GENERALIZED NASH EQUILIBRIUM PROBLEMS WITH QUASI-LINEAR CONSTRAINTS

JIYOUNG CHOI, JIAWANG NIE, XINDONG TANG, AND SUHAN ZHONG

ABSTRACT. We study generalized Nash equilibrium problems (GNEPs) such that objectives are polynomial functions, and each player's constraints are linear in their own strategy. For such GNEPs, the KKT sets can be represented as unions of simpler sets by Carathéodory's theorem. We give a convenient representation for KKT sets using partial Lagrange multiplier expressions. This produces a set of branch polynomial optimization problems, which can be efficiently solved by Moment-SOS relaxations. By doing this, we can compute all generalized Nash equilibria or detect their nonexistence. Numerical experiments are also provided to demonstrate the computational efficiency.

1. Introduction

The generalized Nash equilibrium problem (GNEP) is a class of games that determines strategies for a group of players so that each player's benefit cannot be improved for the given strategy by other players. Suppose there are N players, and the ith player's strategy is represented by the n_i-dimensional real vector $x_i := (x_{i,1}, \dots, x_{i,n_i}) \in \mathbb{R}^{n_i}$. The tuple

$$x := (x_1, \dots, x_N)$$

denotes the set of all player's strategies, with the total dimension

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

When the ith player's strategy x_i is focused, for convenience, we also write

$$x = (x_i, x_{-i}), \quad \text{with} \quad x_{-i} := (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_N).$$

Assume the ith player's decision optimization problem is

$$\mathrm{F}_i(x_{-i}) : \begin{cases} \min\limits_{x_i \in \mathbb{R}^{n_i}} & f_i(x_i, x_{-i}) \\ \quad s.t. & g_i(x_i, x_{-i}) \geq 0, \end{cases} \tag{1.1}$$

where $g_i : \mathbb{R}^n \to \mathbb{R}^{m_i}$ is an m_i-dimensional vector-valued function. For convenience, we only consider inequality constrained GNEPs. The discussion for GNEPs with equality constraints is quite similar. A tuple of strategies $u = (u_1, \dots, u_N)$ is said to be a *generalized Nash equilibrium* (GNE) if each u_i is a minimizer of $\mathrm{F}_i(u_{-i})$. Throughout the paper, an optimizer means a global optimizer, unless its meaning is specified. For each $i = 1, \dots, N$ and for a given x_{-i}, we let $\mathcal{S}_i(x_{-i})$ denote the set of minimizers for $\mathrm{F}_i(x_{-i})$. Therefore, the set $\mathcal{S}$ of all GNEs can be written as

$$\mathcal{S} = \Big\{(u_1, \dots, u_N) : \; u_i \in \mathcal{S}_i(u_{-i}) \text{ for } i = 1, \dots, N\Big\}. \tag{1.2}$$

2020 *Mathematics Subject Classification.* 90C23, 90C33, 91A10, 65K05.
Key words and phrases. GNE, KKT point, pLME, Moment, SOS.

In this paper, we consider a broad class of GNEPs such that all objectives f_i are polynomials in x, while the optimization problem $\mathrm{F}_i(x_{-i})$ has *constraints that are linear in* x_i (they may be polynomial in x_{-i}, so we called them as quasi-linear constraints). We assume that $g_i = (g_{i,1}, \dots, g_{i,m_i})$ is such that for each $j \in [m_i] \coloneqq \{1, \dots, m_i\}$,

$$g_{i,j}(x) \;=\; (\mathbf{a}_{i,j})^T x_i - b_{i,j}(x_{-i}), \tag{1.3}$$

where each $\mathbf{a}_{i,j}$ is a constant n_i-dimensional real vector and $b_{i,j}(x_{-i})$ is a scalar polynomial in x_{-i}. For convenience, denote the coefficient matrix (the superscript ^{T} means the transpose of a matrix or vector)

$$A_i \;\coloneqq\; \begin{bmatrix} \mathbf{a}_{i,1} & \cdots & \mathbf{a}_{i,m_i} \end{bmatrix}^T. \tag{1.4}$$

We also denote the tuple of polynomials in x_{-i}:

$$b_i(x_{-i}) \;\coloneqq\; \begin{bmatrix} b_{i,1}(x_{-i}) & \cdots & b_{i,m_i}(x_{-i}) \end{bmatrix}^T.$$

Then $g_i(x) = A_i x_i - b_i(x_{-i})$ and the ith player's feasible set can be written as

$$X_i(x_{-i}) \;\coloneqq\; \{x_i \in \mathbb{R}^{n_i} : A_i x_i - b_i(x_{-i}) \;\geq\; 0\}. \tag{1.5}$$

The entire feasible set of the GNEP is

$$X \;\coloneqq\; \Big\{(x_1, \dots, x_N) : \, A_i x_i - b_i(x_{-i}) \;\geq\; 0 \text{ for all } i = 1, \dots, N\Big\}. \tag{1.6}$$

The GNEP (1.1) is called a *Nash equilibrium problem* (NEP) if each feasible set $X_i(x_{-i})$ is independent of x_{-i}. A solution to the NEP is then called a *Nash equilibrium* (NE). The GNEP is said to be *convex* if each $\mathrm{F}_i(x_{-i})$ is a convex optimization problem in x_i for every given x_{-i} such that $X_i(x_{-i}) \neq \emptyset$. GNEPs were originally introduced to model economic problems. They are now widely used in various applications, such as transportation, telecommunications, and machine learning. We refer to [5, 11, 12, 13, 35, 36] for applications and surveys of GNEPs.

Solving GNEPs is typically a challenging task, primarily due to the interactions among different players' strategies concerning the objectives and feasible sets. The set of GNEs may be nonconvex, even for strictly convex NEPs [31]. Much earlier work exists to solve GNEPs. Some of them apply classical nonlinear optimization methods, such as the penalty method [2, 12] and Augmented-Lagrangian method [19]. Variational inequality and quasi-variational inequality reformulations are also frequently used to solve GNEPs; see the work [9, 35, 39]. The Nikaido-Isoda function type methods are proposed in [7, 41]. The ADMM-type methods for GNEPs in Hilbert spaces are introduced in [3]. The Gauss-Seidel type methods are proposed in [14]. Methods based on *Karush-Kuhn-Tucker* (KKT) conditions are given in [6, 10]. Certain convexity assumptions are often needed for these methods to be guaranteed to compute a GNE. It is generally quite challenging to solve nonconvex GNEPs. As an alternative, for nonconvex GNEPs, some work aims to find quasi-NEs introduced in [5, 36]. For more detailed introductions to GNEPs, we refer to [11, 13].

Contributions. GNEPs given by polynomial or rational functions are studied in [30, 32, 33]. Particularly, in [30, 33], Moment-SOS relaxation methods are proposed to find GNEs or to detect their nonexistence. These methods require to use *Lagrange multiplier expressions* (LMEs) for some common constraints like simplex, balls, or cubes. However, for more general constraints, LMEs are quite expensive to obtain. In particular, for GNEPs with many linear constraints, the usage of

LMEs is quite inconvenient. In this paper, we study these kinds of GNEPs, which have many quasi-linear constraints. The linear property of constraints can be used to get computationally convenient expressions for Lagrange multipliers. This novel approach greatly improves the efficiency of solving GNEPs.

Note that $x = (x_1, \dots, x_N)$ is a GNE if and only if every $x_i \in \mathcal{S}_i(x_{-i})$. In computation, one can relax $x_i \in \mathcal{S}_i(x_{-i})$ by KKT conditions. For the ith player's decision problem $\mathrm{F}_i(x_{-i})$, these conditions are

$$\boxed{\begin{array}{c} \nabla_{x_i} f_i(x) - A_i^T \lambda_i = 0, \\ 0 \le \lambda_i \perp \big(A_i x_i - b_i(x_{-i})\big) \ge 0. \end{array}} \tag{1.7}$$

In the above, ∇_{x_i} denotes the gradient in the subvector x_i and

$$\lambda_i = \begin{bmatrix} \lambda_{i,1} & \cdots & \lambda_{i,m_i} \end{bmatrix}^T$$

is the vector of Lagrange multipliers. The notation $\lambda_i \perp g_i$ means that λ_i and $g_i(x)$ are perpendicular to each other. The strategy vector x is called a KKT point if for each $i \in [N]$, there exists $\lambda_i \in \mathbb{R}^{m_i}$ such that (x, λ_i) satisfies (1.7).

It is usually not easy to solve (1.7) directly to get a KKT point, since there are Lagrange multiplier variables like λ_i. Moreover, a KKT point may not be a GNE for nonconvex GNEPs. To solve (1.7) more efficiently, LMEs are introduced in [30, 33]. Generally, there exists a vector function $\tau_i(x)$ such that

$$\lambda_i = \tau_i(x) \ \text{ satisfies (1.7) for every KKT point } x. \tag{1.8}$$

Such $\tau_i(x)$ is called a Lagrange multiplier expression. For $A_i \in \mathbb{R}^{m_i \times n_i}$, the transpose A_i^T is n_i-by-m_i. For the special case that $\operatorname{rank} A_i = m_i$,

$$\lambda_i \ = \ (A_i A_i^T)^{-1} A_i \nabla_{x_i} f_i(x).$$

However, for more general cases where $\operatorname{rank} A_i < m_i$, the above LMEs are not applicable.

