Title

Research on Generalized Nash Equilibrium Problems

Author

Tang, Xindong

Publication Date

2021

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA SAN DIEGO

Research on Generalized Nash Equilibrium Problems

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

in

Mathematics

by

Xindong Tang

Committee in charge:

Professor Jiawang Nie, Chair
Professor Zhaowei Liu
Professor Brendon Rhoades
Professor Yixiao Sun
Professor Danna Zhang

2021

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

University of California San Diego

2021

TABLE OF CONTENTS

LIST OF TABLES

ACKNOWLEDGEMENTS

Firstly, I would express my greatest appreciation and respect to my advisor, Professor Jiawang Nie, for his helpfulness and encouragement during my five years of Ph.D. studies. To me, he is not only the dissertation advisor, but also my role model for research. During these five years in San Diego, his wonderful advising and encouragement helped me go through many difficult times in completing this dissertation. I will be forever thankful for this unique opportunity.

I am also grateful to other members of my dissertation committee. It is my great honor to have these great experts from different areas form my dissertation committee. Their expertise offered me valuable help in finishing this dissertation. The Department of Mathematics at UC San Diego provided fertile ground and an unsurpassable research environment to complete this dissertation. Its fantastic faculty and excellent staff gave generous support to my graduate life.

I would like to take this opportunity to thank Professor Lingling Xu, for her help and professional expertise on the topic of Generalized Nash Equilibrium Problems. Also, I want to thank Suhan Zhong, Zi Yang for fruitful discussions and all my friends for their help on my research and my life.

Moreover, I would like to give my deepest gratitude to my family for their endless support and love. My family is always the backbone of my life. Especially, I want to express my appreciation with all my heart to Dr. Xueqian Wu, for her unreserved patience and love in the past eight years.

In this dissertation, some materials have been published, or been submitted for publication. The Chapter 3, in full, has been submitted for publication. The dissertation author coauthored this paper with Nie, Jiawang. The Chapter 4, in full, has been submitted for publication. The dissertation author coauthored this paper with Nie, Jiawang. The Chapter 5, in full, is a reprint of the material as it appears in *Computational Optimization and Applications*, Springer Science+Business Media. The dissertation author coauthored this paper with Nie, Jiawang and Xu, Lingling.

VITA

2016	B. S. in Mathematics, Sichuan University
2021	Ph. D. in Mathematics, University of California San Diego

PUBLICATIONS

J. Nie, X. Tang and L. Xu, "The Gauss-Seidel Method for Generalized Nash Equilibrium Problems of Polynomials", *Computational Optimization and Applications*, 78 (2), 529-557.

J. Nie and X. Tang, "Nash Equilibrium Problems of Polynomials", *Preprint*, 2020. `arXiv:2006.09490`

J. Nie and X. Tang, "Convex Generalized Nash Equilibrium Problems and Polynomial Optimization", *Preprint*, 2020. `arXiv:2101.06504`

ABSTRACT OF THE DISSERTATION

Research on Generalized Nash Equilibrium Problems

by

Xindong Tang
Doctor of Philosophy in Mathematics

University of California San Diego, 2021

Professor Jiawang Nie, Chair

The Generalized Nash Equilibrium Problem (GNEP) is a kind of game to find strategies for a group of players such that each player's objective function is optimized, given other players' strategies. If all the objective and constraining functions involved are polynomials, we call the problem a Generalized Nash Equilibrium Problem of Polynomials (GNEPP). When the constraining functions of each player are independent of other player's strategies, the GNEP is called a (standard) Nash Equilibrium Problem (NEP). The GNEP is said to be convex if each player's optimization is a convex optimization problem, given other players' strategies.

For nonconvex Nash equilibrium problems that are given by polynomial functions, we formulate efficient polynomial optimization problems for computing Nash equilibria. We show that under generic assumptions, the method can find one or even all Nash equilibria if they exist, or detect nonexistence of Nash equilibria. For convex GNEPPs, we introduce rational and parametric expressions for Lagrange multipliers to formulate polynomial optimization for computing Generalized Nash Equilibria (GNEs). We prove that under some specific assumptions, the method can find a GNE if there exists one, or detect nonexistence of GNEs. Numerical experiments are presented to show the efficiency of the methods. The Moment-SOS hierarchy of semidefinite relaxations is used to solve the polynomial optimization.

Moreover, we study the Gauss-Seidel method for solving the nonconvex GNEPPs. We give a certificate for a class of GNEPPs such that the Gauss-Seidel method is guaranteed to converge, and the numerical experiments show that the Gauss-Seidel method can solve many GNEPPs efficiently.

Chapter 1

Introduction

The Generalized Nash Equilibrium Problem (GNEP) is a kind of games to find strategies for a group of players such that each player's objective function is optimized, for given other players' strategies. Suppose there are N players and the ith player's strategy is a vector $x_i \in \mathbb{R}^{n_i}$ (the n_i-dimensional real Euclidean space). We write that

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

The total dimension of all strategies is $n := n_1 + \ldots + n_N$. The main task of the GNEP is to find a tuple $u = (u_1, \ldots, u_N)$ of strategies such that each u_i is a minimizer of the ith player's optimization (denote $u_{-i} := (u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_N)$)

$$\mathrm{F}_i(u_{-i}) : \begin{cases} \min\limits_{x_i \in \mathbb{R}^{n_i}} & f_i(u_1, \ldots, u_{i-1}, x_i, u_{i+1}, \ldots, u_N) \\ s.t. & g_{i,j}(u_1, \ldots, u_{i-1}, x_i, u_{i+1}, \ldots, u_N) = 0 \, (j \in \mathcal{E}_i), \\ & g_{i,j}(u_1, \ldots, u_{i-1}, x_i, u_{i+1}, \ldots, u_N) \geq 0 \, (j \in \mathcal{I}_i), \end{cases} \tag{1.1}$$

where the f_i and $g_{i,j}$ are continuously differentiable functions in x_i, and the $\mathcal{E}_i$, $\mathcal{I}_i$ are disjoint finite (possibly empty) labeling sets. The point u satisfying the above is called a Generalized Nash Equilibrium (GNE). For notational convenience, when the ith player's strategy is considered, we use x_{-i} to denote the subvector of all players' strategies except the ith one, i.e.,

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

and write $x = (x_i, x_{-i})$ accordingly.

This paper focuses on the Generalized Nash Equilibrium Problem of Polynomials (GNEPP), i.e., all the functions f_i and $g_{i,j}$ are polynomials in x. For each $i = 1, \ldots, N$, let

X_i be the point-to-set map such that

$$X_i(x_{-i}) \;:=\; \left\{ x_i \in \mathbb{R}^{n_i} \;\middle|\; \begin{array}{l} g_{i,j}(x_i, x_{-i}) = 0,\, j \in \mathcal{E}_i, \\ g_{i,j}(x_i, x_{-i}) \geq 0,\, j \in \mathcal{I}_i \end{array} \right\}. \tag{1.2}$$

The $X_i(x_{-i})$ is the feasible strategy set of $\mathrm{F}_i(x_{-i})$. The domain of X_i is

$$\mathrm{dom}(X_i) := \{x_{-i} \in \mathbb{R}^{n-n_i} : X_i(x_{-i}) \neq \emptyset\}.$$

The tuple x is said to be a feasible point of the GNEP if $x_i \in X(x_{-i})$ for all i. Denote the set

$$X := \left\{ x \in \mathbb{R}^{n} \;\middle|\; \begin{array}{l} g_{i,j}(x_i, x_{-i}) = 0,\, j \in \mathcal{E}_i,\, i = 1, \ldots, N, \\ g_{i,j}(x_i, x_{-i}) \geq 0,\, j \in \mathcal{I}_i,\, i = 1, \ldots, N \end{array} \right\}. \tag{1.3}$$

Then x is a feasible point for the GNEP if and only if $x \in X$.

Definition 1.1. The GNEP given by (1.1) is called *convex* if for all $i = 1, \ldots, N$ and for all given $x_{-i} \in \mathrm{dom}(X_i)$, the objective $f_i(x_i, x_{-i})$ is convex in x_i on $X_i(x_{-i})$, all $g_{i,j}(x_i, x_{-i})$ ($j \in \mathcal{E}_i$) are affine linear in x_i and all $g_{i,j}(x_i, x_{-i})$ ($j \in \mathcal{I}_i$) are concave in x_i.

For instance, consider the 2-player GNEPP

$$\begin{array}{cl|cl} \min\limits_{x_1 \in \mathbb{R}^3} & \sum\limits_{j=1}^{3} (x_{1,j} - x_{2,j})^2 & \min\limits_{x_2 \in \mathbb{R}^3} & \sum\limits_{j=1}^{3} \left((x_{2,j})^4 - x_{2,j} \prod\limits_{k=1}^{3} x_{1,k} \right) \\ s.t. & x_2^T x_1 - 1 = 0, & s.t. & \|x_1\|^2 - \|x_2\|^2 \geq 0. \\ & (x_{11}, x_{12}, x_{13}) \geq 0; & & \end{array} \tag{1.4}$$

In the above, the $\|\cdot\|$ denotes the Euclidean norm. For each i, the Hessian of f_i with respect to x_i is positive semidefinite for all $x_{-i} \in \mathrm{dom}(X_i)$. All players have convex optimization problems, so this is a convex GNEPP. One can directly check that it has a unique GNE $u = (u_1, u_2)$ with

$$u_1 = \left(\frac{\sqrt[3]{2}}{\sqrt{3}}, \frac{\sqrt[3]{2}}{\sqrt{3}}, \frac{\sqrt[3]{2}}{\sqrt{3}} \right), \; u_2 = \left(\frac{1}{\sqrt[6]{108}}, \frac{1}{\sqrt[6]{108}}, \frac{1}{\sqrt[6]{108}} \right).$$

Generalized Nash equilibrium problems have broad applications, for instance, in the environmental pollution control [19, 34]. Let N be the number of countries involved in the pollution control and $x_{i,0}$ denote the (gross) emissions from the ith country. Assume that the by-product gross emissions are proportional to the industrial output. The revenue of the ith country depends on $x_{i,0}$. Typically, the revenue is $x_{i,0}(b_i - 1/2x_{i,0})$ with a given parameter b_i. The variable $x_{i,j}$ represents the investment from country i to country j. Let $x_i := (x_{i,0}, \dots, x_{i,N})$. For an investor, the benefit of the investment lies in the *emissions reduction units* $\gamma_{i,j} x_{i,j}$ with given parameters $\gamma_{i,j} (i, j = 1, \cdots, N)$. The net emission in country i is $x_{i,0} - \sum_{j=1}^{N} \gamma_{j,i} x_{j,i}$, which is always nonnegative. The accounted-for-emissions for the ith country is $x_{i,0} - \sum_{j=1}^{N} \gamma_{i,j} x_{i,j}$. It must be kept below or equal a certain prescribed level E_i under the environmental control. The pollution in a country may affect other countries. The pollution damage for the ith country is

$$p_i := x_{i,0} - \sum_{j=1}^{N} \gamma_{j,i} x_{j,i} + 2 \prod_{k=1}^{N} (x_{k,0} - \sum_{j=1}^{N} \gamma_{j,k} x_{j,k}).$$

For given parameters $b_i, \gamma_{i,j}, E_i$, the ith country's optimization problem is

$$\begin{cases} \min\limits_{x_i} & -x_{i,0}(b_i - \frac{1}{2} x_{i,0}) + \sum\limits_{j=1}^{N} x_{i,j} + p_i \\ s.t. & x_{i,0} \dots x_{i,j} \geq 0, \\ & x_{i,0} - \sum_{j=1}^{N} \gamma_{i,j} x_{i,j} \leq E_i, \\ & x_{k,0} - \sum_{j=1}^{N} \gamma_{j,k} x_{j,k} \geq 0 \ (k = 1, \dots, N). \end{cases} \tag{1.5}$$

All countries expect to maximize their revenues subtracting investments and pollution damages. Another application of GNEPP is the model for Internet switching (see Example 5.24). More applications for GNEPPs can be found in [3, 21, 22, 104, 128].

1.1 GNEPs and some existing work

The GNEP is an extension of the Nash equilibrium problem (NEP) [82, 83]. For NEPs, the feasible set of each player's strategy is independent of other players. The GNEP originated from economics and was studied in [7, 11, 27, 77, 112]. Robinson [109, 110] established the shadow prices for measuring the effectiveness in an optimization-based combat model. Scotti [121] introduced GNEPs into the study of structural design. Recently, GNEPs have been widely used in many different areas outside economics, such as transportation, telecommunications, pollution control. We refer to [6, 19, 102, 125, 129, 131] for related work.

The following is a classical result about existence of solutions for GNEPs [27, 34]. We refer to [111] for the notion of outer and inner semicontinuity and quasi-convexity.

Theorem 1.2. *[27, 34] Suppose the GNEP of (1.1) satisfies:*

(i) There exist N nonempty, convex and compact sets $K_i \subseteq \mathbb{R}^{n_i}$ such that for every $(x_i, x_{-i}) \in \mathbb{R}^n$ with $x_i \in K_i$ and for every i, the set $X_i(x_{-i})$ is nonempty, closed and convex, $X_i(x_{-i}) \subseteq K_i$, and $X_i(\cdot)$, as a point-to-set map, is both outer and inner semicontinuous.

(ii) For every given x_{-i}, the function $f_i(\cdot, x_{-i})$ is quasi-convex on $X_i(x_{-i})$.

Then, a generalized Nash equilibrium exists.

1.2 Existing work on solving GNEPs

For Nash Equilibrium Problems, when each feasible set X_i is a finite set, the NEP is called a *finite game*. For finite games, Nash Equilibria typically do not exist. People are also interested in mixed strategies, which are probability distributions on the strategy set. Mixed strategy solutions always exist for finite games [83]. For the case that $f_1 + \cdots + f_N = 0$, the NEP is called a *zero-sum game*. NEPs have broad applications in Economics modelling. It is generally hard to solve NEPs [24, 118]. For solving finite games or finding their mixed strategy solutions, we refer to the work [4, 26, 57, 69]. For two-player zero-sum games, the NEPs are equivalent to saddle point problems and there are optimization methods for solving them [20, 101]. More work for solving NEPs can be found in [41, 51, 58, 60, 70, 71, 108, 126]. [10, 76, 113, 123, 127]. Applications outside Economics can be found in [14, 17, 42, 73, 119]. We refer to [5, 13, 25, 80, 103, 112] for more general work on NEPs.

There exists some work for solving GNEPs. Under some convexity assumptions, the GNEP is equivalent to a quasi-variational inequality problem(QVIP) [9, 31, 45, 81, 105]. The Karush-Kuhn-Tucker (KKT) optimality conditions for each player's optimization problem can be used together with the semismooth Newton-type method [28, 29, 32]. A GNEP can be transformed to a NEP with the usage of penalty functions [35, 36, 40, 54]. Gap functions are frequently used for solving GNEPs [53]. A relaxation method for jointly convex GNEPs, based on inexact line search and Nikaido-Isoda functions, is given in [52]. A study on GNEPs with linear coupling constraints and mixed-integer variables is in [117]. Facchinei et al. [37] proposed the Gauss-Seidel method for solving GNEPs. Its main idea is to solve each player's optimization problem alternatively. We also refer to [114, 115] for studies on the Gauss-Seidel method for solving GNEPs with discrete and mixed integer variables. Convergence of the Gauss-Seidel method can be shown for some special GNEPs, such as generalized potential games (GPGs). We refer to [37, 78, 116] for studies on potential games and GPGs. Most of the existing methods assume that each individual player's optimization problem is convex. For more work about GNEPs, we refer to the surveys [34, 39]. It is generally quite difficult to solve GNEPs, even if they are convex. This is because the KKT system of a convex GNEP may still be difficult to solve. The set of GNEs may be nonconvex, even for convex NEPs (see [98]). We refer to [33] for a survey on GNEPs.

1.3 Contribution of this thesis

In this thesis, we study Generalized Nash Equilibrium Problems that are defined by polynomial functions. We formulate efficient polynomial optimization to find the solutions for Nash Equilibrium Problem of Polynomials and Convex Generalized Nash Equilibrium Problems of Polynomials. Moreover, we study the Gauss-Seidel Method for solving the GNEPs of polynomials. The Moment-SOS semidefinite relaxations are used to solve polynomial optimization for finding and verifying GNEs. Our main results are:

- For Nash equilibrium problems that are given by polynomial functions, i.e., the objectives f_i and constraining functions $g_{i,j}$ are polynomial functions in their variables, we formulate efficient polynomial optimization for computing one or more Nash equilibria. Under generic assumptions, we prove the method can find a Nash equilibrium if there exists one. Moreover, we can also find all Nash equilibria if there are finitely many ones of them. The method can also detect nonexistence if there is no Nash equilibrium. For NEPs that are given by generic polynomials, we further show that there are finitely many complex KKT points. Therefore, there are at most finitely many NEs for generic NEPPs. This implies that our method can solve general NEPPs successfully. This is the first method that can guarantee to solve general nonconvex NEPs, to the best of author's knowledge.

- For convex Generalized Nash Equilibrium Problems, we introduce the rational expression for Lagrange multipliers and study their properties. We prove the existence of rational expressions and give a sufficient and necessary condition for positivity of denominators. Moreover, we give parametric expressions for Lagrange multipliers for several cases. For all GNEPs, parametric expressions always exist. Using rational and parametric expressions, we formulate polynomial optimization and propose an algorithm for computing GNEs. Under some general assumptions, we prove that the algorithm can compute a GNE if it exists, or detect nonexistence of GNEs. This is the first numerical method that has these properties, to the best of the authors' knowledge.

- We use the Lasserre type Moment-SOS relaxations [63] to find global minimizers of the occurring polynomial optimization problems in each loop of the Gauss-Seidel method. As demonstrated in section 5.3, the Gauss-Seidel method works well in practice. Moment-SOS relaxations can be used to verify if a computed solution is a GNE or not. There are no other numerical methods for solving GNEPPs efficiently, especially for nonconvex ones, to the best of the authors' knowledge. Moreover, we give a sufficient condition for checking if a given GNEPP is a GPG or not. Based on it, a numerical certificate is given for checking GPGs. This is the first numerical method that can do this, to the best of the authors' knowledge.

Chapter 2

Preliminaries

Notation The symbol $\mathbb{N}$ (resp., $\mathbb{R}$, $\mathbb{C}$) stands for the set of nonnegative integers (resp., real numbers, complex numbers). For a positive integer k, denote the set $[k] := \{1, \dots, k\}$. For a real number t, $\lceil t \rceil$ (resp., $\lfloor t \rfloor$) denotes the smallest integer not smaller than t (resp., the biggest integer not bigger than t). We use e_i to denote the vector such that the ith entry is 1 and all others are zeros. By writing $A \succeq 0$ (resp., $A \succ 0$), we mean that the matrix A is symmetric positive semidefinite (resp., positive definite). For the ith player's strategy vector $x_i \in \mathbb{R}^{n_i}$, the $x_{i,j}$ denotes the jth entry of x_i, for $j = 1, \dots, n_i$. When we write (y, x_{-i}), it means that the ith player's strategy is $y \in \mathbb{R}^{n_i}$, while the vector of all other players' strategy is fixed to be x_{-i}. Let $\mathbb{R}[x]$ denote the ring of polynomials with real coefficients in x, and $\mathbb{R}[x]_d$ denote its subset of polynomials whose degrees are not greater than d. For the ith player's strategy vector x_i, the notation $\mathbb{R}[x_i]$ and $\mathbb{R}[x_i]_d$ are defined in the same way. For ith player's objective $f_i(x)$, the notation $\nabla_{x_i} f_i$, $\nabla^2_{x_i} f_i$ respectively denote its gradient and Hessian with respect to x_i.

In the following, we use the letter z to represent either x, x_i or (x, ω) for some new variables ω, for convenience of discussion. Suppose $z := (z_1, \dots, z_l)$. For a polynomial $p(z) \in \mathbb{R}[z]$, the $p = 0$ means $p(z)$ is identically zero on $\mathbb{R}^l$. We say the polynomial p is nonzero if $p \neq 0$. Let $\alpha := (\alpha_1, \dots, \alpha_l) \in \mathbb{N}^l$, and we denote

$$z^\alpha := z_1^{\alpha_1} \cdots z_l^{\alpha_l}, \quad |\alpha| := \alpha_1 + \dots + \alpha_l.$$

For an integer $d > 0$, denote the monomial power set

$$\mathbb{N}^l_d := \{\alpha \in \mathbb{N}^l : \ |\alpha| \leq d\}.$$

We use $[z]_d$ to denote the vector of all monomials in z whose degree is at most d, ordered in the graded alphabetical ordering. For instance, if $z = (z_1, z_2)$, then

$$[z]_3 = (1, z_1, z_2, z_1^2, z_1 z_2, z_2^2, z_1^3, z_1^2 z_2, z_1 z_2^2, z_2^3).$$

Throughout the paper, a property is said to hold *generically* if it holds for all points in the space of input data except a set of Lebesgue measure zero.

2.1 Ideals and positive polynomials

Let $\mathbb{F} := \mathbb{R}$ or $\mathbb{C}$. For a polynomial $p \in \mathbb{F}[z]$ and subsets $I, J \subseteq \mathbb{F}[z]$, define the product and Minkowski sum

$$p \cdot I := \{pq : \, q \in I\}, \quad I + J := \{a + b : \, a \in I, b \in J\}.$$

The subset I is an ideal if $p \cdot I \subseteq I$ for all $p \in \mathbb{F}[z]$ and $I + I \subseteq I$. For a tuple of polynomials $q = (q_1, \ldots, q_m)$, the set

$$\text{Ideal}[q] := q_1 \cdot \mathbb{F}[z] + \ldots + q_m \cdot \mathbb{F}[z]$$

is the ideal generated by q, which is the smallest ideal containing each q_i.

We review basic concepts in polynomial optimization. A polynomial $\sigma \in \mathbb{R}[z]$ is said to be a sum of squares (SOS) if $\sigma = p_1^2 + \ldots + p_k^2$ for some polynomials $p_i \in \mathbb{R}[z]$. The set of all SOS polynomials in z is denoted as $\Sigma[z]$. For a degree d, we denote the truncation

$$\Sigma[z]_d \, := \, \Sigma[z] \cap \mathbb{R}[z]_d.$$

For a tuple $g = (g_1, \ldots, g_t)$ of polynomials in z, its quadratic module is the set

$$\text{Qmod}[g] \, := \, \Sigma[z] + g_1 \cdot \Sigma[z] + \ldots + g_t \cdot \Sigma[z].$$

Similarly, we denote the truncation of $\text{Qmod}[g]$

$$\text{Qmod}[g]_{2d} \, := \, \Sigma[z]_{2d} + g_1 \cdot \Sigma[z]_{2d-\deg(g_1)} + \ldots + g_t \cdot \Sigma[z]_{2d-\deg(g_t)}.$$

The tuple g determines the basic closed semi-algebraic set

$$\mathcal{S}(g) \, := \, \{z \in \mathbb{R}^l : g_1(z) \geq 0, \ldots, g_t(z) \geq 0\}. \tag{2.1}$$

For a tuple $h = (h_1, \dots, h_s)$ of polynomials in $\mathbb{R}[z]$, its real zero set is

$$\mathcal{Z}(h) := \{z \in \mathbb{R}^l : h_1(z) = \dots = h_s(z) = 0\}.$$

The set $\mathrm{Ideal}[h] + \mathrm{Qmod}[g]$ is said to be *archimedean* if there exists $\rho \in \mathrm{Ideal}[h] + \mathrm{Qmod}[g]$ such that the set $\mathcal{S}(\rho)$ is compact. If $\mathrm{Ideal}[h] + \mathrm{Qmod}[g]$ is archimedean, then $\mathcal{Z}(h) \cap \mathcal{S}(g)$ must be compact. Conversely, if $\mathcal{Z}(h) \cap \mathcal{S}(g)$ is compact, say, $\mathcal{Z}(h) \cap \mathcal{S}(g)$ is contained in the ball $R - \|z\|^2 \geq 0$, then $\mathrm{Ideal}[h] + \mathrm{Qmod}[g, R - \|z\|^2]$ is archimedean and $\mathcal{Z}(h) \cap \mathcal{S}(g) = \mathcal{Z}(h) \cap \mathcal{S}(g, R - \|z\|^2)$. Clearly, if $f \in \mathrm{Ideal}[h] + \mathrm{Qmod}[g]$, then $f \geq 0$ on $\mathcal{Z}(h) \cap \mathcal{S}(g)$. The reverse is not necessarily true. However, when $\mathrm{Ideal}[h] + \mathrm{Qmod}[g]$ is archimedean, if $f > 0$ on $\mathcal{Z}(h) \cap \mathcal{S}(g)$, then $f \in \mathrm{Ideal}[h] + \mathrm{Qmod}[g]$. This conclusion is referenced as Putinar's Positivestellensatz [106]. Interestingly, if $f \geq 0$ on $\mathcal{Z}(h) \cap \mathcal{S}(g)$, we also have $f \in \mathrm{Ideal}[h] + \mathrm{Qmod}[g]$, under some standard optimality conditions [90].

2.2 Localizing and moment matrices

Let $\mathbb{R}^{\mathbb{N}^l_{2d}}$ denote the space of all real vectors that are labeled by $\alpha \in \mathbb{N}^l_{2d}$. A vector $y \in \mathbb{R}^{\mathbb{N}^l_{2d}}$ is labeled as

$$y = (y_\alpha)_{\alpha \in \mathbb{N}^l_{2d}}.$$

Such y is called a *truncated multi-sequence* (tms) of degree $2d$. For a polynomial $f = \sum_{\alpha \in \mathbb{N}^l_{2d}} f_\alpha z^\alpha \in \mathbb{R}[z]_{2d}$, define the operation

$$\langle f, y \rangle := \sum\nolimits_{\alpha \in \mathbb{N}^l_{2d}} f_\alpha y_\alpha. \tag{2.2}$$

The operation $\langle f, y \rangle$ is a bilinear function in (f, y). For a polynomial $q \in \mathbb{R}[z]$, with $\deg(q) \leq 2d$, and the integer $t = d - \lceil \deg(q)/2 \rceil$, the outer product $q \cdot [z]_t([z]_t)^T$ is a symmetric matrix polynomial in z, with length $\binom{n+t}{t}$. We write the expansion as

$$q \cdot [z]_t([z]_t)^T = \sum\nolimits_{\alpha \in \mathbb{N}^l_{2d}} z^\alpha Q_\alpha,$$

for some symmetric matrices Q_α. Then we define the matrix function

$$L_q^{(d)}[y] := \sum\nolimits_{\alpha \in \mathbb{N}^l_{2d}} y_\alpha Q_\alpha. \tag{2.3}$$

It is called the dth *localizing matrix* of q and generated by y. For given q, the matrix $L_q^{(d)}[y]$ is linear in y. Localizing and moment matrices are important for getting semidefinite relaxations of solving polynomial optimization [61, 88, 89]. They are also useful for solving truncated moment problems [38, 92] and tensor decompositions [93, 94]. We refer to [63, 64, 67, 68, 86, 91] for more references about polynomial optimization and moment problems.

2.3 The Moment-SOS hierarchy of semidefinite relaxation for solving polynomial optimization

In this section, we introduce the Moment-SOS hierarchy of semidefinite relaxation for solving polynomial optimization. Consider the following polynomial optimization

$$\begin{cases} \vartheta_{\min} := & \min\limits_{x\in\mathbb{R}^n} & f(x) \\ & s.t. & g_1(x)\ge 0,\ldots,g_{m_1}(x)\ge 0, \\ & & h_1(x)=0,\ldots,h_{m_2}(x)=0. \end{cases} \tag{2.4}$$

Denote the degree

$$d_0 := \max\{\lceil \deg(f)/2\rceil, \lceil \deg(g_1)/2\rceil,\ldots,\lceil \deg(g_{m_1})/2\rceil, \\ \lceil \deg(h_1)/2\rceil,\ldots,\lceil \deg(h_{m_2})/2\rceil\},$$

and we let

$$g=(g_1,\ldots,g_{m_1}),\ \ h=(h_1,\ldots,h_{m_2}).$$

For $d=d_0,d_0+1,\ldots$, the dth moment relaxation for (2.4) is

$$\begin{cases} \vartheta_d := & \min\limits_{y} & \langle f,y\rangle \\ & s.t. & M_d[y]\succeq 0,\ L_{g_1}^{(d)}[y]\succeq 0,\ \ldots\ L_{g_{m_1}}^{(d)}[y]\succeq 0, \\ & & y_0=1,\ L_{h_1}^{(d)}[y]=0,\ \ldots\ L_{h_{m_2}}^{(d)}[y]=0, \\ & & y\in\mathbb{R}^{\mathbb{N}_{2d}^{n_i}}. \end{cases} \tag{2.5}$$

Its dual optimization problem is the SOS relaxation

$$\begin{cases} \max & \gamma \\ s.t. & f-\gamma\in \mathrm{Ideal}[h]_{2d}+\mathrm{Qmod}[g]_{2d}. \end{cases} \tag{2.6}$$

By solving the relaxations (2.5)-(2.6) for $d=d_0,d_0+1,\ldots$, we get the Moment-SOS hierarchy for solving (2.4). The following is the algorithm.

Algorithm 2.1. (The Moment-SOS hierarchy for solving (2.4)). Let f, g, h be as in (2.4). Start with $d := d_0$.

Step 1. Solve the semidefinite relaxation (2.5). If (2.5) is infeasible, then (2.4) has no feasible points and stop; otherwise, solve it for a minimizer y^* and let $t := d_1$, where $d_1 := \max(\max_{i\in[m_1]}\lceil \deg(g_i)/2\rceil, \max_{i\in[m_2]}\lceil \deg(h_i)/2\rceil)$.

Step 2. If y^* satisfies the rank condition

$$\operatorname{rank} M_t[y^*] = \operatorname{rank} M_{t-d_1}[y^*], \tag{2.7}$$

then extract $r := \operatorname{rank} M_t(y^*)$ minimizers for (2.4) and stop.

Step 3. If (2.7) fails to hold and $t < d$, let $t := t + 1$ and then go to Step 2; otherwise, let $d := d + 1$ and go to Step 1.

The rank condition (2.7) is called *flat truncation* in the literature [88]. It is a sufficient (and almost necessary) condition for checking convergence of the Moment-SOS hierarchy. Indeed, the Moment-SOS hierarchy has finite convergence if and only if the flat truncation is satisfied for some relaxation order, under some generic conditions [88]. When (2.7) holds, the method in [48] can be used to extract r minimizers for (2.4). The method is implemented in the software `GloptPoly 3` [49]. We refer to [48], [88] and [63, Chapter 6] for more details.

The convergence properties of Algorithm 2.1 are as follows. By solving the hierarchy of relaxations (2.5)-(2.6), we can get a monotonically increasing sequence of lower bounds $\{\vartheta_d\}_{d=d_0}^{\infty}$ for the minimum value $\vartheta_{\min}$, i.e.,

$$\vartheta_{d_0} \le \vartheta_{d_0+1} \le \cdots \le \vartheta_{\min}.$$

When $\text{Ideal}[h]_{2d} + \text{Qmod}[g]_{2d}$ is archimedean, we have $\vartheta_d \to \vartheta_{\min}$ as $d \to \infty$, as shown in [61]. If $\vartheta_d = \vartheta_{\min}$ for some d, the relaxation (2.5) is said to be exact (or tight) for solving (2.4). For such a case, the Moment-SOS hierarchy is said to have finite convergence. The Moment-SOS hierarchy has finite convergence when the archimedean and some optimality conditions hold [90]. Although there exist special polynomials such that the Moment-SOS hierarchy fails to have finite convergence, such special problems belong to a set of measure zero in the space of input polynomials [90].

2.4 Optimality Conditions for GNEPs

We study optimality conditions for GNEs. Consider the ith player's optimization. For convenience, suppose $\mathcal{E}_i \cup \mathcal{I}_i = [m_i]$ and $g_i = (g_{i,1}, \dots, g_{i,m_i})$. For a given x_{-i}, under some suitable constraint qualifications (e.g., the linear independence constraint qualification (LICQ), Mangasarian-Fromovite constraint qualification (MFCQ), or the Slater's Condition; see [16] for them), if x_i is a minimizer of $\mathrm{F}_i(x_{-i})$, then there exists a Lagrange multiplier vector $\lambda_i := (\lambda_{i,1}, \dots, \lambda_{i,m_i})$ such that

$$\begin{cases} \nabla_{x_i} f_i(x) - \sum_{j=1}^{m_i} \lambda_{i,j} \nabla_{x_i} g_{i,j}(x) = 0, \\ \lambda_i \perp g_i(x),\, g_{i,j}(x) = 0\,(j \in \mathcal{E}_i), \\ \lambda_{i,j} \ge 0\,(j \in \mathcal{I}_i),\, g_{i,j}(x) \ge 0\,(j \in \mathcal{I}_i). \end{cases} \tag{2.8}$$

This is called the first order Karush-Kuhn-Tucker system for $\mathrm{F}_i(x_{-i})$. Such (x_i, λ_i) is called a critical pair of $\mathrm{F}_i(x_{-i})$. Therefore, if x is a GNE, under constraint qualifications, then (3.4) holds for all $i \in [N]$, i.e., there exist Lagrange multiplier vectors $\lambda_1, \dots, \lambda_N$ such that

$$\begin{cases} \nabla_{x_i} f_i(x) - \sum_{j=1}^{m_i} \lambda_{i,j} \nabla_{x_i} g_{i,j}(x) = 0\,(i \in [N]), \\ \lambda_i \perp g_i(x)\,(i \in [N]),\, g_{i,j}(x) = 0\,(i \in [N], j \in \mathcal{E}_i), \\ \lambda_{i,j} \ge 0\,(i \in [N], j \in \mathcal{I}_i),\, g_{i,j}(x) \ge 0\,(i \in [N], j \in \mathcal{I}_i). \end{cases} \tag{2.9}$$

A point x satisfying (2.9) is called a KKT point for the GNEP. For convex GNEPs, each KKT point is a GNE [33, Theorem 4.6]. However, if the GNEP is not convex, then a KKT point may, or may not, be a GNE.

