This article was downloaded by: [2607:f720:1901::1ed] On: 24 August 2023, At: 16:28
Publisher: Institute for Operations Research and the Management Sciences (INFORMS)
INFORMS is located in Maryland, USA

Mathematics of Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Nash Equilibrium Problems of Polynomials

Jiawang Nie, Xindong Tang

To cite this article:
Jiawang Nie, Xindong Tang (2023) Nash Equilibrium Problems of Polynomials. Mathematics of Operations Research

Published online in Articles in Advance 14 Jul 2023

. https://doi.org/10.1287/moor.2022.0334

Full terms and conditions of use: https://pubsonline.informs.org/Publications/Librarians-Portal/PubsOnLine-Terms-and-Conditions

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2023, INFORMS

Please scroll down for article—it is on subsequent pages

With 12,500 members from nearly 90 countries, INFORMS is the largest international association of operations research (O.R.) and analytics professionals and students. INFORMS provides unique networking and learning opportunities for individual professionals, and organizations of all types and sizes, to better understand and use O.R. and analytics tools and methods to transform strategic visions and achieve better outcomes.
For more information on INFORMS, its publications, membership, or meetings visit http://www.informs.org

https://pubsonline.informs.org/journal/moor

MATHEMATICS OF OPERATIONS RESEARCH
Articles in Advance, pp. 1–26
ISSN 0364-765X (print), ISSN 1526-5471 (online)

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

Nash Equilibrium Problems of Polynomials

Jiawang Nie,[a,*] Xindong Tang[b]

[a] Department of Mathematics, University of California San Diego, La Jolla, California 92093; [b] Department of Applied Mathematics, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

*Corresponding author

Contact: njw@math.ucsd.edu, https://orcid.org/0000-0003-2637-8274 (JN); xindong.tang@polyu.edu.hk, https://orcid.org/0000-0002-4951-768X (XT)

Received: May 14, 2021
Revised: November 26, 2022
Accepted: April 24, 2023
Published Online in Articles in Advance: July 14, 2023

MSC2010 Subject Classification: Primary: 90C23, 90C33, 91A10, 65K05

https://doi.org/10.1287/moor.2022.0334

Copyright: © 2023 INFORMS

Abstract. This paper studies Nash equilibrium problems that are given by polynomial functions. We formulate efficient polynomial optimization problems for computing Nash equilibria. The Moment-sum-of-squares relaxations are used to solve them. Under generic assumptions, the method can find a Nash equilibrium, if there is one. Moreover, it can find all Nash equilibria if there are finitely many ones of them. The method can also detect non-existence if there is no Nash equilibrium.

Funding: J. Nie was supported by the National Science Foundation [Grant DMS-2110780].

Keywords: Nash equilibrium • polynomial optimization • Moment-SOS relaxation • Lagrange multiplier expression • tight relaxation

1. Introduction

The Nash equilibrium problem (NEP) is a kind of game for finding strategies for a group of players such that each player's objective is optimized, given other players' strategies. Suppose there are N players, and the ith player's strategy is the variable $x_i \in \mathbb{R}^{n_i}$ (the n_i-dimensional real Euclidean space). We denote that

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

The total dimension of all players' strategies is

$$n := n_1 + \cdots + n_N.$$

When the ith player's strategy x_i is being optimized, we use x_{-i} to denote the subvector of all players' strategies except x_i—that is,

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

and write $x = (x_i, x_{-i})$ accordingly. When the writing x_{-i} appears, the ith player's strategy is being considered for optimization, while the vector of all other players' strategies is fixed to be x_{-i}. In an NEP, the ith player's best strategy x_i is the minimizer for the optimization problem

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

for the given other players' strategies x_{-i}. 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 (1.1) is

$$X_i := \{x_i \in \mathbb{R}^{n_i} : g_{i,j}(x_i) = 0 \ (j \in \mathcal{E}_i), \ g_{i,j}(x_i) \geq 0 \ (j \in \mathcal{I}_i)\}. \tag{1.2}$$

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. We say the strategy vector x is

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

feasible if

$$x = (x_1, \ldots, x_N) \in X := X_1 \times \cdots \times X_n.$$

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

$$\text{find} \quad x^* \in \mathbb{R}^n \quad \text{such that each} \quad x_i^* \quad \text{is a minimizer of } \mathbf{F}_i(x_{-i}^*), \tag{1.3}$$

where $x^* = (x_1^*, \ldots, x_N^*)$. A solution of (1.3) is called a *Nash equilibrium* (NE).[1] When the defining functions f_i and $g_{i,j}$ are continuous, then the NEP is called a *continuous Nash equilibrium problem*. In this paper, we consider cases that each f_i is a polynomial in x and $g_{i,j}$'s are polynomials in x_i. Such an NEP is called a *Nash equilibrium problem of polynomials* (NEPP). The following is an example.

Example 1.1. Consider the two-player NEP with the individual optimization

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

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

In this NEP, each player's objective is strictly convex with respect to its strategy because their Hessian matrices with respect to their own strategies are positive definite. This NEP has only three NEs (see Section 3.3), which are

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

NEPs are challenging problems to solve. Even for the special cases where each player's objective function is multilinear in $(x_1, \ldots, x_N)$, and each feasible set is a simplex, finding an NE is Polynomial Parity Arguments on Directed Graphs-complete (Daskalakis et al. [8]). The problem becomes more difficult when players' optimization problems are nonconvex. This is because an NE $x^* = (x_1^*, \ldots, x_N^*)$ requires that each x_i^* is a global minimizer of $\mathbf{F}_i(x_{-i}^*)$. Indeed, finding a global minimizer of a single polynomial optimization problem is already NP-hard (Lasserre [28]). For polynomial optimization problems, global optimizers can be computed efficiently by the Moment-sum-of-squares (Moment-SOS) hierarchy of semidefinite relaxations (see Lasserre [26], Lasserre [28], and Laurent [30] for related work). Moreover, for some NEPs, there may not exist any NE. Such NEPs are also interesting and have important applications (e.g., NEPs in generative adversarial networks; Farnia and Ozdaglar [12]). If an NE does not exist, how can we detect its nonexistence? This question is mostly open for general NEPs, to the best of the author's knowledge. However, under certain nonsingularity conditions, nonexistence of NEs for NEPPs can be certified by the infeasibility of some semidefinite programs. For the above reasons, this paper focuses on NEPPs.

NEPs are generalizations of *finite games* (Nash [33]), where each X_i is a finite set—that is, $|X_i| < \infty$. In recent years, there has been an increasing number of applications of NEPs in various fields, such as economics, environmental protection, politics, supply chain management, machine learning, etc. We refer to Breton et al. [4], Contreras et al. [6], Farnia and Ozdaglar [12], Goodfellow et al. [14], Maskin [32], and Schofield and Sened [53] for some recent applications of NEPs. In Section 5, we present some concrete applications of NEPs in environmental pollution control and the electricity market. Moreover, we refer to surveys Aubin [3] and Young and Zamir [63] for more general work on NEPs.

In this paper, our primary goal is to find NEs for NEPs. In the following, we review some previous work on solving NEPs. The NEP is called a *zero-sum game* if the sum of objective functions is identically equal to a constant. Two-player zero-sum games are equivalent to *saddle-point problems*. We refer to Chen et al. [5] and Nedić and Ozdaglar [34] for algorithms of solving saddle-point problems under convexity assumptions and Nie et al. [48] for the method of solving nonconvex polynomial saddle-point problems. For finite games, finding mixed-strategy solutions is a special case of NEPs of polynomials; see Ahmadi and Zhang [2], Datta [9], Kontogiannis et al. [21], and Young and Zamir [63] for some related approaches. There exists work on mixed-strategy solutions for continuous games; see Dresher et al. [10], Parrilo [49], and Stein et al. [56] for mixed-strategy solutions to polynomial games

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

and Adam et al. [1] and Kroupa and Votroubek [23] for the recently developed multiple oracle algorithms. For finding pure strategy solutions for general continuous NEPs, we refer to techniques such as variational inequalities (Gürkan and Pang [15] and Kulkarni and Shanbhag [24]), Nikaido-Isososda functions (Krawczyk and Uryasev [22], Uryas'ev and Rubinstein [58]), and manifold optimization tools (Ratliff et al. [52]). In most earlier work, convexity is often assumed for each player's optimization. Moreover, NEPs are special cases of GNEPs (Facchinei and Kanzow [11]), where each player's feasible set is dependent on other players' strategies. For GNEPs given by polynomial functions, the work Couzoudis and Renner [7] introduces a parametric SOS relaxation approach, and the Gauss-Seidel method using Moment-SOS relaxations is studied in Nie et al. [47]. When the GNEPs are further assumed to be convex, the semidefinite relaxation method is introduced in Nie and Tang [45]. At the moment, it is mostly an open question to solve general NEPs, especially when the players' optimization problems are nonconvex.

1.1. Contributions

This paper focuses on Nash equilibrium problems that are given by polynomials. We formulate efficient polynomial optimization for computing one or more Nash equilibria. The Moment-SOS hierarchy of semidefinite relaxations is used to solve the appearing polynomial optimization problems. Our major results are:

- Under some genericity assumptions, we prove that our method can compute a Nash equilibrium if there exists one, or it can detect nonexistence of NEs. Moreover, if there are only finitely many NEs, we show how to find all of them. In the prior existing work, there do not exist similar methods that can achieve such computational goals.
- When the objective and constraining polynomials are generic (i.e., they have generic coefficients), we show that the NEPP has only finitely many KKT points. For such generic NEPPs, our method can compute all NEs, if they exist, or can detect their nonexistence.
- When the objective and constraining polynomials are not generic, our method can still be applied to compute one or more NEs, or to detect their nonexistence. Even if there are infinitely many NEs, our method may still be able to get an NE. In computational practice, there is no need to check whether the NEP is generic or not to implement our algorithms. In fact, our method is self-verifying, that in the actual implementation, the algorithm can check whether the computed point is an NE and check whether the computed solution set is complete or not.

The paper is organized as follows. Some preliminaries about polynomial optimization are given in Section 2. We give efficient polynomial optimization formulations in Section 3. We show how to solve polynomial optimization problems by the Moment-SOS hierarchy in Section 4. Numerical experiments and applications are given in Section 5. Conclusions and discussions are proposed in Section 6. The finiteness of the KKT set for generic NEPs is shown in the appendix.

2. Preliminaries

2.1. Notation

The symbol $\mathbb{N}$ [respectively (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 $[k] := \{1, \ldots, k\}$. 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). For the ith player's strategy variable $x_i \in \mathbb{R}^{n_i}$, the $x_{i,j}$ denotes the jth entry of x_i, $j = 1, \ldots, n_i$. The $\mathbb{R}[x]$ (resp., $\mathbb{C}[x]$) denotes the ring of polynomials with real (resp., complex) coefficients in x. The $\mathbb{R}[x]_d$ (resp., $\mathbb{C}[x]_d$) denotes its subset of polynomials whose degrees are not greater than d. For the ith player's strategy vector x_i, the notations $\mathbb{R}[x_i]$, $\mathbb{C}[x_i]$, $\mathbb{R}[x_i]_d$, $\mathbb{C}[x_i]_d$ are defined in the same way. For the ith player's objective $f_i(x_i, x_{-i})$, the notations $\nabla_{x_i} f_i$, $\nabla^2_{x_i} f_i$, respectively, denote its gradient and Hessian with respect to x_i.

In the following, we use the letter z to represent either x or x_i for the convenience of discussion. Suppose $z := (z_1, \ldots, z_l)$ and $\alpha := (\alpha_1, \ldots, \alpha_l) \in \mathbb{N}^l$; denote

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

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

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

We use $[z]_d$ to denote the vector of all monomials in z and whose degree is at most d, ordered in the graded alphabetical ordering. For example, 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, the word "generic" is used for a property if it holds for all points outside a set of Lebesgue measure zero in the space of input data. For a given multidegree $(d_1, \ldots, d_N)$ (resp., a degree d) in the variable

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

$x = (x_1, \ldots, x_N)$ (resp., in variable x_i), we say a polynomial $p(x)$ (resp., $q(x_i)$) is generic if the coefficient vector of p (resp., q) is generic in the space of coefficients. For multidegrees $a_1, \ldots, a_N$ and degrees $b_{1,1}, b_{1,2}, \ldots, b_{1,m_1}, b_{2,1}, \ldots, b_{N,m_N}$, we say the NEPP is generic if for each i and j, the $f_i(x_1, \ldots, x_N)$ is a generic polynomial with multidegree a_i, and the $g_{i,j}(x_i)$ is a generic polynomial whose degree is $b_{i,j}$.