When GNEPs have quasi-linear constraints, we propose a computationally efficient way to get LMEs. For each fixed GNE x, the KKT system (1.7) must have a Lagrange multiplier vector λ_i which has at most $r_i := \operatorname{rank} A_i$ nonzero entries. This can be implied by Carathéodory's theorem. Suppose $J_i \subseteq [m_i]$ is the label set of nonzero entries of λ_i, with the cardinality $|J_i| = r_i$. Let A_{i,J_i} be the submatrix of A_i whose rows are labelled by J_i, and we define $\lambda_{i,J_i}, b_{i,J_i}$ respectively in a similar way. Then (1.7) implies the equation

$$\nabla_{x_i} f_i(x) - A_{i,J_i}^T \lambda_{i,J_i} = 0.$$

Assume A_{i,J_i} is invertible, then

$$\lambda_{i,J_i} = (A_{i,J_i}^T)^{-1} \nabla_{x_i} f_i(x).$$

The right-hand side is a polynomial function in x. We call it a *partial Lagrange multiplier expressions* (pLME). Then, (1.7) simplifies to

$$0 \le (A_{i,J_i}^T)^{-1} \nabla_{x_i} f_i(x) \ \perp \ \big(A_{i,J_i} x_i - b_{i,J_i}(x_{-i})\big) \ge 0. \tag{1.9}$$

In computational practice, the label set J_i is usually unknown. However, we can enumerate all such $J_i \subseteq [m_i]$ with $|J_i| = r_i$. Therefore, the KKT set $\mathcal{K}$ can be represented as

$$\mathcal{K} = \bigcup_{\substack{i\in[N],\\ J_i\subseteq[m_i],|J_i|=r_i}} \Big\{x \in X : x \text{ satisfies } (1.9)\Big\}. \tag{1.10}$$

Using pLMEs for the KKT set, we propose a method for finding all GNEs, for both convex and nonconvex GNEPs. Our major contributions are:

- We give a pLME representation for the KKT set as in (1.10). The pLMEs can be explicitly given in closed formulae. They are computationally convenient and efficient. Their usage can help solve large GNEPs.
- Based on partial Lagrange multiplier expressions, we give a method for computing GNEs for both convex and nonconvex GNEPs. When the KKT set $\mathcal{K}$ is finite (this is the generic case), we can find *all* GNEs or detect their nonexistence.
- We remark that our method is *not* enumerating active constraining sets since we only consider label sets $J_i \subseteq [m_i]$ with $|J_i| = r_i$. The number of our enumerations depends on the gap $m_i - r_i$. To the best of our knowledge, this is the first work for applying this approach to solve GNEPs.

The paper is organized as follows. Section 2 introduces notation and some basics for polynomial optimization. In Section 3, we introduce partial Lagrange multiplier expressions. In Section 4, we give algorithms for solving GNEPs. Section 5 introduces the Moment-SOS relaxations for solving polynomial optimization problems. Numerical experiments are presented in Section 6. The conclusions and some discussions are given in Section 7.

2. Preliminaries

Notation. The symbol $\mathbb{N}$ (resp., $\mathbb{R}$) represents the set of nonnegative integers (resp., real numbers). The $\mathbb{R}^n$ denotes the n-dimensional Euclidean space. Let $\mathbb{R}[x]$ denote the ring of polynomials with real coefficients in x, and $\mathbb{R}[x]_d$ denote its subset of polynomials whose degrees are not greater than d. For the ith player's strategy vector $x_i \in \mathbb{R}^{n_i}$, the $x_{i,j}$ denotes the jth entry of x_i, for $j = 1, \dots, n_i$. For ith player's objective $f_i(x)$, $\nabla_{x_i} f_i$ denotes its gradient with respect to x_i. For an integer $n > 0$, $[n] \coloneqq \{1, 2, \dots, n\}$. For a vector $u \in \mathbb{R}^n$, $\|u\|$ denotes the standard Euclidean norm. The e_i represents the vector of all zeros except that ith entry is 1, while $\mathbf{1}$ denotes the vector of all ones. The symbol $\mathbf{0}_{n_1\times n_2}$ stands for the zero matrix of dimension $n_1 \times n_2$, and the subscript may be omitted if the dimension is clear in the context. For $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{N}^n$, denote $|\alpha| \coloneqq \alpha_1 + \cdots + \alpha_n$. We write the monomial power $x^\alpha \coloneqq x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ and denote power set $\mathbb{N}_d^n \coloneqq \{\alpha \in \mathbb{R}^n : |\alpha| \le d\}$. The column vector of all monomials in x and of degrees up to d is denoted as

$$[x]_d \coloneqq \begin{bmatrix} 1 & x_1 & \cdots & x_n & x_1^2 & x_1x_2 & \cdots & x_{n-1}x_n^{d-1} & x_n^d \end{bmatrix}. \tag{2.1}$$

For a set T, its cardinality is denoted as $|T|$. For a symmetric matrix A, the inequality $A \succeq 0$ (resp., $A \succ 0$) means that A is positive semidefinite (resp., positive definite).

2.1. **Polynomial optimization.** For a polynomial $p \in \mathbb{R}[x]$ and subsets $I, J \subseteq \mathbb{R}[x]$, define the product and Minkowski sum

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

A subset $I \subseteq \mathbb{R}[x]$ is an ideal of $\mathbb{R}[x]$ if $I + I \subseteq I$ and $p \cdot I \subseteq I$ for all $p \in \mathbb{R}[x]$. For a tuple $h = (h_1, \ldots, h_m)$ of polynomials in $\mathbb{R}[x]$, the ideal generated by h is

$$\mathrm{Ideal}[h] \coloneqq h_1 \cdot \mathbb{R}[x] + \cdots + h_m \cdot \mathbb{R}[x].$$

The real zero set of h is

$$Z(h) \coloneqq \{x \in \mathbb{R}^n : h_1(x) = \cdots = h_m(x) = 0\}.$$

A polynomial $\sigma \in \mathbb{R}[x]$ is said to be a sum of squares (SOS) if there are polynomials $p_1, \ldots, p_k \in \mathbb{R}[x]$ such that $\sigma = p_1^2 + \cdots + p_k^2$. The set of all SOS polynomials in x is denoted as $\Sigma[x]$. In computation, we often work with the degree-d truncation:

$$\Sigma[x]_d \coloneqq \Sigma[x] \cap \mathbb{R}[x]_d.$$

For a polynomial tuple $g \coloneqq (g_1, \ldots, g_t)$, denote

$$S(g) \coloneqq \{x \in \mathbb{R}^n : g_1(x) \geq 0, \ldots, g_t(x) \geq 0\}.$$

Clearly, if there are SOS polynomials $\sigma_0, \ldots, \sigma_t$ such that

$$f = \sigma_0 + \sigma_1 g_1 + \cdots + \sigma_t g_t, \tag{2.2}$$

then $f \geq 0$ on $S(g)$. So, we consider the set

$$\mathrm{QM}[g] \coloneqq \Sigma[x] + g_1 \cdot \Sigma[x] + \cdots + g_t \cdot \Sigma[x].$$

The above set $\mathrm{QM}[g]$ is called the quadratic module generated by g. The degree-d truncation of $\mathrm{QM}[g]$ is similarly defined as

$$\mathrm{QM}[g]_d \coloneqq \Sigma[x]_d + g_1 \cdot \Sigma[x]_{d-\deg(g_1)} + \cdots + g_t \cdot \Sigma[x]_{d-\deg(g_t)}.$$

We are interested in conditions for a polynomial $f \geq 0$ on $Z(h) \cap S(g)$. If $f \in \mathrm{Ideal}[h] + \mathrm{QM}[g]$, then it is easy to see that $f \geq 0$ on $Z(h) \cap S(g)$. The reverse is not necessarily true. The set $\mathrm{Ideal}[h] + \mathrm{QM}[g]$ is said to be *archimedean* if there exists $q \in \mathrm{Ideal}[h] + \mathrm{QM}[g]$ such that $S(q)$ is a compact set. When $\mathrm{Ideal}[h] + \mathrm{QM}[g]$ is archimedean, if $f > 0$ on $Z(h) \cap S(g)$, then $f \in \mathrm{Ideal}[h] + \mathrm{QM}[g]$. This conclusion is referenced as *Putinar's Positivstellensatz* [37]. Interestingly, if $f \geq 0$ on $Z(h) \cap S(g)$, we also have $f \in \mathrm{Ideal}[h] + \mathrm{QM}[g]$, under some standard optimality conditions [27].

2.2. **Localizing and moment matrices.** A real vector y is called a *truncated multi-sequence* (tms) of degree $2k$ if it is labeled as

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

For a tms $y \in \mathbb{R}^{\mathbb{N}^n_{2k}}$ and a polynomial $f = \sum_{\alpha \in \mathbb{N}^n_{2k}} f_\alpha x^\alpha$, define the operation

$$\langle f, y \rangle \coloneqq \sum_{\alpha \in \mathbb{N}^n_{2k}} f_\alpha y_\alpha. \tag{2.3}$$

For $q \in \mathbb{R}[x]_{2k}$ and $t = k - \lceil \deg(q)/2 \rceil$, the product $q \cdot [x]_t [x]_t^T$ is a symmetric matrix polynomial of length $\binom{n+t}{t}$, which can be expressed as

$$q \cdot [x]_t [x]_t^T = \sum_{\alpha \in \mathbb{N}^n_{2k}} x^\alpha Q_\alpha$$

for some symmetric matrices Q_α. For $y \in \mathbb{R}^{\mathbb{N}^n_{2k}}$, denote the matrix

$$L_q^{(k)}[y] \coloneqq \sum_{\alpha \in \mathbb{N}^n_{2k}} y_\alpha Q_\alpha.$$

It is called the kth order *localizing matrix* of q and generated by y. In particular, if $q = 1$ (the constant 1 polynomial), the $L_q^{(k)}[y]$ is reduced to the moment matrix

$$M_k[y] \coloneqq L_1^{(k)}[y].$$

Quadratic modules, moment, and localizing matrices are useful for solving polynomial optimization. We refer to [16, 20, 21, 25] for a more detailed introduction to them.