Chapter 3

Nash Equilibrium Problem of Polynomials

The Nash Equilibrium Problem (NEP) is the game such that every player's feasible set is independent with other players' strategies. In a NEP, the ith player's best strategy x_i is a minimizer for the optimization problem

$$\mathbf{F}_i(x_{-i}) : \begin{cases} \min\limits_{x_i \in \mathbb{R}^{n_i}} & f_i(x_i, x_{-i}) \\ s.t. & g_{i,j}(x_i) = 0 \, (j \in \mathcal{E}_i), \\ & g_{i,j}(x_i) \geq 0 \, (j \in \mathcal{I}_i), \end{cases} \tag{3.1}$$

for given x_{-i} of other players' strategies. In the above, f_i is the ith player's objective function, and $g_{i,j}$ are constraining functions in x_i. The $\mathcal{E}_i$ and $\mathcal{I}_i$ are disjoint labeling sets of finite cardinalities (possibly empty). The feasible set of the optimization $\mathbf{F}_i(x_{-i})$ in (3.1) is

$$X_i \;:=\; \{x_i \in \mathbb{R}^{n_i} : \; g_{i,j}(x_i) = 0 \, (j \in \mathcal{E}_i), \, g_{i,j}(x_i) \geq 0 \, (j \in \mathcal{I}_i)\}. \tag{3.2}$$

It is called the feasible strategy set for the ith player. For NEPs, each set X_i does not depend on x_{-i}. This is different from generalized Nash Equilibrium problems (GNEPs), where each player's feasible set depends on other players' strategies. The entire strategy vector x is a feasible point if

$$x = (x_1, \ldots, x_N) \; \in \; X_1 \times \cdots \times X_n,$$

that is, each $x_i \in X_i$. The NEP can be formulated as

$$\text{find} \quad x^* \in \mathbb{R}^n \quad \text{such that each } x_i^* \text{ is a minimizer of} \, \mathbf{F}_i(x_{-i}^*), \tag{3.3}$$

where $x^* = (x_1^*, \dots, x_N^*)$. A solution of (3.3) is called a *Nash Equilibrium* (NE). When all the defining functions f_i and $g_{i,j}$ are polynomials in x, the NEP is then called a Nash Equilibrium Problem of Polynomials (NEPP). The following is an example.

Example 3.1. Consider the 2-player NEP with the individual optimization

$$
\text{1st player:} \begin{cases} \min\limits_{x_1 \in \mathbb{R}^2} & x_{1,1}(x_{1,1} + x_{2,1} + 4x_{2,2}) + 2x_{1,2}^2, \\ \quad s.t. & 1 - (x_{1,1})^2 - (x_{1,2})^2 \geq 0, \end{cases}
$$
$$
\text{2nd player:} \begin{cases} \min\limits_{x_2 \in \mathbb{R}^2} & x_{2,1}(x_{1,1} + 2x_{1,2} + x_{2,1}) + x_{2,2}(2x_{1,1} + x_{1,2} + x_{2,2}), \\ \quad s.t. & 1 - (x_{2,1})^2 - (x_{2,2})^2 \geq 0. \end{cases}
$$

This NEP has only 3 NEs (see Section 3.1.3), which are

$$
\begin{array}{lll}
\text{1st NE:} & x_1^* = (0, 0), & x_2^* = (0, 0); \\
\text{2nd NE:} & x_1^* = (1, 0), & x_2^* = \frac{1}{\sqrt{5}}(-1, -2); \\
\text{3rd NE:} & x_1^* = (-1, 0), & x_2^* = \frac{1}{\sqrt{5}}(1, 2).
\end{array}
$$

It is interesting to note that each player's objective is strictly convex with respect to its strategy, because their Hessian's with respect to their own strategies are positive definite.

In the current state of the art, it is mostly an open question to solve general NEPs efficiently, especially those whose individual optimization problems are nonconvex.

3.1 Polynomial optimization formulations

In this section, we show how to formulate efficient polynomial optimization for solving the NEP (3.3).

3.1.1 Polynomial expresssions for Lagrange Multipliers

Consider the ith player's individual optimization problem $\mathbf{F}_i(x_{-i})$ in (3.1), for given x_{-i}. For convenience, we write the constraining functions as

$$
g_i(x_i) := (g_{i,1}(x_i), \dots, g_{i,m_i}(x_i)).
$$

Suppose $x = (x_1, \dots, x_N)$ is a NE. Under linear independence constraint qualification condition (LICQC) at x_i, i.e., the set of gradients for active constraining functions are linearly independent, there exist Lagrange multipliers $\lambda_{i,j}$ such that

$$\begin{cases} \sum_{j=1}^{m_i} \lambda_{ij} \nabla_{x_i} g_{i,j}(x_i) = \nabla_{x_i} f_i(x), \\ 0 \le \lambda_{i,j} \perp g_{i,j}(x_i) \ge 0 \, (j \in \mathcal{I}_i). \end{cases} \tag{3.4}$$

The above is the Karush-Kuhn-Tucker (KKT) condition for the optimization $\mathbf{F}_i(x_{-i})$. The x satisfying (3.4) is called a KKT point. So x and $\lambda_{i,j}$ satisfy the following polynomial system ($i = 1, \dots, N$)

$$\underbrace{\begin{bmatrix} \nabla_{x_i} g_{i,1} & \nabla_{x_i} g_{i,2} & \cdots & \nabla_{x_i} g_{i,m_i} \\ g_{i,1}(x) & 0 & \cdots & 0 \\ 0 & g_{i,2}(x) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & g_{i,m_i}(x) \end{bmatrix}}_{G_i(x_i)} \underbrace{\begin{bmatrix} \lambda_{i,1} \\ \lambda_{i,2} \\ \vdots \\ \lambda_{i,m_i} \end{bmatrix}}_{\lambda_i} = \underbrace{\begin{bmatrix} \nabla_{x_i} f_i \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\hat{f}_i(x)}. \tag{3.5}$$

Therefore, if there exists a matrix polynomial $H_i(x_i)$ such that

$$H_i(x_i) G_i(x_i) \, = \, I_{m_i}, \tag{3.6}$$

then we can express λ_i as

$$\lambda_i = H_i(x_i) G_i(x_i) \lambda_i = H_i(x_i) \hat{f}_i(x).$$

Interestingly, the matrix polynomial $H_i(x_i)$ satisfying (3.6) exists under the nonsingularity condition on g_i. The polynomial tuple g_i is said to be *nonsingular* if $G_i(x_i)$ has full column rank for all $x_i \in \mathbb{C}^{n_i}$ [95]. It is a generic condition. We remark that if g_i is nonsingular, then the LICQC holds at every minimizer of (3.1), so there must exist $\lambda_{i,j}$ satisfying (3.4) and we can express $\lambda_{i,j}$ as

$$\lambda_{i,j} = \lambda_{i,j}(x) := \big(H_i(x_i) \hat{f}_i(x) \big)_j \tag{3.7}$$

for all NEs.

Throughout the paper, we assume that every constraining polynomial tuple g_i is nonsingular. This is a generic assumption. So $\lambda_{i,j}(x)$ can be expressed as polynomials as in (3.7). Then, each Nash equilibrium x satisfies the following polynomial systems ($i = 1, \dots, N$)

$$(S_i): \begin{cases} \nabla_{x_i} f_i(x) - \sum_{j=1}^{m_i} \lambda_{ij}(x) \nabla_{x_i} g_{i,j}(x_i) = 0, \\ g_{i,j}(x_i) = 0 \, (j \in \mathcal{E}_i), \; \lambda_{i,j}(x) g_{i,j}(x_i) = 0 \, (j \in \mathcal{I}_i), \\ g_{i,j}(x_i) \ge 0 \, (j \in \mathcal{I}_i), \; \lambda_{i,j}(x) \ge 0 \, (j \in \mathcal{I}_i). \end{cases} \tag{3.8}$$

The above are necessary conditions for NEs. When every optimization in (3.1) is convex, the (3.8) are sufficient conditions for NEs.

3.1.2 Optimization based on KKT conditions

The main task for NEP is to find a tuple $x = (x_1, \dots, x_N)$ such that each x_i is a minimizer for the optimization problem $\mathbf{F}_i(x_{-i})$. We assume each constraining tuple g_i is nonsingular. Then x must satisfy the polynomial system (3.8). Choose a generic positive definite matrix

$$\Theta \in \mathbb{R}^{(n+1)\times(n+1)}.$$

Then we consider the following optimization problem

$$\left\{\begin{array}{ll} \min\limits_{x} & [x]_1^T \cdot \Theta \cdot [x]_1 \\ s.t. & \nabla_{x_i} f_i(x) - \sum_{j=1}^{m_i} \lambda_{ij}(x) \nabla_{x_i} g_{i,j}(x_i) = 0 \, (i \in [N]), \\ & g_{i,j}(x_i) = 0 \, (j \in \mathcal{E}_i, i \in [N]), \\ & \lambda_{i,j}(x) g_{i,j}(x_i) = 0 \, (j \in \mathcal{I}_i, i \in [N]), \\ & g_{i,j}(x_i) \geq 0 \, (j \in \mathcal{I}_i, i \in [N]), \\ & \lambda_{i,j}(x) \geq 0 \, (j \in \mathcal{I}_i, i \in [N]). \end{array}\right. \tag{3.9}$$

In the above, the vector $[x]_1 = \begin{bmatrix} 1 & x^T \end{bmatrix}^T \in \mathbb{R}^{n+1}$. Under the nonsingularity assumptions on g_i, every Nash equilibrium x is a feasible point of (3.9), while the converse is typically not true. However, for every feasible point x of (3.9), the x_i is a critical point for the optimization $\mathbf{F}_i(x_{-i})$. It is important to observe that if (3.9) is infeasible, then there are no NEs. If (3.9) is feasible, then it must have a minimizer, because its objective is a positive definite quadratic function. Moreover, for generic Θ, the minimizer of (3.9) is unique.

Assume that $u := (u_1, \dots, u_N)$ is an optimizer of (3.9). If each u_i is a minimizer for the optimization problem $\mathbf{F}_i(u_{-i})$, then u is a NE. To this end, for each player, consider the optimization problem:

$$\left\{\begin{array}{rl} \omega_i := \min & f_i(x_i, u_{-i}) - f_i(u_i, u_{-i}) \\ s.t. & g_{i,j}(x_i) = 0 \, (j \in \mathcal{E}_i), \\ & g_{i,j}(x_i) \geq 0 \, (j \in \mathcal{I}_i). \end{array}\right. \tag{3.10}$$

If all the optimal values $\omega_i \geq 0$, then u is a Nash Equilibrium. However, if one of them is negative, say, $\omega_i < 0$, then u is not a NE. Let U_i be a set of some optimizers of (3.10), then

u violates the following inequalities

$$f_i(x_i, x_{-i}) \le f_i(v, x_{-i}) \quad (v \in U_i). \tag{3.11}$$

However, every Nash equilibrium must satisfy (3.11).

When u is not a NE, we aim at finding a new candidate by posing the inequalities in (3.11). Therefore, we consider the following optimization problem:

$$\begin{cases} \min\limits_{x} & [x]_1^T \cdot \Theta \cdot [x]_1 \\ s.t. & \nabla_{x_i} f_i(x) - \sum_{j=1}^{m_i} \lambda_{ij}(x) \nabla_{x_i} g_{i,j}(x_i) = 0 \, (i \in [N]), \\ & g_{i,j}(x_i) = 0 \, (j \in \mathcal{E}_i, i \in [N]), \\ & \lambda_{i,j}(x) g_{i,j}(x_i) = 0 \, (j \in \mathcal{I}_i, i \in [N]), \\ & g_{i,j}(x_i) \ge 0 \, (j \in \mathcal{I}_i, i \in [N]), \\ & \lambda_{i,j}(x) \ge 0 \, (j \in \mathcal{I}_i, i \in [N]), \\ & f_i(v, x_{-i}) - f_i(x_i, x_{-i}) \ge 0 \, (v \in \mathcal{K}_i, i \in [N]). \end{cases} \tag{3.12}$$

In the above, each $\mathcal{K}_i$ is a set of some optimizers of (3.10). We can solve (3.12) again for an optimizer. If it is verified to be a NE, then we are done. If it is not, we can add more inequalities like (3.11). Repeating this procedure, we get the following algorithm.

Algorithm 3.2. For the NEP given as in (3.1) and (3.3), do the following

Step 0 Initialize $\mathcal{K}_i := \emptyset$ for all i and $l := 0$. Choose a generic positive definite matrix Θ of length $n + 1$.

Step 1 Solve the polynomial optimization problem (3.12). If it is infeasible, there is no NE and stop; otherwise, solve it for an optimizer u.

Step 2 For each $i = 1, \ldots, N$, solve the optimization (3.10). If all $\omega_i \ge 0$, u is a NE and stop. If one of ω_i is negative, go to the next step.

Step 3 For each i with $\omega_i < 0$, obtain a set U_i of some (may not all) optimizers of (3.10); then update the set $\mathcal{K}_i := \mathcal{K}_i \cup U_i$. Let $l := l + 1$, then go to Step 1.

In the Step 0, we can set $\Theta = R^T R$ for a randomly generated matrix R of length $n+1$. The objective in (3.12) is a positive definite quadratic function, so it has a minimizer if (3.12) is feasible. The case is slightly different for (3.10). If the feasible set X_i is compact or $f_i(x_i, u_{-i})$ is coercive for the given u_{-i}, then (3.10) has a minimizer. If X_i is unbounded and $f_i(x_i, u_{-i})$ is not coercive, it may be difficult for computing the optimal value ω_i. In applications, we are mostly interested in cases that (3.10) has a minimizer, for the existence of a NE. In Section 3.3, we will discuss how to solve the optimization problems in Algorithm 3.2, by the Lasserre type Moment-SOS hierarchy of semidefinite relaxations.

The following is the convergence theorem.

Theorem 3.3. *Assume each constraining polynomial tuple g_i is nonsingular and let $\lambda_{i,j}(x)$ be Lagrange multiplier polynomials as in (3.7). Let $\mathcal{G}$ be the feasible set of (3.9) and $\mathcal{G}^*$ be the set of all NEs. If the complement $\mathcal{G}\backslash\mathcal{G}^*$ is a finite set, i.e., the cardinality $\ell := |\mathcal{G}\backslash\mathcal{G}^*| < \infty$, then Algorithm 3.2 must terminate within at most ℓ loops.*

Proof. Under the nonsingularity assumption of polynomial tuples g_i, the Lagrange multipliers $\lambda_{i,j}$ can be expressed as polynomials $\lambda_{i,j}(x)$ as in (3.7). For each u that is a feasible point of (3.9), every NE must satisfy the constraint

$$f_i(u_i, x_{-i}) - f_i(x_i, x_{-i}) \geq 0.$$

Therefore, every NE must also be a feasible point of (3.12). Since the matrix Θ is positive definite, the optimization (3.12) must have a minimizer, unless it is infeasible. When Algorithm 3.2 goes to a newer loop, say, from the lth to the $(l+1)$th, the optimizer u for (3.12) in the lth loop is no longer feasible for (3.12) in the $(l+1)$th loop. This means that the feasible set of (3.12) must loose at least one point after each loop, unless a NE is reached. Also note that the feasible set of (3.12) is contained in $\mathcal{G}$. If $\mathcal{G}\backslash\mathcal{G}^*$ is a finite set, Algorithm 3.2 must terminate after some loops. The number of loops is at most ℓ. □

When the polynomials $f_i, g_{i,j}$ are generic, the NEP (3.3) has finitely many KKT points, i.e., the feasible set of (3.9) is finite. This is shown in Theorem 3.10 in the Appendix. The assumptions in Theorem 3.3 are general.

3.1.3 Convex NEPs

An important class of NEPs is that each individual optimization $\mathbf{F}_i(x_{-i})$ is a convex optimization problem, i.e., each objective $f_i(x_i, x_{-i})$ is a convex function in x_i for given x_{-i}, the equality constraining function $g_{i,j}(x_i)$ $(j \in \mathcal{E}_i)$ is linear in x_i, and the inequality constraining function $g_{i,j}(x_i)$ $(j \in \mathcal{I}_i)$ is concave in x_i. This requires that each player's strategy set X_i is convex and

$$f_i(\theta a + (1-\theta)b, x_{-i}) \leq \theta f_i(a, x_{-i}) + (1-\theta) f_i(b, x_{-i}), \tag{3.13}$$

for all $a, b \in X_i$, $\theta \in [0,1]$, and for given x_{-i}.

For convex optimization, the optimizers are equivalent to the KKT points, under constraint qualification conditions (e.g., the Slater's condition or the LICQC). In particular, when the constraining polynomial is nonsingular, a point is a minimizer if and only if it is a KKT point, which means that every feasible point of (3.9) is a NE. Therefore, we get the following corollary.

Corollary 3.4. *Assume each g_i is a nonsingular tuple of polynomials. Suppose each $g_{i,j}(x_i)$ ($j \in \mathcal{E}_i$) is linear, each $g_{i,j}(x_i)$ ($j \in \mathcal{I}_i$) is concave, and each $f_i(x_i, x_{-i})$ is convex in x_i for given x_{-i}. Then Algorithm 3.2 must terminate at the first loop with $l = 0$, returning a NE or reporting that there is no NE.*

For convex optimization problems, there are infinitely many optimizers unless the optimizer is unique. Moreover, if the objective is strictly convex (i.e., the inequality (3.13) holds strictly for all $a \neq b$, $0 < \theta < 1$), the optimizer is always unique, if it exists. However, these conclusions are not true for convex NEPs. Even if each player's objective $f_i(x_i, x_{-i})$ is strictly convex in x_i for all given x_{-i}, the NEP might have finitely many NEs. This is the case for Example 3.1.

Example 3.5. Consider the 2-player NEP in Example 3.1. Each individual optimization is strictly convex, because the Hessian's $\nabla^2_{x_1} f_1$ and $\nabla^2_{x_2} f_2$ are positive definite. The constraints are the convex ball conditions. The KKT system is

$$\begin{cases} 2x_{1,1} + x_{2,1} + 4x_{2,2} = -2\lambda_1 x_{1,1}, 4x_{1,2} = -2\lambda_1 x_{1,2}, \\ x_{1,1} + 2x_{1,2} + 2x_{2,1} = -2\lambda_2 x_{2,1}, 2x_{1,1} + x_{1,2} + 2x_{2,2} = -2\lambda_2 x_{2,2}, \\ \lambda_1(1 - (x_{1,1})^2 - (x_{1,2})^2) = 0, \lambda_2(1 - (x_{2,1})^2 - (x_{2,2})^2) = 0, \\ 1 - (x_{1,1})^2 - (x_{1,2})^2 \geq 0, 1 - (x_{2,1})^2 - (x_{2,2})^2 \geq 0, \\ \lambda_1 \geq 0, \lambda_2 \geq 0. \end{cases} \tag{3.14}$$

By solving the above directly, one can show that this NEP has only 3 NEs, together with Lagrange multipliers as follows.

$$\begin{array}{lll} x_1^* = (0,0), & x_2^* = (0,0), & \lambda_1^* = \lambda_2^* = 0; \\ x_1^* = (1,0), & x_2^* = \frac{1}{\sqrt{5}}(-1,-2), & \lambda_1^* = \frac{9\sqrt{5}}{10} - 1, \ \lambda_2^* = \frac{\sqrt{5}}{2} - 1; \\ x_1^* = (-1,0), & x_2^* = \frac{1}{\sqrt{5}}(1,2), & \lambda_1^* = \frac{9\sqrt{5}}{10} - 1, \ \lambda_2^* = \frac{\sqrt{5}}{2} - 1. \end{array}$$

3.1.4 More Nash Equilibria

Algorithm 3.2 aims at finding a single NE. In some applications, people may be interested in more NEs. For the case that there is a unique NE, people are also interested in a certificate for the uniqueness. Here, we give a procedure for computing more NEs or verifying the uniqueness.

Assume that x^* is a Nash Equilibrium produced by Algorithm 3.2, i.e., x^* is also a minimizer of (3.12). Note that all KKT points $\overline{x}$ satisfying

$$[\overline{x}]_1^T\Theta[\overline{x}]_1 \ < \ [x^*]_1^T\Theta[x^*]_1$$

are excluded from the feasible set of (3.12) by the constraints

$$f_i(u_i, x_{-i}) - f_i(x_i, x_{-i}) \ \geq \ 0 \quad (\forall\, u \in \mathcal{K}_i,\, \forall\, i \in [N]).$$

If x^* is an isolated NE (e.g., this is the case if there are finitely many NEs), there exists a scalar $\delta > 0$ such that

$$[x]_1^T\Theta[x]_1 \ \geq \ [x^*]_1^T\Theta[x^*]_1 + \delta$$

for all other NEs x. For such δ, we can try to find a different NE by solving the following optimization problem

$$\left\{\begin{array}{ll}\min\limits_{x} & [x]_1^T\Theta[x]_1 \\ s.t. & \nabla_{x_i} f_i(x) - \sum_{j=1}^{m_i} \lambda_{ij}(x)\nabla_{x_i} g_{i,j}(x_i) = 0\,(i \in [N]), \\ & g_{i,j}(x_i) = 0\,(j \in \mathcal{E}_i, i \in [N]), \\ & \lambda_{i,j}(x) g_{i,j}(x_i) = 0\,(j \in \mathcal{I}_i, i \in [N]), \\ & g_{i,j}(x_i) \geq 0\,(j \in \mathcal{I}_i, i \in [N]), \\ & \lambda_{i,j}(x) \geq 0\,(j \in \mathcal{I}_i, i \in [N]), \\ & f_i(v, x_{-i}) - f_i(x_i, x_{-i}) \geq 0\ (v \in \mathcal{K}_i, i \in [N]), \\ & [x]_1^T\Theta[x]_1 \ \geq \ [x^*]_1^T\Theta[x^*]_1 + \delta. \end{array}\right. \tag{3.15}$$

When an optimizer of (3.15) is computed, we can check if it is a NE or not by solving (3.10). If it is, we get a new NE that is different from x^*. If it is not, we can union new points to the set $\mathcal{K}_i$. Repeating the above process, we are able to get more Nash equilibria.

A concern in computation is how to choose the constant $\delta > 0$ for (3.15). We want a value $\delta > 0$ such that there is no other Nash Equilibrum u such that $[u]_1^T\Theta[u]_1 \leq [x^*]_1^T\Theta[x^*]_1 + \delta$. To this end, we consider the following maximization problem

$$
\begin{cases}
\max\limits_{x} & [x]_1^T \Theta [x]_1 \\
s.t. & \nabla_{x_i} f_i(x) - \sum_{j=1}^{m_i} \lambda_{ij}(x) \nabla_{x_i} g_{i,j}(x_i) = 0 \, (i \in [N]), \\
& g_{i,j}(x_i) = 0 \, (j \in \mathcal{E}_i, i \in [N]), \\
& \lambda_{i,j}(x) g_{i,j}(x_i) = 0 \, (j \in \mathcal{I}_i, i \in [N]), \\
& g_{i,j}(x_i) \geq 0 \, (j \in \mathcal{I}_i, i \in [N]), \\
& \lambda_{i,j}(x) \geq 0 \, (j \in \mathcal{I}_i, i \in [N]), \\
& f_i(v, x_{-i}) - f_i(x_i, x_{-i}) \geq 0 \; (v \in \mathcal{K}_i, i \in [N]), \\
& [x]_1^T \Theta [x]_1 \; \leq \; [x^*]_1^T \Theta [x^*]_1 + \delta.
\end{cases} \tag{3.16}
$$

Interestingly, if x^* is also a maximizer of (3.16), then the feasible set of (3.15) contains all NEs except x^*, under some general assumptions.

Proposition 3.6. *Assume Θ is a generic positive definite matrix and x^* is a minimizer of (3.12).*

(i) If x^* is also a maximizer of (3.16), then there is no other Nash Equilibrium u satisfying $[u]_1^T \Theta [u]_1 \; \leq \; [x^*]_1^T \Theta [x^*]_1 + \delta$.

(ii) If x^* is an isolated KKT point, then there exists $\delta > 0$ such that x^* is also a maximizer of (3.16).

Proof. Note that every NE is a feasible point of (3.12).

(i) If x^* is also a maximizer of (3.16), then the objective $[x]_1^T \Theta [x]_1$ achieves a constant value in the following set of (3.16). If u is a Nash equilibrium with $[u]_1^T \Theta [u]_1 \; \leq \; [x^*]_1^T \Theta [x^*]_1 + \delta$, then

$$
[u]_1^T \Theta [u]_1 = [x^*]_1^T \Theta [x^*]_1.
$$

This means that u is also a minimizer of (3.12). When Θ is a generic positive definite matrix, the optimization (3.12) has a unique optimizer, so $u = x^*$.

(ii) Since Θ is positive definite, there exists $\epsilon > 0$ such that

$$
[x]_1^T \Theta [x]_1 \; \geq \; \epsilon (1 + \|x\|)^2
$$

for all x. Let $C = \sqrt{\big([x^*]_1^T\Theta[x^*]_1\big)/\epsilon}$, then the following set

$$T := \left\{ y = [x]_2 \;\middle|\; \begin{array}{c} \nabla_{x_i} f_i(x) - \sum_{j=1}^{m_i} \lambda_{ij}(x)\nabla_{x_i} g_{i,j}(x_i) = 0 \, (i \in [N]), \\ g_{i,j}(x_i) = 0 \, (j \in \mathcal{E}_i, i \in [N]), \\ \lambda_{i,j}(x) g_{i,j}(x_i) = 0 \, (j \in \mathcal{I}_i, i \in [N]), \\ g_{i,j}(x_i) \geq 0 \, (j \in \mathcal{I}_i, i \in [N]), \\ \lambda_{i,j}(x) \geq 0 \, (j \in \mathcal{I}_i, i \in [N]), \\ f_i(v, x_{-i}) - f_i(x_i, x_{-i}) \geq 0 \;\, (v \in \mathcal{K}_i, i \in [N]), \\ \|x\| \leq C \end{array} \right\}.$$

is compact. Note that $[x^*]_2 \in T$. Let θ be the vector such that

$$[x]_1^T\Theta[x]_1 \;=\; \theta^T y$$

for all $y = [x]_2$. Since x^* is an isolated KKT point, the $y^* := [x^*]_2$ is also an isolated point of T. Then its subset

$$T_1 \;:=\; T\backslash\{y^*\}$$

is also a compact set. Since x^* is a minimizer of (3.12), the hyperplane $H := \{\theta^T y = \theta^T y^*\}$ is a supporting hyperplane for the set T. Since Θ is generic, the optimization (3.12) has a unique minimizer, which implies that y^* is the unique minimizer of the linear function $\theta^T y$ on T. So, H does not intersect T_1 and their distance is positive. There exists a scalar $\tau > 0$ such that

$$[x]_1^T\Theta[x]_1 \;=\; \theta^T y \;\geq\; \theta^T y^* + \tau \;=\; [x^*]_1^T\Theta[x^*]_1 + \tau$$

for all $y = [x]_2 \in T_1$. Then, for the choice $\delta := \tau/2$, the point x^* is the only feasible point for (3.16). Hence, x^* is also a maximizer of (3.16). □

Proposition 3.6 shows the existence of $\delta > 0$ such that (3.12) and (3.16) have the same optimal value. However, it does not give a concrete lower bound for δ. In computational practice, we can first give a priori value for δ. If it does not work, we can decrease δ to a smaller value (e.g., let $\delta := \delta/5$). By repeating this, the optimization (3.16) will eventually have x^* as a maximizer. The following is the algorithm for an NE that is different from x^*.

Algorithm 3.7. Give an initial value for δ (say, 0.1).

Step 1 Solve the maximization problem (3.16). If its optimal value η equals $\upsilon := [x^*]_1^T\Theta[x^*]_1$, then go to Step 2. If η is bigger than υ, then let $\delta = \min(\delta/5, \eta - \upsilon)$ and repeat this step.

Step 2 Solve the optimization problem (3.15). If it is infeasible, then there are no additional NEs; if it is feasible, solve it for a minimizer u.

Step 3 For each $i = 1, \dots, N$, solve the optimization (3.10) for the optimal value ω_i. If all $\omega_i \geq 0$, stop and u is a NE. If one of ω_i is negative, go to Step 4.

Step 4 For each $i \in [N]$, update the set $\mathcal{K}_i := \mathcal{K}_i \cup U_i$, and then go back to Step 2.

When x^* is not an isolated KKT point, there may not exist a satisfactory $\delta > 0$ for the Step 1. For such a case, more investigation is required to verify uniqueness of the NE or to find other NEs. However, for generic NEPs, there are finitely many KKT points (see Theorem 3.10 in the appendix). The following is the convergence result for Algorithm 3.7.

Theorem 3.8. *Under the same assumptions in Theorem 3.3, if Θ is a generic positive definite matrix and x^* is an isolated KKT point, then Algorithm 3.7 must terminate after finitely many steps, returning a NE that is different from x^* or reporting the nonexistence of other NEs.*

Proof. Under the given assumptions, Proposition 3.6(ii) shows the existence of $\delta > 0$ satisfactory for the Step 1 of Algorithm 3.7. Again, by Proposition 3.6(i), the feasible set of (3.15) contains all NEs except x^*. The finite termination of Algorithm 3.7 can be proved in the same way as for Theorem 3.3. □

Once a new NE is obtained, we can repeatedly apply Algorithm 3.7, to compute more NEs, if they exist. In particular, if there are finitely many NEs, we can eventually get all of them. Indeed, for generic NEPPs, the number of NEs is finite (see Theorem 3.10 in the appendix). We can assume the set of equilibria is

$$\{x^{(1)}, \dots, x^{(s)}\}.$$

Without loss of generality, we can assume they are ordered as

$$[x^{(1)}]_1^T \Theta [x^{(1)}]_1 < \cdots < [x^{(s)}]_1^T \Theta [x^{(s)}]_1,$$

since Θ is generic. If the first r NEs, say, $x^{(1)}, \dots, x^{(r)}$, are obtained, there exists $\delta > 0$ such that

$$[x^{(j)}]_1^T \Theta [x^{(j)}]_1 > [x^{(r)}]_1^T \Theta [x^{(r)}]_1 + \delta$$

for all $j = r+1, \dots, s$. Therefore, if we apply Algorithm 3.7 with $x^* = x^{(r)}$, the next Nash equilibrium $x^{(r+1)}$ can be obtained, if it exists. Therefore, we have the following conclusion.

Corollary 3.9. *Under the assumptions of Theorem 3.8, if there are finitely many Nash equilibria, then all of them can be found by applying Algorithm 3.7 repeatedly.*

3.2 On the finiteness of KKT points for generic NEPPs

We discuss the finiteness of KKT points for generic NEPPs. This implies that Algorithm 3.2 and 3.7 has finite convergence. After enumeration of all possibilities of active inequality constraints, we can generally consider the case that (3.1) has only equality constraints. Consequently, the length m_i of the ith player's constraining polynomials can be assumed less than or equal to n_i, the dimension of its strategy x_i. To prove the finiteness, we can ignore the sign conditions $\lambda_{i,j} \geq 0$ for Lagrange multipliers. The KKT system for all players is

$$\begin{cases} \sum_{j=1}^{m_i} \lambda_{ij} \nabla_{x_i} g_{i,j}(x_i) &= \nabla_{x_i} f_i(x) \quad (i \in [N]), \\ g_{i,j}(x_i) &= 0 \qquad\quad (i \in [N], j \in [m_i]). \end{cases} \tag{3.17}$$

When the objectives f_i are generic polynomials in x and each $g_{i,j}$ is a generic polynomial in x_i, we show that (3.17) has finitely many complex solutions.

Theorem 3.10. *Let $d_{i,j} > 0$, $a_{i,j} > 0$ be degrees, for $i \in [N], j \in [m_i]$. If each $g_{i,j}$ is a generic polynomial in x_i of degree $d_{i,j}$ and each f_i is a generic polynomial in x and its degree in x_j is $a_{i,j}$, then the KKT system (3.17) has finitely many complex solutions and hence the NEP has finitely many KKT points.*

Proof. For each player $i = 1, \ldots, N$, denote

$$b_i := a_{i,i} - 1 + d_{i,1} + \cdots + d_{i,m_i} - m_i.$$

$$\tilde{x}_i := (x_{i,0}, x_{i,1}, \ldots, x_{i,n_i}), \quad \tilde{x} := (\tilde{x}_1, \ldots, \tilde{x}_N).$$

The homogenization of $g_{i,j}$ is $\tilde{g}_{i,j}$, a form in $\tilde{x}_i$. Let $\mathbb{P}^{n_i}$ be the n_i dimensional projective space, over the complex field. Consider the projective varieties

$$\mathcal{U}_i := \Big\{ (\tilde{x}_1, \ldots, \tilde{x}_N) \in \mathbb{P}^{n_1} \times \cdots \times \mathbb{P}^{n_N} : \tilde{g}_i(\tilde{x}_i) = 0 \Big\}, i = 1, \ldots, N,$$

$$\mathcal{U} := \mathcal{U}_1 \cap \cdots \cap \mathcal{U}_N.$$

When $g_{i,j}$ are generic polynomials in x_i, the codimension of $\mathcal{U}_i$ is m_i (see [46]), so $\mathcal{U}$ has the codimension $m_1 + \cdots + m_N$.

The ith player's objective f_i is a polynomial in $x = (x_1, \dots, x_N)$, we denote the multi-homogenization of $f_i(x_i, x_{-i})$ as

$$\tilde{f}_i(\tilde{x}_i, \tilde{x}_{-i}) := f_i(x_1/x_{1,0}, \dots, x_N/x_{N,0}) \cdot (\prod_{j=1}^{N}(x_{i,0})^{a_{i,j}}).$$

It is a multi-homogeneous polynomial in $\tilde{x}$. For each i, consider the determinantal variety

$$W_i := \left\{ x \in \mathbb{C}^n \,\middle|\, \operatorname{rank}[\nabla_{x_i} f_i(x) \quad \nabla_{x_i} g_{i,1}(x_i) \quad \cdots \quad \nabla_{x_i} g_{i,m_i}(x_i)] \le m_i \right\}.$$

(The ∇_{x_i} denote the gradient with respect to x_i.) Its multi-homogenization is

$$\widetilde{W}_i := \left\{ \tilde{x} \,\middle|\, \operatorname{rank}[\nabla_{x_i} \tilde{f}_i(\tilde{x}) \quad \nabla_{x_i} \tilde{g}_{i,1}(\tilde{x}_i) \quad \cdots \quad \nabla_{x_i} \tilde{g}_{i,m_i}(\tilde{x}_i)] \le m_i \right\}.$$

The matrix in the above can be explicitly written as

$$J_i(\tilde{x}_i, \tilde{x}_{-i}) := \begin{bmatrix} \partial_{x_{i,1}} \tilde{f}_i(\tilde{x}) & \partial_{x_{i,1}} \tilde{g}_{i,1}(\tilde{x}_i) & \cdots & \partial_{x_{i,1}} \tilde{g}_{i,m_i}(\tilde{x}_i) \\ \partial_{x_{i,2}} \tilde{f}_i(\tilde{x}) & \partial_{x_{i,2}} \tilde{g}_{i,1}(\tilde{x}_i) & \cdots & \partial_{x_{i,2}} \tilde{g}_{i,m_i}(\tilde{x}_i) \\ \vdots & \vdots & \ddots & \vdots \\ \partial_{x_{i,n_i}} \tilde{f}_i(\tilde{x}) & \partial_{x_{i,n_i}} \tilde{g}_{i,1}(\tilde{x}_i) & \cdots & \partial_{x_{i,n_i}} \tilde{g}_{i,m_i}(\tilde{x}_i) \end{bmatrix}.$$

The (m_i+1)-by-(m_i+1) minors of the matrix J_i are homogeneous in $\tilde{x}_i$ of degree b_i. They are homogeneous in $\tilde{x}_j$ of degree $a_{i,j}$, for $j \neq i$. By [87, Proposition 2.1], when $g_{i,j}$ are generic polynomials in x_i, the right m_i columns of J_i are linearly independent for all $\tilde{x}_i \in \mathcal{U}_i$. That is, for every $\tilde{x} \in \mathcal{U}_i$, there must exist a nonzero m_i-by-m_i minor from the right m_i columns of J_i. In the following, we consider fixed generic polynomials $g_{i,j}$.

