. Next, we show that $\phi(x^*, y^*) \ge 0$. For an arbitrary $k' \in \mathbb{N}$, and for all $k_\ell \ge k'$, the point $(x^{(k_\ell)}, y^{(k_\ell)})$ is feasible for $(P_{k'})$, so

$$H(x^{(k_\ell)}, y^{(k_\ell)}, z) \ge 0 \quad \forall z \in \mathcal{V}_{k_\ell}^{(k')}$$

where $\mathcal{V}_{k_\ell}^{(k')}$ is the set defined as

$$\mathcal{V}_{k_\ell}^{(k')} := \left\{ q^{(0)}(x^{(k_\ell)}, y^{(k_\ell)}),\ q^{(1)}(x^{(k_\ell)}, y^{(k_\ell)}), \ldots,\ q^{(k'-1)}(x^{(k_\ell)}, y^{(k_\ell)}) \right\}.$$

As $\ell \to \infty$, we can get

$$H(x^*, y^*, z) \ge 0 \quad \forall z \in \mathcal{V}_*^{(k')}, \tag{5.15}$$

where the set $\mathcal{V}_*^{(k')}$ is

$$\mathcal{V}_*^{(k')} := \left\{q^{(0)}(x^*, y^*),\ q^{(1)}(x^*, y^*),\ \ldots,\ q^{(k'-1)}(x^*, y^*)\right\}.$$

The inequality (5.15) holds for all k', so

$$H(x^*, y^*, z) \geq 0 \quad \forall z \in T := \{q^{(k)}(x^*, y^*)\}_{k\in\mathbb{N}}. \tag{5.16}$$

It follows that

$$H(x^*, y^*, q^{(k_\ell)}(x^*, y^*)) \geq 0.$$

In Algorithm 5.3, each point $z^{(k_\ell)} \in Z(x^{(k_\ell)})$ satisfies

$$\phi(x^{(k_\ell)}, y^{(k_\ell)}) \;=\; H(x^{(k_\ell)}, y^{(k_\ell)}, z^{(k_\ell)}).$$

Therefore, we have

$$\begin{array}{rcr} \phi(x^*, y^*) & = & \phi(x^{(k_\ell)}, y^{(k_\ell)}) + \phi(x^*, y^*) - \phi(x^{(k_\ell)}, y^{(k_\ell)}) \\ & \geq & \Big(H(x^{(k_\ell)}, y^{(k_\ell)}, z^{(k_\ell)}) - H(x^*, y^*, q^{(k_\ell)}(x^*, y^*))\Big) + \\ & & \Big(\phi(x^*, y^*) - \phi(x^{(k_\ell)}, y^{(k_\ell)})\Big). \end{array} \tag{5.17}$$

Since $z^{(k_\ell)} = q^{(k_\ell)}(x^{(k_\ell)}, y^{(k_\ell)})$, by the condition (d), we know that

$$\lim_{\ell\to\infty} z^{(k_\ell)} = \lim_{\ell\to\infty} q^{(k_\ell)}(x^{(k_\ell)}, y^{(k_\ell)}) = \lim_{\ell\to\infty} q^{(k_\ell)}(x^*, y^*),$$

$$H(x^{(k_\ell)}, y^{(k_\ell)}, z^{(k_\ell)}) - H(x^*, y^*, q^{(k_\ell)}(x^*, y^*)) \to 0 \quad \text{as} \quad \ell \to \infty,$$

by the continuity of the polynomial function $H(x, y, z)$ at (x^*, y^*, z^*). By the assumption, $v(x)$ is continuous at x^*, so $\phi(x, y) = v(x) - f(x, y)$ is also continuous at (x^*, y^*). Letting $\ell \to \infty$ in (5.17), we get $\phi(x^*, y^*) \geq 0$. Thus, (x^*, y^*) is feasible for (5.1) and so $F(x^*, y^*) \geq F^*$. In the earlier, we already proved $F(x^*, y^*) \leq F^*$, so (x^*, y^*) is a global optimizer of (5.1), i.e., (x^*, y^*) is a global minimizer of the bilevel optimization (5.1). □

5.3 LMEs and polynomial extensions

The LMEs and polynomial extensions are important in Algorithm 5.3. In this section, we discuss about LMEs and polynomial extensions.

The LMEs are firstly introduced in [85]. The conclusions from [85] implies that, when (5.1) is a SBOP, there exists a polynomial matrix $W(y)$ satisfying $W(y)G(y) = I_{m_2}$ for generic g. For GBOPs, there typically does not exist $W(x,y)$ such that $W(x,y)G(x,y) = I_{m_2}$. This is because the matrix $G(x,y)$ in (5.5) is typically not full column rank for all complex $x \in \mathbb{C}^n$, $y \in \mathbb{C}^p$. However, we can always find a matrix polynomial $W(x,y)$ such that

$$W(x,y)G(x,y) \;=\; \operatorname{diag}[d(x,y)], \tag{5.18}$$

for a denominator polynomial vector

$$d(x,y) \;:=\; \big(d_1(x,y), \ldots, d_{m_2}(x,y)\big)$$

which is nonnegative on $\mathcal{U}$. Such $W(x,y), d(x,y)$ are not unique. In computation, we prefer that $W(x,y), d(x,y)$ have low degrees and $d(x,y) > 0$ on $\mathcal{U}$ (or $d(x,y)$ has as few as possible zeros on $\mathcal{U}$). We would like to remark that there always exist such $W(x,y), d(x,y)$ satisfying (5.18). Note that $H(x,y) := G(x,y)^T G(x,y)$ is a psd matrix polynomial. If the determinant $\det H(x,y)$ is not identically zero (this is the general case), then the adjoint matrix $\operatorname{adj}\big(H(x,y)\big)$ satisfies

$$\operatorname{adj}\big(H(x,y)\big)H(x,y) \;=\; \det H(x,y) I_{m_2}.$$

Then the equation (5.18) is satisfied for

$$W(x,y) := \operatorname{adj}\big(H(x,y)\big)G(x,y)^T, \quad d(x,y) = \det H(x,y)\mathbf{1}_{m_2}.$$

The above choice for $W(x,y), d(x,y)$ may not be very practical in computation, because they typically have high degrees. In applications, there often exist more suitable choices for $W(x,y), d(x,y)$ with much lower degrees.

Example 5.7. *Consider the lower level optimization problem*

$$\begin{cases} \min\limits_{y \in \mathbb{R}^2} & x_1 y_1 + x_2 y_2 \\ \;\; s.t. & (2y_1 - y_2, x_1 - y_1, y_2, x_2 - y_2) \geq 0. \end{cases}$$

The matrix $G(x,y)$ and $\hat{f}(x,y)$ in (5.5) are:

$$G(x,y) = \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 0 & 1 & -1 \\ 2y_1 - y_2 & 0 & 0 & 0 \\ 0 & x_1 - y_1 & 0 & 0 \\ 0 & 0 & y_2 & 0 \\ 0 & 0 & 0 & x_2 - y_2 \end{pmatrix}, \quad \hat{f}(x,y) = \begin{pmatrix} x_1 \\ x_2 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

The equation (5.18) holds for the denominator vector

$$d(x,y) = \big(2x_1 - y_2,\, 2x_1 - y_2,\, x_2(2x_1 - y_2),\, x_2(2x_1 - y_2)\big)$$

and the matrix $W(x,y)$ which is given as follows

$$\begin{pmatrix} x_1 - y_1 & 0 & 1 & 1 & 0 & 0 \\ y_2 - 2y_1 & 0 & 2 & 2 & 0 & 0 \\ (x_2 - y_2)(x_1 - y_1) & (x_2 - y_2)(2x_1 - y_2) & x_2 - y_2 & x_2 - y_2 & 2x_1 - y_2 & 2x_1 - y_2 \\ y_2(y_1 - x_1) & y_2(y_2 - 2x_1) & -y_2 & -y_2 & 2x_1 - y_2 & 2x_1 - y_2 \end{pmatrix}.$$

Note that $d(x,y) \geq 0$ for all feasible (x,y).

We also give a heuristic method to compute $W(x,y)$, $d(x,y)$. Select a point $(\hat{x},\hat{y}) \in \mathcal{U}$. For a priori low degree ℓ, we consider the following convex optimization in $W(x,y)$, $d(x,y)$:

$$\left\{ \begin{array}{rl} \max & \gamma_1 + \cdots + \gamma_{m_2} \\ s.t. & W(x,y)G(x,y) = \operatorname{diag}[d(x,y)], \\ & d(\hat{x},\hat{y}) = \mathbf{1}_{m_2},\ \gamma_1 \geq 0, \ldots, \gamma_{m_2} \geq 0, \\ & W(x,y) \in \Big(\mathbb{R}[x,y]_{2\ell - \deg(G)}\Big)^{m_2 \times (p+m_2)}, \\ & d_j(x,y) - \gamma_j \in \operatorname{Ideal}[\Phi]_{2\ell} + \operatorname{Qmod}[\Psi]_{2\ell}\ (j \in [m_2]). \end{array} \right. \tag{5.19}$$

In the above, the polynomial tuples Φ, Ψ are

$$\Phi := \{h_i\}_{i \in \mathcal{E}_1} \cup \{g_i\}_{i \in \mathcal{E}_2}, \quad \Psi := \{h_j\}_{j \in \mathcal{I}_1} \cup \{g_j\}_{j \in \mathcal{I}_2}. \tag{5.20}$$

We can construct a polynomial extension, required in Assumption 5.2, for many bilevel optimization problems. If (P_x) has linear equality constraints, we can get rid of them by eliminating variables. If (P_x) has nonlinear equality constraints, generally there is no polynomial $q(x,y)$ satisfying Assumption 5.2, unless the corresponding algebraic set is rational. So, we consider cases that (P_x) has no equality constraints, i.e., the label set $\mathcal{E}_2 = \emptyset$. Moreover, we assume the polynomials $g_j(x,z)$ are linear in z, for each $j \in \mathcal{I}_2$. Recall the polynomial tuples Φ, Ψ given as in (5.20). For a priori degree ℓ and for a given triple $(\hat{x},\hat{y},\hat{z})$, we consider the following polynomial system about q:

$$\left\{ \begin{array}{l} q(\hat{x},\hat{y}) = \hat{z}, \\ g_j(x,q) \in \operatorname{Ideal}[\Phi]_{2\ell} + \operatorname{Qmod}[\Psi]_{2\ell}\,(j \in \mathcal{I}_2), \\ q = (q_1, \ldots, q_p) \in \big(\mathbb{R}[x,y]\big)^p. \end{array} \right. \tag{5.21}$$

The second constraint in (5.21) implies that

$$g_j\big(x, q(x,y)\big) \geq 0, \quad \forall (x,y) \in \mathcal{U},\, j \in \mathcal{I}_2.$$

Hence q obtained as above must satisfy Assumption 5.2. The above program can be solved by the software `Yalmip` [70].

Example 5.8. *Consider Example 5.7 with*

$$\begin{aligned}
&\hat{x} = (1,0),\ \hat{y} = (1,0),\ \hat{z} = (0,0),\\
&h(x,y) = (3x_1 - x_2, x_2, x_2 - x_1 + 1),\\
&g(x,y) = (2y_1 - y_2, x_1 - y_1, y_2, x_2 - y_2).
\end{aligned}$$

For $\ell = 2$, a satisfactory $q := (q_1, q_2)$ for (5.21) is

$$q_1(x,y) = x_2/3, \quad q_2(x,y) = 2x_2/3,$$

because $g(x,q) = \frac{1}{3}(0, h_1(x,y), 2h_2(x,y), h_2(x,y))$ and

$$h_1(x,y),\, h_2(x,y) \;\in\; \mathit{Ideal}[\Phi]_{2\ell} + \mathit{Qmod}[\Psi]_{2\ell}.$$

For computational convenience, we prefer explicit expressions for $q(x,y)$. In the following, we give explicit expressions for various cases of bilevel optimization problems.

Simple bilevel optimization. If the lower feasible set is independent of x, i.e., $Z(x) \equiv Z$, then we can just simply choose

$$q(x,y) \;:=\; z$$

in Assumption 5.2, for all $z \in Z$ and all $(x,y) \in \mathcal{U}$. It is a constant polynomial function. Therefore, Assumption 5.2 is always satisfied for all SBOPs.

Box constraints. Suppose the feasible set $Z(x)$ of (P_x) is given as

$$l(x) \;\leq\; Az \;\leq\; u(x),$$

where $A := [a_1, \ldots, a_{m_2}]^T \in \mathbb{R}^{m_2 \times p}$ is a full row rank matrix and $m_2 \leq p$. Let $a_{m_2+1}, \ldots, a_p$ be vectors such that the matrix

$$B \;:=\; [a_1, \ldots, a_{m_2}, a_{m_2+1}, \ldots, a_p]^T \in \mathbb{R}^{p \times p}$$

is invertible. Then the linear coordinate transformation $z = B^{-1}w$ makes the constraints become the box constraints $l_j(x) \le w_j \le u(x)_j$, $j \in [m_2]$. Hence we can choose $q = B^{-1}q'$, where $q' := (q'_1, \dots, q'_p)$ as

$$q'_j(x,y) := \begin{cases} \mu_j l_j(x) + (1-\mu_j)u_j(x), & j = 1, \dots, m_2, \\ (By)_j, & j = m_2+1, \dots, p. \end{cases}$$

where each scalar

$$\mu_j := (u_j(\hat{x}) - (B\hat{z})_j)/(u_j(\hat{x}) - l_j(\hat{x})) \in [0,1].$$

For the special case that $u_j(\hat{x}) - l_j(\hat{x}) = 0$, we just set $\mu_j = 0$. One can simply verify that $q(x,y) \in Z(x)$ for all $(x,y) \in \mathcal{U}$.

Simplex constraints. Suppose that the feasible set $Z(x)$ of (P_x) is given as

$$a^T z \le u(x), \quad z_j \ge l_j(x)\,(j = 1, \dots, p),$$

where $a := (a_1, \dots, a_p) \in \mathbb{R}^p_+$, $u(x)$ and all $l_j(x)$ are polynomials in x. We can choose $q := (q_1, \dots, q_p)$ as

$$q_j(x,y) := c_j \cdot \big(u(x) - a^T l(x)\big) + l_j(x),$$

where each $c_j := (\hat{z}_j - l_j(\hat{x}))/(u(\hat{x}) - a^T l(\hat{x})) \ge 0$. In particular, we set all $c_j = 0$ if $u(\hat{x}) - a^T l(\hat{x}) = 0$. Note that

$$q_j(\hat{x}, \hat{y}) = l_j(\hat{x}) + c_j \cdot (u(\hat{x}) - a^T l(\hat{x})) = \hat{z}_j.$$

For all $(x,y) \in \mathcal{U}$, it is clear that $q(x,y) \ge l(x)$. In addition, we have

$$a^T q(x,y) = a^T l(x)(1 - \sum_{j=1}^{p} a_j c_j) + (\sum_{j=1}^{p} a_j c_j) u(x) \le u(x)$$

since $a^T l(x) \le u(x)$ and $a_1 c_1 + \cdots a_p c_p \le 1$. Therefore, $q(x,y) \in Z(x)$ for all $(x,y) \in \mathcal{U}$.

Annular constraints. Suppose the lower level feasible set is

$$Z(x) = \left\{ y \in \mathbb{R}^p \,\middle|\, r(x) \le \|y - a(x)\|_d \le R(x) \right\},$$

where $\|z\|_d := \sqrt[d]{\sum_{i=1}^{p} |z_i|^d}$ and $a(x) := [a_1(x), \dots, a_p(x)]$ is a polynomial vector, and $r(x), R(x)$ are polynomials such that $0 \le r(x) \le R(x)$ on $\mathcal{U}$. We can choose

$$q(x,y) := a(x) + q'(x)s,$$

where $q'(x) := \mu_1 r(x) + \mu_2 R(x)$, μ_1, μ_2 are scalars such that

$$\|\hat{z} - a(\hat{x})\|_d = \mu_1 r(\hat{x}) + \mu_2 R(\hat{x}), \quad \mu_1, \mu_2 \geq 0, \quad \mu_1 + \mu_2 = 1,$$

and $s := (s_1, \ldots, s_p)$ is the vector such that

$$s_i := \frac{\hat{z}_i - a_i(\hat{x})}{\|\hat{z} - a(\hat{x})\|_d}, \quad i = 1, \ldots, p.$$

(For the special case that $\hat{z} = a(\hat{x})$, we just set all $s_i = p^{-1/d}$.) Then,

$$\begin{aligned} \hat{z} - q(\hat{x}, \hat{y}) &= (\hat{z} - a(\hat{x})) - (q(\hat{x}, \hat{y}) - a(\hat{x})) \\ &= (\hat{z} - a(\hat{x})) - q'(\hat{x})s = 0. \end{aligned}$$

since $q'(\hat{x}) = \|\hat{z} - a(\hat{x})\|_d$. Moreover,

$$\|q(x, y) - a(x)\|_d = \|q'(x)s\|_d = |q'(x)| \cdot \|s\|_d = |q'(x)|.$$

Because $0 \leq r(x) \leq R(x)$ on $\mathcal{U}$, we must have

$$r(x) \leq \|q(x, y) - a(x)\|_d \leq R(x).$$

This means that $q(x, y)$ satisfies Assumption 5.2.

5.4 Numerical experiments

In this section, we report numerical results of applying Algorithm 5.3 to solve bilevel polynomial optimization problems. The computation is implemented in `MATLAB` R2018a, in a Laptop with CPU 8th Generation Intel® Core™ i5-8250U and RAM 16 GB. The software `GloptiPoly 3` [48] and `SeDuMi` [105] are used to solve the polynomial optimization problems in Algorithm 5.3. In this section, we use the following notation.

- The LMEs in Assumption 5.1 are denoted by $\lambda(x, y)$, which are computed by symbolic Gaussian elimination on the equation (5.18).
- The notation (P) denotes the bilevel optimization (5.1). Its optimal value and optimizers are denoted by F^* and (x^*, y^*) respectively.
- The (P_k) denotes the relaxed polynomial optimization in the kth loop of Algorithm 5.3. Its optimal value and minimizers are denoted as F_k^* and $(x^{(k)}, y^{(k)})$ respectively.

- The (Q_k) denotes the lower level optimization problem (5.12) in the kth loop of Algorithm 5.3. Its optimal value and minimizers are denoted as υ_k and $z^{(k)}$ respectively.

- We always have $\upsilon_k \le 0$. Note that $y^{(k)}$ is a minimizer of (5.12) if and only if $\upsilon_k = 0$. Due to numerical round-off errors, we cannot have $\upsilon_k = 0$ exactly. We view $y^{(k)}$ as a minimizer of (5.12) if $\upsilon_k \ge -\epsilon$, for a tiny scalar ϵ (e.g., 10^{-6}).

Example 5.9. *First, we apply Algorithm 5.3 to solve SBOPs. The displayed problems are respectively from [66, Example 5.2], [1, Example 3], [29, Example 3.8], [91, Example 5.2] and [104, Example 2]. All but the first problem are solved successfully in the initial loop $k = 0$. The computational results are shown in Table 5.1.*

Example 5.10. *[77, Example 2] Consider the general bilevel optimization*

$$\begin{cases} \min\limits_{x\in\mathbb{R}^2, y\in\mathbb{R}^3} & y_1^2 + y_3^2 - y_1y_3 - 4y_2 - 7x_1 + 4x_2 \\ s.t. & (x_1, x_2, 1 - x_1 - x_2) \ge 0,\ y \in S(x), \end{cases}$$

where $S(x)$ is the optimizer set of

$$\begin{cases} \min\limits_{z\in\mathbb{R}^3} & z_1^2 + 0.5z_2^2 + 0.5z_3^2 + z_1z_2 + (1 - 3x_1)z_1 + (1 + x_2)z_2 \\ s.t. & (-2z_1 - z_2 + z_3 - x_1 + 2x_2 - 2, z_1, z_2, z_3) \ge 0. \end{cases}$$

The LME can be computed from $D(x, y)\lambda(x, y) = W_1(x, y)\nabla_z f(x, y)$, where

$$W_1(x, y) = \begin{pmatrix} y_1 & y_2 & y_3 \\ 2 + x_1 + 2y_1 - 2x_2 & 2y_2 & 2y_3 \\ y_1 & 2 + x_1 + y_2 - 2x_2 & y_3 \\ -y_1 & -y_2 & 2 + x_1 - 2x_2 - y_3 \end{pmatrix},$$

and $D = diag\{d_1, \dots, d_4\}$ for the denominators $(i = 1, 2, 3, 4)$

$$d_i(x, y) = 2 + x_1 - 2x_2 = 3h_1(x, y) + 2h_3(x, y) \ge 0,\ \forall (x, y) \in \mathcal{U}.$$

By Algorithm 5.3, we get the optimizer for this bilevel optimization in the initial loop $k = 0$. The computational results are shown in Table 5.2.