2.2. 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

$$\mathrm{Ideal}[q] := q_1 \cdot \mathbb{F}[z] + \cdots + q_m \cdot \mathbb{F}[z],$$

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 if $\sigma = s_1^2 + s_2^2 + \cdots + s_t^2$ for some polynomials $s_1, \ldots, s_t \in \mathbb{R}[z]$. The set of all SOS polynomials in z is denoted as $\Sigma[z]$. For a degree k, we denote the truncation

$$\Sigma[z]_{2k} := \Sigma[z] \cap \mathbb{R}[z]_{2k}.$$

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

$$\mathrm{Qmod}[g] := \Sigma[z] + g_1 \cdot \Sigma[z] + \cdots + g_t \cdot \Sigma[z].$$

Similarly, we denote the truncation of $\mathrm{Qmod}(g)$

$$\mathrm{Qmod}[g]_{2k} := \Sigma[z]_{2k} + g_1 \cdot \Sigma[z]_{2k-\deg(g_1)} + \cdots + g_t \cdot \Sigma[z]_{2k-\deg(g_t)}.$$

The tuple g determines the basic closed semialgebraic set

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

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

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

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 [50] Positivestellensatz. 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 (Nie [40]).

2.3. Localizing and Moment Matrices

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

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

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

$$\langle f, y \rangle = \sum_{\alpha \in \mathbb{N}^l_{2k}} f_\alpha y_\alpha. \tag{2.2}$$

The operation $\langle f, y \rangle$ is a bilinear function in (f, y). For a polynomial $q \in \mathbb{R}[z]$ with $\deg(q) \leq 2k$ and the integer

$$t = k - \lceil \deg(q)/2 \rceil,$$

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

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

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

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

$$L_q^{(k)}[y] := \sum_{\alpha \in \mathbb{N}_{2k}^l} y_\alpha Q_\alpha. \tag{2.3}$$

It is called the kth *localizing matrix* of q and generated by y. For given q, $L_q^{(d)}[y]$ is linear in y. Clearly, if $q(u) \geq 0$ and $y = [u]_{2k}$, then

$$L_q^{(k)}[y] = q(u)[u]_t[u]_t^T \succeq 0.$$

For instance, if $l = k = 2$ and $q(z) = 1 - z_1 - z_1 z_2$, then

$$L_q^{(2)}[y] = \begin{bmatrix} y_{00} - y_{10} - y_{11} & y_{10} - y_{20} - y_{21} & y_{01} - y_{11} - y_{12} \\ y_{10} - y_{20} - y_{21} & y_{20} - y_{30} - y_{31} & y_{11} - y_{21} - y_{22} \\ y_{01} - y_{11} - y_{12} & y_{11} - y_{21} - y_{22} & y_{02} - y_{12} - y_{13} \end{bmatrix}.$$

When q is the constant one polynomial, the localizing matrix $L_1^{(k)}[y]$ reduces to a moment matrix, which we denote as

$$M_k[y] := L_1^{(k)}[y]. \tag{2.4}$$

For instance, for $n = 2$ and $y \in \mathbb{R}^{\mathbb{N}_4^2}$, we have $M_0[y] = [y_{00}]$,

$$M_1[y] = \begin{bmatrix} y_{00} & y_{10} & y_{01} \\ y_{10} & y_{20} & y_{11} \\ y_{01} & y_{11} & y_{02} \end{bmatrix}, \quad M_2[y] = \begin{bmatrix} y_{00} & y_{10} & y_{01} & y_{20} & y_{11} & y_{02} \\ y_{10} & y_{20} & y_{11} & y_{30} & y_{21} & y_{12} \\ y_{01} & y_{11} & y_{02} & y_{21} & y_{12} & y_{03} \\ y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22} \\ y_{11} & y_{21} & y_{12} & y_{31} & y_{22} & y_{13} \\ y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04} \end{bmatrix}.$$

Localizing and moment matrices are basic tools to formulate semidefinite relaxations for polynomial optimization problems. They are important tools for solving polynomial, matrix, and tensor optimization problems (Hillar and Nie [20], Nie [35], Nie [36], Nie [41], Nie and Zhang [46]).

2.4. Optimality Conditions for NEPs

Consider the ith player's individual optimization problem $\mathbf{F}_i(x_{-i})$ in (1.1), for given x_{-i}. Suppose $\mathcal{E}_i \cup \mathcal{I}_i = [m_i]$ for some $m_i \in \mathbb{N}$. For convenience, we write the constraining functions as

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

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

$$\begin{cases} \displaystyle\sum_{j=1}^{m_i} \lambda_{i,j} \nabla_{x_i} g_{i,j}(x_i) = \nabla_{x_i} f_i(x), \\ 0 \leq \lambda_{i,j} \perp g_{i,j}(x_i) \geq 0 \, (j \in \mathcal{I}_i). \end{cases} \tag{2.5}$$

In the above, $\lambda_{i,j} \perp g_{i,j}(x_i)$ means that $\lambda_{i,j} \cdot g_{i,j}(x_i) = 0$. The system (2.5) is called the KKT conditions for the optimization $\mathbf{F}_i(x_{-i})$. We say a point $x \in \mathbb{R}^n$ is a *KKT point* if there exist vectors of Lagrange multipliers $\lambda_1, \ldots, \lambda_N$ such that (2.5) holds. For the NE x, if the LICQC of $\mathbf{F}_i(x_{-i})$ holds at x_i for every $i \in [N]$, then x must be a KKT point. Moreover, if each player's optimization problem is convex—that is, the $f_i(x_i, x_{-i})$ is convex in x_i for all $x_{-i} \in X_1 \times \cdots \times X_{i-1} \times X_{i+1} \times \cdots \times X_N$—and every X_i is a convex set, then all KKT points are NEs (Facchinei and Kanzow [11, theorem 4.6]).

Example 2.1. Consider the two-player NEP in Example 1.1. Each individual optimization is strictly convex, because Hessian matrices $\nabla_{x_1}^2 f_1$ and $\nabla_{x_2}^2 f_2$ are positive definite. The constraints are the convex ball conditions. The KKT system is

$$\begin{cases} 2x_{1,1} + x_{2,1} + 4x_{2,2} = -2\lambda_1 x_{1,1}, 4x_{1,2} = -2\lambda_1 x_{1,2}, \\ x_{1,1} + 2x_{1,2} + 2x_{2,1} = -2\lambda_2 x_{2,1}, 2x_{1,1} + x_{1,2} + 2x_{2,2} = -2\lambda_2 x_{2,2}, \\ \lambda_1(1 - (x_{1,1})^2 - (x_{1,2})^2) = 0, \lambda_2(1 - (x_{2,1})^2 - (x_{2,2})^2) = 0, \\ 1 - (x_{1,1})^2 - (x_{1,2})^2 \geq 0, 1 - (x_{2,1})^2 - (x_{2,2})^2 \geq 0, \\ \lambda_1 \geq 0, \lambda_2 \geq 0. \end{cases} \tag{2.6}$$

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

By solving the above directly, one can show that this NEP has only three KKT points, together with Lagrange multipliers as follows:

Nash equilibrium	Lagrange multiplier
$x_1^* = (0,0), \quad x_2^* = (0,0),$	$\lambda_1^* = \lambda_2^* = 0;$
$x_1^* = (1,0), \quad x_2^* = \frac{1}{\sqrt{5}}(-1,-2),$	$\lambda_1^* = \frac{9\sqrt{5}}{10} - 1, \quad \lambda_2^* = \frac{\sqrt{5}}{2} - 1;$
$x_1^* = (-1,0), \quad x_2^* = \frac{1}{\sqrt{5}}(1,2),$	$\lambda_1^* = \frac{9\sqrt{5}}{10} - 1, \quad \lambda_2^* = \frac{\sqrt{5}}{2} - 1.$

All these KKT points are NEs because the NEP is convex. Furthermore, because for each $i = 1, 2$, the LICQC of $\mathbf{F}_i(x_{-i})$ holds for all $x \in X$, these NEs are all solutions to the NEP. This is very different from a single convex optimization problem, where the set of minimizers, if it is nonempty, must be a singleton or have an infinite cardinality if the objective function is convex, and the minimizer has to be unique if the objective function is further assumed to be strictly convex.

However, the KKT point may not be an NE of the NEP when there is no convexity assumed. This is because the KKT Condition (2.5) is typically not sufficient for x_i to be a minimizer of $\mathbf{F}_i(x_{-i})$, which makes nonconvex NEPs quite difficult to solve. In this paper, we mainly focus on finding NEs for nonconvex NEPs of polynomials.

3. Polynomial Optimization Formulations

In this section, we show how to formulate efficient polynomial optimization problems for solving the NEPP (1.3). We first introduce the polynomial expressions for Lagrange multipliers in Section 3.1. Then, in Section 3.2, polynomial optimization problems are formulated for finding NEs, and an algorithm to solve nonconvex NEPs is proposed. Convex NEPs of polynomials are studied in Section 3.3. Last, we further extend our approach to find more NEs in Section 3.4.

3.1. Optimality Conditions and Lagrange Multiplier Expressions

For the NEP (1.3), if x is an NE where the LICQC is satisfied, then it must be a KKT point—that is, x satisfies (2.5) for all $i \in [N]$. Therefore, every NE must satisfy the following equation system:

$$\underbrace{\begin{bmatrix} \nabla_{x_i} g_{i,1}(x_i) & \nabla_{x_i} g_{i,2}(x_i) & \cdots & \nabla_{x_i} g_{i,m_i}(x_i) \\ g_{i,1}(x_i) & 0 & \cdots & 0 \\ 0 & g_{i,2}(x_i) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & g_{i,m_i}(x_i) \end{bmatrix}}_{G_i(x_i)} \underbrace{\begin{bmatrix} \lambda_{i,1} \\ \lambda_{i,2} \\ \vdots \\ \lambda_{i,m_i} \end{bmatrix}}_{\lambda_i} = \underbrace{\begin{bmatrix} \nabla_{x_i} f_i(x) \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\hat{f}_i(x)}. \tag{3.1}$$

If there exists a matrix polynomial $H_i(x_i)$ such that

$$H_i(x_i) G_i(x_i) = I_{m_i}, \tag{3.2}$$

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 (3.2) 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 $x_i \in \mathbb{C}^{n_i}$ (Nie [42]). It is a generic condition (Nie and Ranestad [44, proposition 2.1]). We remark that if g_i is nonsingular, then the LICQC holds at every minimizer of (1.1), so there must exist $\lambda_{i,j}$ satisfying (2.5), and we can express $\lambda_{i,j}$ as

$$\lambda_{i,j} = \lambda_{i,j}(x) := (H_i(x_i) \hat{f}_i(x))_j, \tag{3.3}$$

for all NEs. For example, we consider the following two cases:

- 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

$$g_{i,0} = 1 - \sum_{j=1}^{n_i} x_{i,j}, \; g_{i,1} = x_{i,1}, \; \ldots, \; g_{i,n_i} = x_{i,n_i}.$$

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

If we let

$$H_i(x_i) = \begin{bmatrix} 1-x_{i,1} & -x_{i,2} & \dots & -x_{i,n_i} & 1 & \dots & 1 \\ -x_{i,1} & 1-x_{i,2} & \dots & -x_{i,n_i} & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ -x_{i,1} & -x_{i,2} & \dots & 1-x_{i,n_i} & 1 & \dots & 1 \\ -x_{i,1} & -x_{i,2} & \dots & -x_{i,n_i} & 1 & \dots & 1 \end{bmatrix},$$

then one may check that the (3.2) holds. The Lagrange multipliers $\lambda_{i,j}$ can be accordingly represented as

$$\lambda_{i,0} = -x_i^T \nabla_{x_i} f_i, \quad \lambda_{i,j} = \frac{\partial f_i}{\partial x_{i,j}} - x_i^T \nabla_{x_i} f_i, \quad j = 1, \dots, n_i. \tag{3.4}$$

• For the sphere constraint $1 - x_i^T x_i = 0$ or the ball constraint $1 - x_i^T x_i \geq 0$, the constraining polynomial is $g_{i,1} = 1 - x_i^T x_i$. If we let

$$H_i(x_i) = \begin{bmatrix} -\frac{1}{2}x_{i,1} & -\frac{1}{2}x_{i,2} & \dots & -\frac{1}{2}x_{i,n_i} & 1 \end{bmatrix},$$

then one may check that the (3.2) holds. The Lagrange multiplier can be accordingly expressed as

$$\lambda_{i,1} = -\frac{1}{2} x_i^T \nabla_{x_i} f_i. \tag{3.5}$$

For general nonsingular constraining tuple, one may find $H_i(x_i)$ satisfying (3.2) by solving linear equations. We refer to Nie [42] for more details on getting the polynomial expressions of Lagrange multipliers.