First, we show that $\mathcal{U} \cap \widetilde{W}_1$ have the codimension $n_1 + m_2 + \cdots + m_N$. Let $\mathcal{V}$ be the projective variety consisting of all equivalent classes of the vectors

$$\mathfrak{m}_1(\tilde{x}) := [\tilde{x}_1]^{hom}_{b_1} \otimes [\tilde{x}_2]^{hom}_{a_{1,2}} \otimes \cdots \otimes [\tilde{x}_N]^{hom}_{a_{1,N}}, \tag{3.18}$$

for equivalent classes of $\tilde{x} \in \mathcal{U}$. In the above, $\otimes$ denotes the Kronecker product, $[u]^{hom}_d$ denotes the vector of all monomials in u of degree equal to d. In other words, $[u]^{hom}_d$ is the subvector of of $[u]_d$ for monomials of the highest degree d. Note that $\mathcal{U}$ is birational to $\mathcal{V}$ (consider the natural embedding $\varphi : \mathcal{U} \hookrightarrow \mathcal{V}$ such that $\phi(\tilde{x}) = \mathfrak{m}_1(\tilde{x})$). So $\mathcal{U}$ and $\mathcal{V}$ have the same codimension [122]. For each subset $I \subseteq [n_1]$ of cardinality m_1, we use $\det_I J_1$ to denote the m_1-by-m_1 minor of J_1 for the submatrix whose row indices are in I and whose columns are the right hand side m_1 columns. Then

$$\widetilde{W}_1 \quad = \bigcup_{I \subseteq [n_1], |I| = m_i} \mathcal{X}_I \quad \text{where}$$

$$\mathcal{X}_I := \{\tilde{x} : \operatorname{rank} J_1(x) \leq m_1,\, \det_I J_1(x) \neq 0\}.$$

For each I, we have $\tilde{x} \in \mathcal{X}_I$ if and only if the (m_1+1)-by-(m_1+1) minors of J_1, corresponding to the row indices $I \cup \{\ell\}$ with $\ell \in [n_1] \backslash I$, are equal to zeros. There are totally $n_1 - m_1$ such minors. The vanishing of these (m_1+1)-by-(m_1+1) minors of J_1 gives $n_1 - m_1$ linear equations in the vector $\mathfrak{m}_1(\tilde{x})$ as in (3.18). The coefficients of these linear equations are linearly parameterized by coefficients of f_1. Therefore, when f_1 has generic coefficients, the set

$$\mathcal{Y}_I := \{\mathfrak{m}_1(\tilde{x}) : \tilde{x} \in \mathcal{X}_I \cap \mathcal{U}\}$$

is the intersection of $\mathcal{V}$ with $n_1 - m_1$ generic linear equations. Since $\mathcal{X}_I \cap \mathcal{U}$ is birational to $\mathcal{Y}_I$, they have the same codimension, so the codimension of $\mathcal{X}_I \cap \mathcal{U}$ is $n_1 + m_2 + \cdots + m_N$. This conclusion is true for all the above subsets I. Since

$$\mathcal{U} \cap \widetilde{W}_1 \quad = \bigcup_{I \subseteq [n_1], |I| = m_1} \mathcal{X}_I \cap \mathcal{U},$$

the codimension of $\mathcal{U} \cap \widetilde{W}_1$ is equal to $n_1 + m_2 + \cdots + m_N$.

Second, we can repeat the above argument to show that

$$(\mathcal{U} \cap \widetilde{W}_1) \cap \widetilde{W}_2$$

has codimension $n_1 + n_2 + m_3 + \cdots + m_N$. Let $\mathcal{V}'$ be the projective variety consisting of all equivalent classes of the vectors

$$\mathfrak{m}_2(\tilde{x}) \ := \ [\tilde{x}_1]^{hom}_{a_{2,1}} \otimes [\tilde{x}_2]^{hom}_{b_2} \otimes [\tilde{x}_3]^{hom}_{a_{2,3}} \otimes \cdots \otimes [\tilde{x}_N]^{hom}_{a_{2,N}} \tag{3.19}$$

for equivalent classes of $\tilde{x} \in \mathcal{U} \cap \widetilde{W}_1$. Note that $\mathcal{U} \cap \widetilde{W}_1$ is birational to $\mathcal{V}'$. They have the same codimension. Similarly, we have

$$\widetilde{W}_2 \quad = \bigcup_{I \subseteq [n_2], |I| = m_2} \mathcal{X}'_I \quad \text{where}$$

$$\mathcal{X}'_I \ := \ \{\tilde{x} : \operatorname{rank} J_2(x) \leq m_2,\, \det_I J_2(x) \neq 0\}.$$

When f_2 has generic coefficients, the set

$$\mathcal{Y}'_I := \{\mathfrak{m}_2(\tilde{x}) : \tilde{x} \in \mathcal{X}'_I \cap \mathcal{U} \cap \widetilde{W}_1\}$$

is the intersection of $\mathcal{V}'$ with $n_2 - m_2$ generic hyperplanes. Since $\mathcal{X}'_I \cap \mathcal{U} \cap \widetilde{W}_1$ is birational to $\mathcal{Y}'_I$, they have the same dimension, so the codimension of $\mathcal{X}'_I \cap \mathcal{U} \cap \widetilde{W}_1$ is $n_1 + n_2 + m_3 + \cdots + m_N$. This conclusion is true for all $\mathcal{Y}'_I$. Since

$$\mathcal{U} \cap \widetilde{W}_1 \cap \widetilde{W}_2 \quad = \bigcup_{I \subseteq [n_2], |I| = m_2} \mathcal{X}'_I \cap \mathcal{U} \cap \widetilde{W}_1,$$

we know $\mathcal{U} \cap \widetilde{W}_1 \cap \widetilde{W}_2$ has the codimension $n_1 + n_2 + m_3 + \cdots + m_N$.

Similarly, by repeating the above, we can eventually show that

$$\mathcal{U} \cap \widetilde{W}_1 \cap \widetilde{W}_2 \cap \cdots \cap \widetilde{W}_N$$

has codimension $n_1 + n_2 + \cdots + n_N$. This implies the KKT system (3.17) has codimension $n_1 + n_2 + \cdots + n_N$, i.e., the dimension of the solution set of (3.17) is zero. So, there are finitely many complex KKT points. □

3.3 The Moment-SOS hierarchy for solving optimization

In this section, we discuss how to solve the polynomial optimization problems in Algorithms 3.2 and 3.7 by using the Lasserre type Moment-SOS hierarchy of semidefinite relaxations. We assume the constraining polynomial tuples g_i are all nonsingular. Therefore, the Lagrange multipliers $\lambda_{i,j}$ can be expressed as polynomial functions $\lambda_{i,j}(x)$ as in (3.7) for all Nash equilibria. Note that every Nash equilibrium x^* must satisfy the polynomial system (3.8).

3.3.1 The optimization for all players

We discuss how to solve the polynomial optimization problems (3.12), (3.15) and (3.16), by using the Moment-SOS hierarchy of semidefinite programming relaxations [61, 63, 64, 67, 68].

First, we discuss how to solve (3.15). Suppose the set $\mathcal{K}_i$ is given, for each player.

For notational convenience, denote the polynomial tuples

$$\Phi_i := \Big\{\nabla_{x_i} f_i(x) - \sum_{j=1}^{m_i} \lambda_{ij}(x) \nabla_{x_i} g_{ij}\Big\} \cup \Big\{g_{i,j} : j \in \mathcal{E}_i\Big\}$$
$$\cup \Big\{\lambda_{i,j}(x) \cdot g_{i,j} : j \in \mathcal{I}_i\Big\}, \quad (3.20)$$

$$\Psi_i := \Big\{g_{i,j} : j \in \mathcal{I}_i\Big\} \cup \Big\{\lambda_{i,j}(x) : j \in \mathcal{I}_i\Big\}$$
$$\cup \Big\{f_i(v, x_{-i}) - f_i(x_i, x_{-i}) : \ v \in \mathcal{K}_i\Big\}. \quad (3.21)$$

In the above, for a vector $p = (p_1, \ldots, p_s)$ of polynomials, the set $\{p\}$ stands for $\{p_1, \ldots, p_s\}$, for notational convenience. Denote the unions

$$\Phi \ := \bigcup_{i=1}^{N} \Phi_i, \quad \Psi \ := \bigcup_{i=1}^{N} \Psi_i. \tag{3.22}$$

They are both finite sets of polynomials. Then, the optimization (3.12) can be equivalently written as

$$\left\{\begin{array}{rl} \vartheta_{\min} := \min\limits_{x \in \mathbb{R}^n} & \theta(x) := [x]_1^T \Theta [x]_1 \\ s.t. & p(x) = 0 \ (\forall\, p \in \Phi), \\ & q(x) \geq 0 \ (\forall\, q \in \Psi). \end{array}\right. \tag{3.23}$$

Denote the degree

$$d_0 \ := \ \max\{\lceil \deg(p)/2 \rceil : \ p \in \Phi \cup \Psi\}.$$

For a degree $k \geq d_0$, consider the the kth order moment relaxation for (3.23)

$$\left\{\begin{array}{rl} \vartheta_k \ := \ \min\limits_{y} & \langle \theta, y \rangle \\ s.t. & y_0 = 1, \ L_p^{(k)}[y] = 0 \, (p \in \Phi), \\ & M_d[y] \succeq 0, \ L_q^{(k)}[y] \succeq 0 \, (q \in \Psi), \\ & y \in \mathbb{R}^{\mathbb{N}_{2k}^n}. \end{array}\right. \tag{3.24}$$

Its dual optimization problem is the kth order SOS relaxation

$$\left\{\begin{array}{ll} \max & \gamma \\ s.t. & \theta - \gamma \in \mathrm{Ideal}[\Phi]_{2k} + \mathrm{Qmod}[\Psi]_{2k}. \end{array}\right. \tag{3.25}$$

For relaxation orders $k = d_0, d_0 + 1, \ldots$, we get the Moment-SOS hierarchy of semidefinite relaxations (3.24)-(3.25). This produces the following algorithm for solving the polynomial optimization problem (3.23).

Algorithm 3.11. Let θ, Φ, Ψ be as in (3.23). Initialize $k := d_0$.

Step 1 Solve the moment relaxation (3.24). If it is infeasible, (3.23) has no feasible points and stop; otherwise, solve it for a minimizer y^*.

Step 2 Let $u = (y^*_{e_1}, \dots, y^*_{e_n})$. If u is feasible for (3.23) and $\vartheta_k = \theta(u)$, then u is a minimizer of (3.23). Otherwise, let $k := k+1$ and go to Step 1.

In the Step 2, e_i denotes the labeling vector such that the ith entry is 1 while all other entries are 0. For example, when $n = 4$, then $y_{e_2} = y_{0100}$.

The conclusions of Algorithm 3.11 are justified as follows. The optimization (3.24) is a relaxation of (3.23). This is because if x is a feasible point of (3.23), then $y = [x]_{2k}$ must be feasible for (3.24). Hence, if (3.24) is infeasible, then (3.23) must be infeasible, which also implies the nonexistence of a NE. Moreover, the optimal value ϑ_k of (3.24) is a lower bound for the minimum value of (3.23), i.e., $\vartheta_k \le \theta(x)$ for all x that is feasible for (3.23). In the Step 2, if u is feasible for (3.23) and $\vartheta_k = \theta(u)$, then u must be a minimizer of (3.23). The convergence of Algorithm 3.11 is shown as follows.

Theorem 3.12. *Assume the matrix Θ is a generic positive definite matrix and Ideal$[\Phi]$ + Qmod$[\Psi]$ is archimedean.*

(i) If the optimization (3.23) is infeasible, then the moment relaxation (3.24) must be infeasible when the order k is big enough.

(ii) Suppose the optimization (3.23) is feasible. Let $u^{(k)}$ be the point u produced in the Step 2 of Algorithm 3.11 in the kth loop. Then $u^{(k)}$ converges to the unique minimizer of (3.23). In particular, if the real zero set of Φ is finite, then $u^{(k)}$ is the unique minimizer of (3.23), when k is sufficiently large.

Proof. (i) If (3.23) is infeasible, the constant polynomial -1 can be viewed as a positive polynomial on the feasible set of (3.23). Since Ideal$[\Phi]$ + Qmod$[\Psi]$ is archimedean, we have $-1 \in \text{Ideal}[\Phi]_{2k} + \text{Qmod}[\Psi]_{2k}$, for k big enough, by the Putinar's Positivstellensatz [106]. For such big k, the SOS relaxation (3.25) is unbounded from above, hence the moment relaxation (3.24) must be infeasible.

(ii) When the optimization (3.23) is feasible, it must have a unique minimizer, say, x^*, because its objective is a generic positive definite quadratic polynomial. The convergence of $u^{(k)}$ to x^* is shown in [120] or [88, Theorem 3.3]. For the special case that $\Phi(x) = 0$ has finitely many real solutions, the point $u^{(k)}$ must be equal to x^*, when k is large enough. This is shown in [65] (also see [89]). □

The archimedeanness of Ideal[Φ] + Qmod[Ψ] is essentially requiring that the feasible set of (3.23) is compact. If the real zero set of Φ is compact, then Ideal[Φ] + Qmod[Ψ] must be archimedean. In particular, if the NEPP has finitely many real KKT points, then Ideal[Φ] + Qmod[Ψ] is archimedean. Interestingly, when the objective and constraining polynomials are generic, there are finitely many KKT points. See Theorem 3.10 in the appendix. In fact, as shown in the proof of Theorem 3.10, the zero set of Φ is finite for generic NEPPs, and hence Algorithm 3.11 has finite convergence.

The other polynomial optimization problem (3.15) can be solved in the same way by the Moment-SOS hierarchy of semidefinte relaxations like (3.24)-(3.25). The convergence property is the same. For cleanness of the paper, we omit the details.

For the maximization (3.16), we denote the set of polynomials

$$\Phi := \bigcup_{i=1}^{N} \Phi_i, \quad \Psi := \bigcup_{i=1}^{N} \Psi_i \cup \Big\{ [x^*]_1^T \Theta [x^*]_1 + \delta - [x]_1^T \Theta [x]_1 \Big\}. \tag{3.26}$$

Then (3.16) can be equivalently written as

$$\begin{cases} \vartheta_{\min} := \min\limits_{x \in \mathbb{R}^n} & \theta(x) := -[x]_1^T \Theta [x]_1 \\ \qquad\qquad s.t. & p(x) = 0 \ (\forall\, p \in \Phi), \\ & q(x) \geq 0 \ (\forall\, q \in \Psi). \end{cases} \tag{3.27}$$

Similarly, the hierarchy of moment relaxations (3.24) can be used to solve (3.27). The following is the algorithm.

Algorithm 3.13. Let Φ, Ψ, θ be as in (3.26)-(3.27). Initialize $k := d_0$.

Step 1 Solve the moment relaxation (3.24) for the minimum value ϑ_k and a minimizer y^* and let $t := d_0$.

Step 2 If $\vartheta_k \geq -[x^*]_1^T \Theta [x^*]_1$, stop; otherwise, go to Step 3.

Step 3 If y^* satisfies the rank condition

$$\operatorname{rank} M_t[y^*] \;=\; \operatorname{rank} M_{t-d_0}[y^*], \tag{3.28}$$

then extract a set U_i of $r := \operatorname{rank} M_t(y^*)$ minimizers for (3.27)) and stop.

Step 4 If (3.28) fails to hold and $t < k$, let $t := t + 1$ and then go to Step 3; otherwise, let $k := k + 1$ and go to Step 1.

The optimization (3.27) is always feasible because x^* is a feasible point. Therefore, the moment relaxation (3.24) is also feasible. Since the minimum value ϑ_k is a lower bound of $\vartheta_{\min}$, if $\vartheta_k \geq -[x^*]_1^T \Theta [x^*]_1$, then

$$\vartheta_k = \vartheta_{\min} = -[x^*]_1^T \Theta [x^*]_1$$

and hence x^* is a maximizer of (3.16). In Step 3, the rank condition (3.28) is called *flat truncation* [88]. It is a sufficient (and almost necessary) condition to check convergence of moment relaxations. When (3.28) holds, the method in [48] can be used to extract r minimizers for (3.27)). The method is implemented in the software `GloptPoly` [49]. Moreover, Algorithms 3.11 and 3.13 can be implemented in `GloptPoly`.

The convergence of Algorithm 3.13 is as follows. Note that $\mathrm{Ideal}(\Phi) + \mathrm{Qmod}(\Psi)$ is archimedean, since it contains the polynomial $[x^*]_1^T \Theta [x^*]_1 + \delta - [x]_1^T \Theta [x]_1$. Therefore, we always have $\vartheta_k \to \vartheta_{\min}$ as $k \to \infty$ [61]. Under some classical optimality conditions, we have $\vartheta_k = \vartheta_{\min}$ when k is large enough [90]. Moreover, if the real zero set of Φ is finite, then Algorithm 3.13 has finite convergence [89].

3.3.2 Checking Nash equilibria

Suppose u is a minimizer of (3.12). To check if $u = (u_i, u_{-i})$ is a NE or not, we need to solve the individual optimization (3.10) for each player. For notational convenience, we denote the polynomial tuples

$$\begin{aligned} H_i(u) := \big\{ g_{i,j} : j \in \mathcal{E}_i \big\} \cup \big\{ \lambda_{i,j}(x_i, u_{-i}) \cdot g_{i,j} : j \in \mathcal{I}_i \big\} \\ \cup \Big\{ \nabla_{x_i} f_i(x_i, u_{-i}) - \sum_{j=1}^{m_i} \lambda_{ij}(x_i, u_{-i}) \nabla_{x_i} g_{ij} \Big\}, \end{aligned} \tag{3.29}$$

$$G_i(u) := \big\{ g_{i,j} : j \in \mathcal{I}_i \big\} \cup \big\{ \lambda_{i,j}(x_i, u_{-i}) : j \in \mathcal{I}_i \big\}. \tag{3.30}$$

Like the earlier case, the set $\{p\}$ stands for $\{p_1, \ldots, p_s\}$, when $p = (p_1, \ldots, p_s)$ is a vector of polynomial. The sets $H_i(u), G_i(u)$ are finite collections of polynomials in x_i and parameterized by u. If the optimization (3.10) has a minimizer, then it is equivalent to the following optimization

$$\begin{cases} \omega_i := & \min\limits_{x_i \in \mathbb{R}^{n_i}} & f_i(x_i, u_{-i}) \\ & s.t. & p(x_i) = 0 \, (p \in H_i(u)), \\ & & q(x_i) \geq 0 \, (q \in G_i(u)). \end{cases} \tag{3.31}$$

The above is a polynomial optimization problem in x_i. Denote the degree for its constraining polynomials

$$d_i := \max \left\{ \lceil \deg(p)/2 \rceil : p \in H_i(u) \cup G_i(u) \right\}. \tag{3.32}$$

For a degree $k \geq d_i$, the kth order moment relaxation for (3.23) is

$$\begin{cases} \omega_i^{(k)} := & \min\limits_{y} & \langle f_i(x_i, u_{-i}), y \rangle \\ & s.t. & y_0 = 1, \, L_p^{(k)}[y] = 0 \, (p \in H_i(u)), \\ & & M_k[y] \succeq 0, \, L_q^{(k)}[y] \succeq 0 \, (q \in G_i(u)), \\ & & y \in \mathbb{R}^{\mathbb{N}_{2k}^{n_i}}. \end{cases} \tag{3.33}$$

Its dual optimization problem is the kth order SOS relaxation

$$\begin{cases} \max & \gamma \\ s.t. & f_i(x_i, u_{-i}) - \gamma \in \mathrm{Ideal}[H_i(u)]_{2k} + \mathrm{Qmod}[G_i(u)]_{2k}. \end{cases} \tag{3.34}$$

By solving the above relaxations for $k = d_i, d_i + 1, \ldots$, we get the Moment-SOS hierarchy of relaxations (3.33)-(3.34). This gives the following algorithm.

Algorithm 3.14. For the ith player's individual optimization (3.31), assume u is a minimizer of (3.12).

Step 0 Construct the sets $H_i(u)$, $G_i(u)$ of polynomials as in (3.29), (3.30). Initialize $k := d_i$.

Step 1 Solve the moment relaxation (3.33) for the minumum value $\omega_i^{(k)}$ and a minimizer y^*. If $\omega_i^{(k)} \geq 0$, then $\omega_i = 0$ and stop; otherwise, go to the next step.

Step 2 Let $t := d_i$ as in (3.32). If y^* satisfies the rank condition

$$\operatorname{rank} M_t[y^*] \; = \; \operatorname{rank} M_{t-d_i}[y^*], \tag{3.35}$$

then extract a set U_i of $r := \operatorname{rank} M_t(y^*)$ minimizers for (3.31)) and stop.

Step 3 If (3.35) fails to hold and $t < k$, let $t := t + 1$ and then go to Step 2; otherwise, let $k := k + 1$ and go to Step 1.

We would like to remark that the optimization (3.31) is always feasible, because u is a minimizer of (3.12). The moment relaxation (3.33) is also feasible. Because $\omega_i^{(k)}$ is a lower bound for ω_i, and $\omega_i \le f_i(u_i, u_{-i}) = 0$, if $\omega_i^{(k)} \ge 0$, then ω_i must be 0. The Algorithm 3.14 can be implemented in `GloptPoly`. The following theorem is the convergence for Algorithm 3.14. Its proof follows from [101, Theorem 4.4].

Theorem 3.15. *Assume the ith player's constraining polynomial tuple g_i is nonsingluar and its optimization (3.10) has a minimizer for the given u_{-i}. Assume either one of the following conditions hold:*

(i) The set Ideal$[H_i(u)]$ + Qmod$[G_i(u)]$ is archimedean,

(ii) The real zero set of polynomials in $H_i(u)$ is finite.

If each minimizer of (3.10) is an isolated critical point, then all minimizers of (3.33) must satisfy the flat truncation (3.35), for all k big enough. Therefore, Algorithm 3.14 must terminate within finitely many loops.

We would like to remark the following inclusion

$$\text{Ideal}[g_{i,j} : j \in \mathcal{E}_i] \subseteq \text{Ideal}[H_i(u)], \quad \text{Qmod}[g_{i,j} : j \in \mathcal{I}_i] \subseteq \text{Qmod}[G_i(u)].$$

If $\text{Ideal}[g_{i,j} : j \in \mathcal{E}_i] + \text{Qmod}[g_{i,j} : j \in \mathcal{I}_i]$ is archimedean, then $\text{Ideal}[H_i(u)] + \text{Qmod}[G_i(u)]$ is also archimedean. Therefore, if the archimedeanness holds for the ith player's optimization (3.1), then the condition (i) in Theorem 3.15 is satisfied.

3.4 Numerical Experiments

This section reports numerical experiments for solving NEPs by using Algorithm 3.2 and 3.7. We apply the software `GloptiPoly 3` [49] and `SeDuMi` [124] to solve the Moment-SOS relaxations for the polynomial optimization (3.12), (3.10), (3.15) and (3.16). The computation is implemented in an Alienware Aurora R8 desktop, with an Intel® Core(TM) i7-9700 CPU at 3.00GHz×8 and 16GB of RAM, in a Windows 10 operating system. To implement Algorithm 3.2, we need Lagrange multiplier representations as in [95]. The following cases are frequently used.

- For the constraint $\{x_i \in \mathbb{R}^{n_i} : \sum_{j=1}^{n_i} x_{i,j} \le 1, x_i \ge 0\}$, the constraining polynomials are $g_{i,0} = 1 - \sum_{j=1}^{n_i} x_{i,j}, g_{i,1} = x_{i,1}, \cdots, g_{i,n_i} = x_{i,n_i}$. The Lagrange multipliers $\lambda_{i,j}$ can be represented as

$$\lambda_{i,0} = x_i^T \nabla_{x_i} f_i, \quad \lambda_{i,j} = \frac{\partial f_i}{\partial x_{i,j}} - x_i^T \nabla_{x_i} f_i, \quad j = 1, \ldots, n_i.$$

- For the sphere constraint $1 - x_i^T x_i = 0$ or the ball constraint $1 - x_i^T x_i \ge 0$, the constraining polynomial is $g_{i,1} = 1 - x_i^T x_i$ and the Lagrange multiplier can be expressed as $\lambda_{i,1} = -\frac{1}{2} x_i^T \nabla_{x_i} f_i$.

In Step 2 of Algorithm 3.2 and Step 3 of Algorithm 3.7, if the optimal value $\omega_i \ge 0$ for all players, then the point u is a NE. In numerical computation, we cannot have $\omega_i \ge 0$ exactly, due to round-off errors. Therefore, we use the parameter

$$\omega^* := \min_{i=1,\ldots,N} \omega_i$$

to measure the accuracy of the computed NE. Typically, if ω^* is small, say, $\omega^* \ge -10^{-6}$, we regard the computed solution as an accurate NE.

For the convex NEP in Example 3.1, Algorithm 3.7 found all the 3 NEs correctly with $\omega^* = -1.9512 \cdot 10^{-9}$. The following is another example of convex NEPs.

Example 3.16. Consider the convex NEP

$$\text{1st player:} \begin{cases} \min\limits_{x_1 \in \mathbb{R}^2} & x_{1,1}(x_{1,1} + x_{2,1} + 4x_{2,2}) + 4x_{1,2}^2, \\ s.t. & 1 - (x_{1,1})^2 - (x_{1,2})^2 \ge 0, \end{cases}$$

$$\text{2nd player:} \begin{cases} \min\limits_{x_2 \in \mathbb{R}^2} & 2x_{2,1}^2 + 2x_{2,2}^2 + (x_{1,1} - 2x_{1,2})x_{2,1} \\ & \qquad + (4x_{1,1} + x_{1,2})x_{2,2}, \\ s.t. & 1 - x_{2,1} - x_{2,2} \ge 0,\ x_{2,1} \ge 0,\ x_{2,2} \ge 0. \end{cases}$$

For the second player, the Lagrange multipliers can be represented as

$$\lambda_{2,1} = -\nabla_{x_2} f_2^T x_2,\ \lambda_{2,2} = \frac{\partial f_2}{\partial x_{2,1}} + \lambda_{2,1},\ \lambda_{2,3} = \frac{\partial f_2}{\partial x_{2,2}} + \lambda_{2,1}.$$

For this NEP, Algorithm 3.7, found two NEs:

$$x_1^* = (0.0000, -0.0000),\ x_2^* = (0.0000, 0.0000),\ \omega^* = -7.4772 \cdot 10^{-9};$$

$$x_1^* = (-1.0000, -0.0000),\ x_2^* = (0.1250, 0.8750),\ \omega^* = -3.3640 \cdot 10^{-8}.$$

The computation took about 1 second.

Example 3.17. Consider the 2-player NEP

$$\text{1st player:} \begin{cases} \min\limits_{x_1 \in \mathbb{R}^3} & \sum_{j=1}^3 x_{1,j}(x_{1,j} - j \cdot x_{2,j}) \\ s.t. & 1 - x_{1,1}x_{1,2} \geq 0,\ 1 - x_{1,2}x_{1,3} \geq 0,\ x_{1,1} \geq 0, \end{cases}$$

$$\text{2nd player:} \begin{cases} \min\limits_{x_2 \in \mathbb{R}^3} & \prod_{j=1}^3 x_{2,j} + \sum\limits_{\substack{1 \leq i < j \leq 3 \\ 1 \leq k \leq 3}} x_{1,i}x_{1,j}x_{2,k} + \sum\limits_{\substack{1 \leq i \leq 3 \\ 1 \leq j < k \leq 3}} x_{1,i}x_{2,j}x_{2,k} \\ s.t. & 1 - (x_{2,1})^2 - (x_{2,2})^2 = 0. \end{cases}$$

The first player's optimization is non-convex, with an unbounded feasible set. The Lagrange multipliers for the first player's optimization are

$$\lambda_{1,1} = (1 - x_{1,1}x_{1,2})\tfrac{\partial f_1}{\partial x_{1,1}}, \quad \lambda_{1,2} = -x_{1,1}\tfrac{\partial f_1}{\partial x_{1,2}}, \quad \lambda_{1,3} = x_{1,1}\tfrac{\partial f_1}{\partial x_{1,1}} - x_{1,2}\tfrac{\partial f_1}{\partial x_{1,2}}.$$

Applying Algorithm 3.7, we get four NEs:

$$\begin{array}{ll} x_1^* = (0.3198, 0.6396, -0.6396), & x_2^* = (0.6396, 0.6396, -0.4264); \\ x_1^* = (0.0000, 0.3895, 0.5842), & x_2^* = (-0.8346, 0.3895, 0.3895); \\ x_1^* = (0.2934, -0.5578, 0.8803), & x_2^* = (0.5869, -0.5578, 0.5869); \\ x_1^* = (0.0000, -0.5774, -0.8660), & x_2^* = (-0.5774, -0.5774, -0.5774). \end{array}$$

Their accuracy parameters are respectively

$$-7.1879 \cdot 10^{-8}, -3.5040 \cdot 10^{-7}, -4.3732 \cdot 10^{-7}, -6.4360 \cdot 10^{-7}.$$

It took about 30 seconds. If the second player's objective becomes

$$-\prod_{j=1}^3 x_{2,j} + \sum_{\substack{1 \leq i \leq 3 \\ 1 \leq j < k \leq 3}} x_{1,i}x_{2,j}x_{2,k} - \sum_{\substack{1 \leq i < j \leq 3 \\ 1 \leq k \leq 3}} x_{1,i}x_{1,j}x_{2,k},$$

then there is no NE, which is detected by Algorithm 3.2. It took around 16 seconds.

Example 3.18. Consider the 3-player NEP

$$\text{1st player:} \begin{cases} \min\limits_{x_1 \in \mathbb{R}^2} & (2x_{1,1} - x_{1,2} + 3)x_{1,1}x_{2,1} \\ & \qquad + [(2x_{1,2})^2 + (x_{3,2})^2]x_{1,2} \\ s.t. & 1 - x_1^T x_1 \geq 0, \end{cases}$$

$$\text{2nd player:} \begin{cases} \min\limits_{x_2 \in \mathbb{R}^2} & [(x_{2,1})^2 - x_{1,2}]x_{2,1} \\ & \qquad + [(x_{2,2})^2 + 2x_{3,2} + x_{1,2}x_{3,1}]x_{2,2} \\ s.t. & x_2^T x_2 - 1 = 0,\ x_{2,1} \geq 0,\ x_{2,2} \geq 0, \end{cases}$$

$$
\text{3rd player:} \begin{cases} \min\limits_{x_3 \in \mathbb{R}^2} & (x_{1,1}x_{1,2} - 1)x_{3,1} - [3(x_{3,2})^2 + 1]x_{3,2} \\ & \qquad\qquad +2[x_{3,1} + x_{3,2}]x_{3,1}x_{3,2} \\ s.t. & 1 - (x_{3,1})^2 \geq 0,\ 1 - (x_{3,2})^2 \geq 0. \end{cases}
$$

The Lagrange multipliers can be represented as

$$
\begin{array}{lll}
\lambda_{2,1} = \frac{1}{2}(x_2^T \nabla_{x_2} f_2), & \lambda_{2,2} = \frac{\partial f_2}{x_{2,1}} - 2x_{2,1}\lambda_{2,1}, & \lambda_{2,3} = \frac{\partial f_2}{x_{2,2}} - 2x_{2,2}\lambda_{2,1}, \\
\lambda_{3,1} = -\frac{x_{3,1}}{2}\frac{\partial f_3}{\partial x_{3,1}}, & \lambda_{3,2} = -\frac{x_{3,2}}{2}\frac{\partial f_3}{\partial x_{3,2}}. &
\end{array}
$$

Applying Algorithm 3.7, we get the unique NE

$$
x_1^* = (-0.3558, -0.9346),\ x_2^* = (1.0000, 0.0000),\ x_3^* = (-0.3331, 1.0000).
$$

The accuracy parameter is $-9.2310 \cdot 10^{-9}$. It took around 9 seconds. If the third player's objective becomes $-f_1(x) - f_2(x)$, then the NEP becomes a zero-sum game and there is no NE. Algorithm 3.2 detected the nonexistence. It took around 3 seconds.

Example 3.19. Consider the 2-player NEP

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

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

The Lagrange multipliers can be represented as $(i = 1, 2)$:

$$
\lambda_{i,1} = \frac{1}{2}\nabla_{x_i} f_i^T x_i (2 - x_i^T x_i), \quad \lambda_{i,2} = \frac{1}{4}\nabla_{x_i} f_i^T x_i (1 - x_i^T x_i).
$$

By Algorithm 3.7, we get the unique NE

$$
x_1^* = (-1.3339, 0.4698), \quad x_2^* = (-1.4118, 0.0820),
$$

with the accuracy parameter $-3.5186 \cdot 10^{-8}$. It took around 5 seconds.

Example 3.20. Consider the NEP

$$
\text{1st player:} \begin{cases} \min\limits_{x_1 \in \mathbb{R}^{n_1}} & \sum_{1 \le i \le j \le n_1} x_{1,i} x_{1,j} (x_{2,i} + x_{2,j}) \\ s.t. & 1 - (x_{1,1}^2 + \cdots + x_{1,n_1}^2) = 0, \end{cases}
$$

$$
\text{2nd player:} \begin{cases} \min\limits_{x_2 \in \mathbb{R}^{n_2}} & \sum_{1 \le i \le j \le n_2} x_{2,i} x_{2,j} (x_{1,i} + x_{1,j}) \\ s.t. & 1 - (x_{2,1}^2 + \cdots + x_{2,n_2}^2) = 0, \end{cases}
$$

where the dimension $n_1 = n_2$. The computational results for cases $n_1 = n_2 = 3, 4, 5, 6$ are shown in Table 3.1. The time is displayed in seconds. The accuracy parameter ω^* is

Table 3.1: Computational results for Example 3.20.

n_1	x_1^*	x_2^*	time
3	$(-0.5774,\ -0.5774,\ -0.5774)$	$(-0.5774,\ -0.5774,\ -0.5774)$	1.31
4	$(0.8381,\ 0.5024,\ -0.0328,\ -0.2098)$	$(-0.1791,\ -0.0683,\ 0.4066,\ 0.8933)$	62.85
5	$(0.8466,\ 0.4407,\ 0.1744,\ -0.0101,\ -0.2418)$	$(-0.1944,\ -0.0512,\ 0.1238,\ 0.3370,\ 0.9114)$	682.67
6	$(0.8026,\ 0.4724,\ 0.1799,\ 0.1799,\ -0.0637,\ -0.2527)$	$(-0.1979,\ -0.0772,\ 0.1091,\ 0.1091,\ 0.4040,\ 0.8762)$	18079.99

respectively

$$
-1.0689 \cdot 10^{-7}, \quad -1.4459 \cdot 10^{-9}, \quad -2.7551 \cdot 10^{-9}, \quad -7.0354 \cdot 10^{-9}.
$$

Because of the relatively large amount of computational time, we only compute one NE for each case in the above.