Example 5.11. *[91, Example 5.8] Consider the general bilevel optimization*

$$\begin{cases} \min\limits_{x,y\in\mathbb{R}^4} & (e^Tx)(e^Ty) \\ s.t. & (1 - x^Tx, x_4 - y_3^2, x_1 - y_2y_4) \ge 0, \\ & y \in S(x), \end{cases}$$

Table 5.1: Computational results for some SBOPs.

$\begin{array}{rl} \min\limits_{x,y\in\mathbb{R}^1} & x+y \\ s.t. & (x+1,1-x)\geq 0, \\ & y\in \underset{z\in\mathbb{R}^1}{\operatorname{argmin}} \ \frac{1}{2}xz^2-\frac{1}{3}z^3 \\ & \qquad s.t. \ (z+1,1-z)\geq 0. \end{array}$	$F^*=-1.2380\cdot 10^{-8}$, $v^*=-3.9587\cdot 10^{-8}$, $x^*=-1.0000$, $y^*=1.0000$, time $=0.89$.
$\begin{array}{rl} \min\limits_{x,y\in\mathbb{R}^2} & x_1^2-2x_1+x_2^2-2x_2+y_1^2+y_2^2 \\ s.t. & (x_1,x_2,y_1,y_2,2-x_1)\geq 0, \\ & y\in \underset{z\in\mathbb{R}^2}{\operatorname{argmin}} \ z_1^2-2x_1z_1+z_2^2-2x_2z_2 \\ & \qquad s.t. \ 0.25-(z_1-1)^2\geq 0, \\ & \qquad\qquad 0.25-(z_2-1)^2\geq 0. \end{array}$	$F^*=-1.0000$, $v^*=-1.3113\cdot 10^{-9}$, $x^*=(0.5000,0.5000)$, $y^*=(0.5000,0.5000)$, time $=0.34$.
$\begin{array}{rl} \min\limits_{x,y\in\mathbb{R}^2} & 2x_1+x_2-2y_1+y_2 \\ s.t. & (1+x_1,1-x_1,1+x_2,-0.75-x_2)\geq 0, \\ & y\in \underset{z\in\mathbb{R}^2}{\operatorname{argmin}} \ x^Tz \\ & \qquad s.t. \ (2z_1-z_2,2-z_1)\geq 0, \\ & \qquad\qquad (z_2,2-z_2)\geq 0. \end{array}$	$F^*=-5.0000$, $v^*=-1.4163\cdot 10^{-8}$, $x^*=(-1.0000,-1.0000)$, $y^*=(2.0000,2.0000)$, time $=0.27$.
$\begin{array}{rl} \min\limits_{x\in\mathbb{R}^2,y\in\mathbb{R}^3} & x_1y_1+x_2y_2+x_1x_2y_1y_2y_3 \\ s.t. & (1-x_1^2,1-x_2^2,x_1^2-y_1y_2)\geq 0, \\ & y\in \underset{z\in\mathbb{R}^3}{\operatorname{argmin}} \ x_1z_1^2+x_2^2z_2z_3-z_1z_3^2 \\ & \qquad s.t. \ (z^Tz-1,2-z^Tz)\geq 0. \end{array}$	$F^*=-1.7095$, $v^*=-1.3995\cdot 10^{-9}$, $x^*=(-1.0000,-1.0000)$, $y^*=(1.1097,0.3143,-0.8184)$, time $=6.43$.
$\begin{array}{rl} \min\limits_{x,y\in\mathbb{R}^2} & (x_1-30)^2+(x_2-20)^2-20y_1+20y_2 \\ s.t. & (x_1+2x_2-30,25-x_1-x_2,15-x_2)\geq 0, \\ & y\in \underset{z\in\mathbb{R}^2}{\operatorname{argmin}} \ (x_1-z_1)^2+(x_2-z_2)^2 \\ & \qquad s.t. \ (10-z_1,10-z_2,z_1,z_2)\geq 0. \end{array}$	$F^*=225.0000$, $v^*=-1.6835\cdot 10^{-9}$, $x^*=(20.0000,5.0000)$, $y^*=(10.0000,5.0000)$, time $=0.27$.

where $S(x)$ is the set of optimizer(s) of

$$\left\{\begin{array}{rl} \min\limits_{z\in\mathbb{R}^4} & x_1z_1+x_2z_2+0.1z_3+0.5z_4-z_3z_4 \\ s.t. & x_1^2+x_3^2+x_2+x_4-z_1^2-2z_2^2-3z_3^2-4z_4^2\geq 0, \\ & z_2z_3-z_1z_4\geq 0. \end{array}\right.$$

The LME can be computed from $D(x,y)\lambda(x,y)=W_1(x,y)\nabla_z f(x,y)$, where

$$W_1(x,y)=y_4\cdot\begin{pmatrix} -y_1y_4 & -y_2y_4 & -y_3y_4 & -y_4^2 \\ 2y_1^2-2(x_1^2+x_3^2+x_2+x_4) & 2y_1y_2 & 2y_1y_3 & 2y_1y_4 \end{pmatrix},$$

Table 5.2: Computational results for Example 5.10

(P_0)	$F_0^* = 0.6389$,
	$x^{(0)} = (0.6111,\ 0.3889)$, $\quad y^{(0)} = (0.0000,\ 0.0000,\ 1.8332)$,
(Q_0)	$\upsilon_0 = -6.7295 \cdot 10^{-9} \to$ stop.
Time	1.09 seconds,
Output	$F^* = F_0^*,\ x^* = x^{(0)},\ y^* = y^{(0)}$.

and $D = diag\{d_1, d_2\}$ for the denominators

$$\begin{aligned} d_1(x,y) = d_2(x,y) &= 2y_4^2(x_1^2 + x_3^2 + x_2 + x_4) \\ &\geq 2y_4^2(y_1^2 + 2y_2^2 + 3y_3^2 + 4y_4^2) \geq 0,\ \forall (x,y) \in \mathcal{U}. \end{aligned}$$

By Algorithm 5.3, we get the optimizer for this bilevel optimization in the initial loop $k = 0$. The computational results are shown in Table 5.3.

Table 5.3: Computational results for Example 5.11

(P_0)	$F_0^* = -3.5050$,
	$x^{(0)} = (0.5442, 0.4682, 0.4904, 0.4942)$,
	$y^{(0)} = (-0.7791, -0.5034, -0.2871, -0.1855)$,
(Q_0)	$\upsilon_0 = -1.6143 \cdot 10^{-9} \to$ stop.
Time	49.08 seconds,
Output	$F^* = F_0^*,\ x^* = x^{(0)},\ y^* = y^{(0)}$.

Example 5.12. *Consider the general bilevel optimization*

$$\begin{cases} \min\limits_{x,y \in \mathbb{R}^4} & y_1x_1^2 + y_2x_2^2 - y_3x_3 - y_4x_4 \\ s.t. & (x_1 - 1, x_2 - 1, 4 - x_1 - x_2) \geq 0, \\ & (x_3 - 1, 2 - x_4, x_3^2 - 2x_4, 8 - x^Tx) \geq 0, \\ & y \in S(x), \end{cases}$$

where $S(x)$ is the set of optimizer(s) of

$$\begin{cases} \min\limits_{z \in \mathbb{R}^4} & -z_1z_2 + z_3 + z_4 \\ s.t. & (z_1, z_2, z_3 - x_4, z_4 - x_3) \geq 0, \\ & (4x_1x_2 - x_1z_1 - x_2z_2, 3 - z_3 - z_4) \geq 0. \end{cases}$$

The LME can be computed from $D(x,y)\lambda(x,y) = W_1(x,y)\nabla_z f(x,y)$, where

$$W_1(x,y) \;=\; \begin{pmatrix} x_1(4x_2+y_2)-x_1y_1-x_2y_2 & -x_1y_2 & 0 & 0 \\ -x_1y_1 & 4x_1x_2-x_2y_2 & 0 & 0 \\ 0 & 0 & 3-x_3-y_3 & x_3-y_4 \\ 0 & 0 & x_4-y_3 & 3-x_4-y_4 \\ -y_1 & -y_2 & 0 & 0 \\ 0 & 0 & x_4-y_3 & x_3-y_4 \end{pmatrix},$$

and $D = diag\{d\}$ for the denominator vector $d(x,y)$ as follows

$$\begin{aligned} d(x,y) = (&4x_1x_2+x_1y_2-x_2y_2, 4x_1x_2+x_1y_2-x_2y_2, \\ &\qquad 3-x_3-x_4, 3-x_3-x_4, 4x_1x_2+x_1y_2-x_2y_2, 3-x_3-x_4). \end{aligned}$$

It is clear that $d(x,y)\ge 0$ for all feasible (x,y). The polynomial function $q := (q_1,q_2,q_3,q_4)$ in Assumption 5.2 can be given as

$$q = \big(\mu_1x_2,\ \mu_2x_1,\ x_4+\mu_3(3+x_3+x_4),\ x_3+\mu_4(3+x_3+x_4)\big), \tag{5.22}$$

where

$$\mu = \Big(\frac{\hat z_1}{\hat x_2},\ \frac{\hat z_2}{\hat x_1},\ \frac{\hat z_3-\hat x_4}{3+\hat x_3+\hat x_4},\ \frac{\hat z_4-\hat x_3}{3+\hat x_3+\hat x_4}\Big),$$

for given $(\hat x,\hat y)\in\mathcal{U}$. Since $x_1,x_2,x_3\ge 1$ and $x_4\ge -2\sqrt{2}$, the above μ_1,μ_2,μ_3,μ_4 are well defined. Applying Algorithm 5.3, we get the optimizer for this bilevel optimization in the loop $k=1$. The computational results are shown in Table 5.4.

Example 5.13. *Consider the general bilevel optimization problem*

$$\begin{cases} \min\limits_{x,y\in\mathbb{R}^4} & x_1^2y_4^2 - x_2y_3^2 + x_3y_1 - x_4y_2 \\ s.t. & (4-x_1^2-x_2^2,\ -x_1-x_2^2,\ y_1-x_1,\ \mathbf{1}^Tx)\ge 0, \\ & (x_3+x_4-3,\ 1+x_3-x_4,\ 3-x_3,\ x_4)\ge 0, \\ & y\in S(x), \end{cases}$$

where $S(x)$ is the optimizer(s) set of

$$\begin{cases} \min\limits_{z\in\mathbb{R}^4} & (x_1-z_1)^2+(x_2-z_2)^2+z_3-z_4 \\ s.t. & 4x_3^2-x_1^2-x_2^2+2x_1z_1+2x_2z_2-z^Tz\ge 0\ \cdot \\ & (z_3,\ x_3-z_3,\ z_4,\ x_4-z_4)\ge 0 \end{cases}$$

Table 5.4: Computational results for Example 5.12

(P_0)	$F_0^* = -4.4575$,
	$x^{(0)} = (1.1548,\ 1.1546,\ 1.6458,\ 1.3542)$,
	$y^{(0)} = (0.0000,\ 0.0000,\ 1.3542,\ 1.6458)$,
(Q_0)	$\upsilon_0 = -5.3362 \to$ next loop;
	$z^{(0)} = (2.3093,\ 2.3096,\ 1.3542,\ 1.6458)$,
	$q^{(0)} = (2x_2,\ 2x_1,\ x_4,\ x_3)$ as in (5.22).
(P_1)	$F_1^* = -0.4574$,
	$x^{(1)} = (1.0000,\ 1.0000,\ 1.6458,\ 1.3542)$,
	$y^{(1)} = (2.0000,\ 2.0000,\ 1.3542,\ 1.6458)$,
(Q_1)	$\upsilon_1 = -1.9402 \cdot 10^{-9} \to$ stop.
Time	102.21 seconds,
Output	$F^* = F_1^*,\ x^* = x^{(1)},\ y^* = y^{(1)}$.

The LME can be computed from $D(x,y)\lambda(x,y) = W_1(x,y)\nabla_z f(x,y)$, where

$$W(x,y) = \begin{pmatrix} -1 & 0 & 0 & 0 \\ -(x_3-y_3)y_3 & 0 & (x_3-y_3)(y_1-x_1) & 0 \\ y_3^2 & 0 & -y_3(y_1-x_1) & 0 \\ -(x_4-y_4)y_4 & 0 & 0 & (x_4-y_4)(y_1-x_1) \\ y_4^2 & 0 & 0 & -y_4(y_1-x_1) \end{pmatrix},$$

and $D = diag\{d\}$ for the denominator vector

$$d(x,y) \ = \ (y_1 - x_1) \cdot \big(2,\ x_3 - y_3,\ y_3,\ x_4 - y_4,\ y_4\big).$$

It is clear that $d(x,y) \geq 0$ for all feasible (x,y). The lower level feasible set $Z(x)$ is a mixture of separable and annular constraints:

$$Z(x) = \left\{ z \in \mathbb{R}^4 \left| \begin{array}{c} (z_1-x_1)^2 + (z_2-x_2)^2 + z_3^2 + z_4^2 \leq 4x_3^2, \\ 0 \leq z_3 \leq x_3,\ 0 \leq z_4 \leq x_4 \end{array} \right. \right\}.$$

The polynomial function $q := (q_1, q_2, q_3, q_4)$ in Assumption 5.2 can be given as

$$q = (x_1 + \mu_1 x_3,\ x_2 + \mu_2 x_3,\ \mu_3 x_3,\ \mu_4 x_4), \tag{5.23}$$

where (for a given value $(\hat{x}, \hat{y}, \hat{z})$ of $(x^{(k)},\, y^{(k)},\, z^{(k)})$, q satisfies $q(\hat{x}, \hat{y}) = \hat{z}$)

$$\mu = \Big(\frac{\hat{z}_1 - \hat{x}_1}{\hat{x}_3},\ \frac{\hat{z}_2 - \hat{x}_2}{\hat{x}_3},\ \frac{\hat{z}_3}{\hat{x}_3},\ \frac{\hat{z}_4}{\hat{x}_4}\Big).$$

Table 5.5: Computational results for Example 5.13

(P_0)	$F_0^* = -41.7143$,
	$x^{(0)} = (-1.5616, 1.2496, 3.0000, 4.0000)$,
	$y^{(0)} = (-1.5616, 6.4458, 3.0000, 0.0008)$,
(Q_0)	$v_0 = -33.9991$,
	$z^{(0)} = (-1.5615, 1.2496, 0.0000, 4.0000)$,
	$q^{(0)} = (x_1, x_2, 0, x_4)$ as in (5.23).
(P_1)	$F_1^* = -6.0000$,
	$x^{(1)} = (-2.0000, 0.0001, 3.0000, 0.0001)$,
	$y^{(1)} = (-2.0000, 0.0001, -0.0000, 0.0001)$,
(Q_1)	$v_1 = -2.7612 \cdot 10^{-9} \to$ stop.
Time	3.42 seconds,
Output	$F^* = F_1^*,\ x^* = x^{(1)},\ y^* = y^{(1)}$.

Since $1 \le \hat{x}_3 \le 3$ *and* $0 \le \hat{x}_4 \le 1 + \hat{x}_3$*, we have* $\mu_4 = 0$ *for the special case when* $\hat{x}_4 = 0$*, then the above* q *is well-defined. This bilevel optimization was solved by Algorithm 5.3 in the loop* $k = 1$*. The computational results are shown in Table 5.5.*

Acknowledgments. This Chapter, in full, is a reprint of the material as it appears in *SIAM Journal on Optimization* 2021. The dissertation author coauthored this paper with Nie, Jiawang; Wang, Li and Ye, Jane J.

Chapter 6

Rational Generalized Nash Equilibrium Problems

6.1 Generalized Nash equilibrium problems

The generalized Nash equilibrium problem is a kind of games to find strategies for a group of players such that each player's objective cannot be further optimized, for given strategies of other players. Suppose there are N players and the ith player's strategy is the real vector $x_i \in \mathbb{R}^{n_i}$. We write that

$$x_i := (x_{i,1}, \ldots, x_{i,n_i}), \quad x := (x_1, \ldots, x_N).$$

Let $n := n_1 + \cdots + n_N$. When the ith player's strategy x_i is focused, we also write that $x = (x_i, x_{-i})$, where

$$x_{-i} := (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N).$$

A strategy tuple $u := (u_1, \ldots, u_N)$ is said to be a generalized Nash equilibrium (GNE) if each u_i is the optimizer for the ith player's optimization

$$\mathrm{F}_i(u_{-i}) : \begin{cases} \min\limits_{x_i \in \mathbb{R}^{n_i}} & f_i(x_i, u_{-i}) \\ s.t. & x_i \in X_i(u_{-i}). \end{cases} \tag{6.1}$$

In the above, the $X_i(u_{-i})$ is the feasible set and $f_i(x_i, u_{-i})$ is the ith player's objective. They are parameterized by $u_{-i} = (u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_N)$. Each player's optimization is parameterized by strategies of other players. We denote by $\mathcal{S}$ the set of all GNEs and denote by $\mathcal{S}_i(u_{-i})$ the set of minimizers for the optimization $\mathrm{F}_i(u_{-i})$. The entire feasible strategy set is

$$X := \{(x_1, \dots, x_N) \,|\, x_i \in X_i(x_{-i}) \,, i = 1, \dots, N\}\,. \tag{6.2}$$

A strategy tuple $x = (x_1, \dots, x_N)$ is said to be feasible if each $x_i \in X_i(x_{-i})$.

The *rational generalized Nash equilibrium problems* (rGNEPs) are GNEPs whose all the objectives and constraining functions are rational functions in x. We assume the ith player's feasible set is given as

$$X_i(x_{-i}) \;=\; \left\{ x_i \in \mathbb{R}^{n_i} \;\middle|\; \begin{array}{l} g_{i,j}(x_i, x_{-i}) = 0 \, (j \in \mathcal{I}_0^{(i)}), \\ g_{i,j}(x_i, x_{-i}) \ge 0 \, (j \in \mathcal{I}_1^{(i)}), \\ g_{i,j}(x_i, x_{-i}) > 0 \, (j \in \mathcal{I}_2^{(i)}) \end{array} \right\}, \tag{6.3}$$

where $\mathcal{I}_0^{(i)}, \mathcal{I}_1^{(i)}, \mathcal{I}_2^{(i)}$ are respectively the labelling sets (possibly empty) for equality, weak inequality and strict inequality constraints. For the rational function to be well defined, we assume all denominators are positive in the feasible set. If this is not the case, we can add strict inequality constraints for denominators. Rational functions frequently appear in GNEPs. When functions are polynomials, the GNEPs are studied in the recent work [87–89]. In particular, a special case of GNEPs is the *Nash Equilibrium Problems (NEPs)*: each feasible set $X_i(x_{-i})$ is independent of x_{-i}.

One may reformulate rGNEPs equivalently as polynomial GNEPs by introducing new variables or change the description of the feasible set. However, doing so may loose some useful properties. For instance, the convexity may be lost if we use polynomial reformulations. The following is such an example.