3. Partial Lagrange Multiplier Expressions

This section discusses how to find a convenient representation for the KKT set with partial Lagrange multiplier expressions. We consider GNEPs with quasi-linear constraints. The ith player's decision optimization problem $\mathrm{F}_i(x_{-i})$ reads

$$\left\{ \begin{array}{cl} \min\limits_{x_i \in \mathbb{R}^{n_i}} & f_i(x_i, x_{-i}) \\ s.t. & A_i x_i - b_i(x_{-i}) \geq 0, \end{array} \right. \tag{3.1}$$

where $A_i \in \mathbb{R}^{m_i \times n_i}$ and $b_i(x_{-i})$ is a polynomial vector in x_{-i}. Recall the notation

$$A_i \,=\, \begin{bmatrix} \mathbf{a}_{i,1} & \cdots & \mathbf{a}_{i,m_i} \end{bmatrix}^T.$$

The jth row vector of A_i is $\mathbf{a}_{i,j}^T$. Since the constraints of (3.1) are linear, for every optimizer $x_i \in \mathcal{S}_i(x_{-i})$, there exists the Lagrange multiplier vector $\lambda_i = (\lambda_{i,1}, \ldots, \lambda_{i,m_i})$ such that

$$\boxed{\begin{array}{c} \nabla_{x_i} f_i(x) - A_i^T \lambda_i = 0, \\ 0 \leq \lambda_i \perp (A_i x_i - b_i(x_{-i})) \geq 0. \end{array}} \tag{3.2}$$

The set of all KKT points is

$$\mathcal{K} \,=\, \left\{ x \in X \;\middle|\; \begin{array}{c} \exists\, (\lambda_1, \ldots, \lambda_N) \text{ such that for each } i \in [N], \\ \nabla_{x_i} f_i(x_i, x_{-i}) - A_i^T \lambda_i = 0\ , \\ 0 \leq \lambda_i \perp (A_i x_i - b_i(x_{-i})) \geq 0 \end{array} \right\}. \tag{3.3}$$

When $\operatorname{rank} A_i = m_i \leq n_i$, we can get the following expression for λ_i:

$$\lambda_i \,=\, (A_i A_i^T)^{-1} A_i \nabla_{x_i} f_i(x).$$

When $m_i > n_i$, the above expression is not available since the matrix product $A_i A_i^T$ is singular. Indeed, for the case $m_i > n_i$, a polynomial expression for λ_i typically does not exist, since $b_i(x_{-i})$ depends on x_{-i}, but a rational expression for λ_i always exists. This is shown in [30, 33]. However, such an expression for λ_i may be too complicated to be practical. The expression becomes more complicated if the gap $m_i - n_i$ is large (see [28, Proposition 4.1]).

We look for more convenient expressions for λ_i. Let

$$r_i \coloneqq \operatorname{rank} A_i. \tag{3.4}$$

By Carathéodory's Theorem, for each $x \in \mathcal{K}$, the KKT system (3.2) has a solution λ_i that has at most r_i nonzero entries. This means that $m_i - r_i$ entries of such a λ_i must be zeros. If we know the label set J_i of nonzero entries of λ_i, the expression for λ_i can be simplified. This gives a *partial Lagrange multiplier expression* (pLME) for λ_i.

3.1. **The pLMEs.** To find pLMEs, consider the linear system

$$\nabla_{x_i} f_i(x) \;=\; A_i^T \lambda_i, \quad \lambda_i \ge 0. \tag{3.5}$$

For a subset $J_i \subseteq [m_i]$, let A_{i,J_i} denote the submatrix of A_i whose row labels are in J_i, and so is λ_{i,J_i}. That is

$$A_{i,J_i} \coloneqq \left[\mathbf{a}_{i,j}^T\right]_{j\in J_i}, \quad \lambda_{i,J_i} \coloneqq \left[\lambda_{i,j}\right]_{j\in J_i}.$$

Let r_i be the rank as in (3.4). Denote the set of label sets

$$\mathcal{P}_i \;\coloneqq\; \Big\{ J_i \subseteq [m_i] : |J_i| = r_i,\ \operatorname{rank}(A_{i,J_i}) = r_i \Big\}. \tag{3.6}$$

Since A_i is m_i-by-n_i, we know $r_i \le \min\{m_i, n_i\}$ and each $\mathcal{P}_i$ is nonempty. For each $J_i \in \mathcal{P}_i$, the vector $\lambda_i = (\lambda_{i,1}, \ldots, \lambda_{i,m_i}) \ge 0$ is said to be a *basic feasible solution* of (3.5) with respect to J_i if $\lambda_{i,j} = 0$ for all $j \notin J_i$ and

$$\nabla_{x_i} f_i(x) \;=\; \sum_{j\in J_i} \lambda_{i,j}\mathbf{a}_{i,j} \;=\; (A_{i,J_i})^T \lambda_{i,J_i}. \tag{3.7}$$

Basic feasible solutions can be conveniently expressed by pLMEs. Multiplying A_{i,J_i} on both sides of (3.7), we get

$$(A_{i,J_i} A_{i,J_i}^T)\lambda_{i,J_i} \;=\; A_{i,J_i} \nabla_{x_i} f_i(x).$$

Since $\operatorname{rank} A_{i,J_i} = r_i$, this gives the pLME:

$$\lambda_{i,J_i} \;=\; \lambda_{i,J_i}(x) \;\coloneqq\; (A_{i,J_i} A_{i,J_i}^T)^{-1} A_{i,J_i} \nabla_{x_i} f_i(x). \tag{3.8}$$

In particular, for the case $r_i = n_i$, the above simplifies to

$$\lambda_{i,J_i}(x) \;=\; (A_{i,J_i})^{-T} \nabla_{x_i} f_i(x). \tag{3.9}$$

Here, the superscript denotes the transpose of the inverse.

Example 3.1. Consider the 2-player GNEP with

$$\mathrm{F}_1(x_{-1}) : \left\{ \begin{array}{cl} \min\limits_{x_1\in\mathbb{R}^2} & \|x_1\|^2 \\ s.t. & \begin{bmatrix} -1 & -1 \\ 1 & 0 \\ 0 & 1 \\ 6 & 1 \end{bmatrix} x_1 \ge \begin{bmatrix} -2 \\ 0 \\ 0 \\ 1 + \mathbf{1}^T x_2 \end{bmatrix} \end{array} \right.,$$

$$\mathrm{F}_2(x_{-2}) : \left\{ \begin{array}{cl} \min\limits_{x_2\in\mathbb{R}^2} & \|x_2\|^2 \\ s.t. & \begin{bmatrix} -1 & 1 \\ 1 & 0 \\ 0 & 1 \\ -1 & 0 \end{bmatrix} x_2 \ge \begin{bmatrix} -2x_{1,1} + 2 \\ 0 \\ 0 \\ x_{1,1} - x_{1,2} - 2 \end{bmatrix} \end{array} \right..$$

One can see that $m_1 = m_2 = 4$, $r_1 = r_2 = 2$ and

$$\begin{array}{l|l} \mathcal{P}_1 = \big\{ \{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, & \mathcal{P}_2 = \big\{ \{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \\ \quad \{2,4\}, \{3,4\} \big\}, & \quad \{3,4\} \big\}. \end{array}$$

The pLMEs are given as in (3.9). For instance,

$$\lambda_{1,\{1,4\}} = \begin{bmatrix} -1 & -1 \\ 6 & 1 \end{bmatrix}^{-T} \begin{bmatrix} 2x_{1,1} \\ 2x_{1,2} \end{bmatrix} = \frac{2}{5} \begin{bmatrix} x_{1,1} - 6x_{1,2} \\ x_{1,1} - x_{1,2} \end{bmatrix},$$

$$\lambda_{2,\{1,2\}} = \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix}^{-T} \begin{bmatrix} 2x_{2,1} \\ 2x_{2,2} \end{bmatrix} = 2 \begin{bmatrix} x_{2,2} \\ x_{2,1} + x_{2,2} \end{bmatrix}.$$

In contrast, if we do not use pLMEs, the full Lagrange multiplier expressions as in [30, 33] are much more complicated for this GNEP.

For $x = (x_i, x_{-i}) \in X$ and $J_i \in \mathcal{P}_i$, if $\lambda_{i,J_i}(x) \geq 0$, then x must be a KKT point of $\mathrm{F}_i(x_{-i})$, since (3.7) is satisfied for $\lambda_{i,J_i}(x)$. Therefore, x_i is a KKT point of $\mathrm{F}_i(x_{-i})$ if

$$\boxed{\begin{array}{c} \exists J_i \in \mathcal{P}_i,\ \nabla_{x_i} f_i(x) = A_{i,J_i}^T \lambda_{i,J_i}(x), \\ 0 \leq \lambda_{i,J_i}(x) \perp \big(A_{i,J_i} x_i - b_{i,J_i}(x_{-i})\big) \geq 0. \end{array}} \tag{3.10}$$

Interestingly, the above is also necessary for x_i to be a KKT point of $\mathrm{F}_i(x_{-i})$, as shown in the following theorem.