We would like to remark that our method can also be applied to solve NEPs for which the individual optimization has no constraints, or equivalently, the feasible set X_i for (3.1) is the entire space $\mathbb{R}^{n_i}$. For unconstrained NEPs, the KKT system (3.4) becomes

$$
\nabla_{x_i} f_i(x^*) = 0, \quad i = 1, \ldots, N.
$$

The Algorithms 3.2 and (3.7) can be implemented in the same way.

Example 3.21. Consider the unconstrained NEP

$$
\text{1st player: } \begin{cases} \min & \sum_{i=1}^{n_1} (x_{1,i})^4 + \sum_{0 \le i \le j \le k \le n_1} \frac{x_{1,i}x_{1,j}(x_{1,k}+x_{2,i}+x_{3,j})}{(n_1)^2} \\ s.t. & x_1 \in \mathbb{R}^{n_1}, \end{cases}
$$

$$
\text{2nd player: } \begin{cases} \min & \sum_{i=1}^{n_2} (x_{2,i})^4 + \sum_{0 \le i \le j \le k \le n_2} \frac{x_{2,i}x_{2,j}(x_{2,k}+x_{3,i}+x_{1,j})}{(n_2)^2} \\ s.t. & x_2 \in \mathbb{R}^{n_2}, \end{cases}
$$

$$
\text{3rd player: } \begin{cases} \min & \sum_{i=1}^{n_3} (x_{3,i})^4 + \sum_{0 \le i \le j \le k \le n_3} \frac{x_{3,i}x_{3,j}(x_{3,k}+x_{1,i}+x_{2,j})}{(n_3)^2} \\ s.t. & x_3 \in \mathbb{R}^{n_3}, \end{cases}
$$

where $x_{1,0} = x_{2,0} = x_{3,0} = 1$, and $n_1 = n_2 = n_3$. We implement Algorithm 3.7 for the cases $n_1 = n_2 = n_3 = 2, 3, 4, 5, 6$. The computational results are shown in the following table. For all cases, we computed a NE successfully and obtained that $x_1^* = x_2^* = x_3^*$ (up to round-off errors). There is a unique NE for each case. The computational results are reported in Table 3.2. The time is displayed in seconds.

Table 3.2: The computational results for Example 3.21.

n_1	$x_1^* = x_2^* = x_3^*$	ω^*	time
2	$(-0.8410, \quad -0.7125)$	$-8.8291 \cdot 10^{-9}$	0.34
3	$(-0.6743, -0.6157, -0.5236)$	$-6.6507 \cdot 10^{-9}$	1.58
4	$(-0.5950, -0.5606$ $-0.5097, -0.4363)$	$-1.0577 \cdot 10^{-9}$	16.86
5	$(-0.5476, -0.5247, -0.4919,$ $-0.4472, -0.3860)$	$-4.4438 \cdot 10^{-9}$	177.63
6	$(-0.5157, -0.4992, -0.4762,$ $-0.4457, -0.4060, -0.3534)$	$-3.7536 \cdot 10^{-9}$	1379.27

The following are some examples of NEPs from applications.

Example 3.22. Consider the environmental pollution control problem for three countries for the case *autarky* [19]. Let $x_{i,1} (i = 1, 2, 3)$ denote the (gross) emissions from the ith country. The revenue of the ith country depends on $x_{i,1}$, e.g., a typically one is $x_{i,1}(b_i - \frac{1}{2}x_{i,1})$. The variable $x_{i,2}$ represents the investment by the ith country to local environmental projects. The net emission in country i is $x_{i,1} - \gamma_i x_{i,2}$, which is always nonnegative and must be kept below or equal a certain prescribed level $E_i > 0$ under an environmental constraint. The damage cost of the ith country is assumed to be $d_i(x_{i,1} - \gamma x_{i,2}) + \sum_{j \neq i} c_{i,j} x_{i,2} x_{j,1}$. For given parameters $b_i, c_{i,j}, d_i, \gamma_i, E_i$, the ith $(i = 1,2,3)$ country's optimization problem is

$$\begin{cases} \min\limits_{x_i \in \mathbb{R}^2} & -x_{i,1}(b_i - \frac{1}{2}x_{i,1}) + \frac{(x_{i,2})^2}{2} + d_i(x_{i,1} - \gamma_i x_{i,2}) + \sum\limits_{j \neq i} c_{i,j} x_{i,2} x_{j,1} \\ s.t. & x_{i,2} \geq 0, \ x_{i,1} \leq b_i, \\ & 0 \leq x_{i,1} - \gamma_i x_{i,2} \leq E_i. \end{cases}$$

We consider the general cases that $b_i \neq E_i$. The Lagrange multipliers can be expressed as

$$\begin{aligned} \lambda_{i,4} = & \ \tfrac{1}{(b_i - E_i)E_i} \left(\tfrac{\partial f_i}{\partial x_{i,2}} x_{i,2}(x_{i,1} - \gamma_i x_{i,2}) - \tfrac{\partial f_i}{\partial x_{i,1}}(b_i - x_{i,1})(x_{i,1} - \gamma_i x_{i,2}) \right), \\ \lambda_{i,3} = & \ \tfrac{1}{b_i} \left((b_i - x_{i,1})(\tfrac{\partial f_i}{\partial x_{i,1}} + \lambda_{i,4}) - x_{i,2}(\tfrac{\partial f_i}{\partial x_{i,2}} - \gamma_i \lambda_{i,4}) \right), \\ \lambda_{i,2} = & \ \lambda_{i,3} - \lambda_{i,4} - \tfrac{\partial f_i}{\partial x_{i,1}}, \\ \lambda_{i,1} = & \ \tfrac{\partial f_i}{\partial x_{i,2}} + \gamma_i \lambda_{i,3} - \gamma_i \lambda_{i,4}. \end{aligned}$$

We solve the NEP for the following typical parameters:

$$\begin{array}{llllll} b_1 = 1.5, & b_2 = 2, & b_3 = 1.8, & c_{1,2} = 0.2, & c_{1,3} = 0.3, & c_{2,1} = 0.4, \\ c_{2,3} = 0.2, & c_{3,1} = 0.5, & c_{3,2} = 0.1, & d_1 = 0.8, & d_2 = 1.2, & d_3 = 1.0, \\ E_1 = 3, & E_2 = 4, & E_3 = 2, & \gamma_1 = 0.7, & \gamma_2 = 0.5, & \gamma_3 = 0.9. \end{array}$$

By Algorithm 3.7, we get the unique NE

$$x_1^* = (0.7000, 0.1600), \quad x_2^* = (0.8000, 0.1600), \quad x_3^* = (0.8000, 0.4700),$$

with the accuracy parameter $-1.1059 \cdot 10^{-9}$. It took about 10 seconds.

Example 3.23. Consider the NEP of the electricity market problem [21, 35]. There are three generating companies, and the ith company possesses s_i generating units. For the ith company, the power generation of his jth generating unit is denoted by $x_{i,j}$. Assume $0 \leq x_{i,j} \leq E_{i,j}$, where the nonzero parameter $E_{i,j}$ represents its maximum capacity, and the cost of this generating unit is $\frac{1}{2}c_{i,j}(x_{i,j})^2 + d_{i,j}x_{i,j}$, where $c_{i,j}, d_{i,j}$ are parameters. The electricty price is given by

$$\phi(x) := b - a(\sum_{i=1}^{3} \sum_{j=1}^{s_i} x_{i,j}).$$

The aim of the each company is to maximize its profits, that is, to solve the following optimization problem:

$$i\text{th player:} \begin{cases} \min\limits_{x_i \in \mathbb{R}^{s_i}} & \frac{1}{2} \sum_{j=1}^{s_i} (c_{i,j}(x_{i,j})^2 + d_{i,j} x_{i,j}) - \phi(x)(\sum_{j=1}^{s_i} x_{i,j}). \\ s.t. & 0 \leq x_{i,j} \leq E_{i,j} \, (j \in [s_i]). \end{cases}$$

The Lagrange multipliers according to the constraints $g_{i,2j-1} := E_{i,j} - x_{i,j} \geq 0, g_{i,2j} := x_{i,j} \geq 0$ can be represented as

$$\lambda_{i,2j-1} = -\frac{\partial f_i}{\partial x_{i,j}} \cdot x_{i,j}/E_{i,j}, \lambda_{i,2j} = \frac{\partial f_i}{\partial x_{i,j}} + \lambda_{i,2j-1}. \, (j \in [s_i])$$

We run Algorithm 3.7 for the following setting:

$$\begin{array}{llllll}
s_i = i, & & a = 1, & & b = 10, & \\
c_{1,1} = 0.4, & c_{2,1} = 0.35, & c_{2,2} = 0.35, & c_{3,1} = 0.46, & c_{3,2} = 0.5, & c_{3,3} = 0.5, \\
d_{1,1} = 2, & d_{2,1} = 1.75, & d_{2,2} = 1, & d_{3,1} = 2.25, & d_{3,2} = 3, & d_{3,3} = 3, \\
E_{1,1} = 2, & E_{2,1} = 2.5, & E_{2,2} = 0.67, & E_{3,1} = 1.2, & E_{3,2} = 1.8, & E_{3,3} = 1.6.
\end{array}$$

We found the unique NE

$$x_1^* = 1.7184, \quad x_2^* = (1.8413, 0.6700), \quad x_3^* = (1.2000, 0.0823, 0.0823).$$

The accuracy parameter is $-5.1183 \cdot 10^{-7}$. It took about 8 seconds.

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

Chapter 4

Convex Generalized Nash Equilibrium Problems of Polynomials

Consider the Generalized Nash Equilibrium Problem

$$\mathrm{F}_i(u_{-i}) : \begin{cases} \min\limits_{x_i \in \mathbb{R}^{n_i}} & f_i(u_1, \dots, u_{i-1}, x_i, u_{i+1}, \dots, u_N) \\ s.t. & g_{i,j}(u_1, \dots, u_{i-1}, x_i, u_{i+1}, \dots, u_N) = 0 \, (j \in \mathcal{E}_i), \\ & g_{i,j}(u_1, \dots, u_{i-1}, x_i, u_{i+1}, \dots, u_N) \geq 0 \, (j \in \mathcal{I}_i), \end{cases} \tag{4.1}$$

where the f_i and $g_{i,j}$ are continuously differentiable functions in x_i, and the $\mathcal{E}_i$, $\mathcal{I}_i$ are disjoint finite (possibly empty) labeling sets. The point u satisfying the above is called a Generalized Nash Equilibrium (GNE).

Recall that the GNEP given by (1.1) is called *convex* if for all $i = 1, \dots, N$ and for all given $x_{-i} \in \mathrm{dom}(X_i)$, the objective $f_i(x_i, x_{-i})$ is convex in x_i on $X_i(x_{-i})$, all $g_{i,j}(x_i, x_{-i}) \, (j \in \mathcal{E}_i)$ are affine linear in x_i and all $g_{i,j}(x_i, x_{-i}) \, (j \in \mathcal{I}_i)$ are concave in x_i. In this chapter, we study how to solve convex GNEPPs.

4.1 Rational expressions for Lagrange Multipliers

In Section 3.1, a polynomial expression for the ith player's Lagrange multipliers exists if and only if the matrix $G_i(x)$ is nonsingular. For classical NEPs of polynomials, the nonsingularity holds generically [95,98]. However, this is often not the case for GNEPs. Let $g_i = (g_{i,1}, \dots, g_{i,m_i})$ be the tuple of constraining polynomials in $\mathrm{F}_i(x_{-i})$ and $G_i(x)$ be the matrix polynomial as in (3.5). If there exists a matrix polynomial $\hat{L}_i(x)$ and a nonzero scalar polynomial $q_i(x)$ such that

$$\hat{L}_i(x)G_i(x) \;=\; q_i(x)\cdot I_{m_i}, \tag{4.2}$$

then $q_i(x)\lambda_i = \hat{L}_i(x)\hat{f}_i(x)$ for all critical pairs (x_i,λ_i) of $\mathrm{F}_i(x_{-i})$. Let

$$\hat{\lambda}_i(x) \;:=\; \hat{L}_i(x)\hat{f}_i(x). \tag{4.3}$$

Denote by $\hat{\lambda}_{i,j}(x)$ the jth entry of $\hat{\lambda}_i(x)$.

Definition 4.1. For the ith player's optimization $\mathrm{F}_i(x_{-i})$, if there exist polynomials $\hat{\lambda}_{i,j}, j\in[m_i]$ and a nonzero polynomial q_i such that $q_i(x)\ge 0$ for all $x\in X$, and $\hat{\lambda}_{i,j}(x) = q_i(x)\lambda_{i,j}$ holds for all critical pairs (x_i,λ_i), then we call the tuple

$$\hat{\lambda}_i/q_i := (\hat{\lambda}_{i,1}(x)/q_i(x),\dots,\hat{\lambda}_{i,m_i}(x)/q_i(x))$$

a rational expression for Lagrange multipliers.

The following is an example of rational expression.

Example 4.2. Consider the 2-player convex GNEP

$$\left.\begin{array}{cl} \min\limits_{x_1\in\mathbb{R}^2} & f_1(x_1,x_2)\\ s.t. & 2-x_1^Tx_1-x_2\ge 0; \end{array}\right| \begin{array}{cl} \min\limits_{x_2\in\mathbb{R}^1} & f_2(x_1,x_2)\\ s.t. & 3x_2-x_1^Tx_1\ge 0,\ 1-x_2\ge 0. \end{array} \tag{4.4}$$

The matrices of polynomials $G_1(x)$ and $G_2(x)$ are

$$G_1(x) := \begin{bmatrix} -2x_{1,1}\\ -2x_{1,2}\\ 2-x_1^Tx_1-x_2 \end{bmatrix},\quad G_2(x) := \begin{bmatrix} 3 & -1\\ 3x_2-x_1^Tx_1 & \\ & 1-x_2 \end{bmatrix}.$$

For $x_1=(0,0)$ and $x_2=2$, the $G_1(x)$ is the zero vector. For $x_1=(\sqrt{3},0)$ and $x_2=1$, $\mathrm{rank}(G_2(x))=1$. Both $G_1(x),G_2(x)$ are not nonsingular, so there are no polynomial expressions for Lagrange multipliers. However, the (4.2) holds for

$$\begin{gathered} q_1(x) = 2-x_2,\quad q_2(x) = 1-\tfrac{1}{3}x_1^Tx_1,\\ \hat{L}_1(x) = \begin{bmatrix} -\frac{x_{1,1}}{2} & -\frac{x_{1,2}}{2} & 1 \end{bmatrix},\quad \hat{L}_2(x) = \begin{bmatrix} \frac{1}{3}-\frac{1}{3}x_2 & \frac{1}{3} & \frac{1}{3}\\ \frac{1}{3}x_1^Tx_1-x_2 & 1 & 1 \end{bmatrix}. \end{gathered} \tag{4.5}$$

The Lagrange multiplier expressions are

$$\lambda_1 = \frac{-x_1^T\nabla_{x_1}f_1}{2q_1},\ \lambda_{2,1} = \frac{(1-x_2)}{3q_2}\cdot\frac{\partial f_2}{\partial x_2},\ \lambda_{2,2} = \frac{x_1^Tx_1-3x_2}{3q_2}\cdot\frac{\partial f_2}{\partial x_2}. \tag{4.6}$$

In section 4.1.2, we show that if none of the $g_{i,j}$ is identically zero, then a rational expression for λ_i always exists.

4.1.1 Optimality conditions and rational expressions

Suppose for each i, there exists a rational expression $\hat{\lambda}_i/q_i$ for the ith player's Lagrange multiplier vector. Since $q_i(x)\lambda_{i,j} = \hat{\lambda}_i(x)$ and $q_i(x) \geq 0$ for all $x \in X$, the following holds for all GNEs

$$\begin{cases} q_i(x)\nabla_{x_i} f_i(x) - \sum_{j=1}^{m_i} \hat{\lambda}_{i,j}(x)\nabla_{x_i} g_{i,j}(x) = 0 \, (i \in [N]), \\ \hat{\lambda}_i(x) \perp g_i(x), g_{i,j}(x) = 0 \, (j \in \mathcal{E}_i, i \in [N]), \\ g_{i,j}(x) \geq 0, \hat{\lambda}_{i,j}(x) \geq 0 \, (j \in \mathcal{I}_i, i \in [N]). \end{cases} \tag{4.7}$$

Under some constraint qualifications, if x is a GNE, then it satisfies (4.7). For convex GNEPs, if x satisfies (4.7) and $q_i(x) > 0$, then x must be a GNE, since it satisfies (2.9) with $\lambda_{i,j}$ given by $\lambda_{i,j} = \hat{\lambda}_{i,j}(x)/q_i(x)$. This leads us to consider the following optimization problem

$$\begin{cases} \min\limits_{x \in X} & [x]_1^T \Theta [x]_1 \\ s.t. & q_i(x)\nabla_{x_i} f_i(x) - \sum_{j=1}^{m_i} \hat{\lambda}_{i,j}(x)\nabla_{x_i} g_{i,j}(x) = 0 \, (i \in [N]), \\ & \hat{\lambda}_{i,j}(x) \perp g_{i,j}(x) \, (j \in \mathcal{E}_i \cup \mathcal{I}_i, i \in [N]), \\ & \hat{\lambda}_{i,j}(x) \geq 0 \, (j \in \mathcal{I}_i, i \in [N]). \end{cases} \tag{4.8}$$

In the above, Θ is a generically chosen positive definite matrix. The following proposition is straightforward.

Proposition 4.3. *For the GNEPP given by (1.1), suppose for each $i \in [N]$, the Lagrange multiplier vector λ_i has the rational expression as in Definition 4.1.*

(i) If (4.8) is infeasible, then the GNEP has no KKT points. Therefore, if every GNE is a KKT point, then the infeasibility of (4.8) implies the nonexistence of GNEs.

(ii) Assume the GNEP is convex. If u is a feasible point of (4.8) and $q_i(u) > 0$ for all $i \in [N]$, then u must be a GNE.

In Proposition 4.3 (ii), if $q_i(u) = 0$, then u may not be a GNE. The following is such an example.

Example 4.4. [35, Example A.8] Consider the 3-player convex GNEP

$$\begin{array}{ll|ll|ll} \min\limits_{x_1 \in \mathbb{R}^1} & -x_1 & \min\limits_{x_2 \in \mathbb{R}^1} & (x_2 - 0.5)^2 & \min\limits_{x_3 \in \mathbb{R}^1} & (x_3 - 1.5x_1)^2 \\ s.t. & x_3 \leq x_1 + x_2 \leq 1, & s.t. & x_3 \leq x_1 + x_2 \leq 1, & s.t. & 0 \leq x_3 \leq 2. \\ & x_1 \geq 0; & & x_2 \geq 0; & & \end{array}$$

The first two players have rational expressions for Lagrange multipliers ($i = 1, 2$):

$$\begin{array}{ll} \lambda_{i,1} = \frac{x_i(1-x_1-x_2)}{q_i}\frac{\partial f_i}{\partial x_i}, & \lambda_{i,2} = \frac{-x_i(x_1+x_2-x_3)}{q_i}\frac{\partial f_i}{\partial x_i}, \\ \lambda_{i,3} = \frac{\partial f_i}{\partial x_i} - \lambda_{i,1} + \lambda_{i,2}, & q_i(x) = x_i(1-x_3). \end{array} \tag{4.9}$$

For the third player, we have the polynomial expression

$$\lambda_{3,1} = \frac{2-x_3}{2}\frac{\partial f_3}{\partial x_3}, \quad \lambda_{3,2} = \lambda_{3,1} - \frac{\partial f_3}{\partial x_3}. \tag{4.10}$$

Let $q_3(x) = 1$. Then $u_1 = 0, u_2 = 0.5, u_3 = 0$ satisfy (2.9) with $q_1(u) = 0$. However, $u_1 = 0$ is not a minimizer for the first player's optimization $\mathrm{F}_1(u_{-1})$. It is interesting to note that for $u_1 = \frac{2}{3}, u_2 = \frac{1}{3}, u_3 = 1$, the tuple $u = (u_1, u_2, u_3)$ satisfies (2.9) with $q_1(u) = q_2(u) = 0$, but u is still a GNE [35].

We would like to remark that for some special GNEPs, the equality $q_i(u) = 0$ may imply that u_i is a minimizer of $\mathrm{F}_i(u_{-i})$. See Example 4.8 for such a case.

4.1.2 Existence of rational expressions

We study the existence of rational expressions with nonnegative $q_i(x)$. The following is a useful lemma.

Lemma 4.5. *For the ith player's optimization $F_i(x_{-i})$, if every $g_{i,j}(x)$ is not identically zero, then a rational expression exists for λ_i.*

Proof. Let $H_i(x) = G_i(x)^T G_i(x)$, where $G_i(x)$ is the matrix polynomial in (3.5). If every $g_{i,j}(x)$ is not identically zero, then the determinant $\det H_i(x)$ is also not identically zero. Let $\operatorname{adj} H_i(x)$ denote the adjoint matrix of $H_i(x)$, then

$$H_i(x) \cdot \operatorname{adj} H_i(x) \;=\; \det H_i(x) \cdot I_{m_i}.$$

For $\hat{L}_i(x) := G_i(x)^T \cdot \operatorname{adj} H_i(x)$, we get the rational expression

$$\lambda_{i,j}(x) \;=\; \frac{1}{\det H_i(x)} \hat{L}_i(x) \cdot \hat{f}_i(x). \tag{4.11}$$

Moreover, $q_i(x) \geq 0$ for all x, since $H_i(x)$ is positive semidefinite everywhere. □

The rational expression in (4.11) may not be very practical, because the determinantal polynomials often have high degrees. In practice, we usually have rational expressions with low degrees. If each $q_i(x) > 0$ for all $x \in X$, then every solution of (4.8) is a GNE. One wonders when a rational expression exists with $q_i(x) > 0$ on X. The matrix polynomial G_i is said to be *nonsingular on* X if $G_i(x)$ has full column rank for all $x \in X$. For the GNEP given in Example 4.2, both $G_1(x)$ and $G_2(x)$ are nonsingular on X. The following proposition is useful.

Proposition 4.6. *The matrix $G_i(x)$ is nonsingular on X if and only if there exists a matrix polynomial $\hat{L}_i(x)$ satisfying (4.2) with $q_i(x) > 0$ on X.*

Proof. First, if $G_i(x)$ has full column rank for all $x \in X$, let $H_i(x) := G_i(x)^T G_i(x)$, then $H_i(x)$ is positive definite and the determinant $\det H_i(x) > 0$ for all $x \in X$. Therefore, for $\hat{L}_i(x) := \operatorname{adj} H_i(x)$, the equation (4.11) is satisfied with $q_i(x) := \det H_i(x) > 0$ over X. Second, if (4.2) holds with $q_i(x) > 0$ on X, then $G_i(x)$ is clearly nonsingular on X. □

If $G_i(x)$ is nonsingular on X, then the LICQ must hold for the ith player's optimization. For such a case, every GNE must be a KKT point. We remark that even for the case $q_i(x) < 0$ for some $x \in X$, it is still possible to get a GNE. We refer to Example 4.23 for such a case.

4.1.3 A numerical method for finding rational expressions

We give a numerical method for finding rational expressions for Lagrange multipliers. It was introduced in [100] for solving bilevel optimization problems. Let $G_i(x)$ be the matrix polynomial defined in (3.5). For convenience, denote the tuples

$$g_{\mathcal{E}} := (g_{i,j})_{i\in[N], j\in\mathcal{E}_i}, \quad g_{\mathcal{I}} := (g_{i,j})_{i\in[N], j\in\mathcal{I}_i}.$$

For a priori degree d, consider the following linear convex optimization:

$$\begin{cases} \max\limits_{\hat{L}_i, q_i, \gamma} & \gamma \\ \quad s.t. & \hat{L}_i \cdot G_i = q_i \cdot I_{m_i},\ q_i(v) = 1, \\ & q_i - \gamma \in \operatorname{Ideal}[g_{\mathcal{E}}]_{2d} + \operatorname{Qmod}[g_{\mathcal{I}}]_{2d}, \\ & \hat{L}_i \in (\mathbb{R}[x]_{2d-\deg G_i})^{m_i \times (m_i+n_i)}. \end{cases} \tag{4.12}$$

In the above, the first equality is the same as (4.2). The second equality ensures that q_i is not identically zero, where v is a priori point in X. The constraint $q_i - \gamma \in \operatorname{Ideal}[g_{\mathcal{E}_i}] + \operatorname{Qmod}[g_{\mathcal{E}_i}]$ forces the $q_i(x) \geq \gamma$ on X. Therefore, if the maximum γ is positive, then $q_i(x) > 0$ on X. By

Lemma 4.5, one can always find a feasible $\gamma \geq 0$ satisfying (4.12), for some $d \leq \deg(H(x))$, if none of $g_{i,j}(x)$ is identically zero. By Proposition 4.6, if each $G_i(x)$ is nonsingular on X and the archimedeanness holds for X, then there must exist $\gamma > 0$ satisfying (4.12) for some d. If $(\hat{L}_i, q_i, \gamma)$ is a feasible point of (4.12), then one can get a rational expression for Lagrange multipliers by letting $\hat{\lambda}_{i,j}(x) = \hat{L}_i(x)\hat{f}_i$.

Example 4.7. Consider the GNEP in Example 4.2. We have

$$g_{\mathcal{E}} = \emptyset,\ g_{\mathcal{I}} = (2 - x_1^T x_1 - x_2, 3x_2 - x_1^T x_1, 1 - x_2).$$

Let $\hat{L}_1(x)$ and $\hat{L}_2(x)$ be the matrix polynomials in (4.5), and $q_1(x) = 2-x_2, q_2(x) = 1-\frac{1}{3}x_1^T x_1$. Let $v := (0, 0, 1)$ for both players, and $\gamma_1 = 1$, $\gamma_2 = 1/2$. Then, the $(\hat{L}_1(x), q_1(x), \gamma_1)$ and $(\hat{L}_2(x), q_2(x), \gamma_2)$ are feasible points of (4.12), for $i = 1, 2$ respectively. In fact, we have $q_1(v) = q_2(v) = 1$, and

$$\begin{aligned} &q_1(x) - \gamma_1 = 1 - x_2 = 0 + 1 \cdot (1 - x_2) \in \mathrm{Qmod}[g_{\mathcal{I}}]_2, \\ &q_2(x) - \gamma_2 = \tfrac{1}{2} - \tfrac{1}{3}x_1^T x_1 = 0 + \tfrac{1}{4}(2 - x_1^T x_1 - x_2) + \tfrac{1}{12}(3x_2 - x_1^T x_1) \in \mathrm{Qmod}[g_{\mathcal{I}}]_2. \end{aligned}$$

The rational expressions for Lagrange multipliers are given by (4.6).

Example 4.8. Consider the following GNEP

$$\begin{array}{cl|cl} \min\limits_{x_1 \in \mathbb{R}^3} & f_1(x_1, x_2) & \min\limits_{x_2 \in \mathbb{R}^3} & f_2(x_1, x_2) \\ s.t. & 1 - x_1^T x_1 - x_2^T x_2 \geq 0; & s.t. & 1 - x_1^T x_1 - x_2^T x_2 \geq 0. \end{array}$$

The constraining tuples $g_{\mathcal{E}} := \emptyset,\ g_{\mathcal{I}} := (1 - x_1^T x_1 - x_2^T x_2)$. Let $v := (0, 0, 0)$, $\gamma_1 = \gamma_2 = 0$, $q_1(x) = 1 - x_2^T x_2$, $q_2(x) = 1 - x_1^T x_1$, and

$$\hat{L}_1 = \left[-\frac{1}{2}x_{1,1},\ -\frac{1}{2}x_{1,2},\ -\frac{1}{2}x_{1,3},\ 1 \right],\ \hat{L}_2 = \left[-\frac{1}{2}x_{2,1},\ -\frac{1}{2}x_{2,2},\ -\frac{1}{2}x_{2,3},\ 1 \right].$$

One can verify that $q_1(v) = q_2(v) = 1$ and

$$\begin{aligned} q_1(x) - \gamma_1 = 1 - x_2^T x_2 = x_1^T x_1 + 1 \cdot (1 - x_1^T x_1 - x_2^T x_2) \in \mathrm{Qmod}[g_{\mathcal{I}}]_2, \\ q_2(x) - \gamma_2 = 1 - x_1^T x_1 = x_2^T x_2 + 1 \cdot (1 - x_1^T x_1 - x_2^T x_2) \in \mathrm{Qmod}[g_{\mathcal{I}}]_2. \end{aligned}$$

By Proposition 4.6, we know $(\hat{L}_1(x), q_1(x), \gamma_1)$ and $(\hat{L}_2(x), q_2(x), \gamma_2)$ are minimizers of (4.12) for $i = 1, 2$ respectively. Therefore, we get the rational expression

$$\lambda_1 = \frac{-x_1^T \nabla_{x_1} f_1}{2 \cdot q_1(x)},\ \lambda_2 = \frac{-x_2^T \nabla_{x_2} f_2}{2 \cdot q_2(x)}. \tag{4.13}$$

For each $i = 1, 2$, if $q_i(x) = 0$, then $0 \le {x_i}^T x_i \le 1 - {x_{-i}}^T x_{-i} = 0$. This implies $x_i = (0, 0, 0)$ is the only feasible point of the ith player's optimization and hence it is the minimizer. Therefore, each feasible point of (4.8) is a GNE.

One can solve (4.12) numerically for getting rational expressions. This is done in Example 4.22.

4.2 Parametric expressions for Lagrange multipliers

For some GNEPs, it may be difficult to find convenient rational expressions for Lagrange multipliers. Sometimes, the denominators may have high degrees. This is the case especially when $m_i > n_i$. If some q_i has high degree, the polynomial optimization (4.7) also has a high degree, which makes the result moment SDP relaxations (see subsections 4.3.1 and 4.3.2) very difficult to be solved. To fix such issues, we introduce parametric expressions for Lagrange multipliers.

Definition 4.9. For the ith player's optimization $\mathrm{F}_i(x_{-i})$, a parametric expression for the Lagrange multipliers is a tuple of polynomials

$$\hat{\lambda}_i(x, \omega_i) := (\hat{\lambda}_{i,1}(x, \omega_i), \dots, \hat{\lambda}_{i,m_i}(x, \omega_i)),$$

in x and in a parameter $\omega_i := (\omega_{i,1}, \dots, \omega_{i,s_i})$ with $s_i \le m_i$, such that (x_i, λ_i) is a critical pair if and only if there is a value of ω_i such that (3.4) is satisfied for $\lambda_{i,j} = \hat{\lambda}_{i,j}(x, \omega_i)$ with $j \in [m_i]$.

The following is an example of parametric expressions.

Example 4.10. Consider the 2-player convex GNEP

$$\begin{array}{cl|cl}
\min\limits_{x_1 \in \mathbb{R}^2} & f_1(x_1, x_2) & \min\limits_{x_2 \in \mathbb{R}^2} & f_2(x_1, x_2) \\
s.t. & x_{1,1} - 2x_{1,2} + x_{2,2} \ge 0, & s.t. & x_{1,2} + x_{2,2} - x_{2,1}^2 + 1 \ge 0, \\
 & 1 - x_{2,1} \cdot x_1^T x_1 \ge 0, & & 2 - x_{2,2} \ge 0, 1 + x_{2,2} \ge 0, \\
 & x_{1,1} \ge 0, x_{1,2} \ge 0; & & x_{2,1} \ge 0.
\end{array}$$

The Lagrange multipliers can be expressed as

$$\begin{cases} \lambda_{1,1} = \omega_{1,1}, \\ \lambda_{1,2} = \frac{1}{2}x_{1,1}(\frac{\partial f_1}{\partial x_{1,1}} - \omega_{1,1}) + \frac{1}{2}x_{1,2}(\frac{\partial f_1}{\partial x_{1,2}} + 2\omega_{1,1}), \\ \lambda_{1,3} = \frac{\partial f_1}{\partial x_{1,1}} - \omega_{1,1} + 2x_{2,1}x_{1,1}\lambda_{1,2}, \\ \lambda_{1,4} = \frac{\partial f_1}{\partial x_{1,2}} + 2\omega_{1,1} + 2x_{2,1}x_{1,2}\lambda_{1,2}; \\ \lambda_{2,1} = \omega_{2,1}, \\ \lambda_{2,2} = -\frac{1}{3}\cdot\left[(\frac{\partial f_2}{\partial x_{2,1}} + 2x_{2,1}\omega_{2,1})x_{2,1} + (\frac{\partial f_2}{\partial x_{2,2}} - \omega_{2,1})(x_{2,2}+1)\right], \\ \lambda_{2,3} = \frac{\partial f_2}{\partial x_{2,2}} + \lambda_{2,2} - \omega_{2,1}, \\ \lambda_{2,4} = \frac{\partial f_2}{\partial x_{2,1}} + 2x_{2,1}\omega_{2,1}. \end{cases} \tag{4.14}$$

Parametric expressions are quite useful for solving the GNEPs. The following are some useful cases.