Example 6.1. *Consider the 2-player rational GNEP*

$$\begin{array}{cl|cl} \min\limits_{x_1 \in \mathbb{R}^2} & \frac{2(x_{1,1})^2 + (x_{1,2})^2 + x_{1,1}x_{1,2} \cdot e^T x_2}{x_{1,1}} & \min\limits_{x_2 \in \mathbb{R}^2} & \frac{2(x_{2,1})^2 + (x_{2,2})^2 - x_{2,1}x_{2,2} \cdot e^T x_1}{x_{2,1}} \\ s.t. & x_{1,1} - \frac{x_{2,1}}{x_{1,2}} \ge 0, & s.t. & 1 - e^T(x_2 - x_1) \ge 0, \\ & x_{1,1} > 0,\, x_{1,2} > 0, & & x_{2,1} - 1 \ge 0,\, x_{2,2} - 1 \ge 0. \end{array} \tag{6.4}$$

In the above, $e = [1\ 1]^T$. In the domain $(x_1, x_2) > 0$, each player's optimization is convex in its strategy variable. We can equivalently express this GNEP as polynomial optimization

$$\begin{array}{cl|cl} \min\limits_{x_1 \in \mathbb{R}^3} & x_{3,1}[2(x_{1,1})^2 + (x_{1,2})^2 + x_{1,1}x_{1,2} \cdot \hat{e}^T x_2] & \min\limits_{x_2 \in \mathbb{R}^3} & x_{2,3}[2(x_{2,1})^2 + (x_{2,2})^2 - x_{2,1}x_{2,2} \cdot \hat{e}^T x_1] \\ s.t. & x_{1,1}x_{1,2} - x_{2,1} \ge 0, & s.t. & 1 - \hat{e}^T(x_2 - x_1) \ge 0, \\ & x_{1,1} > 0,\, x_{1,2} > 0, & & x_{2,1} - 1 \ge 0,\, x_{2,2} - 1 \ge 0, \\ & x_{1,1}x_{1,3} = 1, & & x_{2,1}x_{2,3} = 1, \end{array}$$

where $\hat{e} = [1\ 1\ 0]^T$. However, the above two optimization problems are not convex.

6.2 An algorithm for rGNEPs

In this section, we propose a new approach for solving rational GNEPs. It requires to solve a hierarchy of rational optimization problems. They are obtained from Lagrange multiplier expressions and feasible extensions of KKT points that are not GNEs. Under some general assumptions, we prove that this hierarchy either returns a GNE or detects its nonexistence.

One can express Lagrange multipliers as rational functions on the KKT set. Recall the set X as in (6.2). For the ith player's optimization $\mathrm{F}_i(x_{-i})$, we suppose that there is a tuple $\tau_i = (\tau_{i,j})_{j\in\mathcal{I}_0^{(i)}\cup\mathcal{I}_1^{(i)}}$ of rational functions in x, with denominators positive on X, such that

$$\lambda_{i,j} \;=\; \tau_{i,j}(x), \quad j \in \mathcal{I}_0^{(i)} \cup \mathcal{I}_1^{(i)}, \tag{6.5}$$

for each critical pair (x_i, λ_i) of $\mathrm{F}_i(x_{-i})$. Note that the Lagrange multipliers are zero for strict inequality constraints. As discussed in Section 5.3, the rational LMEs exist for general cases. The existence of the LME (6.5) gives the KKT set

$$\mathcal{K} := \left\{ x \in X \;\middle|\; \begin{array}{c} \nabla_{x_i} f_i - \sum\limits_{j\in\mathcal{I}_0^{(i)}\cup\mathcal{I}_1^{(i)}} \tau_{i,j}(x)\nabla_{x_i} g_{i,j}(x) = 0\,(i\in[N]), \\ g_{i,j}(x) \perp \tau_{i,j}(x) \geq 0\,(i\in[N], j\in\mathcal{I}_1^{(i)}) \end{array} \right\}. \tag{6.6}$$

In the above, the symbol $\perp$ denotes the perpendicular relation.

Not every point $u = (u_1, \ldots, u_N) \in \mathcal{K}$ is a GNE. How do we preclude non-GNEs in $\mathcal{K}$? We consider the case that u is not a GNE. Then there exist $i \in [N]$ and a point $v_i \in X_i(u_{-i})$ such that

$$f_i(v_i, u_{-i}) - f_i(u_i, u_{-i}) \;<\; 0.$$

However, if $x := (x_1, \ldots, x_N)$ is a GNE and v_i is also feasible for $\mathrm{F}_i(x_{-i})$, i.e., $v_i \in X_i(x_{-i})$, then x must satisfy the inequality

$$f_i(v_i, x_{-i}) - f_i(x_i, x_{-i}) \geq 0. \tag{6.7}$$

That is, every GNE x satisfies the constraint (6.7) if $v_i \in X_i(x_{-i})$. This is used to solve NEPs in [87]. Unlike NEPs, the feasible set of $X_i(x_{-i})$ depends on x_{-i}. As a result, a point $v_i \in X_i(u_{-i})$ may not be feasible for $\mathrm{F}_i(x_{-i})$, i.e., $v_i \notin X_i(x_{-i})$, for a GNE x. For such a case, the inequality (6.7) may not hold for any GNEs. The following is such an example.

Example 6.2. *Consider the 2-player GNEP*

$$\begin{array}{ll|ll} \min\limits_{x_1 \in \mathbb{R}^2} & (x_{1,1} - x_{1,2})x_{2,1}x_{2,2} - x_1^T x_1 & \min\limits_{x_2 \in \mathbb{R}^2} & 3(x_{2,1} - x_{1,1})^2 + 2(x_{2,2} - x_{1,2})^2 \\ s.t. & 1 - e^T x \geq 0,\ x_1 \geq 0, & s.t. & 2 - e^T x \geq 0,\ x_2 \geq 0. \end{array}$$

It has only two GNEs $x^* = (x_1^*, x_2^*)$:

$$x_1^* = x_2^* = (0.5, 0) \quad \textit{and} \quad x_1^* = x_2^* = (0, 0.5).$$

Consider the point $u = (u_1, u_2) \in \mathcal{K}$, with $u_1 = u_2 = (0, 0)$. The u_1 is not a minimizer of $F_1(u_2)$, so u is not a GNE. The optimizers of $F_1(u_2)$ are $v_1 = (1, 0)$ and $(0, 1)$. One can check that for either GNE x^*, it holds that

$$v_1 \notin X_1(x_2^*), \qquad f_1(v_1, x_2^*) - f_1(x_1^*, x_2^*) = -0.75 < 0.$$

The inequality (6.7) does not hold for any GNE.

The above example shows that the constraint (6.7) may not hold for any GNE. However, if there is a function p_i in x such that

$$v_i = p_i(u), \qquad p_i(x) \in X_i(x_{-i}) \quad \text{for all} \quad x \in \mathcal{K}, \tag{6.8}$$

then the following inequality

$$f_i(p_i(x), x_{-i}) - f_i(x_i, x_{-i}) \,\geq\, 0 \tag{6.9}$$

separates GNEs and non-GNEs. This is because $f_i(x_i, x_{-i}) \leq f_i(p_i(x), x_{-i})$ for every GNE x, since $p_i(x) \in X_i(x_{-i})$. This motivates us to make the following assumption.

Assumption 6.3. *For a given triple (u, i, v_i), with $u \in \mathcal{K}$, $i \in [N]$ and $v_i \in \mathcal{S}_i(u_{-i})$, there exists a vector of rational functions p_i in $x := (x_1, \ldots, x_N)$ such that (6.8) holds.*

The function p_i satisfying (6.8) is called a *feasible extension* of v_i at the point u. This kind of functions are useful for solving bilevel polynomial optimization problems [92]. Based on LMEs and feasible extensions, we propose the following algorithm for solving GNEPs. In the following sections, we will discuss the existence and computation of such p_i.

Algorithm 6.4. For the given GNEP of (6.1), do the following:

Step 0 Find the Lagrange multiplier expressions as in (6.5). Let $\mathscr{U} := \mathcal{K}$ and $k := 0$. Choose a generic positive definite matrix Θ of length $n + 1$.

Step 1 Solve the following optimization (note $[x]_1 = \begin{bmatrix} 1 & x^T \end{bmatrix}^T$)

$$\begin{cases} \min & [x]_1^T \Theta [x]_1 \\ s.t. & x \in \mathscr{U}. \end{cases} \tag{6.10}$$

If (6.10) is infeasible, output that either (6.1) has no GNEs or there is no GNE in the set $\mathcal{K}$. Otherwise, solve it for a minimizer $u := (u_1, \ldots, u_N)$, if it exists.

Step 2 For each $i = 1, \ldots, N$, solve the following optimization

$$\begin{cases} \delta_i := \min & f_i(x_i, u_{-i}) - f_i(u_i, u_{-i}) \\ \qquad\quad s.t. & x_i \in X_i(u_{-i}) \end{cases} \tag{6.11}$$

for a minimizer v_i. Denote the label set

$$\mathcal{N} := \{i \in [N] : \delta_i < 0\}. \tag{6.12}$$

If $\mathcal{N} = \emptyset$, then u is a GNE and stop; otherwise, go to Step 3.

Step 3 For every above triple (u, i, v_i) with $i \in \mathcal{N}$, find a rational feasible extension p_i satisfying (6.8). Then update the set $\mathscr{U}$ as

$$\mathscr{U} := \mathscr{U} \cap \{f_i(p_i(x), x_{-i}) - f_i(x_i, x_{-i}) \geq 0,\ i \in \mathcal{N}\}. \tag{6.13}$$

Then, let $k := k + 1$ and go to Step 1.

We now study the convergence of Algorithm 6.4. First, an interesting case is the convex rational GNEP. A GNEP is said to be convex if every player's optimization problem is convex: for each fixed x_{-i}, the objective $f_i(x_i, x_{-i})$ is convex in x_i, the inequality constraining functions in (6.3) are concave in x_i and all equality constraining functions are linear in x_i. Interestingly, the concavity of constraining functions can be weakened to the convexity of feasible sets under certain assumptions. As in [62], for given x_{-i}, the feasible set $X_i(x_{-i})$ is said to be *nondegenrate* if the gradient $\nabla_{x_i} g_{i,j}(x) \neq 0$ for every $j \in \mathcal{I}_0^{(i)} \cup \mathcal{I}_1^{(i)}$. The set $X_i(x_{-i})$ is said to satisfy the *Slater's condition* if it contains a point that makes all inequalities strictly hold.

Theorem 6.5 ([90]). *Assume the Lagrange multipliers are expressed as in (6.5). Suppose that each objective f_i is convex in x_i, each $g_{i,j}$ is linear in x_i for $j \in \mathcal{I}_0^{(i)}$, and each strategy set $X_i(x_{-i})$ is convex, nondegenerate and satisfies the Slater's condition. Then, Algorithm 6.4 terminates at the initial loop $k = 0$, and it either returns a GNE or detects nonexistence of GNEs.*

Proof. Under the given assumptions, a feasible point is a minimizer of the optimization $\mathrm{F}_i(x_{-i})$ if and only if it is a KKT point. This is shown in [62]. Equivalently, a point is a GNE if and only if it belongs to the set $\mathcal{K}$. If there is a GNE, Algorithm 6.4 can get one in Step 2 for the initial loop $k = 0$, and then it terminates. If there is no GNE, the KKT point set $\mathcal{K}$ is empty, then Algorithm 6.4 terminates in Step 1 for the initial loop. $\square$

Second, we prove that Algorithm 6.4 terminates within finitely many loops under a finiteness assumption on critical points. It is known that a general polynomial optimization problem has finitely many KKT points (see [86]). Recall that $\mathcal{S}$ denotes the set of all GNEs. When the complement $\mathcal{K} \setminus \mathcal{S}$ is a finite set, Algorithm 6.4 must terminate within finitely many loops.

Theorem 6.6 ([90]). *Assume the Lagrange multipliers are expressed as in (6.5). Suppose Assumption 6.3 holds for every triple (u, i, v_i) produced by Algorithm 6.4. If the complement set $\mathcal{K} \setminus \mathcal{S}$ is finite, then Algorithm 6.4 must terminate within finitely many loops, and it either returns a GNE or detects its nonexistence.*

Proof. When $\mathcal{K} \setminus \mathcal{S} = \emptyset$, the algorithm terminates in the initial loop $k = 0$. When $\mathcal{K} \setminus \mathcal{S} \neq \emptyset$ and some $u \in \mathcal{K} \setminus \mathcal{S}$ is the minimizer of (6.10), then $\mathcal{N} \neq \emptyset$. By Assumption 6.3, for each $i \in \mathcal{N}$, there exists $v_i \in \mathcal{S}_i(u)$ such that

$$\delta_i = f_i(v_i, u_{-i}) - f(u_i, u_{-i}) < 0.$$

The set $\mathscr{U}$ is updated with the newly added constraint

$$f_i(p_i(x), x_{-i}) - f(x_i, x_{-i}) \geq 0.$$

The point u does not belong to $\mathscr{U}$ for all future loops. The cardinality of the set $\mathcal{K} \setminus \mathscr{U}$ decreases at least by one, after each loop. Note that $\mathscr{U} \subseteq \mathcal{K}$. Therefore, if $\mathcal{K} \setminus \mathcal{S}$ is a finite set, then Algorithm 6.4 must terminate within finitely many loops.

Next, suppose Algorithm 6.4 terminates with a minimizer u in Step 2. Then $\delta_i \geq 0$ for all i, so every u_i is a minimizer of $\mathrm{F}_i(u_{-i})$, i.e., u is a GNE. $\square$

The KKT point set is finite for general polynomial optimization problems. For some special problems, it may be infinite. When the complement set $\mathcal{K} \setminus \mathcal{S}$ is infinite, Algorithm 6.4 may not be guaranteed to terminate within finitely many loops. However, we can prove its asymptotic convergence under certain assumptions. For each $i = 1, \dots, N$, we define the ith player's value function

$$\nu_i(x_{-i}) := \inf_{x_i \in X_i(x_{-i})} f_i(x_i, x_{-i}). \tag{6.14}$$

The function $\nu_i(x_{-i})$ is continuous under certain conditions, e.g., under the restricted inf-compactness (RIC) condition (see [41, Definition 3.13]). A sequence of functions $\{\phi^{(k)}(x)\}$ is said to be *uniformly continuous* at a point x^* if for each $\epsilon > 0$, there exists $\tau > 0$ such that $\|\phi^{(k)}(x) - \phi^{(k)}(x^*)\| < \epsilon$ for all k and for all x with $\|x - x^*\| < \tau$. The following is the asymptotic convergence result.

Theorem 6.7 ([90]). *For the GNEP (6.1), suppose Lagrange multipliers can be expressed as in (6.5) and Assumption 6.3 holds for every triple (u, i, v_i) produced by Algorithm 6.4. In the kth loop, let $u^{(k)}$, $v_i^{(k)}$ be the minimizers of (6.10), (6.11) respectively and let $p_i^{(k)}$ be the feasible extension in Step 3. Suppose $u^* := (u_1^*, \dots, u_N^*)$ is an accumulation point of the sequence $\{u^{(k)}\}_{k=1}^{\infty}$. If for each $i = 1, \dots, N$,*

i) the strict inequality $g_{i,j}(u^*) > 0$ holds for all $j \in \mathcal{I}_2^{(i)}$, and

ii) the value function $\nu_i(x_{-i})$ is continuous at u_{-i}^*, and

iii) the sequence of feasible extensions $\{p_i^{(k)}\}_{k=1}^{\infty}$ is uniformly continuous at u^*,

then u^* is a GNE for (6.1).

Proof. Up to selection of a subsequence, we can generally assume that $u^{(k)} \to u^*$ as $k \to \infty$. The condition i) implies that $u^* \in X$ and $u_i^* \in X_i(u_{-i}^*)$ for every i. We need to show that each u_i^* is a minimizer for the optimization $\mathrm{F}_i(u_{-i}^*)$. By the definition of ν_i as in (6.14), this is equivalent to showing that

$$\nu_i(u_{-i}^*) - f_i(u^*) \geq 0, \quad i = 1, \dots, N. \tag{6.15}$$

For convenience of notation, let $p_i^{(k)} = x_i$ for each $i \notin \mathcal{N}$, in the kth loop. Since $u^{(k)}$ is feasible for (6.10) in all previous loops, we have that

$$f_i(p_i^{(k')}(u^{(k)}), u_{-i}^{(k)}) - f_i(u^{(k)}) \geq 0, \quad \text{for all} \quad k' \leq k.$$

As $k \to \infty$, the above implies that

$$f_i(p_i^{(k')}(u^*), u_{-i}^*) - f_i(u^*) \geq 0, \quad \text{for all} \quad k'.$$

Then, for every i and for every $k \in \mathbb{N}$,

$$
\begin{aligned}
& \nu_i(u^*_{-i}) - f_i(u^*) \\
= & \big(\nu_i(u^*_{-i}) - f_i(p_i^{(k)}(u^*), u^*_{-i})\big) + \big(f_i(p_i^{(k)}(u^*), u^*_{-i}) - f_i(u^*)\big) \\
\geq & \nu_i(u^*_{-i}) - f_i(p_i^{(k)}(u^*), u^*_{-i}).
\end{aligned} \tag{6.16}
$$

Note that $\nu_i(u_{-i}^{(k)}) = f_i(p_i^{(k)}(u^{(k)}), u_{-i}^{(k)})$ for all k. Under the continuity assumption of ν_i at u^*_{-i}, the convergence $u^{(k)} \to u^*$ implies that

$$\nu_i(u^*_{-i}) = \lim_{k\to\infty} \nu_i(u_{-i}^{(k)}) = \lim_{k\to\infty} f_i(p_i^{(k)}(u^{(k)}), u_{-i}^{(k)}).$$

Becasue $\{p_i^{(k)}\}_{k=1}^{\infty}$ is uniformly continuous at u^*, for every fixed $\epsilon > 0$, there exits $\tau > 0$ such that for all k big enough, we have

$$\|u^* - u^{(k)}\| \leq \tau, \quad \|p_i^{(k)}(u^*) - p_i^{(k)}(u^{(k)})\| < \epsilon.$$

Since f_i is rational and the denominator is positive on X, we have

$$f_i(p_i^{(k)}(u^*), u^*_{-i}) - f_i(p_i^{(k)}(u^{(k)}), u_{-i}^{(k)}) \to 0 \quad \text{as} \quad k \to \infty.$$

In view of the inequality (6.16), we can conclude that $\nu_i(u^*_{-i}) - f_i(u^*) \geq 0$. This shows that u^* is a GNE. □

6.3 Feasible extensions of KKT points

In this section, we discuss the existence and computation of feasible extensions p_i required as in Assumption 6.3. They are important for solving GNEPs. The feasible extensions have explicit expressions for box, simplex and annular constraints. Such expressions were introduced in Section 5.3. Here we give a sufficient condition for the existence of feasible extensions in more general cases.

Theorem 6.8 ([90]). *Assume $\mathcal{K}$ is a finite set. Then, for every triple (u, i, v_i) with $u \in \mathcal{K}$, $i \in [N]$ and $v_i \in X_i(u_{-i})$, there must exist a feasible extension p_i satisfying Assumption 6.3. Moreover, such p_i can be chosen as a polynomial vector function.*

Proof. Since the set $\mathcal{K}$ is finite, by polynomial interpolation, there must exist a real polynomial vector function p_i such that

$$p_i(u) = v_i, \quad p_i(z) = z_i \quad \text{for all} \quad z := (z_1, \ldots, z_N) \in \mathcal{K} \setminus \{u\}.$$

Note that $\mathcal{K} \subseteq X$. For every $x = (x_1, \ldots, x_N) \in \mathcal{K} \setminus \{u\}$, we have $p_i(x) = x_i \in X_i(x_{-i})$. The polynomial function p_i satisfies Assumption 6.3. $\square$

When the set $\mathcal{K}$ is known, we can get a polynomial feasible extension p_i as in Theorem 6.8, by polynomial interpolation. The following is such an example.