Theorem 3.2. *For $x = (x_i, x_{-i}) \in X$, the x_i is a KKT point of $F_i(x_{-i})$ if and only if it satisfies (3.10).*

Proof. Suppose (3.10) is satisfied. Let λ_i be the extension of $\lambda_{i,J_i}(x)$ by adding zero entries. Then (x, λ_i) satisfies the KKT system (3.2), so x_i is a KKT point of $\mathrm{F}_i(x_{-i})$. Conversely, suppose x_i is a KKT point of $\mathrm{F}_i(x_{-i})$. Then there exists λ_i satisfying (3.2). So, the solution set for the linear system (3.5) is nonempty. By Carathéodory's Theorem, $\nabla_{x_i} f_i(x)$ can be represented as a conic combination of linearly independent vectors from A_i^T. Thus, a basic feasible solution must exist for (3.5). This means that (3.10) holds. □

3.2. **Expression of the KKT set.** For the label sets $\mathcal{P}_1, \ldots, \mathcal{P}_N$ as in (3.6), define the Cartesian product

$$\mathcal{P} := \mathcal{P}_1 \times \cdots \times \mathcal{P}_N. \tag{3.11}$$

Table 1 shows some typical instances of $|\mathcal{P}|$ when $\mathrm{rank} A_i = n_i$ for all i. In the table, $|\mathcal{A}|$ represents the number of all possibilities of active constraints. One can see that $|\mathcal{P}| \ll |\mathcal{A}|$.

$(n_1, \ldots, n_N)$ $(m_1, \ldots, m_N)$	$\|\mathcal{P}\|$	$\|\mathcal{A}\|$	$(n_1, \ldots, n_N)$ $(m_1, \ldots, m_N)$	$\|\mathcal{P}\|$	$\|\mathcal{A}\|$
$(2,2)$ $(5,5)$	100	225	$(2,2,2)$ $(5,5,5)$	1000	3375
$(2,4)$ $(5,7)$	350	1470	$(1,2,3)$ $(2,3,4)$	24	168
$(2,4)$ $(4,8)$	420	1620	$(3,3,3)$ $(6,6,6)$	8000	68921
$(3,3)$ $(5,5)$	100	625	$(2,2,2,2)$ $(4,4,4,4)$	1296	10000
$(4,4)$ $(7,7)$	1225	9604	$(3,3,3,3)$ $(5,5,5,5)$	10000	390625

TABLE 1. Some examples of $|\mathcal{P}|$ when $\mathrm{rank} A_i = n_i$ for all i.

For a tuple $J = (J_1, \dots, J_N) \in \mathcal{P}$ with each $J_i \in \mathcal{P}_i$, let $\lambda_{i,J_i}(x)$ be the pLME given by (3.8) and define the set

$$\mathcal{K}_J := \left\{ x \in X \;\middle|\; \begin{array}{c} \nabla_{x_i} f_i(x) - A_{i,J_i}^T \lambda_{i,J_i}(x) = 0, \\ 0 \le \lambda_{i,J_i}(x) \perp \big(A_{i,J_i} x_i - b_{i,J_i}(x_{-i})\big) \ge 0, \\ \text{for all} \quad i = 1, \dots, N \end{array} \right\}. \tag{3.12}$$

Clearly, each $x \in \mathcal{K}_J$ is a KKT point for the GNEP (1.1), so $\mathcal{K}_J \subseteq \mathcal{K}$. Indeed, every KKT point belongs to $\mathcal{K}_J$ for some J. This is shown in the following theorem.

Theorem 3.3. *For the GNEP of (3.1), the KKT set $\mathcal{K}$ can be expressed as*

$$\mathcal{K} = \bigcup_{J \in \mathcal{P}} \mathcal{K}_J. \tag{3.13}$$

Proof. By Theorem 3.2, the KKT set for the optimization $\mathrm{F}_i(x_{-i})$ is

$$\widehat{\mathcal{K}}_i(x_{-i}) := \bigcup_{J_i \in \mathcal{P}_i} \left\{ x_i \;\middle|\; \begin{array}{c} x_i \in X_i(x_{-i}), \quad \nabla_{x_i} f_i(x) - A_{i,J_i}^T \lambda_{i,J_i}(x) = 0, \\ 0 \le \lambda_{i,J_i}(x) \perp \big(A_{i,J_i} x_i - b_{i,J_i}(x_{-i})\big) \ge 0 \end{array} \right\}.$$

In view of (3.12), we have

$$\mathcal{K} = \bigcap_{i=1}^{N} \big\{ x \in X : x_i \in \widehat{\mathcal{K}}_i(x_{-i}) \big\} = \bigcup_{J \in \mathcal{P}} \mathcal{K}_J.$$

So, the equation (3.13) holds. □

When each $\operatorname{rank} A_i = n_i$, the pLME can be given as in (3.9) and Theorem 3.3 implies the following simplified expression.

Corollary 3.4. *If $\operatorname{rank} A_i = n_i$ for each i, then*

$$\mathcal{K}_J = \left\{ x \in X \;\middle|\; \begin{array}{c} 0 \le A_{i,J_i}^{-T} \nabla_{x_i} f_i(x) \perp \big(A_{i,J_i} x_i - b_{i,J_i}(x_{-i})\big) \ge 0, \\ \textit{for all} \quad i = 1, \dots, N \end{array} \right\}$$

for every $J = (J_1, \dots, J_N) \in \mathcal{P}$ and

$$\mathcal{K} = \bigcup_{J \in \mathcal{P}} \left\{ x \in X \;\middle|\; \begin{array}{c} 0 \le A_{i,J_i}^{-T} \nabla_{x_i} f_i(x) \perp \big(A_{i,J_i} x_i - b_{i,J_i}(x_{-i})\big) \ge 0, \\ \textit{for all} \quad i = 1, \dots, N \end{array} \right\}. \tag{3.14}$$

Example 3.5. For the GNEP in Example 3.1, it is clear that $r_i = n_i = 2$ for $i = 1, 2$. For $J = (J_1, J_2)$ with $J_1 = \{1, 4\}$ and $J_2 = \{1, 2\}$, the $\mathcal{K}_J$ is given by

$$\begin{array}{ll}
2 - \mathbf{1}^T x_1 \ge 0, \quad x_1 \ge 0, & 2x_{1,1} - x_{2,1} + x_{2,2} - 2 \ge 0, \\
6x_{1,1} + x_{1,2} - \mathbf{1}^T x_2 - 1 \ge 0, & x_2 \ge 0, \quad 2 - x_{1,1} + x_{1,2} - x_{2,1} \ge 0, \\
x_{1,1} - 6x_{1,2} \ge 0,\; x_{1,1} - x_{1,2} \ge 0, & x_{2,1} + x_{2,2} \ge 0, \\
(x_{1,1} - 6x_{1,2})(2 - \mathbf{1}^T x_1) = 0, & x_{2,2}(2x_{1,1} - x_{2,1} + x_{2,2} - 2) = 0, \\
(x_{1,1} - x_{1,2})(6x_{1,1} + x_{1,2} - \mathbf{1}^T x_2 - 1) = 0, & (x_{2,1} + x_{2,2}) x_{2,1} = 0.
\end{array}$$

Indeed, one can further verify that $\mathcal{K}_J$ is a singleton, i.e.,

$$\mathcal{K}_J = \big\{ (18, 3, 0, 62)/49 \big\}.$$

Furthermore, this point is the unique GNE as well. By Algorithm 4.5, we know that it is contained in $\mathcal{K}_J$ not only for $J = (\{1, 4\}, \{1, 2\})$ but also for $J = (\{2, 4\}, \{1, 2\})$ or $J = (\{3, 4\}, \{1, 2\})$. For the GNE, the active label set is $\hat{J} = (\hat{J}_1, \hat{J}_2)$, with $\hat{J}_1 = \{4\}, \hat{J}_2 = \{1, 2\}$, and $\hat{J}_1 \subsetneqq J_1$. That is, the label set J for finding this KKT point is not the active constraining set $\hat{J}$. We remark that the expressions in (3.13) and (3.14) are not just enumerations of active constraining sets.

4. Solving GNEPs with quasi-linear constraints

We discuss how to solve the GNEP with quasi-linear constraints as in (3.1). Since every GNE x is a KKT point, there exists $J \in \mathcal{P}$ such that $x \in \mathcal{K}_J$. Since $\mathcal{P}$ is a finite set, there are only finitely many choices of J. Moreover, under some genericity assumptions, the KKT set $\mathcal{K}$ is finite. For these cases, the subset $\mathcal{K}_J$ is also finite for every $J \in \mathcal{P}$. This inspires how to find all GNEs.

4.1. Finding all GNEs in $\mathcal{K}_J$.

We introduce how to find GNEs in $\mathcal{K}_J$ for a fixed $J \in \mathcal{P}$. For the given J, pLMEs are given by (3.8), so the set $\mathcal{K}_J$ can be represented by equalities and inequalities of polynomials in the variable x, as shown in (3.12). Let $\Theta \in \mathbb{R}^{(n+1)\times(n+1)}$ be a symmetric positive definite matrix. Consider the following polynomial optimization problem

$$\left\{ \begin{array}{ll} \min & \theta(x) := [x]_1^T \Theta [x]_1 \\ s.t. & x \in \mathcal{K}_J. \end{array} \right. \tag{4.1}$$

If $\mathcal{K}_J \neq \emptyset$, then (4.1) has a unique minimizer u when Θ is generic (see [30, Theorem 5.4]), and u is a KKT point. Otherwise, if (4.1) is infeasible, then $\mathcal{K}_J$ is empty, and there is no GNE in $\mathcal{K}_J$. We will show how to solve (4.1) in Section 5. The following conclusion is obvious.