(i) Suppose the ith player's optimization $\mathrm{F}_i(x_{-i})$ contains the nonnegative constraints, i.e., its constraints are

$$x_{i,1} \geq 0, \dots, x_{i,n_i} \geq 0, \quad g_{i,j}(x) \geq 0\,(j = n_i+1, \dots, m_i).$$

Let $s_i := m_i - n_i$, then a parametric expression is

$$\boxed{\begin{array}{l} (\lambda_{i,1}, \dots, \lambda_{i,n_i}) = \nabla_{x_i} f_i - \sum_{k=1}^{s_i} \omega_{i,k} \cdot \nabla_{x_i} g_{i,k+n_i}, \\ (\lambda_{i,n_i+1}, \dots, \lambda_{i,m_i}) = (\omega_{i,1}, \dots, \omega_{i,s_i}). \end{array}} \tag{4.15}$$

(ii) Suppose the ith player's optimization $\mathrm{F}_i(x_{-i})$ contains box constraints, i.e., its constraints are

$$x_{i,j} - a_{i,j} \geq 0,\; b_{i,j} - x_{i,j} \geq 0, \quad j = 1, \dots, n_i$$
$$g_{i,j}(x) \geq 0. \quad j = n_i+1, \dots, m_i$$

Let $s_i := m_i - 2n_i$, then a parametric expression is

$$\boxed{\begin{array}{ll} \lambda_{i,j} = \frac{b-x_{i,j}}{b-a}\cdot\left(\frac{\partial f_i}{\partial x_{i,j}} - \sum_{k=1}^{s_i}\omega_{i,k}\cdot\frac{\partial g_{i,k+2n_i}}{\partial x_{i,j}}\right), & j = 1, 3, \dots, 2n_i - 1 \\ \lambda_{i,j} = \frac{a-x_{i,j}}{b-a}\cdot\left(\frac{\partial f_i}{\partial x_{i,j}} - \sum_{k=1}^{s_i}\omega_{i,k}\cdot\frac{\partial g_{i,k+2n_i}}{\partial x_{i,j}}\right), & j = 2, 4, \dots, 2n_i \\ \lambda_{i,j} = \omega_{i,j-2n_i}. & j = 2n_i+1, \dots, m_i \end{array}} \tag{4.16}$$

(iii) Suppose the ith player's optimization $\mathrm{F}_i(x_{-i})$ contains simplex constraints, i.e., its constraints are

$$1 - e^T x_i \geq 0, x_{i,1} \geq 0, \dots, x_{i,n_i} \geq 0,\; g_{i,j}(x) \geq 0,\; j = n_i+2, \dots, m_i.$$

Let $s_i := m_i - n_i - 1$, then a parametric expression is

$$\boxed{\begin{array}{ll} \lambda_{i,j} = (\nabla_{x_i} f_i - \sum_{k=1}^{s_i} \omega_{i,k} \cdot \nabla_{x_i} g_{i,k+n_i+1})^T x_i, & j = 1 \\ \lambda_{i,j} = \frac{\partial f_i}{\partial x_{i,j-1}} - \sum_{k=1}^{s_i} \omega_{i,k} \cdot \frac{\partial g_{i,k+n_i+1}}{\partial x_{i,j-1}} - \lambda_{i,1}, & j = 2, \dots, n_i + 1 \\ \lambda_{i,j} = \omega_{i,j-n_i-1}. & j = n_i + 2, \dots, m_i \end{array}} \tag{4.17}$$

(iv) Suppose the ith player's optimization $\mathrm{F}_i(x_{-i})$ contains linear constraints, i.e., its constraints are

$$a_j^T x_i - b_j(x_{-i}) \geq 0,\ j = 1, \dots, r, \quad g_{i,j}(x) \geq 0,\ j = r + 1, \dots, m_i,$$

where each b_j is a polynomial in x_{-i}. Let $A = \begin{bmatrix} a_1 & \cdots & a_r \end{bmatrix}^T$. Assume $\operatorname{rank} A = r$. If we let $s_i := m_i - r$, then a parametric expression is

$$\boxed{\begin{array}{l} (\lambda_{i,1}, \dots, \lambda_{i,r}) = (AA^T)^{-1} A (\nabla_{x_i} f_i - \sum_{k=1}^{s_i} \omega_{i,k} \cdot \nabla_{x_i} g_{i,k+r}), \\ (\lambda_{i,r+1}, \dots, \lambda_{i,m_i}) = (\omega_{i,1}, \dots, \omega_{i,s_i}). \end{array}}$$

(v) Suppose there exists a label subset $\mathcal{T}_i := (t_1, \dots, t_r) \subseteq [m_i]$ such that

$$\hat{G}_i(x) := \begin{bmatrix} \nabla_{x_i} g_{i,t_1}(x) & \dots & \nabla_{x_i} g_{i,t_r}(x) \\ g_{i,t_1}(x) & & \\ & \ddots & \\ & & g_{i,t_r}(x) \end{bmatrix}$$

is nonsingular for all $x \in \mathbb{C}^n$. By [95, Proposition 5.1], there exists a matrix polynomial $D_i(x)$ such that $D_i(x) \cdot \hat{G}_i(x) = I_r$. Let $s_i := m_i - r$, then a parametric expression is

$$\boxed{\begin{array}{l} (\lambda_{i,1}, \dots, \lambda_{i,r}) = D_i(x)(\nabla_{x_i} f_i - \sum_{k=1}^{s_i} \omega_{i,k} \cdot \nabla_{x_i} g_{i,k+r}), \\ (\lambda_{i,r+1}, \dots, \lambda_{i,m_i}) = (\omega_{i,1}, \dots, \omega_{i,s_i}). \end{array}}$$

We would like to remark that a parametric expression always exists. For instance, one can set $\omega_{i,j} = \lambda_{i,j}$ for all j. However, it is preferable to have small s_i, to save computational costs.

4.2.1 Optimality conditions and parametric expressions

Suppose all players have parametric expressions for their Lagrange multipliers as in Definition 4.9. Let $s := s_1 + \ldots + s_N$, and denote

$$\mathbf{x} := (x, \omega_1, \dots, \omega_N).$$

The optimality conditions (2.9) can be equivalently expressed as

$$\begin{cases} \nabla_{x_i} f_i(x) - \sum_{j=1}^{m_i} \hat{\lambda}_{i,j}(x) \nabla_{x_i} g_{i,j}(x) = 0 \, (i \in [N]), \\ \hat{\lambda}_i(x) \perp g_i(x), g_{i,j}(x) = 0 \, (j \in \mathcal{E}_i, i \in [N]), \\ g_{i,j}(x) \geq 0, \hat{\lambda}_{i,j}(x) \geq 0 \, (j \in \mathcal{I}_i, i \in [N]). \end{cases} \tag{4.18}$$

For convex GNEPs, a point x is a GNE if and only if there exists $\omega := (\omega_1, \dots, \omega_N)$ such that x satisfies (4.18). Therefore, we consider the optimization

$$\begin{cases} \min\limits_{x \in X \times \mathbb{R}^s} & [x]_1^T \, \Theta \, [x]_1 \\ \quad s.t. & \nabla_{x_i} f_i(x) - \sum_{j=1}^{m_i} \hat{\lambda}_{i,j}(x) \nabla_{x_i} g_{i,j}(x) = 0 \, (i \in [N]), \\ & \hat{\lambda}_{i,j}(x) \perp g_{i,j}(x) \, (j \in \mathcal{E}_i \cup \mathcal{I}_i, i \in [N]), \\ & \hat{\lambda}_{i,j}(x) \geq 0 \, (j \in \mathcal{I}_i, i \in [N]). \end{cases} \tag{4.19}$$

In the above, the Θ is a generically chosen positive definite matrix. The following proposition is straightforward

Proposition 4.11. *For the GNEPP given by (1.1), suppose each player's optimization has a parametric expression for their Lagrange multipliers as in Definition 4.9.*

(i) If (4.19) is infeasible, then the GNEP has no KKT points. If every GNE is a KKT point, then the infeasibility of (4.19) implies nonexistence of GNEs.

(ii) Assume the GNEP is convex. If (u, w) is a feasible point of (4.19), then u is a GNE.

4.3 The polynomial optimization reformulation

In this section, we give an algorithm for solving convex GNEPs. We assume each λ_i has either a rational or parametric expression, as in Definition 4.1 or 4.9. If λ_i has a polynomial or parametric expression, we let $q_i(x) := 1$. If λ_i has a polynomial or rational expression, then we let $s_i = 0$. Recall the notation

$$x := (x, \omega_1, \dots, \omega_N).$$

Choose a generic positive definite matrix Θ. Then solve the following polynomial optimization

$$\begin{cases} \min\limits_{x} & [x]_1^T \,\Theta\, [x]_1 \\ s.t. & q_i(x)\nabla_{x_i} f_i(x) - \sum_{j=1}^{m_i} \hat{\lambda}_{i,j}(x)\nabla_{x_i} g_{i,j}(x) = 0\,(i \in [N]), \\ & \hat{\lambda}_{i,j}(x) \perp g_{i,j}(x)\,(j \in \mathcal{E}_i \cup \mathcal{I}_i, i \in [N]), \\ & g_{i,j}(x) = 0\,(j \in \mathcal{E}_i, i \in [N]), \\ & g_{i,j}(x) \geq 0\,(j \in \mathcal{I}_i, i \in [N]), \\ & \hat{\lambda}_{i,j}(x) \geq 0\,(j \in \mathcal{I}_i, i \in [N]). \end{cases} \tag{4.20}$$

If (4.20) is infeasible, then there are no KKT points. Since Θ is positive definite, if (4.20) is feasible, then it must have a minimizer, say, $(u, w) \in X \times \mathbb{R}^s$. For convex GNEPs, if $q_i(u) > 0$ for all i, then u must be a GNE. If $q_i(u) \leq 0$ for some i, then u may or may not be a GNE. To check this, we solve the following optimization problem for those i with $q_i(u) \leq 0$

$$\begin{cases} \delta_i := & \min\limits_{x_i} & f_i(x_i, u_{-i}) - f_i(u_i, u_{-i}) \\ & s.t. & g_{i,j}(x_i, u_{-i}) = 0\,(j \in \mathcal{E}_i),\;\; g_{i,j}(x_i, u_{-i}) \geq 0\,(j \in \mathcal{I}_i). \end{cases} \tag{4.21}$$

This is a polynomial optimization in x_i. Since $u \in X$, the point u_i is feasible for (4.21), so $\delta_i \leq 0$. If $\delta_i \geq 0$ for all i, then u must be a GNE. The following is an algorithm for solving the GNEP.

Algorithm 4.12. For the convex GNEP given by (1.1), do the following:

Step 0 Choose a generic positive definite matrix Θ of length $n + s + 1$.

Step 1 Solve the polynomial optimization (4.20). If it is infeasible, then there are no KKT points and stop; otherwise, solve it for a minimizer (u, w).

Step 2 If all $q_i(u) > 0$, then u is a GNE. Otherwise, for those i with $q_i(u) \leq 0$, solve the optimization (4.21) for the minimum value δ_i. If $\delta_i \geq 0$ for all such i, then u is a GNE; otherwise, it is not.

In Step 0, we can choose $\Theta = R^T R$ for a randomly generated square matrix R of length $n + s + 1$. The objective in (4.20) is a positive definite quadratic function, so it must have a minimizer if (4.20) is feasible. Since the objective $f_i(x_i, u_{-i})$ is assumed to be convex in x_i, if it is bounded from below on $X_i(u_{-i})$, then (4.21) must have a minimizer (see [15, Theorem 3]). In applications, we are mostly interested in cases that (4.21) has a minimizer, for the existence of a GNE. In the subsections 4.3.1 and 4.3.2, we will discuss how to solve polynomial optimization problems in Algorithm 4.12, by the Moment-SOS hierarchy of semidefinite relaxations. The convergence of Algorithm 4.12 is shown as follows.

Theorem 4.13. *For the convex GNEPP given by (1.1), suppose each Lagrange multiplier vector λ_i has a rational expression as in Definition 4.1 or a parametric expression as in Definition 4.9.*

(i) If (u, w) is a feasible point of (4.20) such that $q_i(u) > 0$ for all i, then u is a GNE.

(ii) Assume every GNE is a KKT point. If (4.20) is infeasible, then the GNEP has no GNEs. If Θ is positive definite and every $q_i(x) > 0$ for all feasible x of (4.20), then Algorithm 4.12 will find a GNE if it exists.

Proof. (i) This is directly implied by Propositions 4.3 and 4.11.

(ii) If (4.20) is infeasible, then there is no GNE, because every GNE is assumed to be a KKT point and it must be feasible for (4.20). Next, assume (4.20) is feasible. Since Θ is positive definite, the optimization (4.20) has a minimizer, say, (u, w). By the given assumption, we have $q_i(u) > 0$ for all i. So u is a GNE, by the item (i). □

In Theorem 4.13(ii), if $q_i(x) > 0$ for all $x \in X$, then we must have $q_i(x) > 0$ for all feasible x of (4.20). Suppose (u, w) is a computed minimizer of (4.20). If u is not a GNE, i.e., $\delta_i < 0$ for some i, we can let $\mathcal{N} \subseteq [N]$ be the labeling set of i with $\delta_i < 0$. By Theorem 4.13, we know $q_i(u) = 0$ for all $i \in \mathcal{N}$. For a priori small $\varepsilon > 0$, we can add the inequalities $q_i(x) \geq \varepsilon \, (i \in \mathcal{N})$ to the optimization (4.20), to exclude u from the feasible set. Then we solve the following new optimization

$$\left\{ \begin{array}{cl} \min\limits_{x \in X \times \mathbb{R}^s} & [x]_1^T \, \Theta \, [x]_1 \\ s.t. & q_i(x) \nabla_{x_i} f_i(x) - \sum_{j=1}^{m_i} \hat{\lambda}_{i,j}(x) \nabla_{x_i} g_{i,j}(x) = 0 \, (i \in [N]), \\ & \hat{\lambda}_{i,j}(x) \perp g_{i,j}(x) \, (j \in \mathcal{E}_i \cup \mathcal{I}_i, i \in [N]), \\ & \hat{\lambda}_{i,j}(x) \geq 0 \, (j \in \mathcal{I}_i, i \in [N]), \\ & q_i(x) \geq \varepsilon \, (i \in \mathcal{N}). \end{array} \right. \tag{4.22}$$

If $\varepsilon > 0$ is not small, the constraint $q_i(x) \geq \varepsilon$ may also exclude some GNEs. If the new optimization (4.22) is infeasible, one can heuristically get a candidate GNE by choosing a different generic positive definite Θ in (4.20). In computational practice, when a GNE exists, it is very likely that we can get one by doing this. However, how to detect nonexistence of GNEs when (4.20) is feasible can be theoretically difficult. The theoretical side of this problem is mostly open, to the best of the authors' knowledge.

4.3.1 The optimization for all players

We discuss how to solve the polynomial optimization problems in Algorithm 4.12, by using the Moment-SOS hierarchy of semidefinite relaxations [61, 63, 64, 67, 68]. We refer to the notation in subsections 2.1 and 2.2.

First, we discuss how to solve the optimization (4.20). Denote the polynomial tuples

$$\Phi_i := \Big\{q_i(x)\nabla_{x_i} f_i(\mathbf{x}) - \sum_{j=1}^{m_i} \hat{\lambda}_{i,j}(\mathbf{x})\nabla_{x_i} g_{i,j}(x)\Big\} \cup \Big\{g_{i,j}(x) : j \in \mathcal{E}_i\Big\} \cup \Big\{\hat{\lambda}_{i,j}(\mathbf{x}) \cdot g_{i,j}(x) : j \in \mathcal{I}_i\Big\}, \quad (4.23)$$

$$\Psi_i \;:=\; \Big\{g_{i,j}(x) \;:\; j \;\in\; \mathcal{I}_i\Big\} \cup \Big\{\hat{\lambda}_{i,j}(\mathbf{x}) \;:\; j \;\in\; \mathcal{I}_i\Big\}. \quad (4.24)$$

For notational convenience, for a vector $p = (p_1, \ldots, p_s)$, the set $\{p\}$ stands for $\{p_1, \ldots, p_s\}$, in the above. Denote the unions

$$\Phi \;:= \bigcup_{i=1}^{N} \Phi_i, \quad \Psi \;:= \bigcup_{i=1}^{N} \Psi_i.$$

They are both finite sets of polynomials. Then, the optimization (4.20) can be equivalently written as

$$\left\{\begin{array}{rl} \vartheta_{\min} := \min\limits_{\mathbf{x}} & \theta(\mathbf{x}) := [\mathbf{x}]_1^T \Theta [\mathbf{x}]_1 \\ s.t. & p(\mathbf{x}) = 0 \ (\forall\, p \in \Phi), \\ & q(\mathbf{x}) \geq 0 \ (\forall\, q \in \Psi). \end{array}\right. \quad (4.25)$$

Denote the degree

$$d_0 \;:=\; \max\{\lceil \deg(p)/2 \rceil \,:\, p \in \Phi \cup \Psi\}.$$

For a degree $k \geq d_0$, consider the kth order Lasserre type semidefinite moment relaxation for solving (4.25)

$$\left\{\begin{array}{rl} \vartheta_k \;:=\; \min\limits_{y} & \langle \theta, y \rangle \\ s.t. & y_0 = 1,\; L_p^{(k)}[y] = 0\,(p \in \Phi), \\ & M_k[y] \succeq 0,\; L_q^{(k)}[y] \succeq 0\,(q \in \Psi), \\ & y \in \mathbb{R}^{\mathbb{N}_{2k}^{n+s}}. \end{array}\right. \quad (4.26)$$

Its dual optimization problem is the kth order SOS relaxation

$$\begin{cases} \max & \gamma \\ s.t. & \theta - \gamma \in \text{Ideal}[\Phi]_{2k} + \text{Qmod}[\Psi]_{2k}. \end{cases} \tag{4.27}$$

For relaxation orders $k = d_0, d_0 + 1, \ldots$, we get the Moment-SOS hierarchy of semidefinite relaxations (4.26)-(4.27). This produces the following algorithm for solving the polynomial optimization problem (4.25).

Algorithm 4.14. Let θ, Φ, Ψ be as in (4.25). Initialize $k := d_0$.

Step 1 Solve the semidefinite relaxation (4.26). If it is infeasible, then (4.25) has no feasible points and stop; otherwise, solve it for a minimizer y^*.

Step 2 Let $\mathbf{u} = (u, w) := (y^*_{e_1}, \ldots, y^*_{e_{n+s}})$. If $\mathbf{u}$ is feasible for (4.25) and $\vartheta_k = \theta(u)$, then $\mathbf{u}$ is a minimizer of (4.25). Otherwise, let $k := k + 1$ and go to Step 1.

In the Step 2, e_i denotes the labeling vector such that its ith entry is 1 while all other entries are 0. For instance, when $n = s = 2$, $y_{e_3} = y_{0010}$. The optimization (4.26) is a relaxation of (4.25). This is because if $\mathbf{x}$ is a feasible point of (4.25), then $y = [\mathbf{x}]_{2k}$ must be feasible for (4.26). Hence, if (4.26) is infeasible, then (4.25) must be infeasible, which also implies the nonexistence of KKT points. Moreover, the optimal value ϑ_k of (4.26) is a lower bound for the minimum value of (4.25), i.e., $\vartheta_k \le \theta(\mathbf{x})$ for all x that is feasible for (4.25). In the Step 2, if $\mathbf{u}$ is feasible for (4.25) and $\vartheta_k = \theta(\mathbf{u})$, then $\mathbf{u}$ must be a minimizer of (4.25). The Algorithm 4.14 can be implemented in `GloptPoly` [49]. The convergence of Algorithm 4.14 is shown as follows.

Theorem 4.15. *Assume the set Ideal*$[\Phi]$ + *Qmod*$[\Psi] \subseteq \mathbb{R}[\mathbf{x}]$ *is archimedean.*

(i) If (4.25) is infeasible, then the moment relaxation (4.26) must be infeasible when the order k is big enough.

(ii) Suppose (4.25) is feasible and Θ is a generic positive definite matrix. Let $\mathbf{u}^{(k)}$ be the point $\mathbf{u}$ produced in the Step 2 of Algorithm 4.14 in the kth loop. Then $\mathbf{u}^{(k)}$ converges to the unique minimizer of (4.25). In particular, if the real zero set of Φ is finite, then $\mathbf{u}^{(k)}$ is the unique minimizer of (4.25), when k is sufficiently large.

Proof. (i) If (4.25) is infeasible, the constant polynomial -1 can be viewed as a positive polynomial on the feasible set of (4.25). Since Ideal$[\Phi]$ + Qmod$[\Psi]$ is archimedean, we have

$-1 \in \text{Ideal}[\Phi]_{2k} + \text{Qmod}[\Psi]_{2k}$, for k big enough, by the Putinar Positivstellensatz [106]. For such a big k, the SOS relaxation (4.27) is unbounded from above, hence the moment relaxation (4.26) must be infeasible.

(ii) When the optimization (4.25) is feasible, it must have a unique minimizer, say, $\mathbf{x}^*$, because its objective is a generic positive definite quadratic polynomial. The convergence of $\mathbf{u}^{(k)}$ to $\mathbf{x}^*$ is shown in [120] or [88, Theorem 3.3]. For the special case that $\Phi(\mathbf{x}) = 0$ has finitely many real solutions, the point $\mathbf{u}^{(k)}$ must be equal to $\mathbf{x}^*$, when k is large enough. This is shown in [65] (also see [89]). □

The archimedeaness of the set $\text{Ideal}[\Phi] + \text{Qmod}[\Psi]$ is essentially requiring that the feasible set of (4.25) is compact. The archimedeaness is sufficient but not necessary for Algorithm 4.14 to converge. Even if the archimedeaness fails to hold, Algorithm 4.14 is still applicable for solving (4.20). If the point $\mathbf{u}^{(k)}$ is feasible and $\vartheta_k = \theta(\mathbf{u}^{(k)})$, then $\mathbf{u}^{(k)}$ must be a minimizer of (4.20), regardless of the archimedeaness holds or not. Moreover, without archimedeaness, the infeasibility of (4.26) still implies that (4.20) is infeasible. In our computational practice, Algorithm 4.14 almost always has finite convergence.

The polynomial optimization (4.22) can be solved in the same way by the Moment-SOS hierarchy of semidefinite relaxations. The convergence property is the same. For the cleanness of this paper, we omit the details.

4.3.2 Checking Generalized Nash Equilibria

Suppose $\mathbf{u} = (u, w) \in \mathbb{R}^n \times \mathbb{R}^s$ is a minimizer of (4.20). For convex GNEPPs, if all $q_i(u) > 0$, then u is a GNE, by Theorem 4.13(i). If $q_i(u) \le 0$ for some i, we need to solve the optimization (4.21), to check if $u = (u_i, u_{-i})$ is a GNE or not, Note that (4.21) is a convex polynomial optimization problem in x_i. For given u_{-i}, if it is bounded from below, then (4.21) achieves its optimal value at a minimizer.

Consider the ith player's optimization with $q_i(u) \le 0$. For notational convenience, we denote the polynomial tuples

$$\begin{aligned} H_i(u) := &\left\{g_{i,j}(x_i, u_{-i}) : j \in \mathcal{E}_i\right\} \cup \left\{\hat{\lambda}_{i,j}(x_i, u_{-i}) \cdot g_{i,j}(x_i, u_{-i}) : j \in \mathcal{I}_i\right\} \\ &\cup \left\{q_i(x_i, u_{-i})\nabla_{x_i} f_i(x_i, u_{-i}) - \sum_{j=1}^{m_i} \hat{\lambda}_{i,j}(x_i, u_{-i})\nabla_{x_i} g_{i,j}(x_i, u_{-i})\right\}, \end{aligned} \tag{4.28}$$

$$J_i(u) := \left\{g_{i,j}(x_i, u_{-i}) : j \in \mathcal{I}_i\right\} \cup \left\{\hat{\lambda}_{i,j}(x_i, u_{-i}) : j \in \mathcal{I}_i\right\}. \tag{4.29}$$

Like in (4.23)-(4.24), the set $\{p\}$ stands for $\{p_1, \dots, p_s\}$, when $p = (p_1, \dots, p_s)$ is a vector of polynomial. The sets $H_i(u), J_i(u)$ are finite collections.

Under some suitable constraint qualification conditions (e.g., the Slater's Condition), when (4.21) has a minimizer, it is equivalent to

$$\begin{cases} \eta_i := & \min\limits_{x_i \in \mathbb{R}^{n_i}} & \zeta_i(x_i) := f_i(x_i, u_{-i}) - f_i(u_i, u_{-i}) \\ & s.t. & p(x_i) = 0 \, (p \in H_i(u)), \\ & & q(x_i) \geq 0 \, (q \in J_i(u)). \end{cases} \tag{4.30}$$

Denote the degree in variables x_i for its constraining polynomials

$$d_i := \max\big\{\lceil \deg(\zeta_i(x_i, u_{-i}))/2, \deg(p(x_i))/2, \\ \deg(q(x_i))/2 : p \in H_i(u), q \in J_i(u) \rceil\big\}. \tag{4.31}$$

For a degree $k \geq d_i$, the kth order moment relaxation for (4.25) is

$$\begin{cases} \eta_i^{(k)} := & \min\limits_{y} & \langle \zeta_i(x_i), y \rangle \\ & s.t. & y_0 = 1, \; L_p^{(k)}[y] = 0 \, (p \in H_i(u)), \\ & & M_k[y] \succeq 0, \; L_q^{(k)}[y] \succeq 0 \, (q \in J_i(u)), \\ & & y \in \mathbb{R}^{\mathbb{N}_{2k}^{n_i}}. \end{cases} \tag{4.32}$$

The dual optimization problem of (4.32) is the kth order SOS relaxation

$$\begin{cases} \max & \gamma \\ s.t. & \zeta_i(x_i) - \gamma \in \mathrm{Ideal}[H_i(u)]_{2k} + \mathrm{Qmod}[J_i(u)]_{2k}. \end{cases} \tag{4.33}$$

By solving the above relaxations for $k = d_i, d_i + 1, \dots$, we get the Moment-SOS hierarchy of relaxations (4.32)-(4.33). This gives the following algorithm.

Algorithm 4.16. For a minimizer $u = (u_i, u_{-i})$ of (4.20) with $q_i(u) \leq 0$, solve the ith player's optimization (4.30). Initialize $k := d_i$.

Step 1 Solve the moment relaxation (4.32) for the minimum value $\eta_i^{(k)}$ and a minimizer y^*. If $\eta_i^{(k)} \geq 0$, then $\eta_i = 0$ and stop; otherwise, go to the next step.

Step 2 Let $t := d_i$ as in (4.31). If y^* satisfies the rank condition

$$\operatorname{rank} M_t[y^*] \; = \; \operatorname{rank} M_{t-d_i}[y^*], \tag{4.34}$$

then extract a set U_i of $r := \operatorname{rank} M_t(y^*)$ minimizers for (4.30) and stop.

Step 3 If (4.34) fails to hold and $t < k$, let $t := t + 1$ and then go to Step 2; otherwise, let $k := k + 1$ and go to Step 1.

We would like to remark that the optimization (4.30) is always feasible, because u_i is a feasible point since u is a minimizer of (4.20). The moment relaxation (4.32) is also feasible. Because $\eta_i^{(k)}$ is a lower bound for η_i, and $\eta_i \leq \zeta_i(u_i, u_{-i}) = 0$, if $\eta_i^{(k)} \geq 0$, then η_i must be 0. In Step 2, the rank condition (4.34) is called *flat truncation* [88]. It is a sufficient (and almost necessary) condition to check convergence of moment relaxations. When (4.34) holds, the method in [48] can be used to extract r minimizers for (4.30). The Algorithm 4.16 can also be implemented in `GloptPoly` [49]. If Ideal$[H_i(u)]$ + Qmod$[J_i(u)]$ is archimedean, then $\eta_i^{(k)} \to \eta_i$ as $k \to \infty$ [61]. It is interesting to remark that

$$I_1 := \text{Ideal}[g_{i,j}(x_i, u_{-i}) : j \in \mathcal{E}_i] \subseteq \text{Ideal}[H_i(u)],$$

$$I_2 := \text{Qmod}[g_{i,j}(x_i, u_{-i}) : j \in \mathcal{I}_i] \subseteq \text{Qmod}[J_i(u)].$$

If $I_1 + I_2$ is archimedean, then Ideal$[H_i(u)]$ + Qmod$[J_i(u)]$ must also be archimedean. Furthermore, we have the following convergence theorem for Algorithm 4.16.

Theorem 4.17. *For the convex polynomial optimization (4.21), assume its optimal value is achieved at a KKT point. If either one of the following conditions hold,*

(i) The set $I_1 + I_2$ is archimedean, and the Hessian $\nabla^2_{x_i} \zeta_i(x_i^*, u_{-i}) \succ 0$ for a minimizer x_i^* of (4.30); **or**

(ii) The real zero set of polynomials in $H_i(u)$ is finite,

then Algorithm 4.16 must terminate within finitely many loops.

Proof. Since its optimal value is achieved at a KKT point, the optimization problem (4.21) is equivalent to (4.30).

(i) If $I_1 + I_2$ is archimedean and $\nabla^2_{x_i} \zeta_i(x_i^*, u_{-i}) \succ 0$ if x_i^* is a minimizer of (4.30), then $\zeta_i(x_i) - \eta_i \in I_1 + I_2$, by [56, Corollary 3.3]. Since

$$I_1 + I_2 \subseteq \text{Ideal}[H_i(u)] + \text{Qmod}[J_i(u)],$$

we have $\zeta_i(x_i) - \eta_i \in \text{Ideal}[H_i(u)]_{2k} + \text{Qmod}[J_i(u)]_{2k}$ for all k big enough. Therefore, Algorithm 4.16 must terminate within finitely many loops, by the duality theory.

(ii) If the real zero set of polynomials in $H_i(u)$ is finite, then the conclusion is implied by [89, Theorem 1.1] and [88, Theorem 2.2]. □

Remark. If the objective polynomial in (4.21) is SOS-convex and its constraining ones are SOS-concave (see [47] for the definition of SOS-convex polynomials), then Algorithm 4.16 must terminate in the first loop (see [62]). If the optimal value of (4.21) is not achieved at a KKT point, the classical Moment-SOS hierarchy of semidefinite relaxations can be used to solve it. We refer to [56, 61–64, 67, 68] for the work for solving general polynomial optimization.

4.4 Numerical experiments

In this section, we apply Algorithm 4.12 to solve convex GNEPs. To use it, we need Lagrange multiplier expressions. This can be done as follows.

- When polynomial expressions exist, we always use them. In particular, we use polynomial expressions for the first player of the GNEP given by (1.4), the third player in Example 4.4, the production unit and market players in Example 4.25.
- We use rational expressions for all players in Examples 4.19, 4.20 and 4.22. Moreover, rational expressions are used for the second player of the GNEP given by (1.4), the first two players in Example 4.4, the first and third players in Example 4.23 and the consumer players in Example 4.25. For Example 4.22, the rational expression is obtained by solving (4.12) numerically.
- When it is difficult to find convenient or rational expressions, we use parametric expressions for Lagrange multipliers. For all players in Examples 4.21, 4.24, for the first player in Example 4.21, for the second player in Example 4.23, we use parametric expressions.

We apply the software `GloptiPoly 3` [49] and `SeDuMi` [124] to solve the Moment-SOS relaxations for the polynomial optimization (4.25) and (4.30). We use the software `YALMIP` for solving (4.12). The computation is implemented in an Alienware Aurora R8 desktop, with an Intel® Core(TM) i7-9700 CPU at 3.00GHz×8 and 16GB of RAM, in a Windows 10 operating system. For neatness of the paper, only four decimal digits are shown for the computational results.

In Step 2 of Algorithm 4.12, if the optimal values $\delta_i \geq 0$ for each i such that $q_i(u) \leq 0$, then the computed minimizer of (4.20) is a GNE. In numerical computations, we may not have $\delta_i \geq 0$ exactly due to round-off errors. Typically, when δ_i is near zero, say, $\delta_i \geq -10^{-6}$, we regard the computed solution as an accurate GNE. In the following, all the GNEPs are convex.

Example 4.18. (i) For the GNEP given by (1.4), the first player has a polynomial expression for Lagrange multipliers given by

$$\lambda_{1,1} = x_1^T \nabla_{x_1} f_1,\ \lambda_{1,j+1} = \frac{\partial f_1(x)}{\partial x_{1,j}} - \lambda_{1,1} x_{2,j}\ (j = 1, 2, 3).$$

For the second player, the matrix polynomial $G_2(x)$ is not nonsingular, and polynomial expressions do not exist. In section 4.4, we give a rational expression for the second player's Lagrange multipliers. and the second player has a rational expression given as

$$\lambda_{2,1} = \frac{-x_2^T \nabla_{x_2} f_2}{2q_2(x)}, \quad q_2(x) = x_1^T x_1.$$

For each i, the $q_i(x) > 0$ for all $x \in X$. We ran Algorithm 4.12 and obtained the GNE $u = (u_1, u_2)$ with

$$u_1 \approx (0.7274, 0.7274, 0.7274), \quad u_2 \approx (0.4582, 0.4582, 0.4582).$$

It took around 3.06 seconds.

However, if the first player's objective is changed to

$$f_1(x) = (x_{2,1} + x_{2,2} - 2x_{2,3})(x_{1,1} + x_{1,2} - 2x_{1,3})^2 + x_{1,1} + x_{1,2} - 2x_{1,3},$$

then the GNEP has no GNE, detected by Algorithm 4.12. It took around 70 seconds to detect the nonexistence. The matrix polynomials $G_1(x)$ and $G_2(x)$ are nonsingular on X, so all GNEs must be KKT points if they exist.

(ii) For the GNEP in Example 4.4, we use the rational expression given by (4.9) for the first two players, and polynomial expression (4.10) for the third player. By Algorithm 4.12, we obtained a feasible point $\hat{u} = 10^{-4} \cdot (0.1274, 0.4102, 0.3219)$ of (4.20) with $q_1(\hat{u}) \approx 0.1274 \cdot 10^{-4}$ and $q_2(\hat{u}) \approx 0.4102 \cdot 10^{-4}$. We solved (4.21), for $i = 1, 2$, to check if $\hat{u}$ is a GNE or not, and got $\delta_1 \approx -1.0000$, $\delta_2 \approx -1.8996 \cdot 10^{-10}$. Therefore, we solved (4.22) with $\mathcal{N} = \{1\}$ and $\varepsilon = 0.1$, and obtained a GNE $u = (u_1, u_2, u_3)$ with

$$u_1 \approx 0.5000,\ u_2 \approx 0.5000,\ u_3 \approx 0.7500,\ q_1(u) \approx q_2(u) \approx 0.1250.$$

It took around 0.89 second.

Example 4.19. Consider the GNEP in Example 4.2 with objectives

$$f_1(x) = \sum_{j=1}^{2}(x_{1,j} - 1)^2 + x_2(x_{1,1} - x_{1,2}),\ f_2(x) = (x_2)^3 - x_{1,1}x_{1,2}x_2 - x_2.$$

The rational expressions for both players are given by (4.6). For each i, the $q_i(x) > 0$ for all $x \in X$. We ran Algorithm 4.12 and got the GNE $u = (u_1, u_2)$ with

$$u_1 \approx (0.4897, 1.0259),\ u_2 \approx 0.7077.$$

It took around 0.62 second.

Example 4.20. Consider the GNEP in Example 4.8 with objectives

$$f_1(x) = 10x_1^T x_2 - \sum_{j=1}^{3} x_{1,j},\ f_2(x) = \sum_{j=1}^{3}(x_{1,j}x_{2,j})^2 + (3\prod_{j=1}^{3} x_{1,j} - 1)\sum_{j=1}^{3} x_{2,j}.$$

We use rational expressions as in (4.13). From Example 4.8, we know all feasible points of (4.20) are GNEs. By Algorithm 4.12, we got the GNE $u = (u_1, u_2)$ with

$$u_1 \approx (0.9864, 0.0088, 0.0088), \quad u_2 \approx (0.0836, 0.0999, 0.0999).$$

It took around 2.03 seconds.

Example 4.21. Consider the GNEP in Example 4.10 with objectives

$$\begin{aligned} f_1(x) &= x_{2,1}(x_{1,1})^3 + (x_{1,2})^3 - \sum_{j=1}^{2} x_{1,j} \cdot \sum_{j=1}^{2} x_{2,j}, \\ f_2(x) &= (x_{1,1} + x_{1,2})(x_{2,1})^3 - 3x_{2,1} + (x_{2,2})^2 + x_{1,1}x_{1,2}x_{2,2}. \end{aligned}$$

We use parametric expressions as in (4.14). For each i, the $q_i(x) > 0$ for all $x \in X$. By Algorithm 4.12, we got the GNE $u = (u_1, u_2)$ with

$$u_1 \approx (0.6475, 0.2786), \quad u_2 \approx (1.0391, -0.0902).$$

It took around 63.97 seconds.