Example 6.9. Consider Example 6.2. There are four KKT points:

$$\begin{array}{ll} u_1^{(1)} = u_2^{(1)} = (0,0), & u_1^{(2)} = u_2^{(2)} = \left(\frac{\sqrt{17}-3}{4}, \frac{5-\sqrt{17}}{4}\right), \\ u_1^{(3)} = u_2^{(3)} = \left(\frac{1}{2}, 0\right), & u_1^{(4)} = u_2^{(4)} = \left(0, \frac{1}{2}\right). \end{array}$$

The $u^{(1)} = (u_1^{(1)}, u_2^{(1)})$ and $u^{(2)} = (u_1^{(2)}, u_2^{(2)})$ are not GNEs. For $u^{(1)}$, there are two minimizers for $\mathrm{F}_1(u_2^{(1)})$, which are $(1,0)$ and $(0,1)$. The feasible extension p_1 of $(1,0)$ at $u^{(1)}$ is $(1 - x_{1,1} - x_{1,2} - x_{2,2}, x_{2,2})$, and the feasible extension p_1 of $(0,1)$ at $u^{(1)}$ is $(x_{1,1}, 1 - x_{2,1} - x_{2,2} - x_{1,1})$. At $u^{(2)}$, the minimizer of $\mathrm{F}_1(u_2^{(2)})$ is $\left(0, \frac{1}{2}\right)$, and the feasible extension p_1 is

$$\Big(x_{2,1}(x_{2,1} - \frac{\sqrt{17}-3}{4})(x_{2,1} + \frac{3+\sqrt{17}}{2(5-\sqrt{17})}),\ \frac{1}{2} - (x_{2,2} - \frac{1}{2})(x_{2,2} - \frac{5-\sqrt{17}}{4})(x_{2,2} + \frac{4}{5-\sqrt{17}})\Big).$$

When the set $\mathcal{K}$ is not finite, Assumption 6.3 may still hold for some GNEPs. For instance, consider that there are no equality constraints, i.e., $\mathcal{I}_0^{(i)} = \emptyset$. Suppose $\mathcal{K}$ is compact and there exists a continuous map $\rho : \mathbb{R}^n \to \mathbb{R}^{n_i}$ such that $\rho(u) = v_i$ and $g_{i,j}(\rho(x), x_{-i}) > 0$ for all $x \in \mathcal{K}$ and for all $j \in \mathcal{I}_1^{(i)} \cup \mathcal{I}_2^{(i)}$. For every $\epsilon > 0$, one can approximate ρ by a polynomial p_i such that $\|p_i - \rho\| < \epsilon$ on $\mathcal{K}$. Therefore, for ϵ sufficiently small, $g_{i,j}(p_i(x), x_{-i}) > 0$ on $x \in \mathcal{K}$. Such polynomial function p_i is a feasible extension of v_i at u.

We discuss how to compute the rational feasible extension p_i satisfying Assumption 6.3. For the set $\mathcal{K}$ as in (6.6), let E_0 denote the set of its equality constraining polynomials and let E_1 denote the set of its (both weak and strict) inequality ones. Consider the set

$$\mathcal{K}_1 := \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{l} g(x) = 0\,(g \in E_0), \\ g(x) \geq 0\,(g \in E_1) \end{array} \right\}.$$

The set $\mathcal{K}$ may not be closed but $\mathcal{K}_1$ is, and the closure of $\mathcal{K}$ is contained in $\mathcal{K}_1$. For a polynomial $p(x)$, if $p(x) \in X_i(x_{-i})$ for all $x \in \mathcal{K}_1$, then we also have $p(x) \in X_i(x_{-i})$ for all $\mathcal{K}$. Therefore, it is sufficient to get p_i satisfying Assumption 6.3 with $\mathcal{K}$ replaced by $\mathcal{K}_1$. In such cases, we may solve feasible extensions similarly as in (5.21).

Example 6.10. Consider the following 2-player GNEP:

$$
\begin{array}{ll|ll}
\min\limits_{x_1\in\mathbb{R}^2} & \frac{(x_{2,1}+x_{2,2}-2x_{1,1})(x_{1,1})^2+2x_{1,2}}{x_{2,1}} & \min\limits_{x_2\in\mathbb{R}^2} & \frac{x_{2,1}-(x_{2,2})^2}{x_{2,2}+x_{1,1}+x_{1,2}} \\
s.t. & 2x_{1,1}x_{2,1}-x_{1,2}x_{2,2}\geq 0, & s.t. & 2x_{2,1}x_{2,2}-1\geq 0, \\
 & x_{2,1}x_{2,2}-x_{1,1}x_{2,1}\geq 0, & & 1-x_{2,2}\geq 0, \\
 & 2x_{1,2}x_{2,2}-1\geq 0, & & 2-x_{2,1}\geq 0, \\
 & 2-x_{1,2}x_{2,2}\geq 0; & & x_{2,1}\geq 0.
\end{array}
\tag{6.17}
$$

Consider the triple $(u, 1, v_1)$ for $u = (u_1, u_2)$ with

$$u_1 = (0.5, 0.5), \quad u_2 = (0.5, 1), \quad v_1 = (1, 0.5).$$

A feasible p can be chosen in form of $p_i = q/h$, where q is a tuple of polynomials and h is a given scalar polynomial. Let $h = x_{2,1}x_{2,2}$, then a feasible q can be $(x_{2,2}, x_{2,1})/2$. Let $p_1 = \frac{1}{2x_{2,1}x_{2,2}}(x_{2,2}, x_{2,1})$. Then we have

$$
\begin{aligned}
& h\cdot g_{1,1}(p_1, x_2) = 0.25 + 0.25(2x_{2,1}x_{2,2}-1), \\
& h\cdot g_{1,2}(p_1, x_2) = (x_{2,1}x_{2,2}-0.5)^2 + 0.25(2x_{2,1}x_{2,2}-1), \\
& h\cdot g_{1,3}(p_1, x_2) = 0, \quad h\cdot g_{1,4}(p_1, x_2) = 0.75 + 0.75(2x_{2,1}x_{2,2}-1).
\end{aligned}
$$

In the above, each polynomial is nonnegative in the associated $\mathcal{K}_1$.

6.4 Rational optimization problems

This section discusses how to solve the rational optimization problems appearing in Algorithm 6.4. A general rational polynomial optimization problem is

$$
\left\{
\begin{array}{ll}
\min & A(x) := \frac{a_1(x)}{a_2(x)} \\
s.t. & x \in K,
\end{array}
\right.
\tag{6.18}
$$

where $a_1, a_2 \in \mathbb{R}[x]$ and $K \subseteq \mathbb{R}^n$ is a semialgebraic set. We assume the denominator $a_2(x) > 0$ on K, otherwise one can minimize $A(x)$ over two subsets $K \cap \{a_2(x) > 0\}$ and $K \cap \{-a_2(x) > 0\}$ separately. Moment-SOS relaxations can be applied to solve (6.18).

The rational optimization problems in Algorithm 6.4 may have strict inequalities. So we consider the case that K is given as

$$
K = \left\{ x \in \mathbb{R}^n \;\middle|\; \begin{array}{c} p(x) = 0\,(p \in \Psi_0), \\ q(x) \geq 0\,(q \in \Psi_1), \\ q(x) > 0\,(q \in \Psi_2) \end{array} \right\},
\tag{6.19}
$$

where Ψ_0, Ψ_1 and Ψ_2 are finite sets of constraining polynomials in x. Since $a_2(x) > 0$ on K, we have $A(x) \geq \gamma$ on K if and only if $a_1(x) - \gamma a_2(x) \geq 0$ on K, or equivalently, $a_1 - \gamma a_2 \in \mathscr{P}_d(K)$, for the degree

$$d \coloneqq \max\{\deg(a_1), \deg(a_2)\}.$$

The rational optimization (6.18) is then equivalent to

$$\begin{cases} \gamma^* := \max & \gamma \\ \qquad\quad s.t. & a_1(x) - \gamma a_2(x) \in \mathscr{P}_d(K). \end{cases} \tag{6.20}$$

Denote the weak inequality set

$$K_1 := \left\{ x \in \mathbb{R}^n \,\middle|\, \begin{array}{c} p(x) = 0\,(p \in \Psi_0), \\ q(x) \geq 0\,(q \in \Psi_1 \cup \Psi_2) \end{array} \right\}. \tag{6.21}$$

Note that K_1 is closed and $\mathrm{cl}(K) \subseteq K_1$. We consider the moment optimization problem

$$\begin{cases} \min & \langle a_1, w \rangle \\ s.t. & \langle a_2, w \rangle = 1,\, w \in \mathscr{R}_d(K_1). \end{cases} \tag{6.22}$$

It is a moment reformulation for the optimization

$$\begin{cases} a^* \coloneqq \min & A(x) \\ \qquad\quad s.t. & x \in K_1. \end{cases} \tag{6.23}$$

Note that (6.23) is a relaxation of (6.18). It is worthy to observe that if a minimizer of (6.23) lies in the set K, then it is also a minimizer of (6.18).

We apply Moment-SOS relaxations to solve (6.22). Let

$$d_0 \coloneqq \max\left\{\lceil d/2 \rceil,\ \lceil \deg(g)/2 \rceil\ (g \in \Psi_0 \cup \Psi_1 \cup \Psi_2)\right\}. \tag{6.24}$$

For an integer $k \geq d_0$, the kth order SOS relaxation for (6.20) is

$$\begin{cases} \gamma^{(k)} \coloneqq \max & \gamma \\ \qquad\quad s.t. & a_1 - \gamma a_2 \in \mathrm{Ideal}[\Psi_0]_{2k} + \mathrm{Qmod}[\Psi_1 \cup \Psi_2]_{2k}. \end{cases} \tag{6.25}$$

The dual optimization of (6.25) is the kth order moment relaxation

$$\begin{cases} a^{(k)} \coloneqq \min & \langle a_1, y \rangle \\ \qquad\quad s.t. & L_p^{(k)}[y]^{(k)}[y] = 0\,(p \in \Psi_0),\ L_q^{(k)}[y] \succeq 0\,(q \in \Psi_1 \cup \Psi_2), \\ & \langle a_2, y \rangle = 1,\ M_k[y] \succeq 0,\ y \in \mathbb{R}^{\mathbb{N}^n_{2k}}. \end{cases} \tag{6.26}$$

Since (6.26) is a relaxation of (6.22), if (6.26) is infeasible, then (6.18) is also infeasible.

The following is the Moment-SOS algorithm for solving (6.18). It can be conveniently implemented with the software `GloptPoly3` [48].

Algorithm 6.11. For the rational optimization (6.18), let $k := d_0$.

Step 1 Solve the kth order moment relaxation (6.26). If it is infeasible, then (6.18) has no feasible points and stop. Otherwise, solve it for the optimal value $a^{(k)}$ and a minimizer y^*, if they exist. Let $t := d_0$ and go to Step 2.

Step 2 Check whether or not there is an order $t \in [d_0, k]$ such that

$$r := \operatorname{rank} M_t[y^*] \; = \; \operatorname{rank} M_{t-d_0}[y^*]. \tag{6.27}$$

Step 3 If (6.27) fails, let $k := k+1$ and go to Step 1; if (6.27) holds, find points $z_1, \ldots, z_r \in K_1$ and scalars $\mu_1, \ldots, \mu_r > 0$ such that

$$y^*|_{2t} = \mu_1 [z_1]_{2t} + \cdots + \mu_r [z_r]_{2t}. \tag{6.28}$$

Step 4 Output each $z_i \in K$ with $a_2(z_i) > 0$ as a minimizer of (6.18).

Theorem 6.12 ([90]). *Assume $a_2 \geq 0$ on K_1. Suppose y^* is a minimizer of (6.26) and it satisfies (6.27) for some order $t \in [d_0, k]$. Then, each z_i in (6.28), such that $a_2(z_i) > 0$ and $z_i \in K$, is a minimizer of (6.18).*

Proof. Under the rank condition (6.27), the decomposition (6.28) holds for some points $z_1, \ldots, z_r \in K_1$ (see [47, 79]). The constraint $\langle a_2, y^* \rangle = 1$ implies that

$$1 = \langle a_2, y^* \rangle = \mu_1 a_2(z_1) + \cdots + \mu_r a_2(z_r).$$

Since $a_2 \geq 0$ on K_1, we know all $a_2(z_j) \geq 0$. Let $J_1 := \{j : a_2(z_j) > 0\}$ and $J_2 := \{j : a_2(z_j) = 0\}$, then

$$\langle a_1, y^* \rangle = \sum_{j \in J_1} \mu_j a_2(z_j) A(z_j) + \sum_{j \in J_2} \mu_j a_1(z_j).$$

Note that $\sum_{j \in J_1} \mu_j a_2(z_j) = 1$ and each $[z_j]_{2k} \in \mathscr{R}_{2k}(K_1)$. For all nonnegative scalars $\nu_j \geq 0$, $j \in J_1 \cup J_2$ such that $\sum_{j \in J_1} \nu_j a_2(z_j) = 1$, the tms

$$z(\nu) := \nu_1 [z_1]_{2k} + \cdots + \nu_r [z_r]_{2k}$$

is a feasible point for the moment relaxation (6.26). Therefore, the optimality of y^* implies that $A(z_j) = a^{(k)}$ for all $j \in J_1$. Since $a^{(k)} \le a^*$ and each $z_j \in K_1$, we have $A(z_j) \ge a^*$. Hence, $A(z_j) = a^*$ for all $j \in J_1$. Note that (6.22) is a relaxation of (6.23). So each z_j ($j \in J_1$) is a minimizer of (6.23). Therefore, every $z_i \in K$ satisfying $a_2(z_i) > 0$ is a minimizer of (6.18). $\square$

The rational optimization problem in Step 2 of Algorithm 6.4 is

$$\begin{cases} \min & \theta(x) := [x]_1^T \Theta [x]_1 \\ s.t. & x \in \mathscr{U}, \end{cases} \tag{6.29}$$

where Θ is a generic positive definite matrix. The feasible set $\mathscr{U}$ can be expressed as in the form (6.19), with polynomial equalities and weak/strict inequalities, for some polynomial sets Ψ_0, Ψ_1, Ψ_2. That is, (6.29) can be expressed in the form of (6.18), with denominators being 1. Denote the corresponding set

$$\mathscr{U}_1 = \{x \in \mathbb{R}^n \,|\, p(x) = 0\,(p \in \Psi_0),\, q(x) \ge 0\,(q \in \Psi_1 \cup \Psi_2)\}. \tag{6.30}$$

Since Θ is positive definite, the objective θ is coercive and strictly convex. When Θ is also generic, the function θ has a unique minimizer u^* on the set $\mathscr{U}_1$ if it is nonempty. Suppose y^* is a minimizer of the kth order moment relaxation of (6.29). Then, in Algorithm 6.11, the rank condition (6.27) is reduced to

$$\operatorname{rank} M_t[y^*] = 1$$

for some order $t \in [d_0, k]$ and the decomposition (6.28) is equivalent to $y^*|_{2t} = \mu_1 [z_1]_{2t}$ for some $z_1 \in \mathscr{U}_1$. Algorithm 6.11 can be applied to solve (6.29). The following are some special properties of Moment-SOS relaxations for (6.29).

Theorem 6.13 ([90]). *Assume Θ is a generic positive definite matrix.*

i) If the set $\mathscr{U}_1$ is empty and $Ideal[\Psi_0] + Qmod[\Psi_1 \cup \Psi_2]$ is archimedean, then the moment relaxation for (6.29) must be infeasible when the order k is big enough.

ii) Suppose $\mathscr{U}_1 \neq \emptyset$ and $Ideal[\Psi_0] + Qmod[\Psi_1 \cup \Psi_2]$ is archimedean. Denote $u^{(k)} := (y^{(k)}_{e_1}, \ldots, y^{(k)}_{e_n})$, where $y^{(k)}$ is the minimizer of the kth order moment relaxation of (6.29). Then $u^{(k)}$ converges to the unique minimizer of θ on $\mathscr{U}_1$.

iii) Suppose the real zero set of Ψ_0 *is finite. If* $\mathscr{U}_1 \neq \emptyset$*, then we must have* $\operatorname{rank} M_t[y^*] = 1$ *for some* $t \in [d_0, k]$*, when* k *is sufficiently large.*

Proof. i) When $\mathscr{U}_1 = \emptyset$, the constant -1 can be viewed as a positive polynomial on $\mathscr{U}_1$. Since $\operatorname{Ideal}[\Psi_0] + \operatorname{Qmod}[\Psi_1 \cup \Psi_2]$ is archimedean, we have $-1 \in \operatorname{Ideal}[\Psi_0]_{2k} + \operatorname{Qmod}[\Psi_1 \cup \Psi_2]_{2k}$ for k big enough, by Putinar's Positivstellensatz. For such k, the corresponding SOS relaxation (6.25) is unbounded from above, and hence the corresponding moment relaxation must be infeasible.

ii) When $\mathscr{U}_1 \neq \emptyset$, the objective θ has a unique minimizer u^* on $\mathscr{U}_1$. The convergence of $u^{(k)}$ is implied by [79, Theorem 3.3] (also see [101]).

iii) When the real zero set of Ψ_0 is finite and $\mathscr{U}_1 \neq \emptyset$, the conclusion can be implied by [63, Proposition 4.6] (also see [65]). □

Once we get a minimizer u of (6.29), we need to check if it is a GNE or not. For each $i = 1, \ldots, N$, we need to solve the rational optimization problem

$$\begin{cases} \delta_i := \min & f_i(x_i, u_{-i}) - f_i(u_i, u_{-i}) \\ \quad\quad\; s.t. & x_i \in X_i(u_{-i}), \end{cases} \tag{6.31}$$

where $f_i, X_i(u_{-i})$ are given in (6.1). Assume the KKT conditions hold and the Lagrange multiplies can be expressed as in (6.5). Therefore, (6.31) is equivalent to

$$\begin{cases} \min & f_i(x_i, u_{-i}) - f_i(u_i, u_{-i}) \\ s.t. & \nabla_{x_i} f_i(x_i, u_{-i}) = \sum\limits_{j \in \mathcal{I}_0^{(i)} \cup \mathcal{I}_1^{(i)}} \tau_{i,j}(x_i, u_{-i}) \nabla_{x_i} g_{i,j}(x_i, u_{-i}), \\ & \tau_{i,j}(x_i, u_{-i}) g_{i,j}(x_i, u_{-i}) = 0,\ \tau_{i,j}(x_i, u_{-i}) \geq 0,\ (j \in \mathcal{I}_1^{(i)}), \\ & x_i \in X_i(u_{-i}). \end{cases} \tag{6.32}$$

We can equivalently express the feasible set of (6.32) in the form

$$Y_i(u_{-i}) = \left\{ x_i \in \mathbb{R}^{n_i} \left| \begin{array}{l} p(x_i) = 0\,(p \in \Psi_{i,0}), \\ q(x_i) \geq 0\,(q \in \Psi_{i,1}), \\ q(x_i) > 0\,(q \in \Psi_{i,2}) \end{array} \right. \right\}, \tag{6.33}$$

for three sets $\Psi_{i,0}, \Psi_{i,1}, \Psi_{i,2}$ of polynomials in x_i. We can apply a similar version of Algorithm 6.11 to solve the rational optimization problem (6.32). Similar conclusions like in Theorem 6.13 hold for the corresponding Moment-SOS relaxations. A difference is that all rational functions for (6.31) are only in the variable x_i instead of x. It may have several different minimizers, so the rank in (6.27) may be bigger than one.