Throughout the paper, we assume that every constraining polynomial tuple g_i is nonsingular. This is a generic assumption. So, all $\lambda_{i,j}$ can be expressed as polynomials, as in (3.3). Then, each Nash equilibrium satisfies the following polynomial system:

$$\begin{cases} \nabla_{x_i} f_i(x) - \sum_{j=1}^{m_i} \lambda_{i,j}(x) \nabla_{x_i} g_{i,j}(x_i) = 0 \, (i \in [N]), \\ g_{i,j}(x_i) = 0 \, (i \in [N], j \in \mathcal{E}_i), \ \lambda_{i,j}(x) g_{i,j}(x_i) = 0 \ (i \in [N], j \in \mathcal{I}_i), \\ g_{i,j}(x_i) \geq 0 \, (j \in \mathcal{I}_i), \ \lambda_{i,j}(x) \geq 0 \ (i \in [N], j \in \mathcal{I}_i). \end{cases} \tag{3.6}$$

3.2. An Algorithm for Finding an NE

For the NEP of Polynomials (1.3), let $\lambda_{i,j}(x)$ be polynomial Lagrange multiplier expressions as in (3.3) for each $i \in [N]$ and $j \in [m_i]$. Then, every NE must satisfy the polynomial system (3.6). Choose a generic positive definite matrix

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

Then, all NEs are feasible points for the following optimization problem:

$$\begin{cases} \min_x & [x]_1^T \cdot \Theta \cdot [x]_1 \\ s.t. & \nabla_{x_i} f_i(x) - \sum_{j=1}^{m_i} \lambda_{i,j}(x) \nabla_{x_i} g_{i,j}(x_i) = 0 \, (i \in [N]), \\ & g_{i,j}(x_i) = 0 \, (j \in \mathcal{E}_i, i \in [N]), \\ & \lambda_{i,j}(x) g_{i,j}(x_i) = 0 \, (j \in \mathcal{I}_i, i \in [N]), \\ & g_{i,j}(x_i) \geq 0 \, (j \in \mathcal{I}_i, i \in [N]), \\ & \lambda_{i,j}(x) \geq 0 \, (j \in \mathcal{I}_i, i \in [N]). \end{cases} \tag{3.7}$$

In the above, the vector $[x]_1 := (1, x_1, x_2, \dots, x_n)^T \in \mathbb{R}^{n+1}$. Note that $x \in \mathbb{R}^n$ is a KKT point for the NEP if and only if it is feasible for (3.7). It is important to observe that if (3.7) is infeasible, then there are no NEs. If (3.7) is feasible, then it must have a minimizer, because its objective is a positive definite quadratic function. Moreover, for a

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

generic $\Theta \in \mathbb{R}^{(n+1)\times(n+1)}$, the minimizer of (3.7) is unique (see Theorem 4.2). Note that (3.7) is a polynomial optimization problem, which can be solved by the Moment-SOS semidefinite relaxations (see Section 4).

Assume that $u := (u_1, \ldots, u_N)$ is an optimizer of (3.7). Then, u is an NE if and only if each u_i is a minimizer of $\mathbf{F}_i(u_{-i})$. To this end, for each player, consider the optimization problem:

$$\begin{cases} \omega_i := \min & f_i(x_i, u_{-i}) - f_i(u_i, u_{-i}) \\ \quad\quad s.t. & g_{i,j}(x_i) = 0\,(j \in \mathcal{E}_i), \\ & g_{i,j}(x_i) \geq 0\,(j \in \mathcal{I}_i). \end{cases} \tag{3.8}$$

If all the optimal values $\omega_i \geq 0$, then u is a Nash equilibrium. If one of them is negative—say, $\omega_i < 0$—then u is not an NE. For such a case, let U_i be a set of some optimizers of (3.8); then, u violates the following inequalities

$$f_i(x_i, x_{-i}) \leq f_i(v, x_{-i}) \quad (v \in U_i). \tag{3.9}$$

However, every Nash equilibrium must satisfy (3.9).

When u is not an NE, we aim at finding a new candidate by posing the inequalities in (3.9). Therefore, we consider the following optimization problem:

$$\begin{cases} \min\limits_{x} & [x]_1^T \cdot \Theta \cdot [x]_1 \\ s.t. & \nabla_{x_i} f_i(x) - \sum\limits_{j=1}^{m_i} \lambda_{i,j}(x) \nabla_{x_i} g_{i,j}(x_i) = 0\,(i \in [N]), \\ & g_{i,j}(x_i) = 0\,(j \in \mathcal{E}_i, i \in [N]), \\ & \lambda_{i,j}(x) g_{i,j}(x_i) = 0\,(j \in \mathcal{I}_i, i \in [N]), \\ & g_{i,j}(x_i) \geq 0\,(j \in \mathcal{I}_i, i \in [N]), \\ & \lambda_{i,j}(x) \geq 0\,(j \in \mathcal{I}_i, i \in [N]), \\ & f_i(v, x_{-i}) - f_i(x_i, x_{-i}) \geq 0\,(v \in \mathcal{K}_i, i \in [N]). \end{cases} \tag{3.10}$$

In the above, each $\mathcal{K}_i$ is a set of some optimizers of (3.8). We solve (3.10) again for a minimizer—say, $\hat{u}$. If $\hat{u}$ is verified to be an NE, then we are done. If it is not, we can add more inequalities like (3.9) to exclude both u and $\hat{u}$. Repeating this procedure, we get the following algorithm for computing an NE.

Algorithm 3.1 *For the NEP given as in* (1.1) *and* (1.3), *do the following*

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

Step 1. *Solve the Polynomial Optimization Problem* (3.10). *If it is infeasible, then output that there is no NE and stop; otherwise, solve it for an optimizer* u.

Step 2. *For each* $i = 1, \ldots, N$, *solve the Optimization* (3.8). *If all* $\omega_i \geq 0$, *then output the NE* u *and stop. If one of* ω_i *is negative, then 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* (3.8); *then, update the set* $\mathcal{K}_i := \mathcal{K}_i \cup U_i$. *Let* $\ell := \ell + 1$, *then go to* Step 1.

In Step 0, we can set $\Theta = R^T R$ for a randomly generated matrix R of length $n+1$. The objective in (3.10) is a positive definite quadratic function, so it has a minimizer if (3.10) is feasible. The case is slightly different for (3.8). If the feasible set X_i is compact or $f_i(x_i, u_{-i})$ is coercive for the given u_{-i}, then (3.8) has a minimizer. If X_i is unbounded and $f_i(x_i, u_{-i})$ is not coercive, it may be difficult to compute the optimal value ω_i. In applications, we are mostly interested in cases that (3.8) has a minimizer, for the existence of an NE. We discuss how to solve the optimization problems in Algorithm 3.1 by the Moment-SOS hierarchy of semidefinite relaxations in Section 4.

The following is the convergence theorem for Algorithm 3.1.

Theorem 3.2. *Assume each constraining polynomial tuple* g_i *is nonsingular and let* $\lambda_{i,j}(x)$ *be polynomial expressions of Lagrange multipliers as in* (3.3). *Let* $\mathcal{G}$ *be the feasible set of* (3.7) *and* $\mathcal{G}^*$ *be the set of all NEs. If the complement* $\mathcal{G} \backslash \mathcal{G}^*$ *is a finite set—that is, the cardinality* $\ell^* := |\mathcal{G} \backslash \mathcal{G}^*| < \infty$—*then Algorithm* 3.1 *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 (3.3). For each u that is a feasible point of (3.7), every NE must satisfy the constraint

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

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

Therefore, every NE must also be a feasible point of (3.10). Because the matrix Θ is positive definite, the optimization (3.10) must have a minimizer, unless it is infeasible. When Algorithm 3.1 goes to a newer loop—say, from the ℓth to the $(\ell+1)$th—the optimizer u for (3.10) in the ℓth loop is no longer feasible for (3.10) in the $(\ell+1)$th loop. This means that the feasible set of (3.10) must lose at least one point after each loop, unless an NE is met. Also, note that the feasible set of (3.10) is contained in $\mathcal{G}$. If $\mathcal{G}\backslash\mathcal{G}^*$ is a finite set, Algorithm 3.1 must terminate after some loops. The number of loops is at most ℓ^*. □

As shown in the appendix, when the NEP is given by generic polynomials, the NEP has finitely many KKT points (see Theorem A.1). For such cases, $|\mathcal{G}\backslash\mathcal{G}^*| \le |\mathcal{G}| < \infty$, and finite termination of Algorithm 3.1 is guaranteed.

Theorem 3.3. *Let $d_{i,j} > 0$, $a_{i,j} > 0$ be degrees, for all $i \in [N], j \in [m_i]$. 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 whose degree in x_j is $a_{i,j}$, then Algorithm 3.1 terminates within finitely many loops—that is, it either finds an NE if any exist or detects the nonexistence of NEs.*

Proof. The conclusion follows directly from Theorems 3.2 and A.1. □

When there exist infinitely many KKT points that are not NEs, Algorithm 3.1 can still be applied to compute an NE if one exists or to detect the nonexistence of NEs if they do not exist. See Example 5.2(ii) for such a case. However, for such NEPPs, the convergence property of Algorithm 3.1 is not fully understood.

3.3. Convex NEPs

The NEP is said to be *convex* if for every $i \in [N]$, the $f_i(x_i, x_{-i})$ is convex in x_i for all $x_{-i} \in X_{-i} := \prod_{j\in[N]\backslash\{i\}} X_j$, the $g_{i,j}(x_i)$ is linear for each $j \in \mathcal{E}_i$, and is concave for every $j \in \mathcal{I}_i$. For convex NEPs, every KKT point must be an NE because the KKT conditions are sufficient for global optimality.

Moreover, for convex NEPPs, when every constraining tuple g_i is nonsingular, the LICQC holds for all $x \in X$, and a point is an NE if and only if it satisfies the KKT conditions. Note that the Lagrange multipliers can be expressed by polynomials as in (3.3) when nonsingularity is assumed. For such cases, the solution set for (2.5) is exactly the set of NEs. Therefore, if we solve the Polynomial Optimization Problem (3.10) with $\mathcal{K}_i = \emptyset$ for all $i \in [N]$ (i.e., the Polynomial Optimization (3.7)), then every minimizer, if the feasible set is nonempty, must be an NE. On the other hand, if (3.10) is infeasible, then we immediately know the NEs do not exist. This shows that, for convex NEPPs, Algorithm 3.1 must terminate at the initial loop.

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 all $x_{-i} \in X_{-i}$. Then, Algorithm 3.1 must terminate at the first loop with $\ell = 0$, returning an NE or reporting that there is no NE.*

Example 3.5. Consider the convex NEP in Example 1.1. In this NEP, both players have ball constraints, so their Lagrange multipliers can be expressed by polynomials as in (3.5). We ran Algorithm 3.1 for solving this NEP[2] and found the NE $x^* = (x_1^*, x_2^*)$ with

$$x_1^* = (-1.0000, 0.0000), \quad x_2^* = (0.4472, 0.8944),$$

in the initial loop. It took around 0.88 second.

3.4. More Nash Equilibria

Algorithm 3.1 aims at finding a single NE. In some applications, people may be interested in more NEs. Moreover, when there is a unique NE, people are also interested in a certificate for uniqueness.