Theorem 4.1. *For the GNEP as in (3.1), if the optimization problem (4.1) is infeasible, then there is no KKT point in $\mathcal{K}_J$. Otherwise, each minimizer u of (4.1) is a KKT point. Moreover, if the GNEP is convex, u is a GNE.*

For convex GNEPs, once we find a minimizer u for (4.1), then u must be a GNE. However, when the GNEP is nonconvex, u may or may not be a GNE. For nonconvex GNEPs, we can check if u is a GNE or not by solving polynomial optimization problems. By definition, u is a GNE if and only if $\epsilon_i \geq 0$ for every $i \in [m]$, where ϵ_i is the optimal value

$$\left\{ \begin{array}{lll} \epsilon_i := & \min & f_i(x_i, u_{-i}) - f_i(u_i, u_{-i}) \\ & s.t. & x_i \in X_i(u_{-i}). \end{array} \right. \tag{4.2}$$

Therefore, once we get a KKT point u, we solve (4.2) for every $i \in [N]$. If $\epsilon_i \geq 0$ for all $i \in [m]$, then we certify that u is a GNE; otherwise, it is not.

If u is not a GNE, one needs to find other KKT points to solve this GNEP. Also, when $\mathcal{K}_J \neq \emptyset$ but $\mathcal{K}_J \cap \mathcal{S} = \emptyset$, we may need to find all points in $\mathcal{K}_J$ to certify nonexistence of GNEs in $\mathcal{K}_J$. Besides that, people are usually interested in finding all GNEs. In the following, we discuss how to find all GNEs or detect their nonexistence in $\mathcal{K}_J$.

Suppose $u = (u_1, \ldots, u_N) \in \mathcal{K}_J$ is the minimizer of (4.1). Then we have

$$\theta(u) \leq \theta(x) \quad \text{for all } x \in \mathcal{K}_J.$$

When the matrix Θ is generic, the inequality above holds strictly for all $x \in \mathcal{K}_J \backslash \{u\}$. Suppose that u is an isolated point of $\mathcal{K}_J$. This is the case when the GNEP is generic, as shown in [29, Theorem 3.1]. Then, there exists $\delta > 0$ such that

$$\theta(u) + \delta \leq \theta(x), \quad \text{for all } u \neq x \in \mathcal{K}_J. \tag{4.3}$$

For the $\delta > 0$ above, consider the optimization problem

$$\left\{ \begin{array}{ll} \min & \theta(x) \\ s.t. & x \in \mathcal{K}_J, \quad \theta(x) \geq \theta(u) + \delta. \end{array} \right. \tag{4.4}$$

If (4.4) is infeasible, then the set $\mathcal{K}_J$ does not have any KKT points other than u. Otherwise, it must have a minimizer $\hat{u}$ (since Θ is positive definite), which is a KKT point different from u. For the new KKT point $\hat{u}$, we may solve polynomial optimization problems like (4.2) to check if it is a GNE or not.

Indeed, more GNEs can be computed by repeating this process. Suppose we have obtained the KKT points $u^{(1)}, u^{(2)}, \dots, u^{(j)} \in \mathcal{K}_J$ for some $j \geq 1$, in the order that

$$\theta(u^{(1)}) < \theta(u^{(2)}) < \cdots < \theta(u^{(j)}). \tag{4.5}$$

Suppose $u^{(j+1)}$ is a new KKT point such that

$$\theta(u^{(j+1)}) \,=\, \min_{x \in S_j} \theta(x) \quad \text{where} \quad S_j = \mathcal{K}_J \setminus \{u^{(1)}, u^{(2)}, \dots, u^{(j)}\}.$$

If there exists a scalar δ satisfying

$$0 < \delta < \theta(u^{(j+1)}) - \theta(u^{(j)}), \tag{4.6}$$

then $u^{(j+1)}$ can be obtained by computing the minimizer of

$$\begin{cases} \min & \theta(x) \\ s.t. & \theta(x) \geq \theta(u^{(j)}) + \delta, \\ & x \in \mathcal{K}_J. \end{cases} \tag{4.7}$$

The inequality (4.6) can be checked as follows. We can first assign a priori value for δ (say, 0.5), then solve the maximization problem

$$\begin{cases} \theta_{\max} := & \max \quad \theta(x) \\ & s.t. \quad x \in \mathcal{K}_J, \quad \theta(x) \leq \theta(u^{(j)}) + \delta. \end{cases} \tag{4.8}$$

Since $u^{(j)}$ is a feasible point, it always holds $\theta_{\max} \geq \theta(u^{(j)})$. There are two possibilities:

- If $\theta_{\max} = \theta(u^{(j)})$, then $u^{(j)}$ is a maximizer of (4.8). This implies that $u^{(j+1)}$ is infeasible for (4.8), so (4.6) is satisfied.
- If $\theta_{\max} > \theta(u^{(j)})$, then there exists $v \in \mathcal{K}_J$ such that

  $$\theta(u^{(j)}) < \theta(v) \leq \theta(u^{(j)}) + \delta.$$

  This means δ is too large and violates (4.6). We need to decrease the value of δ (e.g., by replacing δ with $\delta/2$) and solve (4.8) again.

In light of the above, we get the following algorithm for finding all GNEs in $\mathcal{K}_J$.

Algorithm 4.2. For the GNEP as in (3.1) and for a given $J \in \mathcal{P}$, select a generic symmetric positive definite matrix Θ and a small positive value (say, 0.5) for δ. Let $\mathcal{S}_J := \emptyset$ and $j := 1$. Then, do the following:

Step 1 Solve the optimization problem (4.1). If it is infeasible, output the nonexistence of GNEs in $\mathcal{K}_J$ and stop. Otherwise, solve (4.1) for a minimizer $u^{(1)}$ and go to Step 2.

Step 2 For each $i \in [N]$, compute the minimum value ϵ_i of (4.2) for $u := u^{(j)}$. If $\epsilon_i \geq 0$ for every i, then update $\mathcal{S}_J := \mathcal{S}_J \cup \{u^{(j)}\}$.

Step 3 Compute the maximum value $\theta_{\max}$ of (4.8).

Step 4 If $\theta_{\max} = \theta(u)$, then go to Step 5; otherwise, let $\delta := \delta/2$ and go to Step 3.

Step 5 Solve the optimization problem (4.7). If it is infeasible, output that $\mathcal{S}_J$ is the set of all GNEs in $\mathcal{K}_J$ and stop. Otherwise, update $j := j + 1$ and solve (4.7) for a minimizer $u^{(j)}$, then go to Step 2.

If the GNEP is convex, every KKT point is a GNE, so Step 2 can be skipped. The properties of Algorithm 4.2 are summarized as follows.

Proposition 4.3. *For the GNEP as in (3.1), the following properties hold for Algorithm 4.2:*

(i) If $\theta_{\max} = \theta(u^{(j)})$ and the optimization (4.7) is infeasible, then $S_J = \{u^{(1)}, \ldots, u^{(j)}\}$ is the set of all GNEs in $\mathcal{K}_J$.

(ii) If $\theta_{\max} = \theta(u^{(j)})$ and $u^{(j+1)}$ is the minimizer of (4.7), then δ satisfies (4.6).

(iii) Assume $u^{(1)}, \ldots, u^{(j)}$ are isolated points of $\mathcal{K}_J$. Suppose Θ is a generic symmetric positive definite matrix, then there exists $\delta > 0$ such that $\theta_{\max} = \theta(u^{(j)})$, i.e., $u^{(j)}$ is the maximizer of (4.8).

Proof. (i) When $\theta_{\max} = \theta(u^{(j)})$, the KKT point $u^{(j)}$ is the maximizer of (4.8). If there is $v \in \mathcal{K}_J$ other than $u^{(1)}, \ldots, u^{(j)}$, then

$$\theta(v) > \theta(u^{(j)}) + \delta.$$

On the other hand, when (4.7) is infeasible, every $x \in \mathcal{K}_J$ must satisfy

$$\theta(x) < \theta(u^{(j)}) + \delta.$$

Therefore, if $\theta_{\max} = \theta(u^{(j)})$ and (4.7) is infeasible, then there are no KKT points in $\mathcal{K}_J$ except $u^{(1)}, \ldots, u^{(j)}$. This implies that all GNEs in $\mathcal{K}_J$ are contained in $\mathcal{S}_J$.

(ii) If $\theta_{\max} = \theta(u^{(j)})$ and $u^{(j+1)}$ exists, then as in (i), we can get

$$\theta(u^{(j+1)}) > \theta(u^{(j)}) + \delta,$$

which means that (4.6) holds.

(iii) For $\epsilon > 0$, let $\mathbb{S}_\epsilon$ denote the set of all $(n+1)$-by-$(n+1)$ symmetric positive definite matrices whose largest eigenvalue equals one and whose smallest eigenvalue is at least ϵ. The set of all $(n+1)$-by-$(n+1)$ symmetric positive definite matrices of unit 2-norm is the union $\bigcup_{l=1}^{\infty} \mathbb{S}_{1/l}$. For each $l \in \mathbb{N}$, we show the conclusion holds for all $\Theta \in \mathbb{S}_{1/l}$ except a set of Lebesgue measure zero.