6.5 Numerical experiments

This section gives numerical experiments for Algorithm 6.4 to solve GNEPs. The rational optimization problems are solved by Moment-SOS relaxations, which are implemented with the software `GloptiPoly3` [48]. The semidefinite programs for the Moment-SOS relaxations are solved by `SeDuMi` [105]. The computation is implemented in MATLAB R2018a, in a Laptop with CPU 8th Generation Intel® Core™ i5-8250U and RAM 16 GB. For neatness of the paper, only four decimal digits are displayed for computational results. The accuracy for a point u to be a GNE is measured by the quantity

$$\delta := \min\{\delta_1, \dots, \delta_N\},$$

where δ_i is the optimal value of (6.11). The point u is a GNE if and only if $\delta = 0$. Due to numerical issues, u can be viewed as a GNE if δ is nearly zero (e.g., $\delta \geq -10^{-6}$). For cleanness of presentation, we do not list the constraining functions $g_{i,j}$ explicitly. Instead, they are ordered row by row, from top to bottom; in each row, they are ordered from left to right. If there is an inequality like $a(x) \leq b(x)$, then the corresponding constraining function is $b(x) - a(x)$.

Example 6.14. (i) Consider the GNEP in Example 6.1. Algorithm 6.4 terminated at the initial loop $k = 0$. The computed GNE is $u = (u_1, u_2)$ with

$$u_1 = (1.3561, 0.7374), \quad u_2 = (1.0000, 1.0468), \quad \delta = -3.44 \cdot 10^{-8}.$$

It took around 8.36 seconds.
(ii) For the GNEP in Example 6.1, if objective functions are changed to

$$f_1(x) = \frac{(x_{1,2})^2 + x_{1,1}x_{1,2}(e^T x_2)}{x_{1,1}}, \quad f_2(x) = \frac{(x_{2,2})^2 - x_{2,1}x_{2,2}(e^T x_1)}{x_{2,1}},$$

then there is no GNE. This is detected by Algorithm 6.4 at the initial loop $k = 0$. It took about 5.47 seconds.
(iii) Consider the GNEP in Example 6.2. By Algorithm 6.4, we got the GNE $u = (u_1, u_2)$ at the loop $k = 1$ with

$$u_1 = (0.0000, 0.5000), \quad u_2 = (0.0000, 0.5000), \quad \delta = -4.47 \cdot 10^{-8}.$$

It took around 3.28 seconds.

(iv) Consider the GNEP in Example 6.10. Algorithm 6.4 terminated at the loop $k = 1$. We got the GNE $u = (u_1, u_2)$ with

$$u_1 = (1.0000, 0.5000), \quad u_2 = (0.5000, 1.0000), \quad \delta = -1.82 \cdot 10^{-8}.$$

It took around 22.73 seconds.

Example 6.15. Consider the 2-player GNEP with the optimization

$$\left.\begin{array}{cl} \min\limits_{x_1 \in \mathbb{R}^3} & x_1^T(x_1 + x_2) + x_{1,1} - x_{1,2} - x_{1,3} \\ s.t. & x_{1,1}x_{1,2}x_{1,3} \le 1 + (e^T x_2)^2, \end{array}\right| \begin{array}{cl} \min\limits_{x_2 \in \mathbb{R}^3} & e^T x_2 + \sum_{j=1}^3 x_{1,j}(x_{2,j})^2 \\ s.t. & (e^T x_1)^2 - x_2^T x_2 \ge 0. \end{array}$$

For the first player's optimization, we have the LME and the feasible extension

$$\lambda_1 = -\frac{x_1^T \nabla_{x_1} f_1}{3 + 3(e^T x_2)^2}, \quad p_1(x) = \left(v_{1,1}, v_{1,2}, \frac{1 + (e^T x_2)^2}{1 + (e^T u_2)^2} \cdot v_{1,3} \right).$$

For the second player, we have the LMEs and feasible extensions for the annular constraints. Algorithm 6.4 terminated at the loop $k = 0$. We got the GNE $u = (u_1, u_2)$,

$$u_1 = (0.3090, 0.8090, 0.8090), \quad u_2 = (-1.6180, -0.6180, -0.6180),$$

with the accuracy parameter $\delta = -2.77 \cdot 10^{-8}$. It took around 5.16 seconds.

Example 6.16. Consider the 3-player GNEP

$$\begin{array}{ll}
\mathrm{F}_1(x_2, x_3) : & \left\{ \begin{array}{cl} \min\limits_{x_1 \in \mathbb{R}^2} & \|x_1 - \frac{1}{2}(x_2 + x_3)\|^2 \\ s.t. & x_{1,1}x_{1,2} = 1 + x_3^T x_3, \; x_{1,1} \ge 0, \; x_{1,2} \ge 0, \end{array} \right. \\
\mathrm{F}_2(x_1, x_3) : & \left\{ \begin{array}{cl} \min\limits_{x_2 \in \mathbb{R}^2} & x_2^T(x_1 + x_3) + (x_{2,1})^3 - 3(x_{2,2})^2 \\ s.t. & \|x_{1,1} \cdot x_2\|^2 = (x_{1,2})^2, \end{array} \right. \\
\mathrm{F}_3(x_1, x_2) : & \left\{ \begin{array}{cl} \min\limits_{x_3 \in \mathbb{R}^2} & x_3^T(x_1 + x_2 + x_3 - e) \\ s.t. & e^T x_3 \le x_1^T x_1, \;\; x_{3,1} \ge 0.1, \;\; x_{3,2} \ge 0.1. \end{array} \right.
\end{array}$$

The LMEs for $\mathrm{F}_1(x_2, x_3)$ and $\mathrm{F}_2(x_1, x_3)$ are

$$\begin{array}{ll} \lambda_{1,1} = \frac{x_1^T \nabla_{x_1} f_1}{2 + 2x_3^T x_3}, & \lambda_{1,2} = \frac{\partial f_1}{\partial x_{1,1}} - x_{1,2}\lambda_{1,1}, \\ \lambda_{1,3} = \frac{\partial f_1}{\partial x_{1,2}} - x_{1,1}\lambda_{1,1}, & \lambda_2 = \frac{-x_2^T \nabla_{x_2} f_2}{2(x_{1,2})^2}. \end{array}$$

We use the LME of simplex constraints for $\mathrm{F}_3(x_1, x_2)$. The first two players have the feasible extension

$$p_1(x) = \left(v_{1,1}, \frac{1 + x_3^T x_3}{v_{1,1}} \right), \quad p_2(x) = \frac{u_{1,1}x_{1,2}}{u_{1,2}x_{1,1}} \cdot (v_{2,1}, v_{2,2}).$$

For the third player, the feasible extension is given for the simplex constraints. Algorithm 6.4 terminated at the initial loop $k = 0$. We got the GNE $u = (u_1, u_2, u_3)$ with

$$u_1 = (1.1401, 1.0461), \quad u_2 = (-0.1743, -0.9009), \quad u_3 = (0.1000, 0.4274)$$

and $\delta = -6.19 \cdot 10^{-8}$. It took around 10.58 seconds.

It is interesting to note that if the third player's objective is changed to

$$x_3^T(x_1 + x_2 - e) + x_{3,1}^2 - x_{3,2}^2$$

then there is no GNE. This is detected by Algorithm 6.4 at the loop $k = 1$. It took around 19.16 seconds.

We remark that Algorithm 6.4 can be generalized to compute more (or even all) GNEs. This can be done with the approach in [87]. Suppose a GNE u is already known. Select a small scalar $\zeta > 0$ and solve the maximization problem

$$\left\{ \begin{array}{rl} \rho := \max & [x]_1^T \Theta [x]_1 \\ s.t. & x \in \mathscr{U},\ [x]_1^T \Theta [x]_1 \le [u]_1^T \Theta [u]_1 + \zeta. \end{array} \right. \tag{6.34}$$

If $\rho > [u]_1^T \Theta [u]_1$, then let $\zeta := \zeta/2$ and solve (6.34) again. Repeat this until ζ is small enough to make $\rho = [u]_1^T \Theta [u]_1$. When u is an isolated KKT point and Θ is generic positive definite, such ζ always exists. This can be proved similarly to that in [87]. Once such ζ is found, we add the new inequality $[x]_1^T \Theta [x]_1 \ge [u]_1^T \Theta [u]_1 + \zeta$ to (6.10). Then Algorithm 6.4 can be applied to get a new GNE, if it exists. It is worthy to note that if the optimization (6.10) is infeasible with the newly added constraints, then there are no other GNEs. By repeating this process, we can get all GNEs if there are finitely many ones. We refer to [87] for more details. The following is an example for getting more GNEs.

Example 6.17. Consider the 2-player GNEP

$$\left. \begin{array}{cl} \min\limits_{x_1 \in \mathbb{R}^2} & \frac{\sum_{j=1}^2 (x_{1,j})^2 x_{2,j} + x_{1,1}x_{1,2}}{(x_{1,1})^2 + 1} \\ s.t. & (1 - e^T x_2)^2 \le \|x_1\|^2 \le 1, \end{array} \right| \begin{array}{cl} \min\limits_{x_2 \in \mathbb{R}^2} & \frac{\sum_{j=1}^2 (x_{2,j})^2 x_{1,j} + x_{2,1}x_{2,2}}{(x_{2,1})^2 + 1} \\ s.t. & (1 - e^T x_1)^2 \le \|x_2\|^2 \le 1. \end{array}$$

We use the LMEs and the feasible extensions of annular constraints for both players. Following the above process, we got two GNEs $u = (u_1, u_2)$ with

$$\begin{array}{llll} u_1 = (0.9250, -0.3799), & u_2 = (0.9250, -0.3799), & \delta = -9.06 \cdot 10^{-8}, & \text{and} \\ u_1 = (-0.2700, 0.9629), & u_2 = (-0.2700, 0.9629), & \delta = -2.67 \cdot 10^{-7}. & \end{array}$$

It took around 29.80 seconds to get both of them. Since each rational LME has a positive denominator on X, we computed all GNEs for this problem.

Acknowledgments. This Chapter, in full, has been submitted for publication. The dissertation author coauthored this paper with Nie, Jiawang and Tang, Xindong.

Chapter 7

Loss Functions for Finite Sets

7.1 Loss functions for finite sets

Let n, k be positive integers. Suppose S is a set of k distinct points in $\mathbb{R}^n$. A function f in $x := (x_1, \ldots, x_n)$ is said to be a loss function for S if the global minimizers of f are precisely the points in S. For convenience, we often select f such that f is nonnegative in $\mathbb{R}^n$ and the minimum value is zero. Mathematically, this is equivalent to that $f \geq 0$ on $\mathbb{R}^n$ and

$$f(x) = 0 \quad \text{if and only if} \quad x \in S. \tag{7.1}$$

When $S = \{u_1, \ldots, u_k\}$, a straightforward choice for the loss function is

$$f = \|x - u_1\|^2 \cdots \|x - u_k\|^2,$$

where $\| \cdot \|$ is the standard Euclidean norm. This loss function is a polynomial of degree $2k$ in the variable x. It requires to use all points of S. In applications, the cardinality k may be big. Moreover, the set S often has noises and it may be given by a large number of samplings around the points in S. For this reason, the above choice of loss function may not be convenient in computational practice.

A frequently used loss function is the class of sum-of-squares polynomials. That is, the loss function f is in the form

$$f = p_1^2 + \cdots + p_m^2,$$

where each p_i is a polynomial in x. Then f is a loss function for S if and only if each $p_i \equiv 0$ on S. For convenience of computation, we prefer that f and each p_i have degrees as low as possible. More preferable is that every local minimizer of f is a global minimizer (i.e., a zero of f). That is, we wish that the loss function f has no *spurious minimizer*. To be precise, a local minimizer that is not a global minimizer is called a spurious minimizer.

In applications, the set S may not be given explicitly. It is often approximately given by a sample set

$$T = \{v_1, \ldots, v_N\},$$

where each v_i is a sample for a point in S and the sample size $N \gg k$. For such a case, we can choose a family $\mathcal{F}$ of loss functions for S, parameterized by some parameters. Since S is approximated by T, we choose a loss function $f \in \mathcal{F}$ such that the average value of f on T is minimum. Mathematically, this is equivalent to solving the optimization

$$\min_{f \in \mathcal{F}} \quad \frac{1}{N} \sum_{i=1}^{N} f(v_i). \tag{7.2}$$

The optimization (7.2) requires that we choose parameters for f such that the average loss on T is minimum. The set S can be determined by parameters for f in the family $\mathcal{F}$.

7.2 A class of loss functions

In this section, we give a general framework of constructing loss functions for finite sets. For convenience, we assume the finite sets are real.

Suppose $S \subseteq \mathbb{R}^n$ is a finite set of cardinality k, say,

$$S \;=\; \{u_1, \ldots, u_k\}.$$

We consider the SOS loss functions

$$f \;=\; p_1^2 + \cdots + p_m^2, \tag{7.3}$$

where each p_i is a polynomial in x. Denote the tuple

$$p \;=\; (p_1, \ldots, p_m).$$

Without loss of generality, one can assume that the minimum value of f is zero, up to shifting of a constant. Note that $f(x) = 0$ if and only if $p(x) = 0$. Therefore, f is a loss function for S if and only if

$$S = \{x \in \mathbb{R}^n : \, p_1(x) = \cdots = p_m(x) = 0\}. \tag{7.4}$$

The above observation gives the following lemma.

Lemma 7.1 ([95]). *Let S, f be as above. Then f is a loss function for S if and only if S is the real zero set of p, i.e., $S = V_{\mathbb{R}}(p)$.*

Then we show how to choose a computationally efficient loss function for S. Let $\mathbb{B}_0$ be the set of first k vectors in the nonnegative power set $\mathbb{N}^n$, in the graded lexicographic ordering, i.e.,

$$\mathbb{B}_0 \coloneqq \Big\{ \underbrace{0,\, e_1,\, \ldots,\, e_n,\, 2e_1,\, e_1 + e_2, \ldots,}_{\text{first } k \text{ of them}} \Big\}. \tag{7.5}$$

Then, we consider the set

$$\mathbb{B}_1 \coloneqq \Big((e_1 + \mathbb{B}_0) \cup \cdots \cup (e_n + \mathbb{B}_0) \Big) \setminus \mathbb{B}_0. \tag{7.6}$$

For convenience of notation, denote the monomial vectors

$$[x]_{\mathbb{B}_0} \coloneqq \big(x^\alpha\big)_{\alpha \in \mathbb{B}_0}, \quad [x]_{\mathbb{B}_1} \coloneqq \big(x^\alpha\big)_{\alpha \in \mathbb{B}_1}.$$

Since S is a finite set of cardinality k, we wish to select $\mathbb{B}_0$ so that the set of equivalent classes of monomials $x^\alpha \, (\alpha \in \mathbb{B}_0)$ is a basis for the quotient space $\mathbb{R}[x]/I(S)$, where $I(S)$ is the vanishing ideal of S. This requires that x^α $(\alpha \in \mathbb{B}_1)$ is a linear combination of monomials $x^\beta \, (\beta \in \mathbb{B}_0)$, modulo $I(S)$. Equivalently, there exist scalars $G(\beta, \alpha)$ such that

$$\varphi[G, \alpha](x) \coloneqq x^\alpha - \sum_{\beta \in \mathbb{B}_0} G(\beta, \alpha) x^\beta \equiv 0 \mod I(S) \tag{7.7}$$

for each $\alpha \in \mathbb{B}_1$. Let $G \coloneqq (G(\beta, \alpha)) \in \mathbb{R}^{\mathbb{B}_0 \times \mathbb{B}_1}$ be the matrix of all such scalars $G(\beta, \alpha)$. The polynomial $\varphi[G, \alpha]$ has coefficients that are linear in entries of G. For convenience, denote that

$$\boxed{\begin{array}{rcl} \varphi[G] & = & \big(\varphi[G, \alpha]\big)_{\alpha \in \mathbb{B}_1}, \\ X_0 & = & \big[[u_1]_{\mathbb{B}_0} \quad \cdots \quad [u_k]_{\mathbb{B}_0} \big], \\ X_1 & = & \big[[u_1]_{\mathbb{B}_1} \quad \cdots \quad [u_k]_{\mathbb{B}_1} \big]. \end{array}} \tag{7.8}$$

The X_0 is a square matrix, which is nonsingular if the points in S are in generic positions. For $\varphi[G]$ to vanish on S, the equation (7.7) implies that

$$X_1 - G^T X_0 = 0.$$

If X_0 is nonsingular, then the matrix G is given as

$$G = X_0^{-T} X_1^T. \tag{7.9}$$

We look for conditions on G such that $\varphi[G]$ has k common zeros in $\mathbb{C}^n$. For each $i = 1, \ldots, n$, define the multiplication matrix $M_{x_i}(G)$ such that

$$[M_{x_i}(G)]_{\mu,\nu} = \begin{cases} 1 & \text{if} \quad x_i \cdot x^\nu \in \mathbb{B}_0,\ \mu = \nu + e_i, \\ 0 & \text{if} \quad x_i \cdot x^\nu \in \mathbb{B}_0,\ \mu \neq \nu + e_i, \\ G(\mu, \nu + e_i) & \text{if} \quad x_i \cdot x^\nu \in \mathbb{B}_1. \end{cases} \tag{7.10}$$

The rows and columns of $M_{x_i}(G)$ are labelled by monomial powers $\mu, \nu \in \mathbb{B}_0$. The following proposition characterizes when $\varphi[G]$ has k common zeros.

Proposition 7.2. *([83, Proposition 2.4]) Let $\mathbb{B}_0$, $\mathbb{B}_1$ be as in (7.5)-(7.6). Then, the polynomial tuple $\varphi[G]$ has k common complex zeros (counting multiplicities) if and only if the multiplication matrices $M_{x_1}(G), \ldots, M_{x_n}(G)$ commute, i.e.,*

$$[M_{x_i}(G), M_{x_j}(G)] = 0 \quad (1 \le i < j \le n). \tag{7.11}$$

In particular, $\varphi[G]$ has k distinct complex zeros if and only if $M_{x_1}(G), \ldots, M_{x_n}(G)$ are simultaneously diagonalizable.

The polynomial tuple $\varphi[G]$ generates the vanishing ideal $I(S)$ of S and $p = \varphi[G]$ has minimum degrees for (7.4) to hold.

Theorem 7.3 ([95])**.** *Assume S is a finite set such that X_0 is nonsingular. Let G be as in (7.9). Then, the ideal Ideal($\varphi[G]$) equals the vanishing ideal of S, i.e.,*

$$\mathit{Ideal}(\varphi[G]) \;=\; \{h \in \mathbb{R}[x] :\, h \equiv 0 \ \mathit{on}\, S\}. \tag{7.12}$$

In particular, if a polynomial h vanishes on S identically, then there are polynomials p_α ($\alpha \in \mathbb{B}_1$) such that

$$h \;=\; \sum_{\alpha \in \mathbb{B}_1} q_\alpha \varphi[G, \alpha]), \quad \deg(q_\alpha) + |\alpha| \le \deg(h). \tag{7.13}$$

Proof. Since X_0 is nonsingular, the set S has k distinct points. Since G is given as in (7.9), the polynomial equation $\varphi[G](x) = 0$ has k distinct solutions. By Proposition 7.2, the multiplication matrices $M_{x_1}(G), \ldots, M_{x_n}(G)$ are simultaneously diagonalizable. Note that the ideal $\mathrm{Ideal}(\varphi[G])$ is zero-dimensional, because the quotient space $\mathbb{C}[x]/\mathrm{Ideal}(\varphi[G])$ has the dimension k. The ideal $\mathrm{Ideal}(\varphi[G])$ must be radical. This can be implied by Corollary 2.7 of [106]. So (7.12) holds.