In this subsection, we study how to find more NEs or check the completeness of solution sets. Assume that x^* is a Nash equilibrium produced by Algorithm 3.1—that is, x^* is also a minimizer of (3.10). Then, all KKT points x satisfying $[x]_1^T\Theta[x]_1 < [x^*]_1^T\Theta[x^*]_1$ are excluded from the feasible set of (3.10) by the constraints

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

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

$$[x]_1^T\Theta[x]_1 \ge [x^*]_1^T\Theta[x^*]_1 + \delta, \tag{3.11}$$

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

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

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

When an optimizer of (3.12) is computed, we can check whether it is an NE or not by solving (3.8) for all $i \in [N]$. If it is, we get a new NE that is different from x^*. If it is not, we update the set $\mathcal{K}_i$ as in Step 3 of Algorithm 3.1. 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 (3.12). We want a value $\delta > 0$ such that (3.11) holds for all unknown NEs. To this end, we consider the following maximization problem:

$$\begin{cases} \max\limits_{x} \ [x]_1^T\Theta[x]_1 \\ s.t. \ \ \nabla_{x_i} f_i(x) - \sum\limits_{j=1}^{m_i} \lambda_{i,j}(x)\nabla_{x_i} g_{i,j}(x_i) = 0\,(i \in [N]), \\ \qquad g_{i,j}(x_i) = 0\,(j \in \mathcal{E}_i, i \in [N]), \\ \qquad \lambda_{i,j}(x) g_{i,j}(x_i) = 0\,(j \in \mathcal{I}_i, i \in [N]), \\ \qquad g_{i,j}(x_i) \geq 0\,(j \in \mathcal{I}_i, i \in [N]), \\ \qquad \lambda_{i,j}(x) \geq 0\,(j \in \mathcal{I}_i, i \in [N]), \\ \qquad f_i(v, x_{-i}) - f_i(x_i, x_{-i}) \geq 0\,(v \in \mathcal{K}_i, i \in [N]), \\ \qquad [x]_1^T\Theta[x]_1 \ \leq \ [x^*]_1^T\Theta[x^*]_1 + \delta. \end{cases} \tag{3.13}$$

Interestingly, if x^* is also a maximizer of (3.13)—that is, the maximum of (3.13) equals $[x^*]^T\Theta[x^*]_1$—then the feasible set of (3.12) 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* (3.10).

i. *If x^* is also a maximizer of* (3.13), *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* (3.13).

Proof. Note that every NE is a feasible point of (3.10).

i. If x^* is also a maximizer of (3.13), then the objective $[x]_1^T\Theta[x]_1$ achieves a constant value in the feasible set of (3.13). 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 (3.10). When Θ is a generic positive definite matrix, the Optimization (3.10) has a unique optimizer, so $u = x^*$.

ii. Because Θ is positive definite, there exists $\epsilon > 0$ such that

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

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

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

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

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 $y = [x]_2$. Because 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. Because x^* is a minimizer of (3.10), the hyperplane $H := \{\theta^T y = \theta^T y^*\}$ is a supporting hyperplane for the set T. Because Θ is generic, the Optimization (3.10) 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 (3.13). Hence, x^* is also a maximizer of (3.13). □

Proposition 3.6 shows the existence of $\delta > 0$ such that (3.10) and (3.13) 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 (3.13) will eventually have x^* as a maximizer. The following is the algorithm for finding an NE that is different from x^*.

Algorithm 3.7 *For the given NEP* (1.3) *and a computed NE x^*, let Θ be the positive definite matrix for computing x^*.*

Step 0. *Give an initial value for δ (say, 0.1).*

Step 1. *Solve the Maximization Problem* (3.13). *If its optimal value η equals $v := [x^*]_1^T \Theta [x^*]_1$, then go to* Step 2. *If η is bigger than v, then let $\delta := \delta/5$ and repeat this step.*

Step 2. *Solve the Optimization Problem* (3.12). *If it is infeasible, then output there are no additional NEs and stop; otherwise, solve* (3.12) *for a minimizer u.*

Step 3. *For each $i = 1, \ldots, N$, solve the Optimization* (3.8) *for the optimal value ω_i. If all $\omega_i \geq 0$, stop and output the new NE u. If one of ω_i is negative, then 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 Step 1. For such a case, more investigation is required to verify the completeness of the solution set or to find other NEs. However, for generic NEPs, there are finitely many KKT points (see Theorem A.1 in the appendix). The following is the convergence result for Algorithm 3.7.

Theorem 3.8. *Under the same assumptions in Theorem 3.2, 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, either returning an 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 Step 1 of Algorithm 3.7. Again, by Proposition 3.6(i), the feasible set of (3.12) contains all NEs except x^*. The finite termination of Algorithm 3.7 can be proved in the same way as for Theorem 3.2. □

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

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, then we enumerate them as

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

Without loss of generality, we assume

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

because Θ is generic. If the first r NEs—say, $x^{(1)}, \ldots, x^{(r)}$—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 $j = r+1, \ldots, s$. Therefore, if we apply Algorithm 3.7 with $x^* = x^{(r)}$, the next Nash equilibrium $x^{(r+1)}$ can be obtained, if it exists. So, 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.*

Remark 3.10. Under the assumption of Theorem 3.3, the NEP has finitely many KKT points. For such cases, Algorithm 3.7 can find all NEs and certify the completeness of solutions set within finitely many steps, by Corollary 3.9.

4. Solve Polynomial Optimization Problems

In this section, we discuss how to solve occurring polynomial optimization problems in Algorithms 3.1 and 3.7. For the NEP, 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 (3.3) for all Nash equilibria. We apply the Moment-SOS hierarchy of semidefinite relaxations (Henrion and Lasserre [17], Lasserre [26], Lasserre [28], Laurent [30]) for solving these polynomial optimization problems. New convergence results for solving these polynomial optimization problems are given due to the usage of polynomial expressions for Lagrange multipliers.

For the variable z, such that $z = x$ or $z = x_i$ for some $i \in [N]$, denote by l the dimension of z. Consider the polynomial optimization problem in the variable z:

$$\begin{cases} \vartheta^* := \min\limits_{z \in \mathbb{R}^l} \ \theta(z) \\ \qquad s.t. \ \ p(z) = 0 \, (\forall p \in \Phi), \\ \qquad\quad q(z) \geq 0 \, (\forall q \in \Psi). \end{cases} \tag{4.1}$$

In the above, Φ and Ψ are sets of equality- and inequality-constraining polynomials, respectively. Denote the degree

$$d_0 := \max\{\lceil \deg(p)/2 \rceil : p \in \{\theta\} \cup \Phi \cup \Psi\}. \tag{4.2}$$

For a degree $k \geq d_0$, recall that the set $\mathrm{Ideal}[\Phi]_{2k} + \mathrm{Qmod}[\Psi]_{2k}$ is introduced in Section 2.2. The kth-order SOS relaxation for (4.1) is

$$\begin{cases} \vartheta_{sos}^{(k)} := \ \max \ \ \gamma \\ \qquad\quad s.t. \ \ \theta - \gamma \in \mathrm{Ideal}[\Phi]_{2k} + \mathrm{Qmod}[\Psi]_{2k}. \end{cases} \tag{4.3}$$

The dual problem of (4.3) is the kth-order moment relaxation

$$\begin{cases} \vartheta_{mom}^{(k)} := \min\limits_{y} \ \langle \theta, y \rangle \\ \qquad\quad s.t. \ \ y_0 = 1, \, L_p^{(k)}[y] = 0 \ (p \in \Phi), \\ \qquad\qquad M_k[y] \succeq 0, \, L_q^{(k)}[y] \succeq 0 \ (q \in \Psi), \\ \qquad\qquad y \in \mathbb{R}^{\mathbb{N}_{2k}^l}, \end{cases} \tag{4.4}$$

where the moment matrix $M_k[y]$ and localizing matrices $L_p^{(k)}[y]$, $L_q^{(k)}[y]$ are given by (2.3) and (2.4). Both (4.3) and (4.4) are semidefinite programs, and the primal-dual pair is called the Moment-SOS relaxations for the Polynomial Optimization Problem (4.1). If $z \in \mathbb{R}^l$ is a feasible point of (4.1), then $[z]_k \in \mathbb{R}_{2k}^l$ must be a feasible point of (4.4). Thus, (4.1) has an empty feasible set if (4.4) is infeasible. When (4.4) has a nonempty feasible set, it is clear

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

that $\vartheta_{mom}^{(k)} \le \vartheta_{sos}^{(k)} \le \vartheta^*$ for all k, and both $\vartheta_{mom}^{(k)}$ and $\vartheta_{sos}^{(k)}$ are monotonically increasing. The following is the Moment-SOS algorithm for solving (4.1).

Algorithm 4.1 *For the Polynomial Optimization Problem* (4.1), *let d_0 be the degree given by* (4.2).

Step 0. *Initialize $k := d_0$.*

Step 1. *Solve the Moment Relaxation* (4.4). *If it is infeasible, then the Polynomial Optimization Problem* (4.1) *is infeasible and stop; otherwise, solve* (4.4) *for the minimum value $\vartheta_{mom}^{(k)}$ and a minimizer $y^{(k)}$.*

Step 2. *Let $t := d_0$. If y^* satisfies the rank condition*

$$\operatorname{rank} M_t[y^*] = \operatorname{rank} M_{t-d_0}[y^*], \tag{4.5}$$

then extract a set U_i of $r := \operatorname{rank} M_t[y^*]$ minimizers for (4.1) *and stop.*

Step 3. *If* (4.5) *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.

Algorithm 4.1 is known as the Moment-SOS hierarchy of semidefinite relaxations (Lasserre [26]). We say the Moment-SOS hierarchy has asymptotic convergence if $\vartheta_{sos}^{(k)} \to \vartheta^*$ as $k \to \infty$, and we say it has finite convergence if $\vartheta_{sos}^{(k)} = \vartheta^*$ for all k that is large enough. For a general polynomial optimization problem, if $\operatorname{Ideal}[\Phi] + \operatorname{Qmod}[\Psi]$ is archimedean, then $\vartheta_{mom}^{(k)} \to \vartheta^*$ as $k \to \infty$ (Lasserre [26]). In Step 2, the Rank Condition (4.5) is called *flat truncation* (Nie [37]). It is a sufficient (and almost necessary) condition to check the finite convergence of moment relaxations. When (4.5) holds, the method in Henrion and Lasserre [17] can be used to extract r minimizers for (4.1). This method and Algorithm 4.1 are implemented in the software `GloptiPoly 3` (Henrion et al. [19]). In the following subsections, we study the convergence result of Algorithm 4.1 when it is applied for solving (3.8), (3.10), (3.12), and (3.13).

4.1. The Optimization for All Players

We discuss the convergence of Algorithm 4.1 for solving (3.10), (3.12), and (3.13).