Let $\Theta \in \mathbb{S}_{1/l}$ be an arbitrary matrix. By the selection of $u^{(1)}, \ldots, u^{(j)}$, it holds that

$$\nu_1 := \theta(u^{(1)}) < \nu_2 := \theta(u^{(2)}) < \cdots < \nu_j := \theta(u^{(j)}).$$

We consider the case $j > 1$ for convenience because the proof is almost the same for $j = 1$. When $\mathcal{K}_J$ has no other points except $u^{(1)}, \ldots, u^{(j)}$, we have $\theta_{\max} = \theta(u^{(j)})$ for all $\delta > 0$. So, we consider the opposite case and suppose $\bar{u}$ is a point in $\mathcal{K}_J$ that is different from $u^{(1)}, \ldots, u^{(j)}$, and that $\nu_j \le \theta(\bar{u})$. If x is a minimizer of (4.7) in previous loops, then

$$e_1^T \Theta e_1 + \|\bar{u}\|^2 \ge [\bar{u}]_1^T \Theta [\bar{u}]_1 \ge [x]_1^T \Theta [x]_1 \ge e_1^T \Theta e_1 + \|x\|^2/l,$$

with $e_1 = (1, 0, \ldots, 0)^T$. So all minimizers of (4.7) in previous loops are contained in the ball

$$B := \left\{ x \in \mathbb{R}^n : \|x\|^2 \le l \cdot \|\bar{u}\|^2 \right\},$$

which implies $u^{(k)} \in B$ for each k. Recall the notation $[x]_d$ as in (2.1). Since each $u^{(k)}$ is an isolated point of $\mathcal{K}_J$, the set

$$T_1 := \left\{ [x]_2 : \, x \in \mathcal{K}_J \cap B, x \ne u^{(k)}, 1 \le k \le j-1 \right\}$$

is compact. Since $\theta(x) = \langle \theta, [x]_2 \rangle$, we have

$$\langle \theta, [u^{(1)}]_2 \rangle < \cdots < \langle \theta, [u^{(j-1)}]_2 \rangle < \min_{y \in T_1} \langle \theta, y \rangle.$$

The right most minimization in the above is equivalent to

$$\left\{ \begin{array}{ll} \min & \langle \theta, y \rangle \\ s.t. & y \in \operatorname{conv}(T_1). \end{array} \right. \tag{4.9}$$

The convex hull $\operatorname{conv}(T_1)$ is a compact convex set. Observe that if (4.9) has more than one minimizer, then θ is a singular normal vector of the convex body $\operatorname{conv}(T_1)$. The set of singular normal vectors of a convex body has Lebesgue measure zero. This is shown in [40, Theorem 2.2.11]. So, when Θ is generic in $\mathbb{S}_{1/l}$, the linear optimization (4.9) has the unique minimizer $u^{(j)}$. Let

$$T_2 \coloneqq T_1 \setminus \{[u^{(j)}]_2\}.$$

Since $u^{(j)}$ is an isolated point of $\mathcal{K}_J$, the set T_2 is also compact, so

$$\langle \theta, [u^{(j)}]_2 \rangle < \min_{y \in T_2} \langle \theta, y \rangle.$$

Then there must exist $\delta > 0$ such that

$$\langle \theta, [u^{(j)}]_2 \rangle + \delta < \min_{y \in T_2} \langle \theta, y \rangle.$$

For the above δ, we must have $\theta_{\max} = \theta(u^{(j)})$. This means that the conclusion holds for all $\Theta \in \mathbb{S}_{1/l}$ except for a set of Lebesgue measure zero, for each $l \in \mathbb{N}$. This completes the proof. $\square$

When the cardinality $|\mathcal{K}_J| < \infty$, all points in $\mathcal{K}_J$ are isolated, so the following follows from Proposition 4.3.

Theorem 4.4. *Consider the GNEP as in (3.1). For the given J, if $|\mathcal{K}_J| < \infty$, then Algorithm 4.2 returns all GNEs contained in $\mathcal{K}_J$ or detects their nonexistence.*

We remark that when the GNEP is given by generic polynomials, the critical set $\mathcal{K}$ (hence its subset $\mathcal{K}_J$ for each $J \in \mathcal{P}$) is finite. This is shown in [29, Theorem 3.1]. So, for generic GNEPs as in (3.1), Algorithm 4.2 can find all GNEs in $\mathcal{K}_J$ or detect their nonexistence.

4.2. **Finding all GNEs.** For a given J, Algorithm 4.2 can compute all GNEs in $\mathcal{K}_J$ or detect their nonexistence. For the GNEP as in (3.1), the set $\mathcal{P}$ is finite. By enumerating $J \in \mathcal{P}$, we can get all GNEs or detect their nonexistence. This gives the following algorithm.

Algorithm 4.5. For the GNEP as in (3.1), formulate the label set $\mathcal{P}$. Let $\mathcal{S} := \emptyset$ For each $J \in \mathcal{P}$, do the following:

Step 1 Formulate the pLME λ_{i,J_i} as in (3.8) for each player i.
Step 2 Apply Algorithm 4.2 to find the set S_J of all GNEs in $\mathcal{K}_J$.
Step 3 Update $\mathcal{S} \coloneqq \mathcal{S} \cup \mathcal{S}_J$.

The following result follows from Theorem 4.4.

Theorem 4.6. *For the GNEP as in (3.1), assume that the critical set $\mathcal{K}$ is finite and Θ is a generic symmetric positive definite matrix. Then, after enumerating all $J \in \mathcal{P}$, Algorithm 4.5 finds all GNEs if $\mathcal{S} \neq \emptyset$, or detects nonexistence of GNEs if $\mathcal{S} = \emptyset$.*

5. Solving Polynomial Optimization

We now show how to solve polynomial optimization problems that appear in Algorithms 4.2 and 4.5. They can be generally expressed in the form:

$$
\left\{ \begin{array}{rcl} f_{min} := & \min\limits_{z} & f(z) \\ & s.t. & p(z) = 0 \ (\forall p \in \Phi), \\ & & q(z) \geq 0 \ (\forall q \in \Psi), \end{array} \right. \tag{5.1}
$$

where the variable z represents either $x \in \mathbb{R}^n$ or $x_i \in \mathbb{R}^{n_i}$ for the ith player, and Φ, Ψ are finite sets of equality and inequality constraining polynomials, respectively. The Moment-SOS relaxations are efficient for solving (5.1) globally. We refer to the books [16, 20, 21, 25] for a more detailed introduction.

Denote the degrees

$$
\begin{array}{rcl} d_0 & := & \max\left\{ \lceil \deg(p)/2 \rceil : p \in \Phi \cup \Psi \right\}, \\ d_1 & := & \max\left\{ \lceil \deg(f)/2 \rceil, d_0 \right\}. \end{array}
$$

Let ℓ be the length of z. For a degree $k \geq d_1$, the kth order moment relaxation for solving (5.1) is

$$
\left\{ \begin{array}{rcl} f_{mom,k} := & \min & \langle f, y \rangle \\ & s.t. & y_0 = 1, L_p^{(k)}[y] = 0 \, (p \in \Phi), \\ & & M_k[y] \succeq 0, \, L_q^{(k)}[y] \succeq 0 \, (q \in \Psi), \\ & & y \in \mathbb{R}^{\mathbb{N}^{\ell}_{2k}}. \end{array} \right. \tag{5.2}
$$

The dual optimization of (5.2) is the kth order SOS relaxation

$$
\left\{ \begin{array}{rcl} f_{sos,k} := & \max & \gamma \\ & s.t. & f - \gamma \in \text{Ideal}[\Phi]_{2k} + \text{QM}[\Psi]_{2k}. \end{array} \right. \tag{5.3}
$$

We refer to Section 2 for the notation $\langle f, y \rangle$, $L_p^{(k)}[y]$, $M_k[y]$, $\text{Ideal}[\Phi]_{2k}$, $\text{QM}[\Psi]_{2k}$ in the above. It is worthy to remark that (5.2)-(5.3) is a primal-dual pair of semidefinite programs. For $k = d_1, d_1 + 1, \ldots$, the primal-dual pair (5.2)-(5.3) is called the Moment-SOS hierarchy. Its convergence property can be summarized as follows. When $\text{Ideal}[\Phi] + \text{Qmod}[\Psi]$ is archimedean, we have $f_{mom,k} \to f_{min}$ as $k \to \infty$. Moreover, if the linear independence constraint qualification, strict complementarity condition, and second order sufficient optimality conditions hold at every minimizer, then $f_{sos,k} = f_{min}$ for all k that is big enough (see [27, 25]).

In the following, we show how to extract minimizers for (5.1) from the moment relaxation. Suppose $y^{(k)}$ is a minimizer of (5.2). If $y^{(k)}$ satisfies the flat truncation: there exists a degree $t \in [d_1, k]$ such that

$$
\operatorname{rank} M_t[y^{(k)}] \;=\; \operatorname{rank} M_{t-d_0}[y^{(k)}], \tag{5.4}
$$

then $f_{min} = f_{mom,k}$ and we can extract $r := \operatorname{rank} M_t[y^{(k)}]$ minimizers for (5.1) (see [17, 24, 25]). Indeed, flat truncation is a sufficient and almost necessary condition for extracting minimizers. This is shown in [24].

The Moment-SOS algorithm for solving (5.1) is as follows.

Algorithm 5.1. For the polynomial optimization (5.1), initialize $k := d_0$.