Suppose h is a polynomial such that $h \equiv 0$ on S. Then the above shows that $h \in \text{Ideal}(\varphi[G])$. So there exist polynomials q_α ($\alpha \in \mathbb{B}_1$) such that

$$h = \sum_{\alpha \in \mathbb{B}_1} q_\alpha \varphi[G, \alpha].$$

The multiplication matrices $M_{x_1}(G), \ldots, M_{x_n}(G)$ commute. One can check that the set of polynomials in the tuple $\varphi[G]$ is a Gröbner basis for $\text{Ideal}(\varphi[G])$, with respect to the graded lexicographical ordering. This can also be implied by the proof of Lemma 2.8 in [83]. Therefore, we can further select polynomials $q_\alpha \in \mathbb{R}[x]$ with degree bounds as in (7.13). $\square$

The condition that X_0 is nonsingular holds when the points of S are in generic positions. The equation (7.13) shows that the polynomial tuple $\varphi[G]$ is a minimum-degree generating set for the vanishing ideal $I(S)$. The following are some examples.

Example 7.4. *Consider the set S in $\mathbb{R}^3$ such that*

$$S = \left\{ \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}, \begin{bmatrix} -1 \\ -2 \\ 4 \end{bmatrix} \right\}, \mathbb{B}_0 = \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \right\}, \mathbb{B}_1 = \left\{ \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \right\}.$$

The matrix G as in (7.9) and $\varphi[G]$ are

$$G = \begin{bmatrix} -1 & \frac{11}{3} & 2 & 2 & -\frac{2}{3} \\ 1 & -\frac{1}{3} & 1 & 0 & \frac{10}{3} \end{bmatrix}, \quad \varphi[G] = \begin{bmatrix} x_2 - x_1 + 1 \\ \frac{x_1}{3} + x_3 - \frac{11}{3} \\ x_1^2 - x_1 - 2 \\ x_1 x_2 - 2 \\ x_1 x_3 - \frac{10 x_1}{3} + \frac{2}{3} \end{bmatrix}.$$

7.3 Simplicial loss functions

For a vector $a := (a_1, \ldots, a_n)$, with each scalar $a_i \neq 0$, consider the standard simplex vertex set

$$\Delta_n(a) := \{0, a_1 e_1, \ldots, a_n e_n\}. \tag{7.14}$$

For the special case that $a = (1, \ldots, 1)$, we denote

$$\Delta_n := \{0, e_1, \ldots, e_n\}. \tag{7.15}$$

When the dimension n is clear in the context, we just write $\Delta = \Delta_n$ for convenience. We consider the special case that $S = \Delta_n(a)$. Then the monomial power sets $\mathbb{B}_0$, $\mathbb{B}_1$ are respectively

$$\mathbb{B}_0 = \{0, e_1, \dots, e_n\}, \quad \mathbb{B}_1 = \{2e_1, e_1 + e_2, \dots, 2e_n\}.$$

For the matrix $G \in \mathbb{R}^{\mathbb{B}_0 \times \mathbb{B}_1}$ given as in (7.9), we have that

$$\boxed{\begin{array}{lcll} \varphi[G, 2e_i] & = & x_i^2 - a_i x_i & (i \in [n]), \\ \varphi[G, e_i + e_j] & = & x_i x_j & (i < j). \end{array}} \tag{7.16}$$

The resulting loss function for the set $\Delta_n(a)$ is

$$f(x) = \sum_{i=1}^{n} x_i^2 (x_i - a_i)^2 + \sum_{1 \le i < j \le n} x_i^2 x_j^2. \tag{7.17}$$

In particular, the above loss function for Δ_n is

$$F(x) \coloneqq \sum_{i=1}^{n} x_i^2 (x_i - 1)^2 + \sum_{1 \le i < j \le n} x_i^2 x_j^2. \tag{7.18}$$

We call f given as in (7.17)-(7.18) the *simplicial loss function.* A nice property is that the simplicial loss function as in (7.17) has no spurious minimizers.

Theorem 7.5 ([95]). *Fix nonzero scalars $a_1, \dots, a_n$, the function f in (7.17) has no spurious minimizers, i.e., every local minimizer of f is also a global minimizer.*

Proof. Suppose $z = (z_1, \dots, z_n)$ is a local minimizer of f. Then z satisfies the optimality conditions

$$\nabla f(z) = 0, \quad \nabla^2 f(z) \succeq 0.$$

This implies that for $i = 1, \dots, n$,

$$\frac{\partial f}{\partial x_i}(z) = 2z_i\big(2z_i^2 - 3a_i z_i + (z^T z - z_i^2 + a_i^2)\big) = 0, \tag{7.19}$$

$$\frac{\partial^2 f}{\partial x_i^2}(z) = 12 z_i^2 - 12 a_i z_i + 2(z^T z - z_i^2 + a_i^2) \ge 0. \tag{7.20}$$

Denote $\delta_i(z) \coloneqq a_i^2 - 8(z^T z - z_i^2)$. The real solutions for (7.19) are $z_i = 0$ and

$$z_i = \frac{3a_i \pm \sqrt{\delta_i(z)}}{4} \quad \text{if} \quad \delta_i(z) \ge 0. \tag{7.21}$$

If each $z_i = 0$, then $z = \mathtt{0}$ is a global minimizer. Suppose some z_i is nonzero, then it satisfies $\delta_i(z) \geq 0$ and $2z_i^2 - 3a_i z_i + (z^T z - z_i^2 + a_i^2) = 0$. So (7.20) can be reformulated as

$$\frac{\partial^2 f}{\partial x_i^2}(z) = 8z_i^2 - 6a_i z_i = 2z_i(4z_i - 3a_i) \geq 0.$$

Plug (7.21) into the above inequality. Since $\sqrt{\delta_i(z)} \leq |a_i| < |3a_i|$ (note $a_i \neq 0$),

$$z_i = \begin{cases} \frac{3a_i - \sqrt{\delta_i(z)}}{4} & \text{if } a_i < 0, \\ \frac{3a_i + \sqrt{\delta_i(z)}}{4} & \text{if } a_i > 0. \end{cases}$$

It is clear that $|z_i| \geq |3a_i/4|$. If z_i is the only nonzero entry of z, then $\sqrt{\delta_i(z)} = |a_i|$ and $z = a_i e_i$, which is a global minimizer. Suppose z has another nonzero entry z_j. By a similar argument, we can get $\delta_j(z) \geq 0$ and $|z_j| \geq |3a_j/4|$. Note that $2a_i^2 - 9a_j^2 \geq 0$ since

$$a_i^2 - 8 \cdot \left|\frac{3a_j}{4}\right|^2 \geq a_i^2 - 8z_j^2 \geq \delta_i(z) \geq 0.$$

Similarly, $2a_j^2 - 9a_i^2 \geq 0$, so

$$2a_j^2 - 9a_i^2 \geq 2a_j^2 - 9 \cdot \frac{9}{2}a_j^2 = -\frac{77}{2}a_j^2 \geq 0.$$

The above holds if and only if $a_j = 0$, which contradicts that each a_i is nonzero. Therefore, every local minimizer of f is a global minimizer, i.e., f has no spurious minimizers. $\square$

When S is not a simplicial vertex set, we can still use the function F in (7.18) to get new loss functions, up to a transformation. These new functions have no spurious minimizers. They are called *transformed* simplicial loss functions. Consider that S is given as

$$S = \{u_1, \ldots, u_k\}. \tag{7.22}$$

We discuss the transformation for two different cases.

Case I: $\mathbf{k \leq n+1}$. Consider the vertex set of a standard simplex set in $\mathbb{R}^{k-1}$

$$\Delta_{k-1} = \{\mathtt{0}, e_1, \ldots, e_{k-1}\}.$$

The loss function as in (7.18) for Δ_{k-1} is

$$F_{k-1}(z) := \sum_{i=1}^{k-1} z_i^2(z_i - 1)^2 + \sum_{1 \leq i < j \leq k-1} z_i^2 z_j^2, \tag{7.23}$$

in the variable $z = (z_1, \ldots, z_{k-1})$. Consider the linear map

$$\ell : \mathbb{R}^{k-1} \to \mathbb{R}^n, \quad \ell(e_i) = u_i - u_k,\ i = 1, \ldots, k-1. \tag{7.24}$$

The representing matrix for the linear map ℓ is

$$U = \begin{bmatrix} u_1 - u_k & \cdots & u_{k-1} - u_k \end{bmatrix}. \tag{7.25}$$

When $u_1, \ldots, u_k$ are in generic positions, the matrix U has full column rank. Let

$$U^\dagger \coloneqq (U^T U)^{-1} U^T$$

be the Pseudo inverse of U. For $x = (x_1, \ldots, x_n)$, consider the loss function

$$f(x) = F_{k-1}\big(U^\dagger (x - u_k)\big). \tag{7.26}$$

Recall that $\mathrm{Null}(U^\dagger)$ denotes the null space of the matrix $U^\dagger$.

Theorem 7.6 ([95]). *Suppose $k \le n+1$ and rank $U = k-1$. Then, the function f as in (7.26) is a loss function for the set*

$$S + Null(U^\dagger) \coloneqq \{x + y :\ x \in S, U^\dagger y = 0\}.$$

Moreover, f has no spurious minimizers.

Proof. The function f as in (7.26) is nonnegative everywhere. Note that $f(x) = 0$ if and only if $U^\dagger(x - u_k) \in \Delta_{k-1}$. It holds that

$$\Delta_{k-1} \; = \; U^\dagger(S - u_k).$$

For $x \in \mathbb{R}^n$, we have $U^\dagger(x - u_k) \in \Delta_{k-1}$ if and only if $x \in S + \mathrm{Null}(U^\dagger)$. This shows that f is a loss function for $S + \mathrm{Null}(U^\dagger)$ in $\mathbb{R}^n$.

The gradient and Hessian of f can be written as

$$\nabla_x f(x) = (U^\dagger)^T \nabla_z F_{k-1}(z), \quad \nabla_x^2 f(x) = (U^\dagger)^T \nabla_z^2 F_{k-1}(z) U^\dagger.$$

Note that $U^\dagger$ has full row rank. If u is a local minimizer of f, then $\nabla_x f(u) = 0$, $\nabla_x^2 f(u) \succeq 0$. Let $z = U^\dagger(u - u_k)$, then the above implies that

$$\nabla_z F_{k-1}(z) = 0, \quad \nabla_z^2 F_{k-1}(z) \succeq 0.$$

As in the proof of Theorem 7.5, one can show that $z \in \Delta_{k-1}$. This implies that z is a global minimizer of F_{k-1} and hence u is a global minimizer of f. So f has no spurious minimizers. $\square$

Here the following is an example of transformed simplicial loss function for the case $k \le n+1$.

Example 7.7. *Consider the set* $S = \left\{ \begin{bmatrix} 4 \\ -2 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 3 \\ -5 \end{bmatrix} \right\}$ *in* $\mathbb{R}^3$*. The matrix* U *as in (7.25) and its Pseudo inverse are*

$$U = \begin{bmatrix} 5 \\ -5 \\ 6 \end{bmatrix}, \quad U^\dagger = \frac{1}{86} \begin{bmatrix} 5 \\ -5 \\ 6 \end{bmatrix}^T.$$

Since $k = 2$*, the simplicial loss function for* Δ_{k-1} *is* $F_1 = z^2(z-1)^2$ *in the univariate variable* z*. Then, the transformed simplicial loss function as in (7.26) is*

$$f(x) = \left(\frac{5x_1}{86} - \frac{5x_2}{86} + \frac{3x_3}{43} + \frac{25}{43} \right)^2 \cdot \left(\frac{5x_1}{86} - \frac{5x_2}{86} + \frac{3x_3}{43} - \frac{18}{43} \right)^2.$$

Case II: $\mathbf{k > n+1}$. Let $\omega : \mathbb{R}^n \to \mathbb{R}^{k-1}$ be the monomial function such that

$$[x]_{\mathbb{B}_0} = \begin{bmatrix} 1 \\ \omega(x) \end{bmatrix}, \tag{7.27}$$

where $\mathbb{B}_0$ is the power set in (7.5). For the set S as in (7.22), denote

$$\hat{S} := \Big\{ \omega(u_1), \ldots, \omega(u_k) \Big\} \subseteq \mathbb{R}^{k-1}. \tag{7.28}$$

Define the linear map $\mathcal{L}$ such that

$$\mathcal{L} : \mathbb{R}^{k-1} \to \mathbb{R}^{k-1}, \quad \mathcal{L}(e_i) = \omega(u_i) - \omega(u_k),\ i = 1, \ldots, k-1.$$

The representing matrix for the linear map $\mathcal{L}$ is

$$L = \Big[\omega(u_1) \quad \cdots \quad \omega(u_{k-1}) \Big] - \Big[\omega(u_k) \quad \cdots \quad \omega(u_k) \Big]. \tag{7.29}$$

When $u_1, \ldots, u_n$ are in generic positions, the matrix L is nonsingular. For such a case, define the function

$$\hat{f}(z) := F_{k-1}\big(L^{-1}(z - \omega(u_k)) \big), \tag{7.30}$$

in the $z = (z_1, \ldots, z_{k-1})$, where F_{k-1} is the simplicial loss function as in (7.23). The above $\hat{f}$ is called a transformed simplicial loss function for $\hat{S}$. The following follows from Theorem 7.6.

Theorem 7.8 ([95]). *Suppose $k > n+1$ and L is nonsingular. Then, the function $\hat{f}$ as in (7.30) is a loss function for $\hat{S}$ and it has no spurious minimizers.*

For $x = (x_1, \dots, x_n)$, define the function

$$f(x) = F_{k-1}\big(L^{-1}(\omega(x) - \omega(u_k))\big). \tag{7.31}$$

Corollary 7.9 ([95]). *Suppose $k > n+1$ and L in (7.29) is nonsingular, then the function f in (7.31) is a loss function for S.*

Proof. The function f as in (7.31) is nonnegative everywhere. By Theorem 7.8, we know $f(x) = 0$ if and only if $\omega(x) \in \hat{S}$. Since ω is a one-to-one map, the f is a loss function for S. □

The following is an example of transformed simplicial loss functions for the case $k \geq n+1$..

Example 7.10. *Consider the $S = \left\{ \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \begin{bmatrix} -1 \\ -2 \end{bmatrix}, \begin{bmatrix} 1 \\ -3 \end{bmatrix}, \begin{bmatrix} -2 \\ 2 \end{bmatrix} \right\}$ in $\mathbb{R}^2$. Since $k = 4 > n+1$, the set $\hat{S}$ in (7.28) is*

$$\hat{S} = \left\{ \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}, \begin{bmatrix} -1 \\ -2 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ -3 \\ 1 \end{bmatrix}, \begin{bmatrix} -2 \\ 2 \\ 4 \end{bmatrix} \right\}.$$

The matrix L as in (7.29) and its inverse are

$$L = \begin{bmatrix} 4 & 1 & 3 \\ 1 & -4 & -5 \\ 0 & -3 & -3 \end{bmatrix}, \quad L^{-1} = \frac{1}{18} \begin{bmatrix} 3 & 6 & -7 \\ -3 & 12 & -23 \\ 3 & -12 & 17 \end{bmatrix}.$$

Since $k = 4$, the simplicial loss function for Δ_{k-1} is

$$F_3(z) = z_1^2(z_1 - 1)^2 + z_1^2 z_2^2 + z_2^2(z_2 - 1)^2 + z_2^2 z_3^2 + z_3^2(z_3 - 1)^2.$$

in the variable $z = (z_1, z_2, z_3)$. Then, the transformed simplicial loss function as in (7.30) is $\hat{f}(z) = F_3(L^{-1}(z - \omega(u_4)))$, with

$$L^{-1}(z - \omega(u_4)) = \frac{1}{18} \begin{bmatrix} 3z_1 + 6z_2 - 7z_3 + 22 \\ -3z_1 + 12z_2 - 23z_3 + 62 \\ 3z_1 - 12z_2 + 17z_3 - 38 \end{bmatrix}.$$

7.4 Finite sets with noises

In this section, we study loss functions for finite sets that are given with noises. Suppose S is approximately given by a sampling set T, say,

$$T = \{v_1, \dots, v_N\}. \tag{7.32}$$

Each point of S is sampled by a certain number of points in T. First, we discuss how to recover the k points of S from sampling points in T.

A finite set can be represented as the optimizer set of a loss function. For convenience, we consider loss functions whose minimum values are zeros. Let $\mathcal{F}$ be a family of loss functions such that each $f \in \mathcal{F}$ has k common zeros. The loss function family $\mathcal{F}$ is parameterized by some parameters. For such given $\mathcal{F}$, we look for the best loss function in $\mathcal{F}$ such that its average value on T is the smallest. This leads to the following definition.

Definition 7.11. *Let $\mathcal{F}$ be a family of loss functions such that each $f \in \mathcal{F}$ is nonnegative and it has k common zeros. A set $S^* = \{u_1^*, \dots, u_k^*\}$ is called the best $\mathcal{F}$-approximation set for T as in (7.32) if S^* is the zero set of f^*, where f^* is the minimizer of the optimization*

$$\begin{cases} \min & \mu(f) := \frac{1}{N} \sum\limits_{i=1}^{N} f(v_i) \\ s.t. & f \in \mathcal{F}. \end{cases} \tag{7.33}$$

we consider the family of the following loss functions

$$f_G := \|\varphi[G]\|^2, \tag{7.34}$$

parameterized by G. We look for the matrix G such that the average of the values of f_G on T is minimum and $\varphi[G]$ has k common zeros.

Then we consider the following matrix optimization problem

$$\begin{cases} \min & \vartheta(G) := \frac{1}{N} \sum\limits_{j=1}^{N} f_G(v_j) \\ s.t. & [M_{x_i}(G), M_{x_j}(G)] = 0 \, (1 \le i < j \le n). \end{cases} \tag{7.35}$$

The value $\varphi[G](v_i)$ is linear in the matrix G. The feasible set of (7.35) is given by a set of quadratic equations. The optimization (7.35) is the specialization of (7.33) such that $\mathcal{F}$ is the family of loss function f_G, with $\varphi[G]$ having k common zeros.

Suppose G^* is the minimizer of (7.35). Let S_0 denote the common zero set of $\varphi[G^*]$. We can use S_0 to approximate the points in S. In some applications, the set S contains only real points and people like to get a real set approximation for S.

First, we study the approximation quality of the optimization (7.35). For each $\alpha \in \mathbb{B}_1$, the sub-Hessian of the objective $\vartheta(G)$ with respect to the αth column $G(:,\alpha)$ is the matrix

$$H := \frac{2}{N}\sum_{j=1}^{N}[v_j]_{\mathbb{B}_0}([v_j]_{\mathbb{B}_0})^{\mathtt{H}}.$$

In the above, the superscript $^{\mathtt{H}}$ denotes the Hermitian transpose.

Theorem 7.12 ([95]). *Let T be as in (7.32) and let $S = \{u_1, \dots, u_k\}$ be such that the matrix X_0 as in (7.8) is nonsingular. Assume there exists $\delta > 0$ such that $H \succeq 2\delta I_k$. Suppose the set T is such that*

$$T \subseteq S + B(0,\epsilon), \quad T \cap B(u_i,\epsilon) \neq \emptyset \, (i = 1, \dots, k), \tag{7.36}$$

for some $\epsilon > 0$. Then, as $\epsilon \to 0$, the optimizer G^* of (7.35) converges to $\hat{G} := X_0^{-T}X_1^T$, and the common zero set S_0 of $\varphi[G^*]$ converges to S.

In particular, when $S, T \subseteq \mathbb{R}^n$, if $\epsilon > 0$ is sufficiently small, the common zero set S_0 contains k distinct real points.