First, we consider (3.10). Let

$$z := x, \quad \theta(x) := [x]_1^T \Theta [x]_1, \tag{4.6}$$

and we denote the polynomial tuples

$$\Phi_i := \left\{ \nabla_{x_i} f_i(x) - \sum_{j=1}^{m_i} \lambda_{i,j}(x) \nabla_{x_i} g_{i,j}(x_i) \right\} \cup \{g_{i,j}(x_i) : j \in \mathcal{E}_i\} \cup \{\lambda_{i,j}(x) \cdot g_{i,j}(x_i) : j \in \mathcal{I}_i\}, \tag{4.7}$$

$$\Psi_i := \{g_{i,j}(x_i) : j \in \mathcal{I}_i\} \cup \{\lambda_{i,j}(x) : j \in \mathcal{I}_i\} \cup \{f_i(v, x_{-i}) - f_i(x_i, x_{-i}) : v \in \mathcal{K}_i\}. \tag{4.8}$$

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

$$\Phi := \bigcup_{i=1}^N \Phi_i, \quad \Psi := \bigcup_{i=1}^N \Psi_i. \tag{4.9}$$

They are both finite sets of polynomials. Then, the Optimization (3.10) can be written as (4.1), and we may apply Algorithm 4.1 for solving it. Recall that e_i is the vector in $\mathbb{R}^n$ such that its ith entry is one and all other entries are zero. For a tms $y \in \mathbb{R}^{\mathbb{N}_{2k}^n}$, the y_{e_i} means the entry of y labelled by e_i. For example, when $n = 4$, then $y_{e_2} = y_{0100}$. Let $y^{(k)}$ be a minimizer of the kth-order Moment Relaxation (4.4) for (3.10), and denote

$$u^{(k)} := (y_{e_1}^{(k)}, y_{e_2}^{(k)}, \ldots, y_{e_n}^{(k)}). \tag{4.10}$$

Then, $u^{(k)}$ is a minimizer of (3.10) if $u^{(k)}$ is feasible for (3.10) and $\langle \theta, y^{(k)} \rangle = \theta(u^{(k)})$. Moreover, we have the following convergence result for solving (3.10):

Theorem 4.2. *For the polynomial optimization Problem* (3.10), *assume Θ is a generic positive definite matrix. Let $z := x$, and let θ, Ψ, Φ be given as in* (4.6)–(4.9). *Suppose* $\operatorname{Ideal}[\Phi] + \operatorname{Qmod}[\Psi]$ *is archimedean.*

i. *If the Optimization* (3.10) *is infeasible, then the moment relaxation* (4.4) *must be infeasible when the order k is big enough.*

ii. *Suppose the Optimization* (3.10) *is feasible. Let $u^{(k)}$ be given as in* (4.10). *Then, $u^{(k)}$ converges to the unique minimizer of* (3.10). *In particular, if the real zero set of Φ is finite, then $u^{(k)}$ is the unique minimizer of* (3.10), *and* (4.5) *holds at $y^{(k)}$ with the rank equal to* 1 *when k is sufficiently large.*

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

Proof.

i. If (3.10) is infeasible, the constant polynomial -1 can be viewed as a positive polynomial on the feasible set of (3.10). Because $\mathrm{Ideal}[\Phi]+\mathrm{Qmod}[\Psi]$ is archimedean, we have $-1\in\mathrm{Ideal}[\Phi]_{2k}+\mathrm{Qmod}[\Psi]_{2k}$, for k big enough, by Putinar's [50] Positivstellensatz. For such a big k, the SOS Relaxation (4.3) is unbounded from above; hence, the Moment Relaxation (4.4) must be infeasible.

ii. When the Optimization (3.10) is feasible, it must have minimizers. Let K be the feasible set of (3.10), and

$$\mathcal{R}_2(K) := cone(\{[u]_2 : u\in K\}).$$

In the above, the *cone* means the conic hull. Consider the moment optimization problem

$$\begin{cases}\min\limits_{w} & \langle\theta, w\rangle\\ s.t. & w_0=1,\ w\in\mathcal{R}_2(K).\end{cases}\tag{4.11}$$

If the matrix Θ is a generic positive definite matrix, then the function θ is generic in $\Sigma_{n,2}$. By Nie [39, proposition 5.2], the Moment Optimization Problem (4.11) has a unique minimizer. When (4.11) has minimizers, its minimum value equals ϑ^*. Suppose (3.10) has two distinct minimizers—say, $x^{(1)}$ and $x^{(2)}$. Then, $[x^{(1)}]_2$ and $[x^{(2)}]_2$ are two distinct minimizers of (4.11), a contradiction to the uniqueness of the minimizer for (4.11). Therefore, (3.10) must have a unique minimizer x^* when Θ is generic.

The convergence of $u^{(k)}$ to x^* is shown in Schweighofer [54] or Nie [37, theorem 3.3]. For the special case that $\Phi(x)=0$ has finitely many real solutions, the point $u^{(k)}$ must equal x^*, when k is large enough. This is shown in Lasserre et al. [29] (also see Nie [38]). □

The archimedeanness of $\mathrm{Ideal}[\Phi]+\mathrm{Qmod}[\Psi]$ is essentially requiring that the feasible set of (3.10) is compact. If the real zero set of Φ is compact, then $\mathrm{Ideal}[\Phi]+\mathrm{Qmod}[\Psi]$ must be archimedean. In particular, if the NEPP has finitely many real KKT points, then $\mathrm{Ideal}[\Phi]+\mathrm{Qmod}[\Psi]$ is archimedean. Interestingly, when the objective and constraining polynomials are generic, there are finitely many KKT points. See Theorem A.1 in the appendix. In fact, as shown in the proof of Theorem A.1, the zero set of Φ is finite for generic NEPPs, and, hence, Algorithm 4.1 has finite convergence. Moreover, by Theorem 4.2, when Θ is generic and the minimizer $y^{(k)}$ for (4.4) is obtained, one may let $u^{(k)}$ be given as in (4.10) and directly check whether $u^{(k)}$ is the unique minimizer or not, instead of checking the Flat Truncation (4.5).

The other Minimization Problem (3.12) can be solved in the same way by Algorithm 4.1. The convergence property is the same. For the cleanness of the paper, we omit the details.

For the Maximization (3.13), we let $z := x$ and

$$\theta(x) := -[x]_1^T\Theta[x]_1.\tag{4.12}$$

Recall that the polynomial tuples Φ_i and Ψ_i are given by (4.7) and (4.8). Denote the set of polynomials

$$\Phi := \bigcup_{i=1}^N \Phi_i,\quad \Psi := \bigcup_{i=1}^N \Psi_i \cup \{[x^*]_1^T\Theta[x^*]_1+\delta-[x]_1^T\Theta[x]_1\}.\tag{4.13}$$

Then, (3.13) can be equivalently written as (4.1). Similarly, Algorithm 4.1 can be used to solve (3.13). The Optimization (3.13) is always feasible because x^* is a feasible point. Therefore, the Moment Relaxation (4.4) is also feasible, and there is no need to check its feasibility in Step 1 of Algorithm 4.1. Because the minimum value $\vartheta_{mom}^{(k)}$ is a lower bound of ϑ^*, if $\vartheta_{mom}^{(k)}\ge -[x^*]_1^T\Theta[x^*]_1$, then

$$\vartheta_{mom}^{(k)} = \vartheta^* = -[x^*]_1^T\Theta[x^*]_1,$$

and x^* is a maximizer of (3.13). When $\vartheta_k < -[x^*]_1^T\Theta[x^*]_1$, the Flat Truncation Condition (4.5) can be applied for checking the finite convergence of the Moment-SOS hierarchy. Under some classical optimality conditions, we have $\vartheta_{mom}^{(k)}=\vartheta^*$ when k is large enough (Nie [40]). Moreover, if the real zero set of Φ is finite, then the Moment-SOS hierarchy has finite convergence, and (4.5) holds (Nie [38]). We would like to remark that when the NEP is given by generic polynomials, the complex zero set of Φ is finite (see Theorem A.1); thus, Algorithm 4.1 has finite convergence.

4.2. Checking Nash Equilibria

Suppose u is a minimizer of (3.10). To check whether $u=(u_i,u_{-i})$ is an NE or not, we need to solve the Individual Optimization (3.8) for all $i\in[N]$.

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

For the given $u \in \mathbb{R}^n$ and $i \in [N]$, (3.8) is a polynomial optimization problem in the variable x_i. If (3.8) is unbounded from below, then u cannot be an NE, and the point v for precluding u can be obtained by adding a suitable extra ball constraint. In the following, we suppose that the minimum of (3.8) is attainable. Because we assume that the polynomial tuple $g_i(x_i)$ is nonsingular, polynomial expressions for Lagrange multiplier expressions exist and can be applied to solve (3.8). Let $\lambda_i(x)$ be the Lagrange multiplier expressions in (3.3). Note that the nonsingularity of g_i implies that the LICQC holds at every $x_i \in X_i$. Every minimizer of (3.8) must be a KKT point of (3.8). Therefore, (3.8) is equivalent to the following polynomial optimization problem:

$$\begin{cases} \omega_i := \min\limits_{x_i \in \mathbb{R}^{n_i}} & f_i(x_i, u_{-i}) - f_i(u_i, u_{-i}) \\ \quad s.t. & \nabla_{x_i} f_i(x_i, u_{-i}) - \sum\limits_{j=1}^{m_i} \lambda_{i,j}(x_i, u_{-i}) \nabla_{x_i} g_{i,j}(x_i) = 0, \\ & g_{i,j}(x_i, u_{-i}) = 0 \, (j \in \mathcal{E}_i), \\ & g_{i,j}(x_i, u_{-i}) \lambda_{i,j}(x_i, u_{-i}) = 0 \, (j \in \mathcal{I}_i), \\ & g_{i,j}(x_i, u_{-i}) \geq 0 \, (j \in \mathcal{I}_i), \\ & \lambda_{i,j}(x_i, u_{-i}) \geq 0 \, (j \in \mathcal{I}_i). \end{cases} \tag{4.14}$$

We introduce the convergence result of Algorithm 4.1 for solving (4.14). Let

$$z := x_i, \quad \theta(x_i) := f_i(x_i, u_{-i}) - f_i(u_i, u_{-i}), \tag{4.15}$$

$$\Phi := \{g_{i,j}(x_i) : j \in \mathcal{E}_i\} \cup \{\lambda_{i,j}(x_i, u_{-i}) \cdot g_{i,j}(x_i) : j \in \mathcal{I}_i\} \cup \left\{ \nabla_{x_i} f_i(x_i, u_{-i}) - \sum_{j=1}^{m_i} \lambda_{i,j}(x_i, u_{-i}) \nabla_{x_i} g_{i,j}(x_i) \right\}, \tag{4.16}$$

$$\Psi := \{g_{i,j}(x_i) : j \in \mathcal{I}_i\} \cup \{\lambda_{i,j}(x_i, u_{-i}) : j \in \mathcal{I}_i\}. \tag{4.17}$$

Like earlier cases, the set $\{p\}$ stands for $\{p_1, \ldots, p_s\}$, when $p = (p_1, \ldots, p_s)$ is a vector of polynomials. Then, the (4.14) can be rewritten as (4.1), and the Moment-SOS relaxations of (4.14) are given by (4.3) and (4.4). We would like to remark that the Optimization (4.14) is always feasible because u_i is in its feasible set. Thus, the Moment Relaxation (4.4) for (4.14) is also feasible, and there is no need to check the feasibility for (4.4) in the first step of Algorithm 4.1. Moreover, the minimum $\vartheta_{mom}^{(k)}$ of (4.4) is a lower bound for ω_i, and $\omega_i \leq 0$. If $\vartheta_{mom}^{(k)} \geq 0$ for some $k \geq d_0$, then ω_i must be 0, and we can stop Algorithm 4.1 immediately because this implies that u_i is the minimizer for $\mathbf{F}_i(u_{-i})$. If $\vartheta_{mom}^{(k)} < 0$, we need to apply the Flat Truncation (4.5) to certify whether the finite convergence for the Moment-SOS hierarchy is achieved or not. The following theorem concerns the finite convergence of Algorithm 4.1 for solving (4.14). Its proof follows from Nie et al. [48, theorem 4.4].

Theorem 4.3. *Assume the ith player's constraining polynomial tuple g_i is nonsingular and its Optimization* (3.8) *has a minimizer for the given u_{-i}. Let $z := x_i$, and let θ, Ψ, Φ be given as in* (4.15)–(4.17). *Assume either one of the following conditions holds*:

i. *The set* $\mathrm{Ideal}[\Phi] + \mathrm{Qmod}[\Psi]$ *is archimedean*,

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

If each minimizer of (4.14) *is an isolated critical point, then all minimizers of* (4.4) *must satisfy the flat truncation* (4.5), *for all k big enough. Therefore, Algorithm* 4.1 *must terminate within finitely many loops.*

We remark that if $\mathrm{Ideal}[g_{i,j} : j \in \mathcal{E}_i] + \mathrm{Qmod}[g_{i,j} : j \in \mathcal{I}_i]$ is archimedean, then $\mathrm{Ideal}[\Phi] + \mathrm{Qmod}[\Psi]$ is also archimedean. Therefore, if the archimedeanness holds for the ith player's optimization (1.1), then condition (i) in Theorem 4.3 is satisfied.