Step 1 Solve the moment relaxation (5.2). If it is infeasible, then (5.1) is infeasible and stop. Otherwise, solve it for a minimizer $y^{(k)}$.

Step 2 Check whether or not $y^{(k)}$ satisfies the rank condition (5.4). If (5.4) holds, then extract $r := \operatorname{rank} M_t[y^{(k)}]$ minimizers of (5.1) and stop. Otherwise, let $k := k+1$ and go to Step 1.

Algorithm 5.1 can be implemented in the software `GloptiPoly3` [18], which calls SDP package like `MOSEK` [1]. For Algorithms 4.2 and 4.5, the optimization problem (5.1) is one of (4.1), (4.2), (4.7), or (4.8). We have the following remarks:

- For the minimization problem (4.1) and (4.7), we have $z := x$ and $f(x) := \theta(x)$, where $\theta(x)$ is defined by the generically selected positive definite matrix Θ. So, if they are feasible, then they have a unique optimizer. Moreover, equality constraints of both (4.1) and (4.7) define finite real varieties when the polynomials for the GNEP have generic coefficients (see [29]). In these cases, flat truncation (5.4) holds with $r = 1$ for all k that is big enough [26].
- For the polynomial optimization problem (4.2) of verifying GNEs, we have $z := x_i$ and $f(x_i) := f_i(x_i, u_{-i}) - f_i(u_i, u_{-i})$. This problem must be feasible, as u_i is a feasible point. If $f_{mom,k} \geq 0$, we can terminate Algorithm 5.1 directly since we don't need to extract minimizers for this case.
- For the maximization problem (4.8), we have $z := x$ and $f(x) := -\theta(x)$. It is always feasible since $u^{(j)}$ is a feasible point. Furthermore, when the GNEP is given by generic polynomials, equality constraints of (4.8) give a finite real variety, so flat truncation (5.4) holds for all k that is big enough.

Recall that e_i represents the vector of all zeros except that ith entry is 1. The notation $y^{(k)}_{e_i}$ denotes the entry of $y^{(k)}$ labeled by e_i. Denote the vector

$$u^{(k)} := (y^{(k)}_{e_1}, y^{(k)}_{e_2}, \ldots, y^{(k)}_{e_n}). \tag{5.5}$$

The following is the convergence property of Algorithm 5.1 when it is applied to solve polynomial optimization problems (4.1), (4.7), or (4.8). They are shown in [30, 31].

Theorem 5.2. *Suppose the optimization problem (5.1) is (4.1), (4.7), or (4.8). Assume Θ is a generic symmetric positive definite matrix and the real variety of Φ is a finite set. Then, we have:*

(i) If (5.1) is infeasible, then the moment relaxation (5.2) must be infeasible when the order k is large enough.

(ii) Suppose (5.1) is feasible. Then $f_{mom,k} = f_{min}$ and the flat truncation holds for all k that is big enough. Furthermore, if the optimization problem (5.1) is (4.1) or (4.7), then $u^{(k)}$ is the unique minimizer of (5.1), when the order k is large enough.

6. Numerical experiments

This section presents the numerical experiments of GNEPs with quasi-linear constraints. Algorithms 4.5 is applied to solve GNEPs. In computations, involved polynomial optimization problems are solved globally with Algorithm 5.1, using the Matlab software `GloptiPoly3` [18]. Additionally, semidefinite programs are solved using the `MOSEK` solver [1] with `Yalmip` [23]. The computations were implemented using Matlab R2023b on a laptop equipped with a 12th Gen Intel(R) Core(TM) i7-1270P 2.20GHz CPU and 32GB RAM. To enhance readability, the computational results are reported with four decimal places. For convenience of expression, the constraints are ordered from left to right and from top to bottom in each problem.

Example 6.1. We use Algorithm 4.5 to solve some GNEPs from existing references. These problems are given explicitly in the Appendix, each with a given citation name. We report our numerical results in Table 2. The notation $\#x^*$ stands for the number of computed GNEs.

Problem	$\#x^*$	All GNEs $x^* = (x_1^*, x_2^*, \dots, x_N^*)$
FKA3	3	$(-0.3805, -0.1227, -0.9932),\ (0.3903, 1.1638),\ (0.0504, 0.0176)$; $(-0.8039, -0.3062, -2.3541),\ (0.9701, 3.1228),\ (0.0751, -0.1281)$; $(1.9630, -1.3944, 5.1888),\ (-3.1329, -10.0000),\ (-0.0398, 1.6392)$
FKA4	1	$(1.0000, 1.0000, 1.0000), (1.0000, 1.0000), (1.0000, 1.0000)$
FKA5	1	$(0.0000, 0.2029, 0.0000), 0.0000, 0.0725), (0.0254, 0.0000)$
FKA8	2	$(0.3333), (0.5000), (0.6667)$ and $(5.3333), (5.3333), (0.6667)$
FKA12	1	$(5.3333), (5.3333)$
NT59	1	$(0.7000, 0.1600), (0.8000, 0.1600), (0.8000, 0.4700)$
NT510	1	$(1.7184), (1.8413, 0.6700), (1.2000, 0.0823, 0.0823)$
NTGS53	2	$(0.0000, 0.5000), (0.5000, 0.0000)$; $(0.0000, 0.5000), (0.0000, 0.5000)$
NTGS54	1	$(0.1000, 0.4000), (0.1000, 0.4000)$
FR33	4	$(0.0000, 2.0000), (0.0000, 6.0000)$; $(0.0000, 0.0000), (0.0000, 0.0000)$; $(1.1876, 1.9062), (1.2481, 0.0000)$; $(1.0000, 2.0000), (1.0000, 2.0000)$
SAG41	1	$(0.5588, 0.5588), (0.2647, 0.2647)$
DSM31	1	$(0.0000, 0.0000), (0.0000, 0.0000)$

TABLE 2. Numerical results for GNEPs in Appendix.

Example 6.2. Consider the 2-player convex GNEP:

$$
\mathrm{F}_1(x_{-1}) : \left\{ \begin{array}{cl} \min\limits_{x_1 \in \mathbb{R}^4} & (x_{1,1}-1)^2 + x_{2,4}(x_{1,2}-1)^2 + (x_{1,3}-1)^2 \\ & \qquad\qquad +(x_{1,4}-2)^2 + (\mathbf{1}^T x_2 - 1)\mathbf{1}^T x_1 \\ s.t. & 0 \le x_1 \le x_2,\ x_{2,3}(x_{2,3}-1)(x_{2,3}-3) \ge x_{1,4}, \end{array} \right.
$$

$$
\mathrm{F}_2(x_{-2}) : \left\{ \begin{array}{cl} \min\limits_{x_2 \in \mathbb{R}^4} & x_{1,1}x_{2,1}^2 - x_{2,2} + x_{1,3}(x_{2,3}-1)^2 + x_{1,4}(x_{2,4}+1)^2 \\ s.t. & x_{2,1} - x_{2,2} - x_{1,2} \ge 0,\ 2x_{1,1} - x_{2,1} + x_{2,2} \ge 0, \\ & x_{2,1} + x_{2,2} + x_{1,1} + x_{1,2} \ge 0, \\ & 4x_{1,1} - 2x_{1,2} - x_{2,1} - x_{2,2} \ge 0,\ x_{2,3} \ge 0, \\ & x_{1,3}(3x_{1,3}-1)(x_{1,3}-1) \ge 3x_{2,4},\ 3 \ge x_{2,3} + x_{2,4}. \end{array} \right.
$$

There are a total of 288 $J \in \mathcal{P}$. It took around 383.28 seconds to find all GNEs by Algorithm 4.5 and the computational time for each $\mathcal{K}_J$ is between 0.01-59.81 seconds. The first GNE was detected within 15.71 seconds. We found 6 GNEs from

16 $\mathcal{K}_J$'s in total, which are

$x_1^* = (0.3333, 0.0000, 0.3333, 0.0000)$,	$x_2^* = (0.6667, 0.6667, 1.0000, 0.0000)$;
$x_1^* = (0.0000, 0.0000, 0.0000, 0.0000)$,	$x_2^* = (0.0000, 0.0000, 3.0000, 0.0000)$;
$x_1^* = (0.5000, 0.0000, 0.0000, 0.0000)$,	$x_2^* = (1.0000, 1.0000, 0.0000, 0.0000)$;
$x_1^* = (0.0000, 0.0000, 1.0000, 0.0000)$,	$x_2^* = (0.0000, 0.0000, 1.0000, 0.0000)$;
$x_1^* = (0.7071, 0.0000, 0.0000, 0.0000)$,	$x_2^* = (0.7071, 0.7071, 0.0000, 0.0000)$;
$x_1^* = (0.0000, 0.0000, 0.0000, 0.0000)$,	$x_2^* = (0.0000, 0.0000, 0.0000, 0.0000)$.