Example 4.22. Consider the 2-player GNEP

$$\begin{array}{cl|cl} \min\limits_{x_1 \in \mathbb{R}^2} & (x_{1,1})^4 + 2(x_{1,2})^2 & \min\limits_{x_2 \in \mathbb{R}^2} & \frac{1}{2}\|x_1\|^2 \cdot \|x_2\|^2 + x_{2,1} - x_{2,2} \\ & \quad + \sum_{j=1}^{2} x_{1,j}(x_{2,j})^2 & & \quad +2x_{1,1}x_{1,2}x_{2,1}x_{2,2} \\ s.t. & x_{1,1} + 2x_{1,2} - x_{2,1} \le 1, & s.t. & (x_{2,1})^2 + x_{1,2}x_{2,2} \le 2, \\ & 3x_{1,1} + 2x_{1,2} - x_{2,1} \le 1.5, & & (x_{1,1})^2 + (x_{2,2})^2 \le 3, \\ & (x_{1,2})^2 + (x_{2,1})^2 \le 3, & & x_{2,2} \ge 0. \\ & x_{1,1} \ge 0, & & \end{array}$$

The objective functions are given by We solve (4.12) numerically for $i = 1, 2$ with $v = (0, 0, 0, 0), d = 2$ to get rational expressions for λ_i's. By Algorithm 4.12, we got the GNE $u = (u_1, u_2)$ with

$$u_1 \approx (0.0000, -0.7500),\ u_2 \approx (-1.0881, 1.7321),\ q_1(u) \approx 0.2591,\ q_2(u) \approx 2.4028.$$

It took around 0.34 second in solving (4.12) for both players, and 8.40 seconds to find the GNE. For neatness of the paper, we do not display Lagrange multiplier expressions obtained by solving (4.12).

Example 4.23. Consider the 3-player GNEP

$$\begin{cases} \min\limits_{x_1 \in \mathbb{R}^2} & x_{2,2}(x_{1,1})^2 - x_{1,2}x_{3,1} \\ s.t. & x_1^T x_1 + x_2^T x_2 + (x_{2,1} + x_{2,2})x_3^T x_3 \le 1; \end{cases}$$

$$\begin{cases} \min\limits_{x_2 \in \mathbb{R}^2_+} & (x_{2,1})^2 + (x_{1,1} - 1)x_{2,1} + (x_{3,2}x_{2,2})^2 - x_{3,1}x_{2,2} \\ s.t. & x_{1,1}x_{2,1}x_{3,1} + x_{1,2}x_{2,2}x_{3,2} + 0.1 \ge 0, 1 - \sum_{j=1}^2 x_{2,j} \ge 0; \end{cases}$$

$$\begin{cases} \min\limits_{x_3 \in \mathbb{R}^2} & (x_3 - x_1 + x_2)^T x_3 \\ s.t. & (x_{3,1} - x_{3,2})^2 \le x_{2,1}, (x_{3,1} + x_{3,2})^2 \le 3. \end{cases}$$

The first player's Lagrange multipliers have a rational expression, that

$$\lambda_1 = \frac{-x_1^T \nabla_{x_1} f_1}{2q_1(x)}, \quad q_1(x) = 1 - x_2^T x_2 - (x_{2,1} + x_{2,2})x_3^T x_3.$$

For the second player, we use the parametric expression in (4.17), with $s_2 = 1$. For λ_3, if we let $q_3 = 2x_{2,1} - 2(x_{3,1})^2 + 2(x_{3,2})^2$, then

$$\lambda_{3,1} = \frac{1}{q_3}\Big(- x_3^T \nabla_{x_3} f_3 + (x_{3,1} + x_{3,2})\frac{\partial f_3}{\partial x_{3,1}}\Big),\ \lambda_{3,2} = \frac{1}{6}\Big(- 2x_{2,1}\lambda_{3,1} - \frac{x_3^T \nabla_{x_3} f_3}{q_3}\Big).$$

Note that $q_3(x) \not\ge 0$ on X. So we change the constraint $\hat{\lambda}_{3,j}(x) \ge 0$ in (4.20) to $q_{3,j} \cdot \hat{\lambda}_{3,j}(x) \ge 0$, to make it work. By Algorithm 4.12, we got the GNE $u = (u_1, u_2, u_3)$ with

$$u_1 \approx (0.0000, -0.7993),\ u_2 \approx (0.5000, 0.0000),\ u_3 \approx (-0.2500, -0.3997),$$
$$q_1(u) \approx 0.6389, \quad q_2(u) = 1, \quad q_3(u) \approx 1.1944.$$

It took around 10.44 seconds.

Example 4.24. [35, Example A.3] Consider the GNEP of 3 players. For $i = 1, 2, 3$, the ith player aims to minimize the quadratic function

$$f_i(x) = \frac{1}{2} x_i^T A_i x_i + x_i^T (B_i x_{-i} + b_i).$$

All variables have box constraints $-10 \le x_{i,j} \le 10$, for all i, j. In addition to them, the first player has linear constraints $x_{1,1}+x_{1,2}+x_{1,3} \le 20$, $x_{1,1}+x_{1,2}-x_{1,3} \le x_{2,1}-x_{3,2}+5$; the second player has $x_{2,1}-x_{2,2} \le x_{1,2}+x_{1,3}-x_{3,1}+7$; and the third player has $x_{3,2} \le x_{1,1}+x_{1,3}-x_{2,1}+4$. The values of parameters are set as follows

$$A_1 = \begin{bmatrix} 20 & 5 & 3 \\ 5 & 5 & -5 \\ 3 & -5 & 15 \end{bmatrix}, A_2 = \begin{bmatrix} 11 & -1 \\ -1 & 9 \end{bmatrix}, A_3 = \begin{bmatrix} 48 & 39 \\ 39 & 53 \end{bmatrix},$$

$$B_1 = \begin{bmatrix} -6 & 10 & 11 & 20 \\ 10 & -4 & -17 & 9 \\ 15 & 8 & -22 & 21 \end{bmatrix}, B_2 = \begin{bmatrix} 20 & 1 & -3 & 12 & 1 \\ 10 & -4 & 8 & 16 & 21 \end{bmatrix},$$

$$B_3 = \begin{bmatrix} 10 & -2 & 22 & 12 & 6 \\ 9 & 19 & 21 & -4 & 20 \end{bmatrix}, b_1 = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}, b_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, b_3 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}.$$

We use parametric expressions for Lagrange multipliers as in (4.16). It is clear $q_i(x)$ for all $x \in X$ and for all $i = 1, 2, 3$. By Algorithm 4.12, we got the GNE $u = (u_1, u_2, u_3)$ with

$$\begin{aligned} &u_1 \approx (-0.0270, -0.1116, -0.0522), \quad u_2 \approx (-0.0796, -0.2692), \\ &u_3 \approx (-0.0018, 0.1245). \end{aligned}$$

It took around 7.63 seconds.

Example 4.25. Consider the GNEP based on the Arrow and Debreu model of a competitive economy [7, 35]. The first N_1 players are consumers, the second N_2 players are production units, and the last player is the market, so $N = N_1 + N_2 + 1$. Each player has P variables. Let $Q_i \in \mathbb{R}^{P \times P}, b_i \in \mathbb{R}^P, \xi_i \in \mathbb{R}_+^P$ and $a_{i,k} \in \mathbb{R}_+$ be parameters. These players' optimization problems are:

$$\text{The } i\text{th player (a consumer):} \begin{cases} \min\limits_{x_i \in \mathbb{R}_+^P} & \frac{1}{2} x_i^T Q_i x_i - b_i^T x_i \\ s.t. & x_N^T x_i \le x_N^T \xi_i + \sum_{k=N_1+1}^{N-1} a_{i,k} x_N^T x_k. \end{cases}$$

$$\text{The } i\text{th player (a production unit):} \begin{cases} \min_{x_i \in \mathbb{R}^P_+} & -x_N^T x_i \\ s.t. & x_i^T x_i \le i - N_1. \end{cases}$$

$$\text{The } N\text{th player (the market):} \begin{cases} \min_{x_N \in \mathbb{R}^P_+} & x_N^T \left(\sum_{k=N_1+1}^{N-1} x_k - \sum_{k=1}^{N_1} (x_k - \xi_k) \right) \\ s.t. & \sum_{j=1}^{P} x_{N,j} = 1. \end{cases}$$

For each $i \in [N_1]$, the Lagrange multipliers have rational expressions as

$$\lambda_{i,1} = \frac{-x_i^T \nabla_{x_i} f_i}{q_i(x)}, \;\; \lambda_{i,j} = \frac{\partial f_i}{\partial x_{i,j}} + x_{N,j} \cdot \lambda_{i,1} \, (j = 1, \ldots, P),$$

where $q_i(x) = x_N^T \xi_i + \sum_{k=N_1+1}^{N-1} a_{i,k} x_N^T x_k > 0$ for all $x \in X$. For each $i = N_1+1, \ldots, N_1+N_2$, the ith player (a production unit) has polynomial expressions

$$\lambda_{i,1} = \frac{-x_i^T \nabla_{x_i} f_i}{2(i - N_1)}, \;\; \lambda_{i,j} = \frac{\partial f_i}{\partial x_{i,j}} + 2x_{i,j} \cdot \lambda_{i,1} \, (j = 1, \ldots, P).$$

For the last player (the market), we substitute $x_{N,P}$ by $1 - \sum_{j=1}^{P-1} x_{N,j}$, then the constraints become $1 - \sum_{j=1}^{P-1} x_{N,j} \ge 0$, $x_{N,1} \ge 0, \ldots, x_{N,P-1} \ge 0$, and hence

$$\lambda_{N,1} = -\sum_{j=1}^{P-1} \frac{\partial f_N}{\partial x_{N,j}} \cdot x_{N,j}, \;\; \lambda_{N,j+1} = \frac{\partial f_N}{\partial x_{N,j}} + \lambda_{N,1} \, (j = 1, \ldots, P-1).$$

For the case $N_1 = 2, N_2 = 2, P = 3$, we run the Algorithm 4.12 with the following parameter setting:

$$Q_1 = \begin{bmatrix} 6 & 1 & 0 \\ 1 & 7 & -5 \\ 0 & -5 & 7 \end{bmatrix}, Q_2 = \begin{bmatrix} 6 & -1 & 1 \\ -1 & 7 & -5 \\ 1 & -5 & 7 \end{bmatrix}, b_1 = \begin{bmatrix} 11 \\ 12 \\ 13 \end{bmatrix}, b_2 = \begin{bmatrix} 12 \\ 13 \\ 11 \end{bmatrix},$$
$$\xi_1 = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}, \xi_2 = \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix}, \;\; a_{1,3} = 0.3, \;\; a_{1,4} = a_{2,3} = a_{2,4} = 0.4.$$

By the algorithm, we got the GNE $u = (u_1, u_2, u_3, u_4, u_5)$ with

$$\begin{aligned} &u_1 \approx (0.0001, 0.9688, 0.2479), \quad u_2 \approx (0.0002, 1.3701, 0.3507), \\ &u_3 \approx (1.1996, 3.8026, 4.3554), \quad u_4 \approx (2.0488, 5.0572, 4.7642), \\ &u_5 \approx (0.0000, 0.7962, 0.2038). \end{aligned}$$

It took around 67.12 seconds.

4.4.1 Comparison with other methods

We compare our Algorithm 4.12 with some other classical methods for solving convex GNEPPs, such as the two-step method in [44] based on Quasi-variational formulation, the penalty method in [35], and the interior point method based on the KKT system in [29]. The tested GNEPPs are those in (1.4), Example 4.19-4.20 and Example 4.22. For the jointly convex GNEP in Example 4.20, we also compare with the relaxation method based on the Nikaido-Isoda function in [52].

For a computed tuple $u := (u_1, \ldots, u_N)$, we use the value

$$\xi := \max\Big\{ \max_{i\in[N], j\in\mathcal{I}_i} \{-g_{i,j}(u)\}, \max_{i\in[N], j\in\mathcal{E}_i} \{|g_{i,j}(u)|\}\Big\}$$

to measure the feasibility violation. Clearly, the point u is feasible if and only if $\xi \le 0$. If we solve (4.21) for all $i \in [N]$, the accuracy parameter of u is $\delta := \max_{i\in[N]} |\delta_i|$. For these methods, we use the following stopping criterion: For each time we get a new iterate u, if its feasibility violation $\xi < 10^{-6}$, then we compute the accuracy parameter δ. If $\delta < 10^{-6}$, then we stop the iteration. For all these methods, the parameters are chosen the same as in [29, 35, 44, 52], except the penalty method, for which the maximum number of inner iterations is 100. Moreover, we allow 1000 maximum iterations for the QVI method and NI-function method, 1000 maximum outer iterations for the penalty method, and $100,000$ maximum iterations for the interior point method. For initial points, we use $(1, 0, 0, 1, 0, 0)$ for (1.4), and the zero vectors for other GNEPs. If the maximum number of iterations is achieved but the stopping criterion is not met, we still solve the (4.21) to check if the latest iterate is a GNE or not.

The computational results are shown in Table 4.1. The "QVI" stands the QVI method,, "Penalty" for the penalty method, "IPM" for the interior point method, "NI" for the NI function method, and "ALG 4.12" is for Algorithm 4.12. The "u" is the latest iterate for each method, "time" is the consumed time (in seconds), and $\max\{\delta, \xi\}$ is the bigger one of the feasibility violation and accuracy parameter of u. "Not convergent" means the sequence cannot reach a limit point, or the limit point is far from being a GNE.

For the GNEP (1.4), the QVI method seems to converge, but the accuracy parameter after 1000 iterations is still around $1.5627 \cdot 10^{-4}$. The penalty method and the interior point failed to converge. For Example 4.19, the QVI method and the interior point method successfully got a GNE in 1.83 and 0.02 seconds respectively, and the penalty method got a candidate GNE with accuracy parameter around $3.86 \cdot 10^{-6}$ after 1000 outer iterations. For

Example 4.20, the penalty method got a candidate GNE with accuracy parameter around $2.80 \cdot 10^{-5}$ at the maximum number of iterations. However, the QVI method, the interior point method and the NI-function method did not converge. For Example 4.22, the QVI method, the penalty method and the interior point method failed to find a GNE. In contrast, Algorithm 4.12 can solve all these convex GNEPPs very quickly, with accuracy parameters less than $6 \cdot 10^{-7}$. Algorithm 4.12 is more reliable for solving convex GNEPs given by polynomials.

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

Table 4.1: Comparison with some methods

Algorithm	u	time	$\max\{\delta, \xi\}$
Problem (1.4)			
QVI	$(0.7273, 0.7273, 0.7273, 0.4582, 0.4582, 0.4582)$	219.05	$1.5627 \cdot 10^{-4}$
Penalty	$10^{-14} \cdot (9.5356, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000)$	not convergent	
IPM	$(1.3857, 1.4202, 1.4202, 0.8602, 0.0587, 0.0587)$	not convergent	
ALG 4.12	$(0.7274, 0.7274, 0.7274, 0.4582, 0.4582, 0.4582)$	2.61	$4.7929 \cdot 10^{-9}$
Example 4.19			
QVI	$(0.4897, 1.0259, 0.7077)$	1.83	$2.5563 \cdot 10^{-7}$
Penalty	$(0.4897, 1.0259, 0.7077)$	6.49	$3.8589 \cdot 10^{-6}$
IPM	$(0.4897, 1.0259, 0.7077)$	0.02	$7.9280 \cdot 10^{-8}$
ALG 4.12	$(0.4897, 1.0259, 0.7077)$	0.62	$9.4049 \cdot 10^{-9}$
Example 4.20			
QVI	$(-0.5199, -0.5199, -0.5199, 0.5771, 0.5771, 0.5771)$	not convergent	
Penalty	$(0.5762, 0.5762, 0.5762, 0.0364, 0.0364, 0.0364)$	3.54	$2.7980 \cdot 10^{-5}$
IPM	$(0.6387, 0.6366, 0.6336, 0.0681, 0.0674, 0.0665)$	not convergent	
NI	$(0.1021, 0.1021, 0.1021, 0.1021, 0.1021, 0.1021)$	not convergent	
ALG 4.12	$(0.9864, 0.0088, 0.0088, 0.0836, 0.0999, 0.0999)$	2.03	$4.3014 \cdot 10^{-7}$
Example 4.22			
QVI	$(0.0000, -0.6300, -1.1339, 1.5875)$	not convergent	
Penalty	$(0.3776, -0.5895, 0.0700, 1.5318)$	not convergent	
IPM	$(0.9407, -0.0471, 1.2194, -0.0360)$	not convergent	
ALG 4.12	$(0.0000, -0.7500, -1.0881, 1.7321)$	8.40	$5.4972 \cdot 10^{-7}$

Chapter 5

The Gauss-Seidel Method for Generalized Nash Equilibrium Problems of Polynomials

5.1 The Gauss-Seidel method for GNEPPs

In this chapter, we study the Gauss-Seidel method for solving Generalized Nash Equilibrium Problems of Polynomials. We consider the case that the GNEPP only has inequality constraints, i.e., find $x = (x_1, \ldots, x_N)$ such that each x_i is an optimizer of the ith player's optimization problem

$$\begin{cases} \min\limits_{x_i \in \mathbb{R}^{n_i}} & f_i(x_i, x_{-i}) \\ s.t. & g_{i,j}(x_i, x_{-i}) \geq 0 \, (j = 1, \ldots, s_i), \end{cases} \tag{5.1}$$

where all $f_i(x_i, x_{-i})$ and $g_{i,j}(x_i, x_{-i})$ are polynomial functions in $\mathbf{x}$. A solution $\mathbf{x}$ satisfying the above is called a generalized Nash equilibrium (GNE).

Let $g_i = (g_{i,1}, \ldots, g_{i,s_i}) : \mathbb{R}^n \to \mathbb{R}^{s_i}$ be the vector-valued function. The inequality $g_i(x_i, x_{-i}) \geq 0$ is defined componentwisely. Then (5.1) can be rewritten as

$$\begin{cases} \min\limits_{x_i \in \mathbb{R}^{n_i}} & f_i(x_i, x_{-i}) \\ s.t. & g_i(x_i, x_{-i}) \geq 0. \end{cases} \tag{5.2}$$

For given x_{-i}, the feasible strategy set for the ith player is

$$X_i(x_{-i}) := \{x_i \in \mathbb{R}^{n_i} : g_i(x_i, x_{-i}) \geq 0\}. \tag{5.3}$$

The Gauss-Seidel method was introduced in [37] for solving GNEPs. The following is the general framework of the Gauss-Seidel method.

Algorithm 5.1. For the GNEP of (5.1), do the following:

Step 1. Choose a feasible starting point $x^{(0)} = (x_1^{(0)}, \dots, x_N^{(0)})$, a positive regularization parameter $\tau^{(0)}$ and let $k := 0$.

Step 2. If $x^{(k)}$ satisfies a suitable termination criterion, stop.

Step 3. For $i = 1, \dots, N$, compute a global minimizer $x_i^{(k+1)}$ of the optimization

$$\left\{ \begin{array}{ll} \min\limits_{x_i \in \mathbb{R}^{n_i}} & f_i(x_1^{(k+1)}, \dots, x_{i-1}^{(k+1)}, x_i, x_{i+1}^{(k)}, \dots, x_N^{(k)}) + \tau^{(k)} \|x_i - x_i^{(k)}\|^2 \\ s.t. & g_i(x_1^{(k+1)}, \dots, x_{i-1}^{(k+1)}, x_i, x_{i+1}^{(k)}, \dots, x_N^{(k)}) \geq 0. \end{array} \right. \tag{5.4}$$

Step 4. Choose a new regularization parameter $\tau^{(k+1)} \in [0, \tau^{(k)}]$.

Step 5. Let $x^{(k+1)} := (x_1^{(k+1)}, \dots, x_N^{(k+1)})$, $k := k+1$, and go to Step 2.

In practice, Algorithm 5.1 performs well for solving GNEPPs. It can compute equilibria for many problems. This is demonstrated in numerical experiments in Section 5.3. The GNEPPs are very hard to be solved by other existing methods, to the best of the authors' knowledge. On the other hand, Algorithm 5.1 is not theoretically guaranteed to converge for all GNEPPs. Its convergence can be shown for some special GNEPs, such as GPGs. In the following, we show how to implement Algorithm 5.1 when the defining functions are polynomials. After that, we review some properties of Algorithm 5.1.

5.1.1 Moment-SOS relaxations for polynomial optimization

We discuss how to implement Algorithm 5.1 when all the objective and constraining functions are given by polynomials. In its Step 3, the sub-optimization (5.4) is a polynomial optimization problem whose variable is $x_i \in \mathbb{R}^{n_i}$. For convenience, denote

$$\left\{ \begin{array}{l} f_i^{(k)} := f_i(x_1^{(k+1)}, \dots, x_{i-1}^{(k+1)}, x_i, x_{i+1}^{(k)}, \dots, x_N^{(k)}) + \tau^{(k)} \|x_i - x_i^{(k)}\|^2, \\ g_i^{(k)} := g_i(x_1^{(k+1)}, \dots, x_{i-1}^{(k+1)}, x_i, x_{i+1}^{(k)}, \dots, x_N^{(k)}). \end{array} \right. \tag{5.5}$$

They are polynomials in x_i. One can rewrite (5.4) equivalently as

$$\left\{ \begin{array}{rll} \vartheta_{\min} = & \min\limits_{x_i \in \mathbb{R}^{n_i}} & f_i^{(k)}(x_i) \\ & s.t. & g_i^{(k)}(x_i) \geq 0. \end{array} \right. \tag{5.6}$$

Denote the degree

$$d_0 := \max\{\lceil \deg(f_i^{(k)})/2 \rceil, \lceil \deg(g_i^{(k)})/2 \rceil\}.$$

For $d = d_0, d_0+1, \ldots$, the dth moment relaxation for (5.6) is

$$\begin{cases} \vartheta_d := \min\limits_{y} & \left\langle f_i^{(k)}, y \right\rangle \\ \quad\quad\quad s.t. & M_d[y] \succeq 0,\ L_{g_i^{(k)}}^{(d)} \succeq 0, \\ & y_0 = 1, y \in \mathbb{R}^{\mathbb{N}_{2d}^{n_i}}. \end{cases} \tag{5.7}$$

Its dual optimization problem is the SOS relaxation

$$\begin{cases} \max & \gamma \\ s.t. & f_i^{(k)} - \gamma \in \mathrm{Qmod}(g_i^{(k)})_{2d}. \end{cases} \tag{5.8}$$

By solving the relaxations (5.7)-(5.8) for $d = d_0, d_0+1, \ldots$, we get the Moment-SOS hierarchy for solving (5.6). The following is the algorithm.

Algorithm 5.2. (The Moment-SOS hierarchy for solving (5.6)). Let $f_i^{(k)}, g_i^{(k)}$ be as in (5.5). Start with $d := d_0$.

Step 1. Solve the semidefinite relaxation (5.7). If (5.7) is infeasible, then (5.6) has no feasible points and stop; otherwise, solve it for a minimizer y^* and let $t := d_1$, where $d_1 := \lceil \deg(g_i^{(k)})/2 \rceil$.

Step 2. If y^* satisfies the rank condition

$$\operatorname{rank} M_t[y^*] = \operatorname{rank} M_{t-d_1}[y^*], \tag{5.9}$$

then extract $r := \operatorname{rank} M_t(y^*)$ minimizers for (5.6) and stop.

Step 3. If (2.7) fails to hold and $t < d$, let $t := t+1$ and then go to Step 2; otherwise, let $d := d+1$ and go to Step 1.

The rank condition (5.9) is called *flat truncation* in the literature [88]. It is a sufficient (and almost necessary) condition for checking convergence of the Moment-SOS hierarchy. When $\mathrm{Qmod}(g_i^{(k)})$ is archimedean, we have $\vartheta_d \to \vartheta_{\min}$ as $d \to \infty$, as shown in [61]. If $\vartheta_d = \vartheta_{\min}$ for some d, the relaxation (5.7) is said to be exact (or tight) for solving (5.4). For such a case, the Moment-SOS hierarchy is said to have finite convergence. The Moment-SOS hierarchy has finite convergence when the archimedean and some optimality conditions hold [90]. We refer to Section 2.3 for more details about the Moment-SOS relaxations, for cleaness of this thesis.

5.1.2 Some properties of Algorithm 5.1

Although Algorithm 5.1 converges for many problems, it is possible that it does not converge for some special ones. For instance, it is possible that (5.4) becomes infeasible after some loops even if the starting point $x^{(0)}$ is feasible. The following is such an example.

Example 5.3. Consider the 2-player GNEP

$$\begin{array}{cc|cc} \min\limits_{x_1 \in \mathbb{R}^1} & -x_1 - x_2 & \min\limits_{x_2 \in \mathbb{R}^1} & x_1 x_2 \\ s.t. & 0 \le x_1 \le 2 & s.t. & x_1 + (x_2)^2 \le 1. \end{array} \tag{5.10}$$

If Algorithm 5.1 begins with $(x_1^{(0)}, x_2^{(0)}) = (0,1)$ and uses the constant $\tau^{(k)} = 0.05$, then $x_1^{(1)} = 2$ and (5.4) is infeasible for $k = 1$ and $i = 2$.

When a GNEP has a shared constraint, i.e., there exists a set $X \subseteq \mathbb{R}^n$ such that $X_i(x_{-i}) = \{x_i : (x_i, x_{-i}) \in X\}$ for all players, then the suboptimization (5.4) is feasible for all k, provided that the initial point $x^{(0)}$ is feasible [37]. Beyond the concern of infeasibility, the sequence of $x^{(k)}$ produced by Algorithm 5.1 might be alternating and does not converge. Let's see the following example.

Example 5.4. Consider the 2-player GNEP

$$\begin{array}{cc|cc} \min\limits_{x_1 \in \mathbb{R}^1} & x_1 & \min\limits_{x_2 \in \mathbb{R}^1} & x_1 x_2 \\ s.t. & x_1 \ge x_2 & s.t. & (x_1)^2 + (x_2)^2 = 2. \end{array} \tag{5.11}$$

If Algorithm 5.1 starts with $(x_1^{(0)}, x_2^{(0)}) = (1,1)$ and uses the constant $\tau^{(k)} = 0.001$, the sub-optimization (5.4) for the first player is

$$\left\{ \begin{array}{rl} \min\limits_{x_1 \in \mathbb{R}^1} & x_1 + 0.001(x_1 - 1)^2 \\ s.t. & x_1 \ge 1. \end{array} \right.$$

Its minimizer $x_1^{(1)} = 1$. After plugging $(x_1^{(1)}, x_2^{(0)})$ into (5.4), the sub-optimization (5.4) for the second player is

$$\left\{ \begin{array}{rl} \min\limits_{x_2 \in \mathbb{R}^1} & x_2 + 0.001(x_2 - 1)^2 \\ s.t. & x_2^2 = 1, \end{array} \right.$$

whose minimizer $x_2^{(1)} = -1$. After one iteration, Algorithm 5.1 produced the point $x^{(1)} = (1, -1)$. For the loop of $k = 1$, the sub-optimization problem (5.4) for the first player is

$$\left\{ \begin{array}{rl} \min\limits_{x_1 \in \mathbb{R}^1} & x_1 + 0.001(x_1 - 1)^2 \\ s.t. & x_1 \ge -1, \end{array} \right.$$

whose minimizer $x_1^{(2)} = -1$, and the sub-optimization (5.4) for the second player is

$$\begin{cases} \min\limits_{x_2 \in \mathbb{R}^1} & -x_2 + 0.001(x_2+1)^2 \\ s.t. & x_2^2 = 1, \end{cases}$$

whose minimizer $x_2^{(2)} = 1$. So, $(x_1^{(2)}, x_2^{(2)}) = (-1, 1)$. Continuing this process, one can show that $x^{(k)}$ is alternating in the pattern

$$(1,1) \longrightarrow (1,-1) \longrightarrow (-1,-1) \longrightarrow (-1,1) \longrightarrow (1,1) \longrightarrow \cdots .$$

Algorithm 5.1 does not converge for this GNEP.

We would like to remark that even for the case that Algorithm 5.1 converges, the limit of $x^{(k)}$ is not necessarily a GNE for (5.1). This is shown in the following example.

Example 5.5. For instance, the following GNEP with two players

$$\begin{array}{cc|cc} \min\limits_{x_1 \in \mathbb{R}^1} & x_1 & \min\limits_{x_2 \in \mathbb{R}^1} & (x_2)^2 - (x_1 - 1)x_2 \\ s.t. & x_2(x_1 - x_2 - 1) \geq 0, & s.t. & (x_1)^2 + (x_2)^2 \leq 3, \\ & x_1 \geq 0, & & x_2 \geq 0 \end{array} \tag{5.12}$$

is a GNEPP. The dimensions $n_1 = n_2 = 1$ and

$$\begin{aligned} X_1(x_{-1}) &= \{x_1 \in \mathbb{R} : x_2(x_1 - x_2 - 1) \geq 0, x_1 \geq 0\}, \\ X_2(x_{-2}) &= \{x_2 \in \mathbb{R} : (x_1)^2 + (x_2)^2 \leq 3, x_2 \geq 0\}. \end{aligned}$$

For the first player, when $x_2 > 0$, its feasible set is $x_1 \geq x_2 + 1$, and its best strategy is $x_1 = x_2 + 1$. When $x_2 = 0$, the first player's best strategy is $x_1 = 0$. For any fixed x_1 with $x_1^2 \leq 3$, the second player's problem is feasible and its best strategy is $\max((x_1 - 1)/2, 0)$. One can verify that $(0, 0)$ is a GNE for this GNEPP.

For the first player, when $x_2 > 0$, its feasible set is $x_1 \geq x_2 + 1$, so the sub-optimization (5.4) in the kth loop is

$$\begin{cases} \min\limits_{x_1 \in \mathbb{R}^1} & x_1 + \tau^{(k)}(x_1 - x_1^{(k)})^2 \\ s.t. & x_1 \geq 1 + x_2^{(k)}. \end{cases} \tag{5.13}$$

For $0 < \tau^{(k)} < 0.5$, the minimizer of (5.13) is $1 + x_2^{(k)}$. For the second player, the sub-optimization (5.4) in the kth loop is

$$\begin{cases} \min\limits_{x_2 \in \mathbb{R}^1} & (x_2)^2 - x_2 x_2^{(k)} + \tau^{(k)}(x_2 - x_2^{(k)})^2 \\ s.t. & (x_2)^2 \leq 3 - (x_2^{(k)} + 1)^2, \; x_2 \geq 0. \end{cases} \tag{5.14}$$

When (5.14) is feasible, its minimizer is

$$\min\left\{\frac{1+2\tau^{(k)}}{2+2\tau^{(k)}}x_2^{(k)}, \sqrt{3-(x_2^{(k)}+1)^2}\right\}.$$

Therefore, for any constant $0<\tau^{(k)}<0.5$ or a decreasing $\tau^{(k)}$ with $\tau^{(0)}<0.5$, if $0<x_2^{(0)}\leq \sqrt{3}-1$ (to make (5.14) feasible), then $x^{(k)}\to(1,0)$ as $k\to\infty$. However, $(1,0)$ is not a GNE, because when $x_2=0$, $x_1=0$ is feasible and it is the minimizer. Indeed, $(0,0)$ is a GNE. This shows that a limit point produced by Algorithm 5.1 is not necessarily a GNE.

In practice, however, the performance of Algorithm 5.1 is good. Under certain conditions, Algorithm 5.1 converges and the limit is a GNE. This requires some assumptions on the feasible sets of (5.2). Let G be a set-valued map defined on a set U, i.e., $G(x)$ is a subset of a range Y, for all $x\in U$. Its domain, $\operatorname{dom} G$, is the set of $x\in U$ such that $G(x)\neq\emptyset$ [8]. The map G is said to be *inner semicontinuous* at $x\in U$ relative to $\operatorname{dom} G$ if for all $y\in G(x)$ and for all sequences $\{x_\ell\}\subseteq \operatorname{dom} G$ such that that $x_\ell\to x$, there exists a sequence of $y_\ell\in G(x_\ell)$ converging to y. The map G is called inner semicontinuous relative to $\operatorname{dom} G$ if it is inner semicontinuous relative to $\operatorname{dom} G$ at every point in $\operatorname{dom} G$. For instance, if the set $X_i(x_{-i})=\{x_i:(x_i,x_{-i})\in C_i\}$ for $C_i\subseteq\mathbb{R}^n$ being a polyhedron or a ball, then the set-valued map $x_{-i}\mapsto X_i(x_{-i})$ is inner semicontinuous relative to its domain at all points x_{-i} [111]. However, for the GNEP in (5.12), the set-valued map $x_2\to X_1(x_2)$ is not inner semicontinuous at $(0,0)$ (see the end of this section). We refer to [12, 111] for the inner semicontinuity of set-valued maps. The following is a useful lemma about inner semicontinuity.

Lemma 5.6. *For two closed sets U and V, let $f:U\times V\to\mathbb{R}$ and $h:U\times V\to\mathbb{R}^m$ be two continuous functions. For $y\in V$, define the set-valued map*

$$G(y)=\{x\in U:h(x,y)\geq 0\}.$$

Consider two sequences $\{x^{(k)}\}\subseteq \operatorname{dom} G$ and $\{y^{(k)}\}\subseteq V$ such that $x^{(k)}\to x^*$ and $y^{(k)}\to y^*$. Suppose $0\leq\tau^{(k)}\to 0$ as $k\to\infty$. Assume that each $x^{(k)}$ is a minimizer of the optimization problem

$$\begin{cases} \min\limits_{x\in U} & f(x,y^{(k)})+\tau^{(k)}\|x-x^{(k-1)}\|^2,\\ s.t. & h(x,y^{(k)})\geq 0. \end{cases} \tag{5.15}$$

If the set-valued map $G(y)$ is inner semicontinuous relative to $\operatorname{dom} G$*, then x^* is also a minimizer of*

$$\begin{cases} \min\limits_{x \in U} & f(x, y^*) \\ s.t. & h(x, y^*) \geq 0. \end{cases} \tag{5.16}$$

Proof. We prove it by a contradiction argument. Suppose otherwise that x^* is not a minimizer of (5.16), then there exists $z^* \in G(y^*)$ such that

$$f(z^*, y^*) < f(x^*, y^*). \tag{5.17}$$

Since the mapping G is inner semicontinuous, there exists a sequence of $z^{(k)}$ such that $z^{(k)} \to z^*$ and $z^{(k)} \in G(y^{(k)})$. The sequence $\{z^{(k)}\}$ is clearly bounded. Because x_k is a minimizer of

$$\begin{array}{ll} \min & f(x, y^{(k)}) + \tau^{(k)} \|x - x^{(k-1)}\|^2 \\ s.t. & x \in G(y^{(k)}), \end{array}$$

we have that

$$f(z^{(k)}, y^{(k)}) + \tau^{(k)} \|z^{(k)} - x^{(k-1)}\|^2 \geq f(x^{(k)}, y^{(k)}) + \tau^{(k)} \|x^{(k)} - x^{(k-1)}\|^2. \tag{5.18}$$

Because $f(x, y)$ is continuous, it holds that

$$f(x^{(k)}, y^{(k)}) \to f(x^*, y^*), \quad f(z_k, y_k) \to f(z^*, y^*)$$

as $k \to \infty$. For all $\varepsilon > 0$, there exists K_1 such that

$$\begin{array}{rcl} f(x^{(k)}, y^{(k)}) - f(x^*, y^*) & > & -\frac{\varepsilon}{4}, \\ f(z^{(k)}, y^{(k)}) - f(z^*, y^*) & < & \frac{\varepsilon}{4} \end{array}$$

for all $k > K_1$. Combining the two inequalities, we can get

$$f(x^{(k)}, y^{(k)}) - f(z^{(k)}, y^{(k)}) + f(z^*, y^*) - f(x^*, y^*) > -\frac{\varepsilon}{2}.$$

Therefore, we have

$$\begin{aligned} f(z^*, y^*) - f(x^*, y^*) + \frac{\varepsilon}{2} &> f(z^{(k)}, y^{(k)}) - f(x^{(k)}, y^{(k)}) \\ &\geq \tau^{(k)} \Big(\|x^{(k)} - x^{(k-1)}\|^2 - \|z^{(k)} - x^{(k-1)}\|^2 \Big). \end{aligned} \tag{5.19}$$

The last inequality follows from (5.18). Because $\{x^{(k)}\}$, $\{z^{(k)}\}$ are convergent sequences and $\tau^{(k)} \to 0$, there must exist K_2 such that

$$\tau^{(k)} (\|x^{(k)} - x^{(k-1)}\|^2 - \|z^{(k)} - x^{(k-1)}\|^2) > -\frac{\varepsilon}{2}$$

whenever $k > K_2$. Let $K := \max\{K_1, K_2\}$, then for all $k > K$

$$f(z^*, y^*) - f(x^*, y^*) + \varepsilon > 0.$$

Since ε can be arbitrarily small, the above implies that

$$f(z^*, y^*) - f(x^*, y^*) \geq 0,$$

which contradicts (5.17). Therefore, x^* is a minimizer of (5.16). □

Lemma 5.6 immediately implies the following result.