5. Numerical Experiments

This section reports numerical experiments for solving NEPs by Algorithms 3.1 and 3.7. For all polynomial optimization problems appearing in the algorithms, we apply the software `GloptiPoly 3` (Henrion et al. [19]) to formulate Moment-SOS semidefinite relaxations and use `SeDuMi` (Sturm [57]) for solving these semidefinite programs. The computation is implemented in an Alienware Aurora R8 desktop, with an Intel® Core(TM) i7-9700 CPU at 3.00 GHz×8 and 16 GB of RAM, in a Windows 10 operating system.

For ball and simplex constraints, the Lagrange multiplier expressions are given by (3.4) and (3.5), respectively. Polynomial expressions of Lagrange multipliers for other types of constraints are given in the descriptions of each example. In Step 2 of Algorithm 3.1 and Step 3 of Algorithm 3.7, if the optimal value $\omega_i \geq 0$ for all players,

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

then the point u is an NE. In numerical computation, we cannot have $\omega_i \geq 0$ 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, $\omega^* \geq -10^{-6}$—then we regard the computed solution as an NE.

Example 5.1. For the convex NEP in Example 1.1, Algorithm 3.1 found the NE

$$x_1^* = (-1.0000, 0.0000), \quad x_2^* = (0.4472, 0.8944),$$

in the first loop, as shown in Example 3.5. The accuracy parameter is $\omega^* = -7.9793 \cdot 10^{-9}$. Then, we ran Algorithm 3.7 and found two more NEs, which are

$$\begin{aligned} &x_1^* = (-0.0000, 0.0000), x_2^* = (0.0000, 0.0000), && \omega^* = -1.4147 \cdot 10^{-10}; \\ &x_1^* = (1.0000, -0.0000), x_2^* = (-0.4472, -0.8944), && \omega^* = -1.7829 \cdot 10^{-8}. \end{aligned}$$

Moreover, Algorithm 3.7 certified that these three NEs are all solutions to this NEP. It took around 1.40 seconds to find these two additional NEs and certify the completeness of the solution set.

In the following example, we show that our algorithm can find NEs for NEPs that have infinitely many KKT points.

Example 5.2.

i. Consider the convex NEP

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

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

then one may check that for each $\alpha \in [0, 1/2]$, $x_1 = (2\alpha, 1 - 2\alpha)$, $x_2 = (\alpha, 1 - \alpha)$ is an NE. Applying Algorithm 3.1, we got the NE:

$$x_1^* = (0.9247, 0.0753), x_2^* = (0.4624, 0.5376), \ \omega^* = -2.1940 \cdot 10^{-8}.$$

The computation took about 0.19 second.

ii. Consider the NEP

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

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

One may check that for each $\alpha \in \left[-\frac{1}{4}, 0\right)$, the $x_1 = (\alpha, 0)$, $x_2 = \left(-\frac{1}{4\alpha}, \frac{1}{2}\right)$ is a KKT point that is not an NE. Applying Algorithm 3.1, we got the NE

$$x_1^* = (1.0000, -0.0000), x_2^* = (0.4259, 0.5000), \ \omega^* = -6.2187 \cdot 10^{-9}.$$

The computation took about 0.33 second.

Example 5.3. In this example, we consider NEPs with box constraints such that every $x_i \in \mathbb{R}^1$. For each $i \in [N]$, the ith player's feasible set is given by

$$1 + x_i \geq 0, \quad 1 - x_i \geq 0.$$

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

Then, the associated Lagrange multipliers can be expressed as

$$\lambda_{i,1} = \frac{1}{2}\frac{\partial f_i}{\partial x_i}\cdot(1-x_i), \quad \lambda_{i,2} = \lambda_{i,1} - \frac{\partial f_i}{\partial x_i}.$$

i. Consider the two-player zero-sum game with box constraints in Parrilo [49, example 3.1] (see also Kroupa and Votroubek [23, example 1]), where the objective functions are

$$f_1(x_1,x_2) = (x_1)^2 - 2x_1(x_2)^2 + x_2, \quad f_2(x_1,x_2) = -f_2(x_1,x_2).$$

Applying Algorithm 3.1, we got the NE:

$$x_1^* = 0.3969,\ x_2^* = 0.6300,\ \omega^* = -2.9179\cdot 10^{-11},$$

in the initial loop. It took around 0.54 second.

ii. Consider the two-player game with box constraints in Stein et al. [56, example 2.3] (see also Kroupa and Votroubek [23, example 2]), where the objective functions are

$$\begin{aligned} f_1(x_1,x_2) &= 2(x_1)^3 + 3(x_1x_2)^2 - 2x_1x_2 + x_1 - 3(x_2)^3,\\ f_2(x_1,x_2) &= 4(x_2)^3 - 2(x_1x_2)^2 + (x_1)^2 - (x_1)^2x_2 - 4x_2. \end{aligned}$$

Applying Algorithm 3.1, we detected nonexistence of NEs in the third loop.[3] It took around 0.85 second.

iii. Consider the generalization of separable network games in Kroupa and Votroubek [23, example 5]. The objective functions are

$$\begin{aligned} f_1(x_1,x_2,x_3) &= 2(x_1)^2 + 2x_1(x_2)^2 - 5x_1x_2 + 4x_1x_3 + x_2 + 2x_3,\\ f_2(x_1,x_2,x_3) &= 2(x_2)^2 - 2x_1(x_2)^2 + 5x_1x_2 - 5x_2x_3 + 2x_2(x_3)^2 - x_2 + 2(x_1)^2,\\ f_3(x_1,x_2,x_3) &= -2x_2(x_3)^2 - 4x_1x_3 + 5x_2x_3 - 2x_3 - 4(x_1)^2 - 2(x_2)^2. \end{aligned}$$

Applying Algorithm 3.1, we detected nonexistence of NEs in the second loop. It took around 0.90 second.

For all NEPs in the following examples except Example 5.7, our method found all NEs with certified completeness of solution sets. In the following, we only report the numerical result of finding all solutions, unless specifically mentioned, for the neatness of this paper.

Example 5.4. Consider the two-player NEP

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

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

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

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

Applying Algorithm 3.7, we got four NEs:

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

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

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.

However, if the second player's objective becomes

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

then there is no NE, which was detected by Algorithm 3.1. It took around 16 seconds.

Example 5.5. Consider the three-player NEP

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

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

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

The Lagrange multipliers can be represented as

$$\lambda_{2,1} = \frac{1}{2}(x_2^T \nabla_{x_2} f_2), \quad \lambda_{2,2} = \frac{\partial f_2}{x_{2,1}} - 2x_{2,1}\lambda_{2,1}, \quad \lambda_{2,3} = \frac{\partial f_2}{x_{2,2}} - 2x_{2,2}\lambda_{2,1},$$
$$\lambda_{3,1} = -\frac{x_{3,1}}{2}\frac{\partial f_3}{\partial x_{3,1}}, \quad \lambda_{3,2} = -\frac{x_{3,2}}{2}\frac{\partial f_3}{\partial x_{3,2}}.$$

Applying Algorithm 3.7, we got 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.

Nonetheless, 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, which was detected by Algorithm 3.1. It took around 3 seconds.

Example 5.6. Consider the two-player NEP

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

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

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 got 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.

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

Example 5.7. Consider the NEP

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

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

where $n_1 = n_2$. We ran Algorithm 3.7 for cases $n_1 = n_2 = 3, 4, 5, 6$. The computational results are shown in Table 1. In the table, n_1 is the dimension for variables x_1 and x_2, x_1^* and x_2^* are computed solutions to the NEP, and w^* is the accuracy parameter. All time consumptions are displayed in seconds, Because of the relatively large amount of computational time, we only compute one NE for each case above.

We would like to remark that our method can also be applied to solve unconstrained NEPs, where all individual optimization problems have no constraints, or, equivalently, the feasible set X_i for (1.1) is the entire space $\mathbb{R}^{n_i}$. For unconstrained NEPs, the KKT System (2.5) becomes

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

and Algorithms 3.1 and 3.7 can be implemented in the same way.

Example 5.8. Consider the unconstrained NEP

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

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

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

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 $n_1 = n_2 = n_3 = 2, 3, 4, 5, 6$. For all cases, we computed an 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 2. The time is displayed in seconds.

The following are some examples of NEPs from applications.

Example 5.9. Consider the environmental pollution control problem for three countries for the case *autarky* (Breton et al. [4]). Let $x_{i,1} (i = 1, 2, 3)$ denote the (gross) emissions from the ith country. The revenue of the ith country depends on $x_{i,1}$—for example, a typical one is $x_{i,1}\left(b_i - \frac{1}{2} x_{i,1}\right)$. 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 to a certain prescribed level $E_i > 0$ under an environmental constraint. The damage cost of the ith country is assumed to be $d_i(x_{i,1} - \gamma x_{i,2}) + \sum_{j \ne i} c_{i,j} x_{i,2} x_{j,1}$. 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

Table 1. Computational results for Example 5.7.

n_1	x_1^*	x_2^*	ω^*	Time
3	(−0.5774, −0.5774, −0.5774)	(−0.5774, −0.5774, −0.5774)	$-1.0689 \cdot 10^{-7}$	1.31
4	(0.8381, 0.5024, −0.0328, −0.2098)	(−0.1791, −0.0683, 0.4066, 0.8933)	$-1.4459 \cdot 10^{-9}$	62.85
5	(0.8466, 0.4407, 0.1744, −0.0101, −0.2418)	(−0.1944, −0.0512, 0.1238, 0.3370, 0.9114)	$-2.7551 \cdot 10^{-9}$	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)	$-7.0354 \cdot 10^{-9}$	18,079.99

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

Table 2. The computational results for Example 5.8.

n_1	$x_1^* = x_2^* = x_3^*$	ω^*	Time
2	$(-0.8410, -0.7125)$	$-8.8291 \cdot 10^{-9}$	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	$(-0.5157, -0.4992, -0.4762, -0.4457, -0.4060, -0.3534)$	$-3.7536 \cdot 10^{-9}$	1,379.27

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

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

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

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 got 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 5.10. Consider the NEP of the electricity market problem (Contreras et al. [6]). 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 electricity price is given by

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

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

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

The Lagrange multipliers associated 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

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

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

For the following parameters

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

we ran Algorithm 3.7 and 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.

6. Conclusions and Discussions

This paper studies Nash equilibrium problems that are given by polynomial functions. Algorithms 3.1 and 3.7 are proposed for computing one or all NEs. The Moment-SOS hierarchy of semidefinite relaxations is used to solve the appearing polynomial optimization problems. Under generic assumptions, we can compute a Nash equilibrium if it exists and detect its nonexistence if there is none. Moreover, we can get all Nash equilibria if there are finitely many ones of them.