Proof. First, we show the convergence $G^* \to \hat{G}$ as $\epsilon \to 0$. Since the set $\hat{B} := \cup_{i=1}^{k} B(u_i, 1)$ is compact, the polynomial function $\varphi[\hat{G}](x)$ is Lipschitz continuous on $\hat{B}$. There exists $R > 0$ such that for all $i \in [k]$ and for all $x \in B(u_i,\epsilon)$,

$$\|\varphi[\hat{G}](x) - \varphi[\hat{G}](u_i)\| \le R\|x - u_i\| \le R\epsilon.$$

Since $T \subseteq S + B(0,\epsilon)$, each $v_j \in T$ belongs to some $B(u_{i_j},\epsilon)$ for $i_j \in \{1,\dots,k\}$. So the above inequality implies that (note that each $\varphi[\hat{G}](u_{i_j}) = 0$)

$$\vartheta(\hat{G}) = \frac{1}{N}\sum_{j=1}^{N}\|\varphi[\hat{G}](v_j)\|^2 = \frac{1}{N}\sum_{j=1}^{N}\|\varphi[\hat{G}](v_j) - \varphi[\hat{G}](u_{i_j})\|^2 \le (R\epsilon)^2.$$

Since G^* is the minimizer of (7.35), we have

$$0 \le \vartheta(G^*) \le \vartheta(\hat{G}) \le (R\epsilon)^2. \tag{7.37}$$

Moreover, it holds that

$$\begin{aligned}\vartheta(G^*) &= \frac{1}{N}\sum_{j=1}^{N}\|\varphi[G^*](v_j) - \varphi[\hat{G}](v_j) + \varphi[\hat{G}](v_j)\|^2, \\ &\geq \frac{1}{N}\sum_{j=1}^{N}\Big(\|\varphi[G^*](v_j) - \varphi[\hat{G}](v_j)\| - \|\varphi[\hat{G}](v_j)\|\Big)^2 \\ &\geq \frac{1}{N^2}\Big(\sum_{j=1}^{N}\|\varphi[G^*](v_j) - \varphi[\hat{G}](v_j)\| - \sum_{j=1}^{N}\|\varphi[\hat{G}](v_j)\|\Big)^2.\end{aligned}$$

In the above, the first inequality follows from that $\|a+b\|^2 \geq (\|a\| - \|b\|)^2$ and the second inequality follows from the Cauchy-Schwartz inequality. Then, we have

$$\sum_{j=1}^{N}\|\varphi[G^*](v_j) - \varphi[\hat{G}](v_j)\| \leq N\sqrt{\vartheta(G^*)} + \sum_{j=1}^{N}\|\varphi[\hat{G}](v_j)\|$$

By the formula of $\varphi[G](x)$ and using Cauchy-Schwartz inequality again, we get

$$\sum_{j=1}^{N}\|(G^* - \hat{G})^T[v_j]_{\mathbb{B}_0}\| \leq N\Big(\sqrt{\vartheta(G^*)} + \sqrt{\vartheta(\hat{G})}\Big).$$

Since $\sum_{j=1}^{N}\|(G^* - \hat{G})^T[v_j]_{\mathbb{B}_0}\|^2 \leq \big(\sum_{j=1}^{N}\|(G^* - \hat{G})^T[v_j]_{\mathbb{B}_0}\|\big)^2$, we have

$$\frac{1}{N}\sum_{j=1}^{N}\|(G^* - \hat{G})^T[v_j]_{\mathbb{B}_0}\|^2 \leq N\Big(\sqrt{\vartheta(G^*)} + \sqrt{\vartheta(\hat{G})}\Big)^2.$$

By the assumption $H \succeq 2\delta I_k$, the above implies

$$\|G^* - \hat{G}\| \leq \sqrt{\frac{N}{\delta}}\Big(\sqrt{\vartheta(G^*)} + \sqrt{\vartheta(\hat{G})}\Big).$$

Therefore, as $\epsilon \to 0$, we have G^* converges to $\hat{G}$.

In the following, we assume that $S, T \subseteq \mathbb{R}^n$. Since X_0 is nonsingular, S has k distinct real points. Recall the multiplication matrices $M_{x_i}(G^*), M_{x_i}(\hat{G})$ given as in (7.10). Since $G^* \to \hat{G}$, the common zero set of $\varphi[G^*]$ converges to that of $\varphi[\hat{G}]$. The zero set of $\varphi[\hat{G}]$ is S, which consists of k distinct real points. Hence, $\varphi[G^*]$ also has k distinct common zeros when $\epsilon > 0$ is sufficiently small. Then it remains for us to show that all common zeros of $\varphi[G^*]$ are real. For a vector $\xi = (\xi_1, \ldots, \xi_n)$, define the matrices

$$M_1 = \sum_{i=1}^{n}\xi_i M_{x_i}(G^*), \quad M_2 = \sum_{i=1}^{n}\xi_i M_{x_i}(\hat{G}).$$

Their characteristic polynomials are

$$p_1(\lambda) := \det(M_1 - \lambda I), \quad p_2(\lambda) := \det(M_2 - \lambda I).$$

Fix a generic real value for ξ so that M_2 has k distinct real eigenvalues. This is because $\varphi[\hat{G}](x)$ has real distinct solutions and by the Stickelberger's Theorem (see (7.39) as in [65, 106]). Note that both $p_1(\lambda)$, $p_2(\lambda)$ have degree k and all coefficients are real. The $p_2(\lambda)$ has k distinct real roots. They are ordered as

$$\hat{\lambda}_1 < \hat{\lambda}_2 < \cdots < \hat{\lambda}_k.$$

We can choose real scalars $b_0, \ldots, b_k$ such that

$$b_0 < \hat{\lambda}_1 < b_1 < \cdots < b_{k-1} < \hat{\lambda}_k < b_k.$$

As $\epsilon \to 0$, the coefficients of p_1 converge to those of p_2. So, when $\epsilon > 0$ is small enough, $p_1(b_j)$ has the same sign as $p_2(b_j)$ does. Since each $p_2(b_{j-1})p_2(b_j) < 0$, we have

$$p_1(b_{j-1})p_1(b_j) < 0, \quad j = 1, \ldots, k+1.$$

This implies that p_1 has k distinct real roots. Equivalently, M_1 has k distinct real eigenvalues for $\epsilon > 0$ sufficiently small. By Proposition 7.2, all the multiplication matrices $M_{x_i}(G^*)$ are simultaneously diagonalizable. Also note that M_1 is diagonalizable and there is a unique real eigenvector (up to scaling) for each real eigenvalue. This shows that $M_{x_1}(G^*), \ldots, M_{x_n}(G^*)$ can be simultaneously diagonalized by common real eigenvectors. All $M_{x_1}(G^*), \ldots, M_{x_n}(G^*)$ have real entries, so they have only real eigenvalues. Therefore, by Stickelberger's Theorem, $\varphi[G^*]$ has k distinct real common zeros if $\epsilon > 0$ is sufficiently small. $\square$

When the set S is approximately given by the sampling set T, we can solve (7.35) for an optimizer matrix G^*, to get loss functions. Let S_0 be the common zero set of the polynomial tuple $\varphi[G^*]$. If T is far from S, S_0 may have non-real points. If real points are wanted, we can choose the real part set

$$S^{re} := \{\mathrm{Re}(u) : u \in S_0\}. \tag{7.38}$$

First, we show how to compute the common zero set S_0. By Stickelberger's Theorem (see [65, 106]), the set S_0 can be expressed as

$$S_0 = \left\{ (\lambda_1, \ldots, \lambda_n) \;\middle|\; \begin{array}{c} \exists q \in \mathbb{C}^k \setminus \{0\} \text{ such that} \\ M_{x_i}(G^*)q = \lambda_i q,\ i = 1, \ldots, n \end{array} \right\}. \tag{7.39}$$

To get S_0 numerically, people often use Schur decompositions. Let

$$M_1 = \xi_1 M_{x_1}(G^*) + \cdots + \xi_n M_{x_n}(G^*), \tag{7.40}$$

where $\xi_1, \ldots, \xi_n$ are generically chosen scalars. Then, compute the Schur decomposition for M_1:

$$Q^{\mathtt{H}} M_1 Q \;=\; P, \quad Q = \begin{bmatrix} q_1 & \cdots & q_k \end{bmatrix}. \tag{7.41}$$

In the above, $Q \in \mathbb{C}^{k\times k}$ is a unitary matrix and $P \in \mathbb{C}^{k\times k}$ is upper triangular. Based on the Schur decomposition (7.41), the common zeros $\hat{u}_1, \ldots, \hat{u}_k$ of $\varphi[G^*]$ can be given as

$$\hat{u}_i := \big(q_i^{\mathtt{H}} M_{x_1}(G^*) q_i, \ldots, q_i^{\mathtt{H}} M_{x_n}(G^*) q_i\big), \quad i = 1, \ldots, k. \tag{7.42}$$

We refer to [21] for how to use Schur decompositions to compute common zeros of zero-dimensional polynomial systems. For general cases, the set S_0 contains k distinct points. It holds when $S, T \subseteq \mathbb{R}^n$ and the points in T are close to S; see Theorem 7.12.

Based on the above discussions, we get the following algorithm for obtaining loss functions when S is approximately given by the sampling set T.

Algorithm 7.13. *For the given set T as in (7.32) and the cardinality k, do the following:*

Step 1 Solve quadratic optimization (7.35) for the optimizer G^*.

Step 2 Compute the common zero set $S_0 = \{\hat{u}_1, \ldots, \hat{u}_k\}$ of $\varphi[G^*]$. Let S^* be the set S_0 or S^{re} be as in (7.38) if the real points are wanted.

Step 3 Get a loss function for the set S^*, by the method in Section 7.2 or Section 7.3.

In Step 1, the optimization (7.35) has a convex quadratic objective, but its constraints are given by quadratic equations, in the matrix variable G. So (7.35) is a quadratically constrained quadratic program (QCQP). It can be solved as a polynomial optimization problem (e.g., by the software `GloptiPoly 3` [48]). The classical nonlinear optimization methods, (e.g., Gauss-Newton, trust region, and Levenberg-Marquardt type methods) can also be applied to solve (7.35). We refer to [51, 76, 117] for such references.

In Step 2, the common zero set S_0 can be computed as in (7.42), by using the Schur decomposition (7.41) for the matrix M_1 in (7.40), for generically chosen scalars $\xi_1, \ldots, \xi_n$.

In Step 3, there are two options for obtaining loss functions for the set S^*, given in Sections 7.2 and 7.3 respectively. One is to choose $f = \|\varphi[G]\|^2$; the other one is to apply a transformation first and then choose f similarly. After the transformation, there are no spurious optimizers for the loss function.

7.5 Applications

In this section, we present numerical experiments for loss functions. The computation is implemented in `MATLAB` R2018a, in a Laptop with CPU 8th Generation Intel® Core™ i5-8250U and RAM 16 GB. The optimization problem (7.35) can be solved by the polynomial optimization software `GloptiPoly 3` (with the SDP solver `SeDuMi`), or it can be solved by classical nonlinear optimization solvers (e.g., the `MATLAB` function `fmincon` can be used for convenience). First, we explore the numerical performance of Algorithm 7.13.

Example 7.14. *Consider the set*

$$S = \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 3 \\ 2 \end{bmatrix}, \begin{bmatrix} 1.5 \\ 2.5 \end{bmatrix}, \begin{bmatrix} 2.5 \\ 3 \end{bmatrix}, \begin{bmatrix} 2 \\ 1.5 \end{bmatrix}, \begin{bmatrix} 3 \\ 1 \end{bmatrix} \right\}.$$

Suppose T is a sampling set of S such that

$$T \subseteq S + \epsilon[-1,1]^2, \quad |T \cap \{u_i + \epsilon[-1,1]^2\}| = N_i \, (i = 1, \ldots, 6).$$

We apply Algorithm 7.13 for cases $N_i \in \{50, 100\}$ and $\epsilon \in \{0.05, 0.1, 0.5\}$. The samples are generated with `MATLAB` function `randn`. We summarize the computational results in Table 7.1 and Figure 7.1. In Table 7.1, the symbol S^* denotes the computed approximation set as in (7.38). We use the distance

$$\|S - S^*\| := \max_{v \in S^*} \min_{u_i \in S} \|v - u_i\|$$

to measure the approximation quality of S^* to S. We use the loss function $f = \|\varphi[G]\|^2$ for S^*. The maximum value of f on S is shown in the third column. In Figure 7.1, the sampling points in T are plotted in dots, the points in S are plotted in the diamond symbol and the points in S^* are plotted in the square symbol. The left column from top to bottom shows cases for $N_i = 50$ and $\epsilon = 0.05, 0.1, 0.5$ respectively. The right column shows cases for $N_i = 100$ accordingly.

Example 7.15. *We use Algorithm 7.13 and the transformed simplicial loss functions in Section 7.3 to learn Gaussian mixture models (GMMs). Each GMM has parameters (w_i, μ_i, Σ_i), $i = 1, \ldots, k$, where each weight $w_i > 0$, the mean vector $\mu_i \in \mathbb{R}^n$ and the covariance matrix $\Sigma_i \in \mathcal{S}^n_{++}$ (the cone of real symmetric positive definite n-by-n matrices), such that $w_1 + \cdots + w_k = 1$. We explore the performance of transformed simplicial loss functions for two cases*

$$I) : \, n = 4, \, k \in \{4, 5\}, \quad II) : \, n = 5, \, k \in \{3, 4\}.$$

Table 7.1: The numerical results of Example 7.14

N_i	ϵ	$\|S - S^*\|$	$\max_{u \in S} f(u)$
50	0.05	0.0064	$1.27 \cdot 10^{-4}$
	0.1	0.0145	$2.98 \cdot 10^{-4}$
	0.5	0.1821	0.0862
N_i	ϵ	$\|S - S^*\|$	$\max_{u \in S} f(u)$
100	0.05	0.0055	$8.06 \cdot 10^{-5}$
	0.1	0.0067	$1.89 \cdot 10^{-4}$
	0.5	0.1080	0.0359

In particular, we compare the results for diagonal Gaussian mixture models (each Σ_i is diagonal) and non-diagonal Gaussian mixture models (each Σ_i is non-diagonal). For each instance, 1000 samples are generated. The weights $w_1, \dots, w_k$ are also computed from sampling: we first use the `MATLAB` *command* ***randi*** *getting 1000 integers from $[k]$, and then counting each w_i based on the occurrence probability of $i \in [k]$. We generate each covariance matrix as $\Sigma_i = R^T R$, for some randomly generated square matrix R. The clustering accuracy rate counts the percentage of samples belonging to the correct cluster. (For a point $v \in T$, apply a nonlinear optimization method to minimize f with the starting point v. Once a minimizer u is returned, we cluster v to the group labeled by the point $u \in S^*$.) We run* 10 *instances for each case and count the accuracy rate as an average for all instances. The computational results are reported in Table 7.2. Algorithm 7.13 together with transformed simplicial loss functions has good performance for both diagonal and non-diagonal Gaussian mixture models. The clustering accuracy rate is higher for non-diagonal Gaussian mixtures than that for diagonal ones. In particular, for $(n, k) = (4, 5)$, the clustering accuracy rate can be as high as 98.92%.*

Table 7.2: The accuracy rates for Example 7.15.

n	k	diagonal	non-diagonal
4	4	77.66%	85.34%
	5	88.73%	98.92%
5	3	80.93%	84.04%
	4	82.40%	89.58%

Acknowledgments. This Chapter, in full, has been submitted for publication. The dissertation author coauthored this paper with Nie, Jiawang.

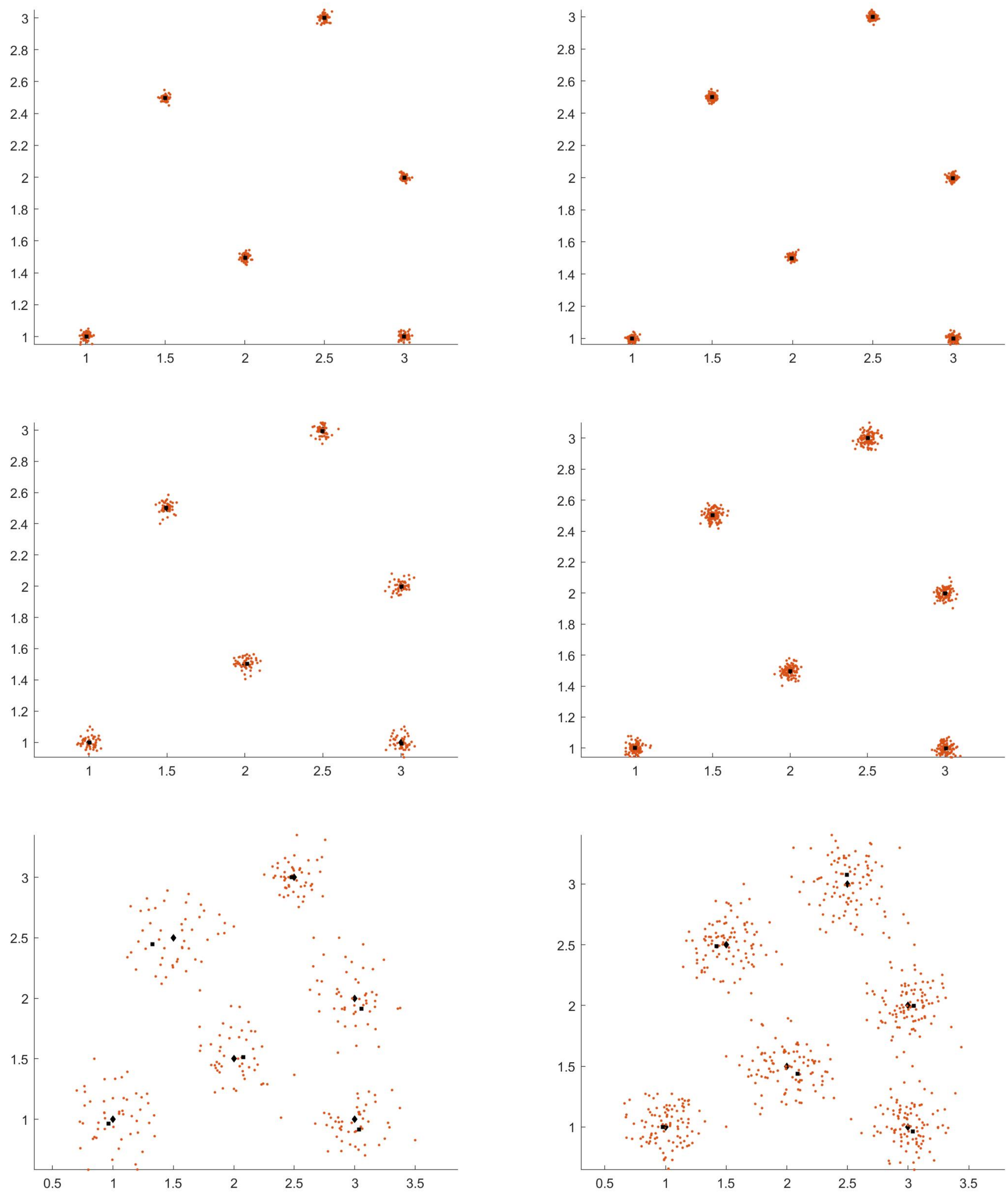

Figure 7.1: The visualization of Example 7.14. The left column is for $N_i = 50$, and the right column is for $N_i = 100$. The first row is for $\epsilon = 0.05$, the second row is for $\epsilon = 0.1$, and the third row is for $\epsilon = 0.5$.

Bibliography

[1] G. Allende and G. Still, Solving bilevel programs with the KKT-approach, *Math. Program.*, 138 (2013), pp. 309-332.

[2] D. Ardagna, M. Ciavotta and M. Passacantando, Generalized Nash equilibria for the service provisioning problem in multi-cloud systems, *IEEE Transactions on Services Computing* 10, 381–395, 2017.

[3] J. Bard, *Practical Bilevel Optimization: Algorithms and Applications,* Dordrecht: Kluwer Academic Publishers, 1998.

[4] F. Bastin, C. Cirillo and P. Toint, Convergence theory for nonconvex stochastic programming with an application to mixed logit, *Math. Program.*, 108(2–3), pp. 207–234, 2006.