Theorem 5.7. *Let $x^{(k)}$ be the sequence produced by Algorithm 5.1 for the GNEP of (5.1). Assume that $x^{(k)} \to x^*$ and $\tau^{(k)} \to 0$. If for each i the set-valued map $G_i : x_{-i} \mapsto X_i(x_{-i})$ is inner semicontinuous relative to its domain $\operatorname{dom} G_i$, then the limit x^* is a GNE for the GNEP of (5.1).*

Remark: Theorem 5.7 assumes that the sequence of $x^{(k)}$ produced by Algorithm 5.1 converges. However, the theorem does not give a sufficient condition for this sequence to converge. To ensure convergence, we need to assume the GNEPs are GPGs; see Theorems 5.10 and 5.11. There exists a convergence result [116, Lemma 1] that is similar to Lemma 5.6 and Theorem 5.7.

In the proof of Lemma 5.6, it is required that $\tau^{(k)} \to 0$, which is also assumed in Theorem 5.7. However, in the implementation of Algorithm 5.1, we do not need $\tau^{(k)} \to 0$. Sometimes, a constant $\tau^{(k)}$ works very well. We refer to Theorem 5.11 and examples in Section 5.3.

For Examples 5.3 and 5.4, Algorithm 5.1 does not produce a convergent sequence. For Example 5.5, the set-valued map $G_1 : x_2 \mapsto X_1(x_2)$ for the first player is not inner semicontinuous relative to its domain $\operatorname{dom} G_1$. In fact, at the point $(x_1, x_2) = (0, 0)$, it is clear that $x_1 \in G_1(x_2)$. However, for every sequence $\{x_2^{(k)}\}$ such that $0 < x_2^{(k)} \to x_2 = 0$, $G_1(x_2^{(k)}) = [x_2^{(k)}+1, \infty)$. Since each $x_2^{(k)} > 0$, there does not exist a sequence $\{x_1^{(k)}\}$ converging to $x_1 = 0$ and $x_1^{(k)} \in G_1(x_2^{(k)}) = [x_2^{(k)}+1, \infty)$. Therefore, the inner semicontinuity assumption in Theorem 5.7 fails for Example 5.5.

5.2 Generalized potential games

The Gauss-Seidel method is frequently used for solving GNEPs. However, its convergence is not guaranteed for all of them. One wonders for what kind of GNEPs the

Gauss-Seidel method converges. The generalized potential game (GPG) is such a GNEP. The following is the definition of GPGs in [37].

Definition 5.8. *([37]) The GNEP of (5.1) is a generalized potential game if:*

(i) There exists a closed set $\emptyset \neq X \subseteq \mathbb{R}^n$ *such that*

$$X_i(x_{-i}) \equiv \{x_i \in D_i : (x_i, x_{-i}) \in X\}$$

for all players, where each $D_i \subseteq \mathbb{R}^{n_i}$ *is a closed set such that* $(D_1 \times \cdots \times D_N) \cap X \neq \emptyset$.

(ii) There exist a continuous function $P(x) : \mathbb{R}^n \to \mathbb{R}$ *and a forcing function* $\sigma : \mathbb{R}_+ \to \mathbb{R}_+$ *(i.e.,* $\lim_{k\to\infty} \sigma(t_k) = 0$ *implies* $\lim_{k\to\infty} t_k = 0$*) such that for all* $y_i, x_i \in X_i(x_{-i})$

$$\begin{aligned} f_i(y_i, x_{-i}) - f_i(x_i, x_{-i}) > 0 \quad \Longrightarrow \\ P(y_i, x_{-i}) - P(x_i, x_{-i}) \geq \sigma(f_i(y_i, x_{-i}) - f_i(x_i, x_{-i})). \end{aligned} \tag{5.20}$$

The item (i) in Definition 5.8 is from the concept of *shared constraint* [34]. It implies that if $x^{(0)}$ is feasible, then the sub-optimization problem (5.4) is feasible for all k and i. The item (ii) means that there exists a single "dominant function" P that measures the changes on each player's objective functions [37].

Some special GNEPs can be directly verified as GPGs. For instance, for the GNEP of (5.1), if each objective f_i can be expressed as

$$f_i(x) = f_0(x) + \sum_{j=1}^{M} f_{i,j}(x_j) \tag{5.21}$$

for some functions f_0 and $f_{i,j}$ and the item (i) holds, then the GNEP of (5.1) is a GPG because P, σ can be chosen as

$$P(x) = f_0(x) + \sum_{i=1}^{N} \sum_{j=1}^{M} f_{i,j}(x_j), \quad \sigma(t) = t.$$

One can easily check that the above $P(x)$ and $\sigma(t)$ satisfy (5.20) [78].

GPGs are extensions of potential games, which were originally introduced for NEPs [78]. They have broad applications [84]. The following is an example of GPG arising from applications.

Example 5.9. The GNEPP from the environmental pollution control, described in the introduction, is a GPG. The functions P and σ can be chosen as

$$\begin{aligned} P(x) = &\ 2\prod_{i=1}^{N}(x_{i,0} - \sum_{j=1}^{N}\gamma_{j,i}x_{j,i}) + \sum_{i=1}^{N}\left[\sum_{j=0}^{N}x_{i,j} - \sum_{j=1}^{N}\gamma_{j,i}x_{j,i} - x_{i,0}(b_i - 1/2x_{i,0})\right], \\ \sigma(t) = &\ t. \end{aligned}$$

The numerical results of Algorithm 5.1 are shown in the next section.

The following is the convergence result for Algorithm 5.1 when it is applied to solve GPGs.

Theorem 5.10. *([37, Theorem 5.2]) Consider the GNEP of (5.1) such that all the functions are continuous. Assume that (5.1) is a GPG and each set-valued map $G_i : x_{-i} \mapsto X_i(x_{-i})$ is inner semicontinuous relative to its domain. In Algorithm 5.1, suppose each $x_i^{(k+1)}$ is a minimizer of (5.4) and the parameters $\tau^{(k)}$ are updated as*

$$\tau^{(k+1)} := \max\left\{\min\left[\tau^{(k)}, \max_{i=1,\dots,N}(\|x_i^{(k+1)} - x_i^{(k)}\|)\right], 0.1\tau^{(k)}\right\}. \tag{5.22}$$

Then every limit point of the sequence $\{x^{(k)}\}_{k=0}^{\infty}$ produced by Algorithm 5.1 is a GNE for (5.1).

The updating scheme (5.22) for $\tau^{(k)}$ is a bit complicated. However, if each player's optimization problem (5.2) is convex, then the parameter $\tau^{(k)}$ can be chosen to be constant.

Theorem 5.11. *([37, Theorem 4.3]) Consider the GNEP of (5.1) such that all the functions are continuous. Assume that (5.1) is a GPG and each set-valued map $G_i : x_{-i} \mapsto X_i(x_{-i})$ is inner semicontinuous relative to its domain. Suppose the objectives $f_i(\,\cdot\,, x_{-i})$ and the feasible sets $X_i(x_{-i})$ are all convex. In Algorithm 5.1, suppose each $x_i^{(k+1)}$ is a minimizer of (5.4) and the parameter $\tau^{(k)} = \tau > 0$ is a constant. Then every limit point of the sequence $\{x^{(k)}\}_{k=0}^{\infty}$ produced by Algorithm 5.1 is a GNE for (5.1).*

Beyond GPGs, the Gauss-Seidel method has convergence for GNEPs with discrete strategy sets [114] or mixed-integer variables [115]. In general, when (5.1) is not a GPG, the convergence of Algorithm 5.1 is not known very much. We have seen examples in Section 5.1 such that Algorithm 5.1 fails to converge. On the other hand, the performance of Algorithm 5.1 is actually very good in our computational experiments (see Section 5.3). In the following, we discuss how to certify that a GNEP is a GPG.

5.2.1 A certificate for GPGs

Generally, it is hard to check whether a GNEP is a GPG or not. The main challenge is to verify the item (ii) in Definition 5.8. In this subsection, we give a certificate for (5.20) to hold. For the ith player, denote the set

$$K_i = \left\{ (x_i, y_i, x_{-i}) \in \mathbb{R}^{n_i} \times \mathbb{R}^{n_i} \times \mathbb{R}^{n-n_i} \;\middle|\; \begin{array}{c} x_i, y_i \in X_i(x_{-i}) \\ f_i(y_i, x_{-i}) - f_i(x_i, x_{-i}) \geq 0 \end{array} \right\}. \tag{5.23}$$

For convenience, denote the differences of functions

$$\left\{ \begin{array}{rcl} \triangle P_i & := & P(y_i, x_{-i}) - P(x_i, x_{-i}), \\ \triangle f_i & := & f_i(y_i, x_{-i}) - f_i(x_i, x_{-i}). \end{array} \right. \tag{5.24}$$

The following lemma is straightforward for verification.

Lemma 5.12. *For the GNEP of (5.1), if the item (i) in Definition 5.8 holds, and there exist polynomials* $P \in \mathbb{R}[x]$, $p_{i,0}, p_{i,1} \in \mathbb{R}[x_i, y_i, x_{-i}]$ $(i = 1, \ldots, N)$ *such that* $p_{i,0} \geq 0$, $p_{i,1} \geq 0$ *on* K_i *and*

$$\triangle P_i \;=\; (p_{i,0} + 1)\triangle f_i + p_{i,1} \tag{5.25}$$

for all i*, then (5.1) is a GPG.*

In the equation (5.25), we can replace the constant 1 by any positive number $\epsilon > 0$, up to scaling coefficients. For numerical reasons, we prefer the constant 1. Lemma 5.12 gives a certificate for GPGs. The following are examples of GPGs certified by (5.25).

Example 5.13. Consider the 2-player GNEP with the sets

$$X = \{(x_1, x_2) : 1 \leq x_1, x_2 \leq 10,\; x_1 \geq x_2\},$$

$$X_1(x_{-1}) = \{x_1 : (x_1, x_2) \in X\}, \quad X_2(x_{-2}) = \{x_2 : (x_1, x_2) \in X\}.$$

The two players' optimization problems are respectively

$$\begin{array}{cl|cl} \min\limits_{x_1} & x_1 + x_2 & \min\limits_{x_2} & -x_1 x_2 \\ s.t. & (x_1, x_2) \in X, & s.t. & (x_1, x_2) \in X. \end{array} \tag{5.26}$$

Let $P(x_1, x_2) = (x_1)^3 - x_1 x_2 + x_1$, we have

$$\left\{ \begin{array}{ll} \triangle P_1 & = (y_1 - x_1)[(y_1 - x_1)^2 + 1] + (3y_1 x_1 - x_2)(y_1 - x_1), \\ \triangle P_2 & = -x_1(y_2 - x_2), \\ \triangle f_1 & = y_1 - x_1, \\ \triangle f_2 & = -x_1(y_2 - x_2). \end{array} \right. \tag{5.27}$$

The equation (5.25) is satisfied for

$$p_{1,0} = (y_1 - x_1)^2,\ p_{1,1} = (3y_1x_1 - x_2)(y_1 - x_1),\ p_{2,0} = p_{2,1} = 0.$$

It is clear that $p_{1,0}, p_{2,0}, p_{2,1}$ are nonnegative. By the definition of K_1, $\triangle f_1 \geq 0$, and $3y_1x_1 - x_2 \geq 3y_1 - x_2 \geq 0$, so $p_{1,1} \geq 0$ on K_1.

Example 5.14. Consider the 2-player GNEP with the sets

$$X = \{(x_1, x_2) : (x_1)^3 + (x_2)^3 \leq 2, x_1 \geq 6x_2\},$$

$$X_1(x_{-1}) = \{x_1 : (x_1, x_2) \in X\}, \quad X_2(x_{-2}) = \{x_2 : (x_1, x_2) \in X\}.$$

The two players' optimization problems are respectively

$$\begin{array}{cl|cl} \min\limits_{x_1} & (x_1)^2x_2 + (x_2)^2x_1 - 4(x_1)^4 & \min\limits_{x_2} & x_1x_2 - 3(x_2)^2 \\ s.t. & x_1 \geq 0,\ (x_1, x_2) \in X, & s.t. & x_2 \geq 0.125,\ (x_1, x_2) \in X. \end{array} \tag{5.28}$$

For $P(x_1, x_2) = (x_1)^2x_2 + (x_2)^2x_1 - 4(x_1)^4$,

$$\left\{ \begin{array}{rcl} \triangle P_1 & = & \triangle f_1 = (y_1)^2x_2 + (y_2)^2x_1 - 4(y_1)^4 \\ & & \qquad -(x_1)^2x_2 - (x_2)^2x_1 + 4(x_1)^4, \\ \triangle P_2 & = & (x_1)^2(y_2 - x_2) + x_1((y_2)^2 - (x_2)^2), \\ \triangle f_2 & = & x_1(y_2 - x_2) - 3(y_2)^2 + 3(x_2)^2. \end{array} \right. \tag{5.29}$$

The equation (5.25) holds with

$$p_{1,0} = 0,\ p_{1,1} = 0, p_{2,0} = x_1, p_{2,1} = (y_2 - x_2)[4x_1(y_2 + x_2) + 3(y_2 + x_2) - x_1].$$

Clearly, $p_{1,0}$, $p_{1,1}$, $p_{2,0} \geq 0$. Note that

$$\triangle f_2 = (y_2 - x_2)(x_1 - 3(y_2 + x_2)) \geq 0$$

on K_2. Then, either $x_1 - 3(y_2 + x_2) > 0$ hence $y_2 - x_2 \geq 0$, or $x_1 - 3(y_2 + x_2) = 0$, which forces $y_2 - x_2 = 0$. This is because $x_1 \geq 6y_2$, $x_1 \geq 6x_2$ and $y_2, x_2 > 0$, if $x_1 - 3(y_2 + x_2) = 0$, then the only possible case is $x_1 = 6y_2 = 6x_2$. Thus from

$$4x_1(y_2 + x_2) + 3(y_2 + x_2) - x_1 > x_1(4y_2 + 4x_2 - 1) \geq 0,$$

we know $p_{2,1} \geq 0$ on K_2.

Example 5.15. Consider the 2-player GNEP with the sets

$$X = \left\{ (x_1, x_2) \;\middle|\; \begin{array}{l} x_1 = (x_{1,1}, x_{1,2}) \in \mathbb{R}^2, x_2 \in \mathbb{R}, \\ x_{1,1}, x_{1,2}, x_2 \geq 0.5, \\ x_2 - 0.3 \leq x_{1,1} + x_{1,2} \leq x_2 + 0.3 \end{array} \right\},$$

and $X_1(x_{-1}) = \{x_1 : (x_1, x_2) \in X\}$, $X_2(x_{-2}) = \{x_2 : (x_1, x_2) \in X\}$. The optimization problems are respectively

$$\begin{array}{cl|cl} \min\limits_{x_1} & x_{1,1}x_2 + x_{1,2}x_2 & \min\limits_{x_2} & x_{1,1} \cdot x_{1,2} \cdot x_2 \\ s.t. & \|x_1\| = 2,\ (x_1, x_2) \in X, & s.t. & (x_1, x_2) \in X. \end{array} \tag{5.30}$$

For $P(x_1, x_2) = (x_{1,1} + x_{1,2} + 1)^3 x_2$,

$$\begin{cases} \triangle P_1 & = x_2((y_{1,1} + y_{1,2} + 1)^3 - (x_{1,1} + x_{1,2} + 1)^3), \\ \triangle P_2 & = (y_2 - x_2)(x_{1,1} + x_{1,2} + 1)^3, \\ \triangle f_1 & = x_2(y_{1,1} + y_{1,2} - x_{1,1} - x_{1,2}), \\ \triangle f_2 & = x_{1,1}x_{1,2}(y_2 - z_2). \end{cases} \tag{5.31}$$

The equation (5.25) holds with

$$\begin{aligned} p_{1,0} &= (y_{1,1} + y_{1,2} + 1)^2 + (x_{1,1} + x_{1,2} + 1)^2 + (y_{1,1} + y_{1,2})(x_{1,1} + x_{1,2}) \\ &\quad + y_{1,1} + y_{1,2} + x_{1,1} + x_{1,2}, \\ p_{1,1} &= 0, \quad p_{2,0} = 3x_{1,1} + 3x_{1,2} + 5, \\ p_{2,1} &= (y_2 - x_2)\big[(x_{1,1})^3 + (x_{1,2})^3 + 3(x_{1,1})^2 + 3(x_{1,2})^2 + 3x_{1,1} + 3x_{1,2} + 1\big]. \end{aligned}$$

The equality $\triangle P_1 = (1 + p_{1,0})\triangle f_1$ follows from the identity

$$a^3 - b^3 = (a - b)(a^2 + ab + b^2)$$

with $a = y_{1,1} + y_{1,2} + 1$ and $b = x_{1,1} + x_{1,2} + 1$. Clearly, $p_{1,0}, p_{1,1} \geq 0$ on K_1, and $p_{2,0} \geq 0$ on K_2. Since $x_{1,1}x_{1,2} > 0$, $\triangle f_2 \geq 0$ implies $y_2 - x_2 \geq 0$, so $p_{2,1} \geq 0$ on K_2.

5.2.2 Putinar Positivstellensatz for the certificate

Lemma 5.12 gives a convenient certificate for checking GPGs. One needs to find polynomials $p_{i,0}, p_{i,1}$ and P satisfying (5.25) and $p_{i,0}, p_{i,1} \geq 0$ on K_i. For a polynomial tuple h, we have seen that if $p \in \text{Qmod}(h)$, then $p \geq 0$ on the semialgebraic set $\mathcal{S}(h)$. This motivates us to use Putinar's Positivstellensatz for verifying that.

For the set K_i as in (5.23), let $h_i := (h_{i,t})_{t=1}^{m_i}$ be a tuple of polynomials in $\mathbb{R}[x_i, y_i, x_{-i}]$ such that

$$K_i = \{(x_i, y_i, x_{-i}) : h_i(x_i, y_i, x_{-i}) \geq 0\}.$$

Moreover, let $h_{i,0} = 1$ for all i. When the item (i) in Definition 5.8 holds, the GNEP of (5.1) is a GPG if there exist $P \in \mathbb{R}[x]$ and $q_{i,0}, q_{i,1} \in \text{Qmod}(h_i)$ such that

$$\triangle P_i = (q_{i,0} + 1)\triangle f_i + q_{i,1} \tag{5.32}$$

for all players. For an even degree $2d$, we parameterize $P, q_{i,0}, q_{i,1}$ as

$$P(x) = \mathbf{p}^T[x]_{2d}, \quad q_{i,0} = \sum\nolimits_{t=0}^{m_i}\big([x, y_i]_{d-d_{it}}\big)^T \cdot Q_{i,0}^t \cdot \big([x, y_i]_{d-d_{it}}\big) \cdot h_{i,t},$$

$$q_{i,1} = \sum\nolimits_{t=0}^{m_i}\big([x, y_i]_{d-d_{it}}\big)^T \cdot Q_{i,1}^t \cdot \big([x, y_i]_{d-d_{it}}\big) \cdot h_{i,t}.$$

In the above, the degree $d_{it} = \lceil \deg(h_{i,t})/2 \rceil$. One can show that $q_{i,0}, q_{i,1} \in \text{Qmod}(h_i)$ if and only if there exist psd matrices $Q_{i,0}^t, Q_{i,1}^t$ in the above parametrization, for some d [63, Chapter 2]. For notational convenience, denote

$$Q \;:=\; \big(Q_{i,0}^t, Q_{i,1}^t\big)_{i=1,\ldots,N, t=1,\ldots,m_i}. \tag{5.33}$$

Therefore, the certificate (5.25) in Lemma 5.12 can be checked by solving the semidefinite program

$$\begin{cases} \min\limits_{\mathbf{p},Q} & \sum\limits_{i,t} \text{trace}\Big(Q_{i,0}^t + Q_{i,1}^t\Big) \\ s.t. & \triangle P_i \equiv (q_{i,0} + 1)\triangle f_i + q_{i,1}\,(\forall\, i), \\ & P \in \mathbb{R}[x]_{2d}, \\ & Q_{i,0}^t \succeq 0,\; Q_{i,1}^t \succeq 0\,(\forall\, i, t). \end{cases} \tag{5.34}$$

The certificate given by solving (5.34) does not require to have priori polynomials $P, q_{i,0}$ and $q_{i,1}$. Instead, the coefficients of these polynomials are variables in (5.34) that are awaiting to be solved numerically.

Example 5.16. Consider the 2-player GNEP such that the two players' optimization problems are

$$\begin{array}{cl|cl} \min\limits_{x_1 \in \mathbb{R}^1} & 2x_2 - x_1 & \min\limits_{x_2 \in \mathbb{R}^1} & (x_1)^2 - 2x_1x_2 - (x_2)^2 \\ s.t. & (x_1)^2 + (x_2)^2 \leq 1, & s.t. & (x_1)^2 + (x_2)^2 \leq 1, \\ & x_1 \geq 0, & & x_2 \geq 0. \end{array} \tag{5.35}$$

The $\Delta f_1 = x_1 - y_1$, $\Delta f_2 = 2x_1(x_2 - y_2) + (x_2)^2 - (y_2)^2$, and the defining polynomial tuples h_1, h_2 for the sets K_1, K_2 are respectively,

$$h_1 = \{1 - (x_1)^2 - (x_2)^2,\, 1 - (y_1)^2 - (x_2)^2,\, x_1,\, y_1,\, \Delta f_1\},$$
$$h_2 = \{1 - (x_1)^2 - (x_2)^2,\, 1 - (x_1)^2 - (y_2)^2,\, x_2,\, y_2,\, \Delta f_2\}.$$

For the degree $d = 2$, the semidefinite program (5.34) becomes

$$\begin{cases} \min\limits_{\mathbf{p},Q} & \sum\limits_{i=1}^{2}\sum\limits_{t=0}^{6} \mathrm{trace}\Big(Q_{i,0}^t + Q_{i,1}^t\Big) \\ s.t. & \mathbf{p}^T([y_1,x_2]_4 - [x]_4) = \left(\sum\limits_{t=0}^{6}[x,y_1]_{2-d_{it}}^T Q_{1,0}^t [x,y_1]_{2-d_{it}} h_{1,t} + 1\right)\cdot \Delta f_1 \\ & \qquad\qquad + \sum\limits_{t=0}^{6}[x,y_1]_{2-d_{it}}^T Q_{1,1}^t [x,y_1]_{2-d_{it}} h_{1,t}, \\ & \mathbf{p}^T([x_1,y_2]_4 - [x]_4) = \left(\sum\limits_{t=0}^{6}[x,y_2]_{2-d_{it}}^T Q_{2,0}^t [x,y_2]_{2-d_{it}} h_{2,t} + 1\right)\cdot \Delta f_2 \\ & \qquad\qquad + \sum\limits_{t=0}^{6}[x,y_2]_{2-d_{it}}^T Q_{2,1}^t [x,y_2]_{2-d_{it}} h_{2,t}, \\ & \mathbf{p} \in \mathbb{R}^{\mathbb{N}_4^2} \cong \mathbb{R}^{15},\, Q_{i,0}^t \succeq 0,\, Q_{i,1}^t \succeq 0\,(\forall\, i,t). \end{cases} \tag{5.36}$$

By solving (5.36), we can numerically verify that this GNEP is a GPG. The computed solution of (5.36) is displayed as follows (the coefficients are displayed with 6 decimal digits)

$$\begin{aligned} P(x) = {}& -3.176290x_1 - 1.000000(x_2)^2 - 0.000006(x_1)^2 \\ & -2.000000x_1x_2 + 0.219805(x_1)^3 - 0.000002(x_1)^4, \\ q_{10}(x,y_1) = {}& 2.176289 + 0.000006x_1 + 2.000000x_2 + 0.000006y_1 \\ & -0.219805(y_1)^2 + 0.000002(y_1)^3 - 0.219805(x_1)^2 + 0.000002(x_1)^3 \\ & -0.219805x_1y_1 + 0.000002x_1(y_1)^2 + 0.000002(x_1)^2y_1, \\ q_{11}(x,y_1) = {}& q_{20}(x,y_2) = q_{21}(x,y_2) = 0. \end{aligned}$$

The semidefinite program (5.34) is useful for checking GPGs. For instance, GNEPPs in Example 5.19 and 5.20 are numerically checked to be GPGs by solving (5.34). Moreover, the problems A11-13,A15,A17-A18 in [35] (see section 5.1) can be verified to be GPGs in the same way.

5.3 Numerical experiments

This section reports numerical experiments for solving GNEPPs by using Algorithm 5.1. The subproblem (5.4) is a polynomial optimization problem. We apply the software `GloptiPoly 3`

[49] and SeDuMi [124] to solve Moment-SOS relaxations of (5.4). The semidefinite program (5.34) for certifying GPGs is implemented by the software YALMIP [72]. The computation is implemented in a Dell XPS 15 9550 Laptop, with an Intel® Core(TM) i7-6700HQ CPU at 2.60GHz×4 and 16GB of RAM, in a Windows 10 operating system. In the computation, the sequence $\{x^{(k)}\}_{k=0}^{\infty}$ is regarded to converge if for some k it holds that

$$\|x^{(i)} - x^{(j)}\|_{\infty} \leq 10^{-8} \text{ for all } i, j \in \{k-10, \ldots, k\}. \tag{5.37}$$

The point $x^{(k)}$ is regarded as a GNE with the accuracy parameter $\varepsilon > 0$ if

$$|f_i(x^{(k)}) - f_i{}^*| \leq \varepsilon \tag{5.38}$$

for all players, where $f_i{}^*$ is the minimum value of (5.2) with $x_{-i} = x_{-i}^{(k)}$. Our computational results show that Algorithm 5.1 performs very well for solving GNEPPs, even if for nonconvex ones. First, we see some examples of the GPGs from Section 5.2.

Example 5.17. Consider the environmental pollution problem in the introduction and Example 5.9. We have seen that it is a GPG. Assume the number of players is $N = 2$ and the parameters $b_1 = b_2 = 2, E_1 = E_2 = 1, \gamma_{1,1} = 0.7, \gamma_{1,2} = 0.9, \gamma_{2,1} = \gamma_{2,2} = 0.8$. We run Algorithm 5.1 with $x_{1,0}^{(0)} = x_{1,1}^{(0)} = \cdots = x_{2,2}^{(0)} = 0.5$, and $\tau^{(0)} = 0.1$, $\tau^{(k+1)}$ updated as in (5.22). After 21 iterations, we get

$$x_{1,0}^{(21)} = 0.9999,\ x_{1,1}^{(21)} = 0,\ x_{1,2}^{(21)} = 0,\ x_{2,0}^{(21)} = 0.7500,\ x_{2,1}^{(21)} = 0,\ x_{2,2}^{(21)} = 0.9375.$$

Its accuracy parameter $\varepsilon = 1.7856 \cdot 10^{-8}$. It costs about 7 seconds.

Example 5.18. i) Consider the GNEP in Example 5.13. It is a GPG. All the individual optimization problems are convex. We run Algorithm 5.1 with the initial point $(x_1^{(0)}, x_2^{(0)}) = (3, 2)$ and fixed $\tau^{(k)} = 0.02$, and get a GNE $(2.0000, 2.0000)$ with $\varepsilon = 6.1541 \cdot 10^{-8}$. It runs 12 iterations and costs 2.6289 seconds.

ii) Consider the GNEP in Example 5.14. It is a GPG. We run Algorithm 5.1 with the initial point $(x_1^{(0)}, x_2^{(0)}) = (1, 0.125)$ and fixed $\tau^{(k)} = 0.02$. It returns the GNE $(1.2595, 0.1250)$ with $\varepsilon = 2.2891 \cdot 10^{-9}$. It runs 12 iterations and costs around 2 seconds.

iii) Consider the GNEP in Example 5.15. It is a GPG. All the individual optimization problems are convex. We run Algorithm 5.1 with the initial point $(x_{1,1}^{(0)}, x_{1,2}^{(0)}, x_2^{(0)}) = (1, 1, 2)$ and fixed $\tau^{(k)} = 0.02$. It returns the GNE $(1.3229, 0.5000, 1.5229)$ with $\varepsilon = 1.3631 \cdot 10^{-7}$. It runs 12 costs around 3 seconds.

iv) Consider the GNEP in Example 5.16. It is numerically verified to be a GPG. We run Algorithm 5.1 with the initial point $(x_1^{(0)}, x_2^{(0)}) = (0.2, 0.3)$ and fixed $\tau^{(k)} = 0.02$. For $k = 12$, we get $x^{(12)} = (0.9539, 0.3)$. The iteration difference is $2.1792 \cdot 10^{-8}$ and the GNE accuracy $\varepsilon = 5.4170 \cdot 10^{-9}$. It costs about 1.6 seconds.

Example 5.19. Consider the 2-player GNEP such that the individual optimization problems are respectively

$$
\begin{array}{cl|cl}
\min\limits_{x_1 \in \mathbb{R}^2} & x_{1,1}(x_{1,2} + 2x_{2,1} + 2x_{2,2}) & \min\limits_{x_2 \in \mathbb{R}^2} & (x_{1,1})^2 + (x_{1,2})^2 \\
& \quad +x_{1,2}(x_{2,1} + x_{2,2}) + 2x_{2,1}x_{2,2} & & \quad -(x_{2,1})^2 - (x_{2,2})^2 \\
s.t. & \sum\limits_{i=1}^{2}\sum\limits_{j=1}^{2} x_{i,j} = 1, & s.t. & \sum\limits_{i=1}^{2}\sum\limits_{j=1}^{2} x_{i,j} = 1, \\
& x_{1,1} \geq 0,\ x_{1,2} \geq 0, & & x_{2,1} \geq 0,\ x_{2,2} \geq 0.
\end{array}
$$

By solving the semidefinite program (5.34), we can numerically check that this GNEP is a GPG. Run Algorithm 5.1 with the initial points $x_1^{(0)} = (0.2, 0.3)$, $x_2^{(0)} = (0.2, 0.3)$ and fixed $\tau^{(k)} = 0.02$. After 19 loops, we get that

$$x^{(19)} = (0, 0.5, 0, 0.5).$$

It is a GNE with accuracy parameter $\varepsilon = 5.1857 \cdot 10^{-7}$. It costs about 5.36 seconds.

Example 5.20. Consider the 2-player GNEP whose optimization problems are

$$
\begin{array}{cl|cl}
\min\limits_{x_1 \in \mathbb{R}^2} & -2(x_{1,2})^2 + x_{2,1}x_{1,2} + x_{1,1}x_{2,1} & \min\limits_{x_2 \in \mathbb{R}^2} & (x_{2,1})^2 - 2x_{1,2}x_{2,2} \\
& & & \quad -2x_{1,1}x_{2,2} + (x_{2,2})^2 \\
s.t. & x_{1,1} + x_{1,2} + x_{2,1} + x_{2,2} = 1 & s.t. & x_{1,1} + x_{1,2} + x_{2,1} + x_{2,2} = 1 \\
& x_{1,1}, x_{1,2} \geq 0.1 & & x_{2,1}, x_{2,2} \geq 0.1
\end{array}
$$

By solving the semidefinite program (5.34), one can numerically check that this GNEP is a GPG. We run Algorithm 5.1 with

$$x^{(0)} = (0.25, 0.25, 0.25, 0.25), \quad \tau^{(0)} = 0.1,$$

and $\tau^{(k+1)}$ updated as (5.22). For $k = 12$, we get

$$x^{(12)} = (0.1000, 0.4000, 0.1000, 0.4000),$$

which is a GNE. The accuracy $\varepsilon = 1.14611 \cdot 10^{-8}$. It costs around 2.7 seconds.

Example 5.21. Consider the GNEP whose optimization problems are

$$\begin{array}{cl|cl} \min\limits_{x_1 \in \mathbb{R}^2} & (x_{1,1})^2 + (x_{1,2})^2 + x_{1,1} + x_{1,2} & \min\limits_{x_2 \in \mathbb{R}^2} & (x_{2,2})^2 - x_{2,1}x_{2,2} \\ s.t. & \|x_1\|^2 + \|x_2\|^2 \leq 1, & s.t. & \|x_1\|^2 + \|x_2\|^2 \leq 1, \\ & x_{1,1} \geq 0,\ x_{1,2} \leq 0.5, & & x_{2,1} \leq 0,\ 0.3 \leq x_{2,2} \leq 0.8. \end{array}$$

This is a GPG [37]. We run Algorithm 5.1 with

$$x_1^{(0)} = (0.5, 0.5), \quad x_2^{(0)} = (-0.6, 0.6), \quad \tau^{(0)} = 0.1,$$

and $\tau^{(k+1)}$ updated as (5.22). For $k = 16$, we get $x^{(16)} = (0, -0.5, 0, 0.3)$ as a GNE with accuracy parameter $\varepsilon = 4.1908 \cdot 10^{-10}$. It costs around 4.11 seconds.