In Nie and Tang [45], a semidefinite relaxation method using rational and parametric Lagrange multiplier expressions is proposed for solving convex GNEPs. Under some general conditions, the method in Nie and Tang [45] is guaranteed to find one GNE or detect nonexistence of GNEs. The NEPs considered in this work are special cases of GNEPs, because they can be viewed as GNEPs where every player's feasible set is independent of other players' strategies. Moreover, for convex NEPs, Algorithm 3.1 reduces to Nie and Tang [45, algorithm 5.3] and terminates at Step 2 in the first loop, as shown in Corollary 3.4. In contrast, this paper mainly focuses on solving nonconvex NEPs, and the main difficulty of problems in the scope of this paper is brought by nonconvexity. Major differences between contributions in this paper and those in Nie and Tang [45] are as follows:

- In this paper, we primarily focus on nonconvex NEPs of polynomials. One of our main contributions in this work is that we proposed an algorithm that finds NEs for nonconvex NEPs, if they exist. Note when there is no convexity being assumed, every block x_i^* of the NE x^* is the global minimizer for $\mathbf{F}_i(x_{-i}^*)$, which is usually nonconvex. For nonconvex NEPs, the KKT conditions are typically not sufficient for global optimality; thus, the updating scheme $\mathcal{K}_i := \mathcal{K}_i \cup \mathcal{U}_i$ in Step 3 of Algorithm 3.1 is applied to preclude KKT points that are not NEs. Therefore, we usually need to solve a sequence of polynomial optimization problems to get NEs. In comparison, Nie and Tang [45] concerns GNEPs where every player solves a convex optimization problem. Therefore, once a KKT point is obtained with some constraint qualification conditions being satisfied, this KKT point must be a GNE. So, there is no need to preclude any KKT point, and we usually only need to solve one polynomial optimization problem for a GNE. Indeed, convex NEPs are studied in Section 3.3, which is the intersection of problems considered in this work and in Nie and Tang [45]. One can easily see that it is way more difficult to solve NEPs without any convexity assumption from our discussion in Sections 3.2 and 3.3.
- The goal of the method in Nie and Tang [45] is to find just one GNE, and it cannot check whether the computed GNE is unique or not. In comparison, Algorithm 3.7 proposed in Section 3.4 aims to find more NEs. Furthermore, when there are finitely many NEs, Algorithm 3.7 can find all NEs and check the completeness of the computed solution set, under some general conditions. We would like to remark that there is no other numerical method that can achieve such computational goals for general NEPs given by polynomials, to the best of the authors' knowledge.
- Algorithms 3.1 and 3.7 assume that all constraining polynomial tuples g_i are nonsingular, so that there exist polynomial expressions for Lagrange multipliers. When the NEP is given by generic polynomials, nonsingularity is satisfied for all $i \in [N]$. However, polynomial Lagrange multiplier expressions typically do not exist for GNEPs. For such cases, one may consider the corresponding Lagrange multipliers as new variables, but this is often computationally expensive, especially when there are a lot of constraints. In Nie and Tang [45], rational and parametric Lagrange multiplier expressions are studied for solving convex GNEPs. For NEPs, when constraints are singular, rational and parametric Lagrange multiplier expressions can also be applied to find NEs. Nonetheless, convergence results in Theorem 3.2 and Corollary 3.9 may no longer hold, because there may exist NEs that are not KKT points when polynomial expressions for Lagrange multipliers do not exist.

There is much interesting future work to do. If there are only finitely many KKT points that are not NEs, Algorithm 3.1 must terminate within finitely many loops. This is shown in Theorem 3.2. For generic NEPPs, the finiteness of KKT points is shown in Theorem A.1. However, the convergence property of Algorithm 3.1 is not known

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

when there are infinitely many KKT points. In Example 5.2(ii), there are infinitely many KKT points that are not NEs, but Algorithm 3.1 is still able to get an NE in a few loops. If there are infinitely many KKT points that are not NEs, does Algorithm 3.1 still converge to find an NE? This question is mostly open to the authors.

It is important to compute NEs efficiently for large-scale NEPs. Even for unconstrained NEPs, the kth-order moment relaxation for (3.7) is a semidefinite program with $\mathcal{O}(n^{2k})$ variables. Algorithm 3.1 may not be computationally practical for solving large-scale NEPs. Sparse polynomial optimization problems are studied in Lasserre [27], Nie and Demmel [43], Waki et al. [59], Wang et al. [60], Wang et al. [61], and Wang et al. [62]. Recently, the software `TSSOS` (Magron and Wang [31]) that implements the term and correlative sparse Moment-SOS relaxations is developed. In Algorithms 3.1 and 3.7, polynomial optimization problems are formulated to find NEs, and one may implement sparse Moment-SOS relaxations for solving these polynomial optimization problems. However, even for the NEPP where each player's optimization problem $\mathbf{F}_i(x_{-i})$ is sparse, the Polynomial Optimization Problem (3.7) may not be sparse. This is because both the polynomial expressions of Lagrange multipliers and the KKT system may consist of dense polynomials (see Qu and Tang [51] for more details). Therefore, how to exploit sparsity to find NEs efficiently for large-scale NEPs is important for future work.

Nonconvex NEPs may or may not have NEs, even if all feasible sets are compact. For each $i \in [N]$, let $\mathcal{B}_i$ be the set of Borel probability measures supported in X_i. Define the measure function

$$\Gamma_i(\mu_1, \ldots, \mu_N) := \int_{X_1} \cdots \int_{X_N} f_i(x_1, \ldots, x_N) d\mu_1 \cdots d\mu_N.$$

The mixed-strategy extension for the NEP (1.3) is to find $(\mu_1^*, \ldots, \mu_N^*) \in \mathcal{B}_1 \times \cdots \times \mathcal{B}_N$ such that

$$\Gamma_i(\mu_1^*, \ldots, \mu_{i-1}^*, \mu_i^*, \mu_{i+1}^*, \mu_N^*) \le \Gamma_i(\mu_1^*, \ldots, \mu_{i-1}^*, \mu_i, \mu_{i+1}^*, \mu_N^*) \tag{6.1}$$

holds for all $i \in [N]$ and for all $\mu_i \in \mathcal{B}_i$. Such a $(\mu_1^*, \ldots, \mu_N^*)$ is called a *mixed-strategy solution*, and it always exists (Glicksberg [13]). Mixed-strategy solutions to finite games are studied in Ahmadi and Zhang [2], Aubin [3], Daskalakis et al. [8], Kontogiannis et al. [21], Nash [33], and Young and Zamir [63]. The mixed-strategy extensions of general continuous NEPs are typically difficult to solve because it is a computational challenge to do operations with measures. However, when the functions are polynomials, the mixed-strategy extension can be equivalently expressed in terms of moment variables. We discuss how this can be done in the following.

For the NEPP (1.3), let $a_{i,j}$ be the degree of f_i in x_j and let

$$b_j = \max\{a_{1,j}, \ldots, a_{N,j}\}.$$

Let $T^{(i)}$ be the Nth-order tensor such that for all $u_j = [x_j]_{b_j}$ and $j \in [N]$,

$$f_i(x) = T^{(i)}(u_1, \ldots, u_N) := \sum_{k_1, \ldots, k_N} T^{(i)}_{k_1, \ldots, k_N} (u_1)_{k_1} \ldots (u_N)_{k_N}.$$

Denote the set $\mathcal{X}_i := \{[x_i]_{b_i} : x_i \in X_i\}$. Let $\operatorname{conv}(\mathcal{X}_i)$ be the convex hull of $\mathcal{X}_i$. For a probability measure $\mu_i \in \mathcal{B}_i$, if $u_j = \int_{X_i} [x_j]_{b_j} d\mu_i$, then we have $u_j \in \operatorname{conv}(\mathcal{X}_i)$ (see Henrion et al. [18], Lasserre [28], and Laurent [30]). Because f_i is a polynomial, for every $(\mu_1, \ldots, \mu_N) \in \mathcal{B}_1 \times \cdots \times \mathcal{B}_N$, there exists $(u_1, \ldots, u_N) \in \operatorname{conv}(\mathcal{X}_1) \times \cdots \times \operatorname{conv}(\mathcal{X}_N)$ such that

$$\int_{X_1} \cdots \int_{X_N} f_i(x_1, \ldots, x_N) d\mu_1 \ldots d\mu_N = T^{(i)}(u_1, \ldots, u_N). \tag{6.2}$$

Conversely, for each $(u_1, \ldots, u_N) \in \operatorname{conv}(\mathcal{X}_1) \times \cdots \times \operatorname{conv}(\mathcal{X}_N)$, there exist probability measures $\mu_1, \ldots, \mu_N$ such that each $\mu_i \in \mathcal{B}_i$, and (6.2) holds. Therefore, the mixed-strategy extension of the NEPP (1.3) is equivalent to its convex moment relaxation: Find a tuple

$$(u_1^*, \ldots, u_N^*) \in \operatorname{conv}(\mathcal{X}_1) \times \cdots \times \operatorname{conv}(\mathcal{X}_N),$$

such that for each $i = 1, \ldots, N$,

$$T^{(i)}(u_1^*, \ldots, u_{i-1}^*, u_i, u_{i+1}^*, \ldots, u_N^*) \ge T^{(i)}(u_1^*, \ldots, u_N^*),$$

for all $u_i \in \operatorname{conv}(\mathcal{X}_i)$. Moreover, if each u_i^* is an extreme point of $\operatorname{conv}(\mathcal{X}_i)$, then one can get an NE for the original NEPP from $(u_1^*, \ldots, u_N^*)$. We refer to Laraki and Lasserre [25] for moment game problems and Dresher et al. [10], Parrilo [49], and Stein et al. [56] for more details on mixed-strategy solutions to polynomial games.

Acknowledgments

The authors thank the editors and anonymous referees for fruitful suggestions.

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

Appendix. Finiteness of KKT Points for Generic NEPPs

The finiteness of KKT points implies that Algorithms 3.1 and 3.7 have finite termination. In the following, we discuss the finiteness of KKT points for generic NEPPs.

After the enumeration of all possibilities of active inequality constraints, we can generally consider the case that (1.1) only has 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. Then, the KKT system for all players is

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

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 (A.1) has finitely many complex solutions.

Theorem A.1. *Let $d_{i,j} > 0$, $a_{i,j} > 0$ be degrees for all $i \in [N]$ and $j \in [m_i]$. 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, whose degree in x_j is $a_{i,j}$, then the KKT System (A.1) has finitely many complex solutions, and, hence, the NEP has finitely many KKT points.*

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

$$b_i := a_{i,i} - 1 + d_{i,1} + \cdots + d_{i,m_i} - m_i,$$
$$\widetilde{x}_i := (x_{i,0}, x_{i,1}, \ldots, x_{i,n_i}), \quad \widetilde{x} := (\widetilde{x}_1, \ldots, \widetilde{x}_N).$$

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

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

When all $g_{i,j}$ are generic polynomials in x_i, the codimension of $\mathcal{U}_i$ is m_i (see Harris [16]), so $\mathcal{U}$ has the codimension $m_1 + \cdots + m_N$.

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

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

It is a multihomogenous polynomial in $\widetilde{x}$. For each i, consider the determinantal variety (the ∇_{x_i} denotes the gradient with respect to x_i)

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

Its multihomogenization is

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

The matrix in the above can be explicitly written as

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

The $(m_i + 1)$-by-$(m_i + 1)$ minors of the matrix J_i are homogeneous in $\widetilde{x}_i$ of degree b_i. They are homogeneous in $\widetilde{x}_j$ of degree $a_{i,j}$, for $j \neq i$. By Nie and Ranestad [44, 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 $\widetilde{x}_i \in \mathcal{U}_i$. That is, for every $\widetilde{x} \in \mathcal{U}_i$, 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

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

for equivalent classes of $\widetilde{x} \in \mathcal{U}$. In the above, $\otimes$ denotes the Kronecker product; $[u]_d^{hom}$ denotes the vector of all monomials in u of degrees equal to d. In other words, $[u]_d^{hom}$ is the subvector of $[u]_d$ for monomials of the highest degree d. Note that $\mathcal{U}$ is birational to $\mathcal{V}$ (consider the natural embedding $\varphi : \mathcal{U}, \to \mathcal{V}$ such that $\phi(\widetilde{x}) = \mathfrak{m}_1(\widetilde{x})$). So, $\mathcal{U}$ and $\mathcal{V}$ have the same

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

codimension (Shafarevich [55]). 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,

$$\widetilde{W}_1 = \bigcup_{I \subseteq [n_1],\, |I| = m_1} \mathcal{X}_I \quad \text{where}$$
$$\mathcal{X}_I := \{\widetilde{x} : \operatorname{rank} J_1(x) \le m_1,\ \det_I J_1(x) \neq 0\}.$$

For each I, we have $\widetilde{x} \in \mathcal{X}_I$ 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. Vanishing of these (m_1+1)-by-(m_1+1) minors of J_1 gives $n_1 - m_1$ linear equations in the vector $\mathfrak{m}_1(\widetilde{x})$ as in (A.2). The coefficients of these linear equations are linearly parameterized by coefficients of f_1. Therefore, when f_1 has generic coefficients, the set

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

is the intersection of $\mathcal{V}$ with hyperplanes given by $n_1 - m_1$ generic linear equations. Because $\mathcal{X}_I \cap \mathcal{U}$ 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. Because

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

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

Second, we 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

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

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

$$\widetilde{W}_2 = \bigcup_{I \subseteq [n_2],\, |I| = m_2} \mathcal{X}'_I \quad \text{where}$$
$$\mathcal{X}'_I := \{\widetilde{x} : \operatorname{rank} J_2(x) \le m_2,\ \det_I J_2(x) \neq 0\}.$$

When f_2 has generic coefficients, the set

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

is the intersection of $\mathcal{V}'$ with $n_2 - m_2$ generic hyperplanes of codimension one. Because $\mathcal{X}'_I \cap \mathcal{U} \cap \widetilde{W}_1$ 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 $\mathcal{Y}'_I$. Last, because

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

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 that the KKT System (A.1) has codimension $n_1 + n_2 + \cdots + n_N$—that is, the dimension of the solution set of (A.1) is zero. So, there are finitely many complex KKT points. □

Endnotes

[1] In some literature, this is also called a *pure strategy Nash equilibrium*, in contrast to *mixed-strategy Nash equilibria*, which are probability measures supported on the feasible strategy sets. We refer to Section 6 for more details on mixed-strategy NEs; also see Dresher et al. [10], Kroupa and Votroubek [23], Nash [33], Parrilo [49], Stein et al. [56], and Young and Zamir [63].