[5] J. T. Barron, A general and adaptive robust loss function, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[6] A. Ben-Tal, D. Den Herto, et al., Robust solutions of optimization problems affected by uncertain probabilities, *Management Science 59*, no. 2 (2013): 341-357.

[7] A. Ben-Tal, and A. Nemirovski, *Lectures on Modern Convex Optimization (2012)*, SIAM, Philadelphia, PA. Google Scholar Google Scholar Digital Library Digital Library (2011).

[8] D. Bertsekas, *Nonlinear Programming*, second edition, Athena Scientific, 1995.

[9] D. Bertsimas, M. Sim, and M. Zhang, Adaptive distributionally robust optimization, *Management Science* 65.2 (2019): 604-618.

[10] M. Bjørndal and K. Jørnsten, The deregulated electricity market viewed as a bilevel programming problem, *J. Global Optim.*, 33 (2005), pp. 465-475.

[11] G. Boglárka and K. Kovács, Solving a Huff-like Stackelberg location problem on networks, *J. Global Optim.*, 64 (2016), pp. 233-247.

[12] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[13] M. Branda, Sample approximation technique for mixed-integer stochastic programming problems with several chance constraints, *Oper. Res. Lett.*, 40(3), pp. 207–211, 2012.

[14] X. Chen, Y. Shi and X. Wang, Equilibrium oil market share under the COVID-19 pandemic, *Preprint*, 2020. arXiv:2007.15265

[15] Z. Chen, M. Sim, and H. Xu, Distributionally robust optimization with infinitely constrained ambiguity sets, *Oper. Res.* 67.5 (2019): 1328-1344.

[16] D. Cheng, Y. Gong, S. Zhou, et al., Person re-identification by multi-channel parts-based CNN with improved triplet loss function, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[17] P. Christoffersen and K. Jacobs, The importance of the loss function in option valuation, *Journal of Financial Economics* 72(2) 291–318, 2004.

[18] F. Clarke, *Optimization and Nonsmooth Analysis,* Society for Industrial and Applied Mathematics, 1990.

[19] B. Colson, P. Marcotte and G. Savard, An overview of bilevel optimization, *Ann. Oper. Res.*, 153 (2007), pp. 235-256.

[20] J. Contreras, M. Klusch and J.B. Krawczyk, Numerical solutions to Nash-Cournot equilibria in coupled constraint electricity markets, *IEEE Transactions on Power Systems* 19(1), 195–206, 2004.

[21] R. M. Corless, P. M. Gianni and B. M. Trager, A reordered Schur factorization method for zero-dimensional polynomial systems with multiple roots, *Proceedings of the Internaltional Symposium on Symbolic and Algebraic Computation*, pp. 133–140, Maui, Hawaii, 1977.

[22] D. Cox, J. Little, and D. OShea. *Ideals, Varieties, and Algorithms: An introduction to Computational Algebraic Geometry and Commutative Algebra*, Springer Science & Business Media, 2013.

[23] R. Curto and L. Fialkow, Recursiveness, positivity, and truncated moment problems, *Houston Journal of Mathematics* 17 (1991), pp. 603-635.

[24] R. Curto and L. Fialkow, Truncated K-moment problems in several variables. *Journal of Operator Theory*, 54(2005), pp. 189-226.

[25] E. Delage and Y. Ye, Distributionally robust optimization under moment uncertainty with application to data-driven problems, *Oper. Res.* 58.3 (2010): 595-612.

[26] S. Dempe, *Bilevel Optimization: Theory, Algorithms and Applications*, Vol. 3, TU Bergakademie Freiberg, Fakultät für Mathematik und Informatik, 2018.

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

[28] S. Dempe and A. Zemkoho, Bilevel optimization: Advances and next challenges, *Springer Optimization and its Applications*, vol. 161, 2020.

[29] S. Dempe and S. Franke, Solution algorithm for an optimistic linear Stackelberg problem, *Comput. Oper. Res.*, 41 (2014), pp. 277-281.

[30] S. Dempe, V. Kalashnikov, G. Pérez-Valdés and N. Kalashnykova, *Bilevel Programming Problems,* Energy Systems, Springer, Berlin. 2015.

[31] A. Dreves, F. Facchinei, C. Kanzow, and S. Sagratella, On the solution of the KKT conditions of generalized Nash equilibrium problems, *SIAM J. Optim.*, 21, 1082–1108, 2011.

[32] J. C. Duchi and H. Namkoong, Learning models with uniform performance via distributionally robust optimization, *The Annals of Statistics*, 49.3 (2021): 1378-1406.

[33] P. Esfahani and D. Kuhn, Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations, *Math. Program.*, 171.1-2 (2018): 115-166.

[34] F. Facchinei, A. Fischer, and V. Piccialli, Generalized Nash equilibrium problems and Newton methods, *Math. Program.*, 117(1), 163–194, 2009.

[35] F. Facchinei and C. Kanzow, Generalized Nash equilibrium problems, *Ann. Oper. Res.*, 175(1), 177–211, 2010.

[36] L. Franceschi, P. Frasconi, S. Salzo, R. Grazzi and M. Pontil, Bilevel programming for hyperparameter optimization, *International Conference on Machine Learning*, (2018), pp. 1563-1572.

[37] M. Fu (eds.), *Handbook of Simulation Optimization*, volume 216, Springer, New York, 2015.

[38] J. Gauvin and F. Dubeau, Differential properties of the marginal function in mathematical programming, *Math. Program. Stu.*, 19 (1982), pp. 101-119.

[39] S. Ghadimi and G. Lan, Stochastic first-and zeroth-order methods for nonconvex stochastic programming, *SIAM J. Optim.*, 23(4), pp. 2341–2368, 2013.

[40] S. Ghadimi, G. Lan and H. Zhang, Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization, *Math. Program.*, 155(1-2), pp. 267–305, 2016.

[41] L. Guo, G. Lin, J.J. Ye and J. Zhang, Sensitivity analysis of the value function for parametric mathematical programs with equilibrium constraints, *SIAM J. Optim.*, 24 (2014), pp. 1206-1237.

[42] B. Guo, J. Nie, and Z. Yang, Learning diagonal Gaussian mixture models and incomplete tensor decompositions, *Vietnam J. Math.*, 50.2 (2022): 421–446.

[43] M. Gürbüzbalaban, A. Ruszczyński, and L. Zhu, A stochastic subgradient method for distributionally robust non-convex learning, *Preprint*, 2020. `arXiv:2006.04873`.

[44] G. Hanasusanto, V. Roitch, et al, A distributionally robust perspective on uncertainty quantification and chance constrained programming, *Math. Program.*, 151.1 (2015): 35-62.

[45] J. W. Helton and J. Nie, Semidefinite representation of convex sets, *Math. Program.*, 122 (1), 21–64, 2010.

[46] J. Helton and J. Nie, A semidefinite approach for truncated K-moment problem, *Found. Comput. Math.*, 12(6), pp. 851–881, 2012.

[47] D. Henrion and J. Lasserre, Detecting global optimality and extracting solutions in GloptiPoly. *Positive polynomials in control*, 293–310, Lecture Notes in Control and Inform. Sci., 312, Springer, Berlin, 2005.

[48] D. Henrion, J. Lasserre, and J. Löfberg, `Gloptipoly 3:` moments, optimization and semidefinite programming. *Optim. Meth. Softw.*, 24(4-5), pp. 761–779, 2009.

[49] A. von Heusinger, C. Kanzow, and M. Fukushima, Newton's method for computing a normalized equilibrium in the generalized Nash game through fixed point formulation, *Math. Program.* 132(1), 99–123, 2012.

[50] K. Judd, The law of large numbers with a continuum of iid random variables, *Journal of Economic theory*, 35(1), pp. 19–25, 1985.

[51] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, Frontiers in Applied Mathematics 16, SIAM, Philadelphia, 1995.

[52] A. Kesselman, S. Leonardi, and V. Bonifaci, Game-theoretic analysis of internet switching with selfish users, *International Workshop on Internet and Network Economics*, (pp. 236-245), Springer, Berlin, Heidelberg, 2005.

[53] E. de. Klerk, D. Kuhn, and K. Postek, Distributionally robust optimization with polynomial densities: theory, models and algorithms, *Math. Program.*, (2019): 1-32.

[54] A. J. Kleywegt, A. Shapiro and T. Homem-de-Mello, The sample average approximation method for stochastic discrete optimization, *SIAM J. Optim.*, 12.2 (2002): 479-502.

[55] Y. H. Ko, K. J. Kim, and C. H. Jun, A new loss function-based method for multiresponse optimization, *Journal of Quality Technology* 37(1), 50–59, 2005.

[56] M. Krein and D. Louvish, *The Markov Moment Problem and Extremal Problems*, American Mathematical Society, 1977.

[57] G. Kunapuli, K. Bennett, J. Hu and J-S. Pang, Classification model selection via bilevel programming, *Optim. Meth. Softw.*, 23 (2008), pp. 475-489.

[58] G. Lan, *First-oder and Stochastic Optimization Methods for Machine Learning*, Switzerland: Springer, 2020.

[59] J. Lasserre, Global optimization with polynomials and the problem of moments, *SIAM J. Optim.*, 11, pp. 796–817, 2001.

[60] J. Lasserre, Convexity in semi-algebraic geometry and polynomial optimization, *SIAM J. Optim.*, 19 (2009): 1995-2014.

[61] J. Lasserre, *Moment, Positive Polynomials and Their Applications*, Imperial College Press, 2009.

[62] J. Lasserre, On representations of the feasible set in convex optimization, *Optim. Lett.*, 4(1), 1-5, 2010.

[63] J. Lasserre, M. Laurent and P. Rostalski, Semidefinite characterization and computation of zero-dimensional real radical ideals, *Foundations of Computational Mathematics* 8(5), 607–647, 2008.

[64] J. Lasserre and T. Weisser, Distributionally robust polynomial chance-constraints under mixture ambiguity sets, *Math. Program.* 185.1 (2021): 409-453.

[65] M. Laurent, Sums of squares, moment matrices and optimization over polynomials, *Emerging Applications of Algebraic Geometry of IMA Volumes in Mathematics and its Applications*, 149, pp. 157–270, Springer, 2009.

[66] G. Lin, M. Xu and J. Ye, On solving simple bilevel programs with a nonconvex lower level program, *Math. Program.*, 144 (2014), 277-305.

[67] J. Linderoth, A. Shapiro and S. Wright, The empirical behavior of sampling methods for stochastic programming, *Ann. Oper. Res.*, 142 (1): 215–241, 2006.

[68] R. Liu, P. Mu, X. Yuan, S. Zeng and J. Zhang, A generic first-order algorithmic Framework for bi-level programming beyond lower-level singleton, *International Conference on Machine Learning*, 2020.

[69] M. Liu and O. Tuzel, Coupled generative adversarial networks, *Advances in Neural Information Processing Systems*, 29 (2016): 469-477.

[70] J. Lofberg, YALMIP: A toolbox for modeling and optimization in MATLAB, *IEEE international conference on robotics and automation* (IEEE Cat. No. 04CH37508), IEEE, 2004.

[71] J. Luedtke and S. Ahmed, A sample approximation approach for optimization with probabilistic constraints, *SIAM J. Optim.*, 19(2), pp. 674–699, 2008.

[72] H. Markowitz, Portfolio selection, *Journal of Finance*, 7, 77-91, 1952.

[73] M. Mevissen, E. Ragnoli and J. Yu, Data-driven distributionally robust polynomial optimization, *Advances in Neural Information Processing Systems*, 26 (2013).

[74] J. Mirrlees, The theory of moral hazard and unobservable behaviour: part I, *Rev. Eco. Stud.*, 66 (1999), pp. 3-22.

[75] M. Mohri, G. Sivek and A. T. Suresh, Agnostic federated learning, *International Conference on Machine Learning*, PMLR, 2019.

[76] J. J. More, The Levenberg-Marquardt algorithm: implementation and theory, in: G. A. Watson, ed., *Lecture Notes in Mathematics 630: Numerical Analysis*, Springer-Verlag, Berlin, 1978, 105–116.

[77] L. Muu and N. Quy, A global optimization method for solving convex quadratic bilevel programming problems, *J. Global Optim.* 26 (2003), pp. 199-219.

[78] T. Netzer, On semidefinite representations of non-closed sets, *Linear Algebra and its Applications* 432 (2010), 3072–3078.

[79] J. Nie, Certifying convergence of Lasserre's hierarchy via flat truncation. *Math. Program.*, Ser. A, 142 (2013), no. 1-2, pp. 485–510.

[80] J. Nie, Optimality conditions and finite convergence of Lasserre's hierarchy, *Math. Program.* 146 (2014), pp. 97-121.

[81] J. Nie, The $\mathcal{A}$-Truncated K-Moment Problem, *Foundations of Computational Mathematics* 14.6 (2014): 1243-1276.

[82] J. Nie, Linear optimization with cones of moments and nonnegative polynomials, *Math. Program.*, 153.1 (2015): 247-274.

[83] J. Nie, Generating polynomials and symmetric tensor decompositions, *Foundation of Computational Mathematics* 17, 423–465, 2017.

[84] J. Nie, Low rank symmetric tensor approximations, *SIAM Journal on Matrix Analysis and Applications* 38(4), 1517–1540, 2017.

[85] J. Nie, Tight relaxations for polynomial optimization and Lagrange multiplier expressions, *Math. Program.*, 178 (2019), pp. 1-37.

[86] J. Nie and R. Kristian, Algebraic degree of polynomial optimization, *SIAM J. Optim.* 20, 485–502, 2009.

[87] J. Nie and X. Tang, Nash equilibrium problems of polynomials, *Preprint*, 2020. `arXiv:2006.09490`

[88] J. Nie and X. Tang, Convex generalized Nash equilibrium problems and polynomial optimization, *Math. Program.*, (2021): 1-34.

[89] J. Nie, X. Tang and L. Xu, The Gauss-Seidel method for generalized Nash equilibrium problems of polynomials, *Computational Optimization and Applications* 78, 529–557, 2021.

[90] J. Nie, X. Tang and S. Zhong, Rational generalized Nash equilibrium problems, *Preprint*, 2021. `arXiv:2110.12120`

[91] J. Nie, L. Wang and J. J. Ye, Bilevel polynomial programs and semidefinite relaxation methods, *SIAM J. Optim.*, 27 (2017), pp. 1728-1757.

[92] J. Nie, L. Wang, J. J. Ye and S. Zhong, A Lagrange multiplier expression method for bilevel polynomial optimization, *SIAM J. Optim.*, 31.3 (2021): 2368–2395.

[93] J. Nie, L. Yang and S. Zhong, Stochastic polynomial optimization, *Optim. Meth. Softw.*, 35.2 (2020): 329–347.

[94] J. Nie, L. Yang, S. Zhong and G. Zhou, Distributionally robust optimization with moment ambiguity sets, *Preprint*, 2021. `aerXiv:2103.12315`

[95] J. Nie and S. Zhong, Loss functions for finite sets, *Preprint*, 2021. `arXiv:2112:05927`

[96] J. Pang, G. Scutari, F. Facchinei and C. Wang, Distributed power allocation with rate constraints in Gaussian parallel interference channels, *IEEE Transactions on Information Theory* 54(8), 3471–3489, 2008.

[97] G. Pflug and D. Wozabal, Ambiguity in portfolio selection, *Quantitative Finance* 7.4 (2007): 435-442.

[98] K. Postek, A. Ben-Tal, et al, Robust optimization with ambiguous stochastic constraints under mean and dispersion information, *Operations Research* 2018 Jun;66(3):814-33.

[99] M. Putinar, Positive polynomials on compact semi-algebraic sets, *Indiana Univ. Math. J.*, 42 (1993), pp. 969–984.

[100] F. Schorfheide, Loss function-based evaluation of DSGE models, *Journal of Applied Econometrics* 15(6), 645–670, 2000.

[101] M. Schweighofer, Optimization of polynomials on compact semialgebraic sets, *SIAM J. Optim.*, 15(3), 805–825, 2005.

[102] A. Shapiro, D. Dentcheva and A. Ruszczynski, *Lectures on Stochastic Programming: Modeling and Theory*, Society for Industrial and Applied Mathematics, 2021.

[103] K. Shimizu, Y. Ishizuka and J. Bard, *Nondifferentiable and Two-level Mathematical Programming,* Kluwer Academic, 1997.

[104] K. Shimizu and E. Aiyoshi, A new computational method for Stackelberg and min-max problems by use of a penalty method, *IEEE Trans. Automat. Contr.*, 26 (1981), pp. 460-466.

[105] J. Sturm, SeDuMi 1.02: a MATLAB toolbox for optimization over symmetric cones, *Optimization Methods and Software*, 11&12, pp. 625–653, 1999. `http://sedumi.mcmaster.ca/`

[106] B. Sturmfels, Solving systems of polynomial equations, *CBMS Regional Conference Series in Mathematics*, 97, AMS, Providence, RI, 2002.

[107] C. H. Sudre, W. Li, T. Vercauteren, et al., Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations, *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, 240–248, Springer, Cham, 2017.

[108] B. Verweij, S. Ahmed, A. Kleywegt, G. Nemhauser, and A. Shapiro, The sample average approximation method applied to stochastic routing problems: a computational study, *Computational Optimization and Applications*, 24(2-3):289–333, 2003.

[109] Q. Wang, Y. Ma, K. Zhao, and Y. Tian, A comprehensive survey of loss functions in machine learning, *Annals of Data Science*, 2020. `doi.org/10.1007/s40745-020-00253-5`

[110] W. Wiesemann, D. Kuhn and M. Sim, Distributionally robust convex optimization, *Operations Research* 62.6 (2014): 1358-1376.

[111] Z. Wu, M. Shamsuzzaman and E. S. Pan, Optimization design of control charts based on Taguchi's loss function and random process shifts, *International Journal of Production Research* 42(2), 379–390, 2004.

[112] H. Xu, Y. Liu and H. Sun, Distributionally robust optimization with matrix moment constraints: Lagrange duality and cutting plane methods, *Mathematical Programming* 169.2 (2018): 489-529.

[113] Y. Yang and W. Wu, A Distributionally Robust Optimization Model for Real-Time Power Dispatch in Distribution Networks, *IEEE Transactions on Smart Grid* 10.4 (2018): 3743-3752.

[114] J. J. Ye, *Constraint qualifications and optimality conditions in bilevel optimization*, Bilevel Optimization: Advances and Next Challenges, Ch. 8, Springer Optimization and its Applications, vol. 161, 2020.

[115] J. J. Ye, X. Yuan, S. Zeng and J Zhang, Difference of convex algorithms for bilevel programs with applications in hyperparameter selection, *Preprint*, 2021. `arXiv:2102.09006`

[116] J. J. Ye and D. Zhu, Optimality conditions for bilevel programming problems, *Optim.*, 33 (1995), pp. 9-27.

[117] Y. X. Yuan, Recent advances in numerical methods for nonlinear equations and nonlinear least squares, *Numerical Algebra Control and Optimization*, 1, 15–34, 2011.

[118] Z. Zhang, S. Ahmed and G. Lan, Efficient algorithms for distributionally robust stochastic optimization with discrete scenario support, *SIAM J. Optim.*, 31.3 (2021), 1690–1721.

[119] J. Zhang, H. Xu and L. Zhang, Quantitative stability analysis for distributionally robust optimization with moment constraints, *SIAM J. Optim.*, 26.3 (2016): 1855-1862.

[120] S. Zhu and Masao Fukushima, Worst-case conditional value-at-risk with application to robust portfolio management, *Oper. Res.*, 57.5 (2009): 1155-1168.

[121] S. Zymler, D. Kuhn and B. Rustem, Distributionally robust joint chance constraints with second-order moment information, *Mathematical Programming* 137.1-2 (2013): 167-198.