Example 5.22. Consider the 3-player GNEP whose optimization problems are

$$\begin{array}{cl|cl|cl} \min\limits_{x_1 \in \mathbb{R}^1} & (x_1 - x_2)^2 & \min\limits_{x_2 \in \mathbb{R}^1} & (x_2 - x_3)^2 & \min\limits_{x_3 \in \mathbb{R}^1} & (x_3 - x_1)^2 \\ s.t. & \sum_{i=1}^3 (x_i)^2 \leq 10 & s.t. & x_2 \leq 3 & s.t. & \sum_{i=1}^3 x_i \leq 6. \end{array}$$

Any feasible point x with $x_1 = x_2 = x_3$ is a GNE, with optimal value 0 for all players. If we run Algorithm 5.1 with $(x_1^{(0)}, x_2^{(0)}, x_3^{(0)}) = (0, 1, 2)$ and $\tau^{(k)} = 0$ (which is actually not allowed since we require $\tau > 0$, but we still show the result of $\tau = 0$ in order to show the necessity of a positive τ), then we get an alternating sequence

$$(0,1,2) \longrightarrow (1,1,2) \longrightarrow (1,2,2) \longrightarrow (1,2,1) \longrightarrow (2,2,1)$$
$$\longrightarrow (2,1,1) \longrightarrow (2,1,2) \longrightarrow (1,1,2) \longrightarrow \cdots.$$

If we run Algorithm 5.1 with the same initial point $x^{(0)} = (0, 1, 2)$ but different regularization parameter $\tau^{(k)}$, the computational results are reported in Table 5.1. We run it for five different $\tau^{(k)}$. Two of them are fixed values $0.1, 0.05$, and the other one is $\tau_0 = 0.5$, $\tau^{(k+1)}$ updated as (5.22). In the table, "Iteration Difference" is the value of

$$\max_{291 \leq i < j \leq 300} \|x^{(i)} - x^{(j)}\|_\infty$$

since none of these five sequence satisfies (5.37) at the 300 iteration. The "ε" is the value such that $x^{(300)}$ can be verified up to accuracy ε as a GNE.

Example 5.23. Consider the following 2-player GNEP

$$\begin{array}{cl|cl} \min\limits_{x_1 \in \mathbb{R}^2} & (x_{1,1})^3 + x_{1,2}x_{2,1} + x_{1,1}x_{1,2} + x_{2,2} & \min\limits_{x_2 \in \mathbb{R}^2} & -(x_{2,1})^4 + x_{1,1}(x_{2,2})^2 \\ s.t. & (x_{1,1})^2 + (x_{1,2})^2 \leq 1 & s.t. & x_{1,1} \leq (x_{2,1})^2 + (x_{2,2})^2 \leq 1. \end{array}$$

Table 5.1: Computational Results for Example 5.22

$\tau^{(k+1)}$	x_1	x_2	x_3	Iteration Difference	ε
0.1000	1.4289	1.4289	1.4289	$1.9139 \cdot 10^{-5}$	10^{-7}
0.0500	1.4494	1.4494	1.4494	$5.2015 \cdot 10^{-5}$	10^{-7}
(5.22)	1.4116	1.4106	1.4116	0.0020	10^{-6}

It can be observed that both objective functions are nonconvex. Further the feasible set of the second player is not convex neither. We run Algorithm 5.1 with $x^{(0)} = (0.5, 0.5, 0.6, 0.6)$ and fixed $\tau^{(k)} = 0.02$. For $k = 16$, we get

$$x^{(16)} = (-0.9342,\ -0.3568,\ 1.0000,\ -3.7839 \cdot 10^{-5}),$$

which is a GNE. The accuracy $\varepsilon = 6.1075 \cdot 10^{-8}$. It costs around 3.68 seconds.

Example 5.24. ([32, 55]) Consider the example of a model for Internet switching [32, 55]. Assume there are N users, and the maximum capacity of the buffer is B. The x^i denotes the amount of ith user's "packets" in the buffer. It is clear $x^i \geq 0$ for any i. We also suggest the buffer is managed with *"drop-tail" policy*, which means if the buffer is full, further packets will be lost and resent. Let $\frac{x_i}{x_1+\cdots+x_N}$ be the *transmission rate* of user i, and $\frac{x_1+\cdots+x_N}{B}$ represent the *congestion level* of the buffer, and $1 - \frac{x_1+\cdots+x_N}{B}$ measure the decrease in the utility of the ith user as the congestion level increases. The ith user's optimization problem is

$$\begin{cases} \min\limits_{x_i} & f_i(x) = -\frac{x_i}{x_1+\cdots+x_N}(1 - \frac{x_1+\cdots+x_N}{B}) \\ s.t. & x_i \geq 0,\ x_1 + \cdots x_N \leq B. \end{cases}$$

It can be transformed into a polynomial optimization problem by introducing a new variable y_i for each player. The GNEP is then equivalent to that

$$\begin{cases} \min\limits_{x_i, y_i} & -x_i y_i (1 - \frac{\sum x_i}{B}) \\ s.t. & x_i \geq 0,\ x_1 + \cdots x_N \leq B \\ & (x_1 + \cdots + x_N) y_i = 1. \end{cases}$$

Here, we consider the case that $B = 1$ and $N = 10$, and run Algorithm 5.1 with the initial point

$$(0.4, \underbrace{0.01, 0.01, \ldots, 0.01}_{9 \text{ times}}, \underbrace{1/0.49, 1/0.49, \ldots, 1/0.49}_{10 \text{ times}}),$$

and $\tau^{(0)} = 0.1$. The parameters $\tau^{(k)}$ are updated as in (5.22). After 47 iterations, Algorithm 5.1 returned the point (here we only show the result of $x_1, \ldots, x_{10}$)

$$(0.09,\ 0.09,\ \ldots,\ 0.09).$$

with accuracy parameter $\varepsilon = 1.6344 \cdot 10^{-8}$. It costs around 61.93 seconds.

Example 5.25. ([35, A. 1]) Consider a variation of the GNEP in the last example that we change the constraints of the first player to $0.3 \leq x_i \leq 0.5$. This GNEP can also be transformed into a GNEPP by introducing a new variable y_i for each player and it is then equivalent to that

$$\begin{array}{cl|cl} & \text{player } \ i = 1 & & \text{player } \ i > 1 \\ \min\limits_{x_1, y_1 \in \mathbb{R}} & -x_1 y_1 (1 - \frac{\sum x_1}{B}) & \min\limits_{x_i, y_i \in \mathbb{R}} & -x_i y_i (1 - \frac{\sum x_i}{B}) \\ s.t. & 0.3 \leq x_1 \leq 0.5 & s.t. & x_1 + \cdots + x_N \leq B,\, x_i \geq 0.001 \\ & (x_1 + \cdots + x_N) y_1 = 1 & & (x_1 + \cdots + x_N) y_i = 1. \end{array}$$

Here, we consider the case that $B = 1$ and $N = 10$, the same as in [35]. We run Algorithm 5.1 with the initial point

$$(0.3, \underbrace{0.01, 0.01, \ldots, 0.01}_{9 \text{ times}}, \underbrace{1/0.39, 1/0.39, \ldots, 1/0.39}_{10 \text{ times}})$$

and $\tau^{(0)} = 0.1$. The parameters $\tau^{(k)}$ are updated as in (5.22). After 47 iterations, Algorithm 5.1 returned the point (here we only show the result of $x_1, \ldots, x_{10}$)

$$(0.3, 0.06943, 0.06943, \ldots, 0.06943).$$

with accuracy parameter $\varepsilon = 1.1261 \cdot 10^{-8}$. It costs around 60.69 seconds.

Example 5.26. Consider the GNEP which is the same as in Example 5.25 except we change the objective function to

$$f_i(x) = \frac{x_i}{x_1 + \cdots + x_N}(1 - \frac{x_1 + \cdots + x_N}{B}).$$

We still consider the case that $B = 1$ and $N = 10$, and the same technique to transform each player's subproblem into polynomial optimization problems. Start from the initial point

$$(0.3, \underbrace{0.01, 0.01, \ldots, 0.01}_{9 \text{ times}}, \underbrace{1/0.39, 1/0.39, \ldots, 1/0.39}_{10 \text{ times}})$$

with $\tau^{(0)} = 0.1$ and $\tau^{(k)}$ updated as in (5.22). After 44 iterations, Algorithm 5.1 returns the GNE (here we only show the result of $x_1, \ldots, x_{10}$)

$$(0.5000,\ 0.4920,\ 0.0010,\ \ldots,\ 0.0010)$$

with the accuracy parameter $\epsilon = 3.7773 \cdot 10^{-7}$. It took about 59 seconds.

Example 5.27. (Random GNEPPs with joint simplex/ball constraints) We randomly generate objective polynomials for each player with the joint simplex/ball constraint

$$\sum_{i=1}^{N} \sum_{j=1}^{n_i} x_{i,j} = 1, x_{i,j} \geq 0, \qquad \text{or} \qquad \|x_1\|^2 + \cdots + \|x_N\|^2 \leq 1.$$

We generate 100 random instances and count the number of problems that was solved successfully by Algorithm 5.1. The accuracy parameter is set to be $\varepsilon = 10^{-6}$ for checking $x^{(k)}$ as a GNE, i.e., we regard $(x^{(k)})$ as a GNE if (5.38) was satisfied with $\varepsilon = 10^{-6}$. For each instance, we run Algorithm 5.1 for at most 200 loops with $\tau^{(0)} = 0.1$, $\tau^{(k+1)}$ updated as in (5.22). If it does not return a GNE with required accuracy, we regard that it fails to solve the GNEP. The performance of Algorithm 5.1 is reported in Table 5.2. The number N is the number of players, n_i is the dimension of the ith player's strategy vector, and d is the degree of objective polynomials. The time is measured in seconds.

Table 5.2: Computational Results for Example 5.27

			Joint Simplex		Joint Ball	
N	$(n_1, \ldots, n_N)$	d	Succ. Rate	Ave. Time	Succ. Rate	Ave. Time
3	(2,2,2)	3	100%	9.97	94 %	16.71
3	(2,2,2)	4	92%	46.10	83 %	37.88
3	(3,3,3)	2	95%	11.21	97 %	9.93
3	(3,3,3)	3	92%	36.21	96 %	38.44
3	(3,3,3)	4	84%	98.76	88 %	88.98
4	(3,3,3,3)	2	94%	19.50	96 %	19.10
2	(4,3)	3	97%	13.53	92 %	17.55
2	(4,3)	4	92%	52.54	94 %	55.65
3	(3,2,4)	2	96%	9.43	97 %	9.09
3	(3,2,4)	3	92%	44.53	98 %	26.06
4	(3,2,4,2)	2	93%	19.52	95 %	22.73
4	(3,2,4,2)	3	94%	70.76	96 %	89.46

5.3.1 Test problems in [35]

We apply Algorithm 5.1 to solve the GNEPs in [35] that are GNEPPs or that can be transformed into GNEPPs. We normalize the objective functions such that the greatest absolute values of the coefficients are equal to one. For example, the problem A.17 in [35] is normalized as follows:

$$\begin{array}{cl} \min\limits_{x_1 \in \mathbb{R}^2} & \frac{1}{38}((x_{1,1})^2 + x_{1,1}x_{1,2} + (x_{1,2})^2) \\ & \quad +(x_{1,1} + x_{1,2})x_{2,1} - \frac{25x_{1,1}}{38} - x_{1,2} \\ s.t. & x_{1,1} + 2x_{1,2} - x_{2,1} \leq 14, \\ & 3x_{1,1} + 2x_{1,2} + x_{2,1} \leq 30, \\ & x_1 \geq 0, \end{array} \quad \Bigg| \quad \begin{array}{cl} \min\limits_{x_2 \in \mathbb{R}^1} & \frac{1}{25}(x_{1,1} + x_{1,2})x_{2,1} \\ & \quad +\frac{1}{25}(x_{2,1})^2 - x_{2,1} \\ s.t. & x_{1,1} + 2x_{1,2} - x_{2,1} \leq 14, \\ & 3x_{1,1} + 2x_{1,2} + x_{2,1} \leq 30, \\ & x_1 \geq 0. \end{array}$$

For the test problem A.2 and A.14, we use the same technique as shown in Example 5.25 to transform these non-polynomial GNEPs into GNEPPs. For the test problem A.10a, we run Algorithm 5.1 with the same initial point as in [35] and yield an alternative sequence that that is not convergent. Moreover, we also run Algorithm 5.1 with randomly generated feasible initial points for 100 times and no convergent sequence can be obtained. All the parameters are settled the same as in [35] for each problem. The computational results are shown in Table 5.3, where e denotes the vector of all ones. All the problems, except problem A.10a, were solved successfully by Algorithm 5.1.

Acknowledgments. This Chapter, in full, is a reprint of the material as it appears in *Computational Optimization and Applications*, Springer Science+Business Media. The dissertation author coauthored this paper with Nie, Jiawang and Xu, Lingling.

Table 5.3: Computational Results for test problems in [35]

problem	initial point	τ_0	$\tau^{(k+1)}$	iterations	time	ε
A.2	$0.05e$	0.1	(5.22)	27	37.61	$0.73 \cdot 10^{-7}$
A.3	$0.1e$	0.1	(5.22)	46	13.13	$0.10 \cdot 10^{-5}$
A.4	$0.1e$	0.1	(5.22)	12	10.96	$0.32 \cdot 10^{-6}$
A.5	$0.1e$	0.1	(5.22)	25	10.21	$0.16 \cdot 10^{-7}$
A.6	e	0.1	(5.22)	38	11.38	$0.41 \cdot 10^{-6}$
A.7	e	0.1	(5.22)	17	10.74	$0.21 \cdot 10^{-7}$
A.8	$0.5e$	0.1	(5.22)	54	14.84	$0.52 \cdot 10^{-6}$
A.10a	see [35]	0.1	(5.22)	200	not convergent	
A.11	$0.5e$	0.1	(5.22)	37	4.86	$0.11 \cdot 10^{-6}$
A.12	e	0.1	(5.22)	65	7.22	$0.17 \cdot 10^{-7}$
A.13	e	0.1	(5.22)	12	2.01	$0.71 \cdot 10^{-8}$
A.14	$0.1e$	0.1	(5.22)	42	50.50	$0.56 \cdot 10^{-8}$
A.15	e	0.0001	(5.22)	200	45.51	$0.13 \cdot 10^{-5}$
A.17	e	0.001	(5.22)	200	25.82	$0.19 \cdot 10^{-7}$
A.18	e	0.5	(5.22)	200	59.11	$0.29 \cdot 10^{-5}$

Bibliography

[1] A.A. Ahmadi and P.A. Parrilo. Sum of squares and polynomial convexity, *Proc. IEEE Conf. Decision Control*, 2009.

[2] A.A. Ahmadi and P.A. Parrilo. A complete characterization of the gap between convexity and SOS-convexity *SIAM Journal on Optimization*, 23.2 (2013): 811-833.

[3] J. Anselmi, D. Ardagna and M. Passacantando, Generalized nash equilibria for saas/paas clouds, *European Journal of Operational Research*, 236.1 (2014): 326-339.

[4] A. A. Ahmadi and J. Zhang, Semidefinite programming and Nash equilibria in bimatrix games, *INFORMS Journal on Computing*, to appear.

[5] J.P. Aubin, *Optima and equilibria: an introduction to nonlinear analysis*, vol. 140, Springer Science & Business Media, 2013.

[6] 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 (2017), pp. 381–395.

[7] K. Arrow and G. Debreu, Existence of an equilibrium for a competitive economy, *Econometrica: Journal of the Econometric Society*, 22 (1954), pp. 265–290.

[8] J. Aubin and H. Frankowska, *Set-Valued Analysis*, Springer Science & Business Media, 2009.

[9] D. Aussel and S. Sagratella, Sufficient conditions to compute any solution of a quasi-variational inequality via a variational inequality, *Mathematical Methods of Operations Research*, 85(1), 3-18, (2017).

[10] R. Aumann, and A. Brandenburger, Epistemic conditions for Nash equilibrium, *Econometrica: Journal of the Econometric Society* (1995): 1161-1180.

[11] T. Basar and G. J. Olsder, *Dynamic Noncooperative Game Theory*, SIAM, 23 (1999).

[12] B. Bank, J. Guddat, D. Klatte, B. Kummer and K. Tammer, *Non-Linear Parametric Optimization*, Akademie-Verlag, Berlin (1982).

[13] T. Başar and G. J. Olsder, *Dynamic noncooperative game theory*, SIAM, 1998.

[14] M. G. Bell, A game theory approach to measuring the performance reliability of transport networks, *Transportation Research Part B: Methodological*, 34(6), 533-545, 2000.

[15] E.G. Belousov and D. Klatte, A Frank–Wolfe type theorem for convex polynomial programs, *Computational Optimization and Applications*, 22.1 (2002): 37-48.

[16] D. Bertsekas. *Nonlinear programming*, second edition, Athena Scientific, 1995.

[17] J. Björnerstedt and J.W. Weibull, Nash equilibrium and evolution by imitation, *No. 407, IUI Working Paper*, 1994.

[18] J. Bochnak, M. Coste, and M. F. Roy, *Real algebraic geometry*, 293.C310, Vol. 36. Springer Science & Business Media, 2013.

[19] M. Breton, G. Zaccour, and M. Zahaf, A game-theoretic formulation of joint implementation of environmental projects, *European Journal of Operational Research*, 168 (2006), pp. 221–239.

[20] Y. Chen, G. Lan, and Y. Ouyang. Optimal primal-dual methods for a class of saddle point problems. *SIAM J. Optim.*, 24(4), 1779–1814, 2014.

[21] 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 (2004): 195-206.

[22] E. Couzoudis and P. Renner, Computing generalized Nash equilibria by polynomial programming, *Mathematical Methods of Operations Research*, 77.3 (2013), pp. 459-472.

[23] R. E. Curto, and L. A. Fialkow, Truncated K-moment problems in several variables, *Journal of Operator Theory*, pp.189-226, 2005.

[24] C. Daskalakis, P.W. Goldberg, and C.H. Papadimitriou, The complexity of computing a Nash equilibrium, *SIAM Journal on Computing*, 39(1), 195-259, 2009.

[25] R.S. Datta, Universality of Nash equilibria, *Mathematics of Operations Research*, 28 (2003), no. 3, 424–432.

[26] R.S. Datta, Finding all Nash equilibria of a finite game using polynomial algebra, *Economic Theory*, 42 (2010), no. 1, 55–96.

[27] G. Debreu, A social equilibrium existence theorem, *Proceedings of the National Academy of Sciences*, 38 (1952), pp. 886–893.

[28] A. Dreves, F. Facchinei, A. Fischer, and M. Herrich, A new error bound result for Generalized Nash Equilibrium Problems and its algorithmic application, *Computational Optimization and Applications*, 59 (2014), pp. 63–84.

[29] A. Dreves, F. Facchinei, C. Kanzow, and S. Sagratella, On the solution of the KKT conditions of Generalized Nash Equilibrium Problems, *SIAM Journal on Optimization*, 21 (2011), pp. 1082–1108.

[30] F. Facchinei, A. Fischer, and V. Piccialli, On generalized nash games and variational inequalities, *Operations Research Letters*, 35 (2007), pp. 159–164.

[31] F. Facchinei, A. Fischer, and V. Piccialli, On generalized nash games and variational inequalities, *Operations Research Letters*, 35 (2007), pp. 159–164.

[32] F. Facchinei, A. Fischer, and V. Piccialli, Generalized Nash Equilibrium Problems and Newton methods, *Mathematical Programming*, 117 (2009), pp. 163–194.

[33] F. Facchinei and C. Kanzow, Generalized Nash Equilibrium Problems, *Annals of Operations Research*, 175.1 (2010): 177-211.

[34] F. Facchinei and C. Kanzow, Generalized nash equilibrium problems, *4OR*, 5 (2007), pp. 173–210.

[35] F. Facchinei and C. Kanzow, Penalty methods for the solution of Generalized Nash Equilibrium problems, *SIAM Journal on Optimization*, 20 (2010), pp. 2228–2253.

[36] F. Facchinei and L. Lampariello, Partial penalization for the solution of Generalized Nash Equilibrium Problems, *Journal of Global Optimization*, 50(1):39-57, 2011.

[37] F. Facchinei, V. Piccialli, and M. Sciandrone, Decomposition algorithms for generalized potential games, *Computational Optimization and Applications*, 50 (2011), pp. 237–262.

[38] L. Fialkow and J. Nie, The truncated moment problem via homogenization and flat extensions, *Journal of Functional Analysis*, 263(6), 1682–1700, 2012.

[39] A. Fischer, M. Herrich, and K. Schönefeld, Generalized Nash Equilibrium Problems-recent advances and challenges, *Pesquisa Operacional*, 34 (2014), pp. 521–558.

[40] M. Fukushima, Restricted generalized Nash equilibria and controlled penalty algorithm, *Computational Management Science*, 8 (2011), pp. 201–208.

[41] G. Gürkan and J.S. Pang, Approximations of Nash equilibria, *Mathematical Programming*, 117(1-2), pp. 223–253, 2009.

[42] M.R. Garfinkel, Domestic politics and international conflict, *The American Economic Review*, 1294-1309, 1994.

[43] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, Generative adversarial nets, *In Advances in neural information processing systems* (pp. 2672-2680), 2014

[44] D. Han, H. Zhang, G. Qian, and L. Xu, An improved two-step method for solving Generalized Nash Equilibrium Problems, *European Journal of Operational Research*, 216(3), 613-623, 2012.

[45] P. Harker, Generalized nash games and quasi-variational inequalities, *European Journal of Operational Research*, 54 (1991), pp. 81–94.

[46] J. Harris, *Algebraic Geometry A First Course.*, Springer Verlag, Berlin, 1992.

[47] J.W. Helton and J. Nie, Semidefinite representation of convex sets, *Mathematical Programming*, 122.1 (2010): 21-64.

[48] D. Henrion and J. Lasserre, Detecting global optimality and extracting solutions in GloptiPoly, *Positive polynomials*, 293.C310, in controlLecture Notes in Control and Inform. Sci. 312,Springer, Berlin, 2005.

[49] D. Henrion, J. Lasserre, and J. Löfberg, Gloptipoly 3: moments, optimization and semidefinite programming, *Optimization Methods and Software*, 24(4-5):761–779, 2009.

[50] B. He and X. Yuan. Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective. *SIAM J. Imaging Sci.* 5 (2012), no. 1, 119–149.

[51] P.J.J. Herings, and R. Peeters. *Homotopy methods to compute equilibria in game theory*, Economic Theory 42.1 (2010): 119-156.

[52] A. von Heusinger and C. Kanzow, Relaxation methods for Generalized Nash Equilibrium Problems with inexact line search, *Journal of Optimization Theory and Applications*, 143 (2009), pp. 159–183.

[53] A. von Heusinger and C. Kanzow, Optimization reformulations of the Generalized Nash Equilibrium Problem using Nikaido-Isoda-type functions, *Computational Optimization and Applications*, 43 (2009), pp. 353–377.

[54] C. Kanzow and D. Steck, Augmented Lagrangian methods for the solution of Generalized Nash Equilibrium Problems, *SIAM Journal on Optimization*, 26 (2016), pp. 2034–2058.

[55] A. Kesselman, S. Leonardi and V. Bonifaci, Game-theoretic analysis of internet switching with selfish users, In: *International Workshop on Internet and Network Economics*, 26 (2005), pp. 236-245.

[56] E De Klerk, and M Laurent, On the Lasserre hierarchy of semidefinite programming relaxations of convex polynomial optimization problems, *SIAM Journal on Optimization*, 21.3 (2011): 824-832.

[57] S.C. Kontogiannis, P.N. Panagopoulou and P.G. Spirakis. Polynomial algorithms for approximating nash equilibria of bimatrix games, *International Workshop on Internet and Network Economics*, Springer, Berlin, Heidelberg, 2006.

[58] J.B. Krawczyk, and U. Stanislav, Relaxation algorithms to find Nash equilibria with economic applications, *Environmental Modeling & Assessment*, 5.1 (2000): 63-73.

[59] J.B. Krawczyk, *ECON 407 - Economic Dynamics B, Introduction to Dynamic Games with Application - Course Notes*, Works and Services, Printing Unit, Victoria University of Wellington (May 1995) p. 1774.

[60] R. Laraki and J. Lasserre. Semidefinite programming for min-max problems and games, *Mathematical Programming*, 131, pp. 305–332, 2012).

[61] J. Lasserre, Global optimization with polynomials and the problem of moments, *SIAM Journal on Optimization*, 11 (2001), pp. 796–817.

[62] J. Lasserre, Convexity in semialgebraic geometry and polynomial optimization, *SIAM Journal on Optimization* 19.4 (2009): 1995-2014.

[63] J. Lasserre, *An introduction to polynomial and semi-algebraic optimization*, Cambridge University Press, Volume 52, 2015.

[64] J. Lasserre, The Moment-SOS Hierarchy, *Proceedings of the International Congress of Mathematicians (ICM 2018)*, vol. 3, B. Sirakov, P. Ney de Souza and M. Viana (Eds.), pp. 3761–3784, World Scientific, 2019.

[65] 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.

[66] M. Laurent, Revisiting two theorems of Curto and Fialkow on moment matrices, *Proceedings of the American Mathematical Society*, 133(10), 2965-2976, 2005.

[67] M. Laurent, Sums of squares, moment matrices and optimization over polynomials, *Emerging Applications of Algebraic Geometry of IMA Volumes in Mathematics and its Applications*, vol. 149, pp. 157–270, Springer, 2009.

[68] M. Laurent, Optimization over polynomials: Selected topics, *Proceedings of the International Congress of Mathematicians*, ICM 2014, S. Jang, Y. Kim, D-W. Lee, and I. Yie (eds.), pp. 843-869, 2014.

[69] C.E. Lemke, J.T. Howson, Jr, Equilibrium points of bimatrix games. Journal of the Society for industrial and Applied Mathematics, *Journal of the Society for industrial and Applied Mathematics*, 12(2), pp.413-423,1964.

[70] S. Li and T. Başar, Distributed algorithms for the computation of noncooperative equilibria, *Automatica*, vol. 23, no. 4, pp. 523–533, 1987.

[71] R.J. Lipton and E. Markakis, Nash Equilibria via Polynomial Equations, *In: Farach-Colton M. (eds) LATIN 2004: Theoretical Informatics*, LATIN 2004. Lecture Notes in Computer Science, vol 2976. Springer, Berlin, Heidelberg.

[72] J. Lofberg, YALMIP: A toolbox for modeling and optimization in MATLAB, *2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508)*, IEEE, 2004.

[73] A.B. MacKenzie and S.B. Wicker, Game theory in communications: Motivation, explanation, and application to power control, *In GLOBECOM'01. IEEE Global Telecommunications Conference (Cat. No. 01CH37270)* (Vol. 2, pp. 821-826). IEEE.

[74] X. Mao, Q. Li, H. Xie, R.Y. Lau, Z. Wang, and S. Paul Smolley, Least squares generative adversarial networks, *In Proceedings of the IEEE International Conference on Computer Vision*, (pp. 2794-2802), 2017.

[75] D. Maistroskii. Gradient methods for finding saddle points, *Matekon* 13, 3–22 (1977).

[76] E. Maskin, Nash equilibrium and welfare optimality, *The Review of Economic Studies*, 6(1), 23-38, 1999.

[77] L. McKenzie, On the existence of a general equilibrium for a competitive market, *Econometrica*, 27 (1959), pp. 54-C71.

[78] D. Monderer and L. Shapley, Potential games, *Games Econ. Behav.*, 14 (1996), pp. 124–143.

[79] R.B. Myerson, Nash equilibrium and the history of economic theory, *Journal of Economic Literature*, 37.3 (1999): 1067-1082.

[80] R.B. Myerson, *Game theory*, Harvard University Press, 2013.

[81] K. Nabetani, P. Tseng, and M. Fukushima, Parametrized variational inequality approaches to Generalized Nash Equilibrium Problems with shared constraints, *Computational Optimization and Applications*, 48 (2011), pp. 423–452.

[82] J. Nash, Equilibrium points in n-person games, *Proc. Natl. Acad. Sci.*, 36 (1950), pp. 48–49.

[83] J. Nash, Non-cooperative games, *Annals of mathematics*, (1951): 286-295.

[84] J. Neel, J. Reed, and R. Gilles, The role of game theory in the analysis of software radio networks, *In Proceedings of SDR forum technical conference*, (2002), 2250–2255.

[85] A. Nemirovski. Prox-method with rate of convergence O(1/t) for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM J. Optim.* 15 (2004), no. 1, 229–251.

[86] J. Nie and B. Sturmfels, Matrix cubes parameterized by eigenvalues, *SIAM journal on matrix analysis and applications*, 31 (2), 755–766, 2009.

[87] J. Nie and R. Kristian, Algebraic degree of polynomial optimization, *SIAM Journal on Optimization*, 20.1 (2009): 485-502.

[88] J. Nie, Certifying convergence of Lasserre's hierarchy via flat truncation, *Mathematical Programming*, 142(1-2):485–510, 2013.

[89] J. Nie, Polynomial optimization with real varieties, *SIAM Journal On Optimization* 23.3 (2013): 1634-1646.

[90] J. Nie, Optimality conditions and finite convergence of Lasserre's hierarchy. *Mathematical programming*, 146(1-2):97–121, 2014.

[91] J. Nie, The hierarchy of local minimums in polynomial optimization, *Mathematical Programming* 151 (2), pp. 555–583, 2015.

[92] J. Nie, Linear optimization with cones of moments and nonnegative polynomials, *Mathematical Programming*, 153(1), 247–274, 2013.

[93] J. Nie, Generating polynomials and symmetric tensor decompositions, *Foundations of Computational Mathematics* 17 (2), 423–465, 2017.

[94] J. Nie, Low rank symmetric tensor approximations, *SIAM Journal on Matrix Analysis and Applications*, 38(4), 1517–1540, 2017.

[95] J. Nie, Tight relaxations for polynomial optimization and Lagrange multiplier expressions, *Mathematical Programming* 178, no.1-2, pp. 1–37, 2019.

[96] J. Nie and K. Ye, Hankel tensor decompositions and ranks, *SIAM Journal on Matrix Analysis and Applications*, vol. 40, no. 2, pp. 486–516, 2019.

[97] J. Nie, X. Tang and L. Xu, The Gauss-Seidel Method for Generalized Nash Equilibrium Problems of Polynomials, *Computational Optimization and Applications*, (2020).

[98] J. Nie and X. Tang, Nash Equilibrium Problems of Polynomials, *Preprint*, 2020. `arXiv:2006.09490`

[99] J. Nie and X. Tang, Convex Generalized Nash Equilibrium Problems and Polynomial Optimization, *Preprint*, 2021. `arXiv:2101.06504`

[100] J. Nie, L. Wang, J. Ye and S. Zhong, A Lagrange Multiplier Expression Method for Bilevel Polynomial Optimization, *Preprint*, 2020. `arXiv:2007.07933`

[101] J. Nie, Z. Yang and G. Zhou, The Saddle Point Problem of Polynomials, *Preprint*, 2018. `arXiv:1809.01218`

[102] G. Oggioni, Y. Smeers, E. Allevi, and S. Schaible, A Generalized Nash Equilibrium model of market coupling in the european power system, *Networks and Spatial Economics*, 12 (2012), pp. 503–560.

[103] M.J. Osborne and A. Rubinstein, *A course in game theory*, MIT press, 1994.

[104] D. Paccagnan, B. Gentile, F. Parise, M. Kamgarpour and J. Lygeros, Distributed computation of generalized Nash equilibria in quadratic aggregative games with affine coupling constraints, *IEEE 55th Conference on Decision and Control (CDC)*, pp. 6123–6128, IEEE, 2016.

[105] J. Pang and M. Fukushima, Quasi-variational inequalities, generalized nash equilibria, and multi-leader-follower games, *Computational Management Science*, 2 (2005), pp. 21–56.

[106] M. Putinar, Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal*, 42(3):969–984, 1993.

[107] A. Radford, L. Metz and S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, *arXiv preprint*, arXiv:1511.06434, 2015

[108] L.J. Ratliff, S.A. Burden, and S.S Sastry. Characterization and computation of local nash equilibria in continuous games, *2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, IEEE, 2013.

[109] S. Robinson, Shadow prices for measures of effectiveness, I: Linear model, *Operations Research*, 41 (1993), pp. 518–535.

[110] S. Robinson, Shadow prices for measures of effectiveness, II: General model, *Operations Research*, 41 (1993), pp. 536–548.

[111] R. Rockafellar and R. Wets, *Variational Analysis*, Springer Science & Business Media, Volume 317, 2009.

[112] J. Rosen, Existence and uniqueness of equilibrium points for concave n-person games, *Econometrica*, 33 (1965), 520–534.

[113] A. Rubinstein, *Perfect equilibrium in a bargaining model* Econometrica: Journal of the Econometric Society, 97-109, 1982.

[114] S. Sagratella, Computing all solutions of Nash equilibrium problems with discrete strategy sets, *SIAM Journal on Optimization*, 26(4): 2190-2218, 2016.

[115] S. Sagratella, Computing equilibria of Cournot oligopoly models with mixed-integer quantities, *Mathematical Methods of Operations Research* 86, no. 3 (2017): 549-565.

[116] S. Sagratella, Algorithms for generalized potential games with mixed-integer variables, *Computational Optimization and Applications* 68, no. 3 (2017): 689-717.

[117] S. Sagratella, On Generalized Nash Equilibrium Problems with linear coupling constraints and mixed-integer variables, *Optimization*, 68.1 (2019): 197-226.

[118] R. Savani, and B. Von Stengel. Hard-to-solve bimatrix games, *Econometrica*, 74.2 (2006): 397-429.

[119] N. Schofield and I. Sened, Local Nash equilibrium in multiparty politics, *Annals of Operations Research*, 109(1-4), 193-211.

[120] M. Schweighofer, Optimization of polynomials on compact semialgebraic sets, *SIAM J. Optim.*, vol. 15, no. 3, pp. 805–825, 2005.

[121] S. Scotti, *Structural design using equilibrium programming formulations*, Ph.D Thesis, 1995.

[122] I. R. Shafarevich, *Basic Algebraic Geometry 1*, Springer-Verlag Berlin Heidelberg 2013

[123] A. Shaked, Existence and computation of mixed strategy Nash equilibrium for 3-firms location problem *The Journal of Industrial Economics*, (1982): 93-96.

[124] J. Sturm, Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization methods and software*, 11(1-4):625–653, 1999.

[125] L. Sun and Z. Gao, An equilibrium model for urban transit assignment based on game theory, *European Journal of Operational Research*, 181 (2007), pp. 305–314.

[126] S. Uryas'ev and R.Y. Rubinstein, On relaxation algorithms in computation of non-cooperative equilibria, *IEEE Transactions on Automatic Control*, vol. 39, no. 6, pp. 1263–1267, 1994.

[127] D.E. Wildasin, Nash equilibria in models of fiscal competition. *Journal of public economics, 35(2), 229-240, 1988.*

[128] H. Yin, U.V. Shanbhag, and P.G. Mehta, Nash equilibrium problems with congestion costs and shared constraints, *In Proceedings of the 48h IEEE Conference on Decision and Control (CDC)*, held jointly with 2009 28th Chinese Control Conference (pp. 4649-4654). IEEE.

[129] D. Yue and F. You, Game-theoretic modeling and optimization of multi-echelon supply chain design and operation under stackelberg game and market equilibrium, *Computers & Chemical Engineering*, 71 (2014), 347–361.

[130] I. Zabotin. A subgradient method for finding a saddle point of a convex-concave function. *Issled. Prikl. Mat.* 15, 6–12 (1988).

[131] J. Zhou, W. Lam, and B. Heydecker, The Generalized Nash Equilibrium model for oligopolistic transit market with elastic demand, *Transportation Research Part B: Methodological*, 39 (2005), pp. 519–544.