[2] See Section 4 for how to solve polynomial optimization problems and Section 5 for computational information.

[3] We remark that for this NEP, as well as the NEP in Example 5.3(iii), though a (pure strategy) NE does not exist, there exist mixed-strategy solutions. See Kroupa and Votroubek [23] and Stein et al. [56] for more details.

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

References

[1] Adam L, Horčík R, Kasl T, Kroupa T (2021) Double oracle algorithm for computing equilibria in continuous games. *Proc. Conf. AAAI Artificial Intelligence* 35(6):5070–5077.
[2] Ahmadi AA, Zhang J (2021) Semidefinite programming and Nash equilibria in bimatrix games. *INFORMS J. Comput.* 33(2):607–628.
[3] Aubin JP (2002) *Optima and Equilibria: An Introduction to Nonlinear Analysis*, *Graduate Texts in Mathematics*, volume 140 (Springer Science & Business Media, Berlin, Heidelberg).
[4] Breton M, Zaccour G, Zahaf M (2006) A game-theoretic formulation of joint implementation of environmental projects. *Eur. J. Oper. Res.* 168(1):221–239.
[5] Chen Y, Lan G, Ouyang Y (2014) Optimal primal-dual methods for a class of saddle point problems. *SIAM J. Optim.* 24(4):1779–1814.
[6] Contreras J, Klusch M, Krawczyk JB (2004) Numerical solutions to Nash-Cournot equilibria in coupled constraint electricity markets. *IEEE Trans. Power Systems* 19(1):195–206.
[7] Couzoudis E, Renner P (2013) Computing generalized Nash equilibria by polynomial programming. *Math. Methods Oper. Res.* 77(3):459–472.
[8] Daskalakis C, Goldberg PW, Papadimitriou CH (2009) The complexity of computing a Nash equilibrium. *SIAM J. Comput.* 39(1):195–259.
[9] Datta RS (2010) Finding all Nash equilibria of a finite game using polynomial algebra. *Econom. Theory* 42(1):55–96.
[10] Dresher M, Karlin S, Shapley L (2016) Polynomial games. Kuhn HW, Tucker AW, eds. *Contributions to the Theory of Games*, volume 1 (Princeton University Press, Princeton, NJ), 161–180.
[11] Facchinei F, Kanzow C (2010) Generalized Nash equilibrium problems. *Ann. Oper. Res.* 175(1):177–211.
[12] Farnia F, Ozdaglar A (2020) Do GANs always have Nash equilibria? Daumé III H, Singh A, eds. *Proc. 37th Internat. Conf. Machine Learning* (PMLR, New York), 3029–3039.
[13] Glicksberg IL (1952) A further generalization of the Kakutani fixed point theorem, with application to Nash equilibrium points. *Proc. Amer. Math. Soc.* 3(1):170–174.
[14] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2020) Generative adversarial networks. *Commun. ACM* 63(11):139–144.
[15] Gürkan G, Pang JS (2009) Approximations of Nash equilibria. *Math. Program.* 117(1):223–253.
[16] Harris J (2013) *Algebraic Geometry: A First Course*, *Graduate Texts in Mathematics*, vol. 133 (Springer Science & Business Media, New York).
[17] Henrion D, Lasserre JB (2005) Detecting global optimality and extracting solutions in Gloptipoly. Henrion D, Garulli A, eds. *Positive Polynomials in Control*, *Lecture Notes in Control and Information Science*, vol. 312 (Springer, Berlin, Heidelberg), 293–310.
[18] Henrion D, Korda M, Lasserre JB (2020) *The Moment-SOS Hierarchy: Lectures in Probability, Statistics, Computational Geometry, Control and Nonlinear PDEs*, *Optimization and its Applications*, vol. 4 (World Scientific, Singapore).
[19] Henrion D, Lasserre JB, Löfberg J (2009) Gloptipoly 3: Moments, optimization and semidefinite programming. *Optim. Methods Software* 24(4–5):761–779.
[20] Hillar C, Nie J (2008) An elementary and constructive solution to Hilbert's 17th problem for matrices. *Proc. Amer. Math. Soc.* 136(1):73–76.
[21] Kontogiannis SC, Panagopoulou PN, Spirakis PG (2006) Polynomial algorithms for approximating Nash equilibria of bimatrix games. Spirakis PG, Mavronicolas M, Kontogiannis SC, eds. *WINE 2006 Internat. Workshop Internet Network Econom.* (Springer, Berlin, Heidelberg), 286–296.
[22] Krawczyk JB, Uryasev S (2000) Relaxation algorithms to find Nash equilibria with economic applications. *Environ. Model. Assessment* 5(1):63–73.
[23] Kroupa T, Votroubek T (2023) Multiple oracle algorithm to solve continuous games. *Proc. Decision Game Theory Security 13th Internat. Conf. GameSec 2022* (Springer, Berlin, Heidelberg), 149–167.
[24] Kulkarni AA, Shanbhag UV (2012) On the variational equilibrium as a refinement of the generalized Nash equilibrium. *Automatica* 48(1):45–55.
[25] Laraki R, Lasserre JB (2012) Semidefinite programming for min–max problems and games. *Math. Program.* 131(1):305–332.
[26] Lasserre JB (2001) Global optimization with polynomials and the problem of moments. *SIAM J. Optim.* 11(3):796–817.
[27] Lasserre JB (2006) Convergent SDP-relaxations in polynomial optimization with sparsity. *SIAM J. Optim.* 17(3):822–843.
[28] Lasserre JB (2015) *An Introduction to Polynomial and Semi-Algebraic Optimization*, *Cambridge Texts in Applied Mathematics*, vol. 52 (Cambridge University Press, Cambridge, UK).
[29] Lasserre JB, Laurent M, Rostalski P (2008) Semidefinite characterization and computation of zero-dimensional real radical ideals. *Foundations Comput. Math.* 8(5):607–647.
[30] Laurent M (2009) Sums of squares, moment matrices and optimization over polynomials. Putinar M, Sullivant S, eds. *Emerging Applications of Algebraic Geometry*, *The IMA Volumes in Mathematics and its Applications*, vol. 149 (Springer, New York), 157–270.
[31] Magron V, Wang J (2021) TSSOS: A Julia library to exploit sparsity for large-scale polynomial optimization. Preprint, submitted March 1, https://arxiv.org/abs/2103.00915.
[32] Maskin E (1999) Nash equilibrium and welfare optimality. *Rev. Econom. Stud.* 66(1):23–38.
[33] Nash J (1951) Non-cooperative games. *Ann. Math.* 54(2):286–295.
[34] Nedić A, Ozdaglar A (2009) Subgradient methods for saddle-point problems. *J. Optim. Theory Appl.* 142(1):205–228.
[35] Nie J (2011) Polynomial matrix inequality and semidefinite representation. *Math. Oper. Res.* 36(3):398–415.
[36] Nie J (2012) Sum of squares methods for minimizing polynomial forms over spheres and hypersurfaces. *Front. Math. China* 7(2):321–346.
[37] Nie J (2013a) Certifying convergence of Lasserre's hierarchy via flat truncation. *Math. Program.* 142(1):485–510.
[38] Nie J (2013b) Polynomial optimization with real varieties. *SIAM J. Optim.* 23(3):1634–1646.
[39] Nie J (2014a) The $\mathcal{A}$-truncated $\mathcal{K}$-moment problem. *Foundations Comput. Math.* 14(6):1243–1276.
[40] Nie J (2014b) Optimality conditions and finite convergence of Lasserre's hierarchy. *Math. Program.* 146(1):97–121.
[41] Nie J (2017) Symmetric tensor nuclear norms. *SIAM J. Appl. Algebra Geom.* 1(1):599–625.
[42] Nie J (2019) Tight relaxations for polynomial optimization and Lagrange multiplier expressions. *Math. Program.* 178(1):1–37.
[43] Nie J, Demmel J (2009) Sparse SOS relaxations for minimizing functions that are summations of small polynomials. *SIAM J. Optim.* 19(4):1534–1558.

Downloaded from informs.org by [2607:f720:1901::1ed] on 24 August 2023, at 16:28 . For personal use only, all rights reserved.

[44] Nie J, Ranestad K (2009) Algebraic degree of polynomial optimization. *SIAM J. Optim.* 20(1):485–502.
[45] Nie J, Tang X (2023) Convex generalized Nash equilibrium problems and polynomial optimization. *Math. Program.* 198:1485–1518.
[46] Nie J, Zhang X (2018) Real eigenvalues of nonsymmetric tensors. *Comput. Optim. Appl.* 70(1):1–32.
[47] Nie J, Tang X, Xu L (2021) The Gauss–Seidel method for generalized Nash equilibrium problems of polynomials. *Comput. Optim. Appl.* 78(2):529–557.
[48] Nie J, Yang Z, Zhou G (2022) The saddle point problem of polynomials. *Foundations Comput. Math.* 22(4):1133–1169.
[49] Parrilo PA (2006) Polynomial games and sum of squares optimization. *Proc. 45th IEEE Conf. Decision Control* (IEEE, Piscataway, NJ), 2855–2860.
[50] Putinar M (1993) Positive polynomials on compact semi-algebraic sets. *Indiana Univ. Math. J.* 42(3):969–984.
[51] Qu Z, Tang X (2022) A correlative sparse Lagrange multiplier expression relaxation for polynomial optimization. Preprint, submitted August 8, https://arxiv.org/abs/2208.03979.
[52] Ratliff LJ, Burden SA, Sastry SS (2013) Characterization and computation of local Nash equilibria in continuous games. *2013 51st Annu. Allerton Conf. Commun. Control Comput.* (IEEE, Piscataway, NJ), 917–924.
[53] Schofield N, Sened I (2002) Local Nash equilibrium in multiparty politics. *Ann. Oper. Res.* 109(1):193–211.
[54] Schweighofer M (2005) Optimization of polynomials on compact semialgebraic sets. *SIAM J. Optim.* 15(3):805–825.
[55] Shafarevich IR (2013) *Basic Algebraic Geometry 1: Varieties in Projective Space* (Springer Science & Business Media, Berlin, Heidelberg).
[56] Stein ND, Ozdaglar A, Parrilo PA (2008) Separable and low-rank continuous games. *Internat. J. Game Theory* 37(4):475–504.
[57] Sturm JF (1999) Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optim. Methods Software* 11(1–4):625–653.
[58] Uryas'ev S, Rubinstein RY (1994) On relaxation algorithms in computation of noncooperative equilibria. *IEEE Trans. Automatic Control* 39(6):1263–1267.
[59] Waki H, Kim S, Kojima M, Muramatsu M (2006) Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity. *SIAM J. Optim.* 17(1):218–242.
[60] Wang J, Magron V, Lasserre JB (2021a) Chordal-TSSOS: A moment-SOS hierarchy that exploits term sparsity with chordal extension. *SIAM J. Optim.* 31(1):114–141.
[61] Wang J, Magron V, Lasserre JB (2021b) TSSOS: A moment-SOS hierarchy that exploits term sparsity. *SIAM J. Optim.* 31(1):30–58.
[62] Wang J, Magron V, Lasserre JB, Mai NHA (2022) CS-TSSOS: Correlative and term sparsity for large-scale polynomial optimization. *ACM Trans. Math. Software* 48(4):1–26.
[63] Young P, Zamir S (2014) *Handbook of Game Theory* (Elsevier, Amsterdam).