In particular, we found 5 GNEs in $\mathcal{K}_J$ for

$$J = \big(\{2,5,7,8\},\, \{1,4,5,7\}\big) \text{ or } \big(\{2,5,7,9\},\, \{1,4,5,7\}\big).$$

Example 6.3. Consider the 2-player nonconvex NEP

$$\mathrm{F}_i(x_{-i}) : \begin{cases} \min\limits_{x_i \in \mathbb{R}^{n_i}} & f_i(x_i, x_{-i}) \\ s.t. & A_i x_i \ge b_i, \end{cases}$$

where $n_1 = 7$, $n_2 = 5$, and

$$f_1(x) = 3x_{1,1}^2 + 4x_{1,2}^2 + 4x_{1,2}x_{2,1} + 3x_{1,4}x_{2,4} + 4x_{1,6}x_{2,4},$$
$$f_2(x) = x_{1,2}x_{2,2} + 3x_{1,5}x_{2,4} + x_{1,6}x_{2,2} + x_{2,1}^2 + 2x_{2,1}x_{2,2} + x_{2,3}^2,$$

$$A_1 = \begin{bmatrix} 0 & -3 & 0 & 2 & 3 & 1 & 3 \\ 2 & -1 & 2 & -2 & 1 & 1 & -2 \\ -1 & -1 & 0 & 2 & 2 & 1 & -3 \\ 1 & 1 & 0 & 1 & 0 & -1 & 2 \\ 1 & 2 & 0 & 2 & -3 & -2 & -2 \\ -1 & 0 & -2 & 3 & 1 & -1 & -3 \\ 0 & -1 & -3 & -2 & -2 & -3 & 2 \\ -3 & 2 & 0 & 1 & -3 & -2 & -3 \\ 1 & 1 & 1 & 2 & 3 & 0 & 1 \end{bmatrix}, \; b_1 = \begin{bmatrix} 1 \\ 5 \\ 4 \\ 2 \\ 2 \\ 2 \\ 2 \\ 1 \\ -1 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 2 & -3 & -1 & -1 & -1 \\ -3 & 4 & 3 & 2 & -3 \\ 1 & 2 & 1 & 0 & 2 \\ 2 & -3 & 2 & 3 & -1 \\ -3 & 1 & 2 & 2 & 2 \\ 2 & 1 & -2 & -3 & 4 \\ 0 & 2 & 3 & 1 & 2 \end{bmatrix}, \; b_2 = \begin{bmatrix} 1 \\ 3 \\ 1 \\ 1 \\ 3 \\ 0 \\ -1 \end{bmatrix}.$$

There are a total of 756 $J \in \mathcal{P}$. It took around 1774.40 seconds to find all NEs by Algorithm 4.5 and the computational time for each $\mathcal{K}_J$ is between 1.29-12.51 seconds. The first NE was detected within 151.88 seconds, which is

$$x_1^* = (1.7344, -1.2108, 0.8670, 0.9041, 0.9669, -3.2800, -1.3538),$$
$$x_2^* = (-0.4706, -1.0941, 4.4392, -3.3294, 0.2314).$$

This is the unique NE. It is contained in $\mathcal{K}_J$ for

$$J = \big(\{1,2,3,4,5,7,8\},\, \{1,2,4,5,6\}\big), \quad \big(\{1,2,3,4,6,7,8\},\, \{1,2,4,5,6\}\big),$$
$$\text{or} \quad \big(\{1,2,3,4,7,8,9\},\, \{1,2,4,5,6\}\big).$$

In addition, x^* is the unique KKT point for this problem.

Example 6.4. Consider the 2-player nonconvex GNEP

$$\mathrm{F}_i(x_{-i}): \begin{cases} \min\limits_{x_i \in \mathbb{R}^{n_i}} & (-1)^i \|x_1 + \mathbf{1}\|^2 + (-1)^{i+1}\|x_2 + \mathbf{1}\|^2 \\ s.t. & \alpha_{i,1}^T x_1 + \beta_{i,1}^T x_2 + \gamma_{i,1} \geq 0, \\ & \alpha_{i,2}^T x_1 + \beta_{i,2}^T x_2 + \gamma_{i,2} \geq 0, \\ & 1 \geq \mathbf{1}^T x_i,\ x_i \geq 0, \end{cases}$$

where $n_1 = 4$, $n_2 = 2$, and

$$\alpha_{1,1} = \begin{bmatrix} -1 \\ -3 \\ 4 \\ 2 \end{bmatrix},\ \alpha_{1,2} = \begin{bmatrix} -5 \\ 0 \\ -1 \\ 0 \end{bmatrix},\ \alpha_{2,1} = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix},\ \alpha_{2,2} = \begin{bmatrix} 0 \\ -1 \\ -1 \\ 0 \end{bmatrix},$$

$$\beta_{1,1} = \begin{bmatrix} 1 \\ -1 \end{bmatrix},\ \beta_{1,2} = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ \beta_{2,1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ \beta_{2,2} = \begin{bmatrix} -5 \\ 5 \end{bmatrix},$$

$$\gamma_{1,1} = -2,\ \gamma_{1,2} = 1,\ \gamma_{2,1} = 1,\ \gamma_{2,2} = -1.$$

There are a total of 279 $J \in \mathcal{P}$. It took around 37.60 seconds to find all GNEs by Algorithm 4.5 and the computational time for each $\mathcal{K}_J$ is between 0.08-2.18 seconds. The first GNE was detected within 2.84 seconds, which is

$$x_1^* = (0.0000, 0.0000, 1.0000, 0.0000), \quad x_2^* = (0.0000, 1.0000).$$

This is the unique GNE. It is contained in $\mathcal{K}_J$ for

$$\begin{aligned} J = & \big(\{1,2,3,4\}, \{3,4\}\big), \quad \big(\{2,3,4,5\}, \{3,4\}\big), \\ & \big(\{2,3,4,7\}, \{3,4\}\big), \quad \text{or} \quad \big(\{3,4,5,7\}, \{3,4\}\big). \end{aligned}$$

We remark that this problem only has two KKT points.

Example 6.5. Consider the 2-player nonconvex GNEP

$$\mathrm{F}_1(x_{-1}): \begin{cases} \min\limits_{x_1 \in \mathbb{R}^2} & x_{1,1}x_{2,1}^3 + x_{1,2}x_{2,2}^3 - x_{1,1}^2 x_{1,2}^2 \\ s.t. & A_1 x_1 \geq B_1 \begin{bmatrix} 1 \\ x_2 \end{bmatrix} + C_1 \begin{bmatrix} x_{2,1}^2 \\ x_{2,1}x_{2,2} \\ x_{2,2}^2 \end{bmatrix}, \end{cases}$$

$$\mathrm{F}_2(x_{-2}): \begin{cases} \min\limits_{x_2 \in \mathbb{R}^2} & x_{2,2}\|x_2\|^2 - 2x_{1,2}x_{2,1} - x_{1,1}x_{1,2}x_{2,2} \\ s.t. & A_2 x_2 \geq B_2 \begin{bmatrix} 1 \\ x_1 \end{bmatrix} + C_2 \begin{bmatrix} x_{1,1}^2 \\ x_{1,1}x_{1,2} \\ x_{1,2}^2 \end{bmatrix}, \end{cases}$$

where

$$A_1 = \begin{bmatrix} 1 & 0 \\ 0 & -2 \\ 3 & -1 \\ -4 & 3 \\ -6 & -5 \\ 0 & -5 \end{bmatrix},\ B_1 = \begin{bmatrix} 0 & 0 & 0 \\ -1 & -1 & 0 \\ -2 & -1 & -1 \\ -3 & -2 & -3 \\ -1 & -1 & -2 \\ 0 & 1 & -1 \end{bmatrix},\ C_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} -2 & 0 \\ -4 & 4 \\ -2 & 7 \\ -1 & 4 \\ -3 & 4 \\ 2 & 1 \end{bmatrix},\ B_2 = \begin{bmatrix} 0 & -1 & -1 \\ -6 & 0 & -1 \\ 4 & -3 & -3 \\ -4 & 1 & -3 \\ 3 & 0 & -1 \\ -1 & 0 & 0 \end{bmatrix},\ C_2 = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

There are a total of 210 $J \in \mathcal{P}$. It took around 9.45 seconds to find all GNEs by Algorithm 4.5 and the computational time for each $\mathcal{K}_J$ is between 0.01–0.82 second. The first GNE was detected within 3.69 seconds, which is

$$x_1^* = (0.4447, -0.3256), \quad x_2^* = (-0.6094, 0.3249).$$

It is contained in $\mathcal{K}_J$ for $J = (\{1,4\},\{3,6\})$ or $(\{4,5\},\{3,6\})$. We found 2 GNEs in total. The other GNE is

$$x_1^* = (0.3612, -0.8078), \quad x_2^* = (-0.4776, 0.6078).$$

It is contained in $\mathcal{K}_J$ for $J = (\{4,5\},\{5,6\})$. We remark these GNEs are also the only KKT points for this problem.

Example 6.6. Consider the 2-player nonconvex GNEP

$$\mathrm{F}_i(x_{-i}) : \begin{cases} \min\limits_{x_i \in \mathbb{R}^3} & f_i(x) \\ s.t. & A_i x_i \ge B_i[x_{-i}]_1 + d_i(x_{-i}), \end{cases}$$

where

$$f_1(x) = x_{1,1}(x_{1,1} - 2x_{2,1}) + x_{1,2} \cdot \mathbf{1}^T x_2,$$
$$f_2(x) = x_{2,1}(2 + 2x_{2,2}) + x_{2,3} \cdot \mathbf{1}^T x_1,$$

$$A_1 = \begin{bmatrix} -4 & -1 & -2 \\ 0 & 3 & 4 \\ -1 & 5 & -3 \\ 5 & -1 & -3 \\ 5 & -4 & 0 \\ 0 & 4 & -5 \\ -3 & 4 & -5 \end{bmatrix},\ B_1 = \begin{bmatrix} 0 & -5 & -4 & -2 \\ -1 & 3 & 6 & -1 \\ -6 & 0 & 2 & 0 \\ -5 & 2 & 3 & 0 \\ -5 & 3 & 0 & 5 \\ 0 & 0 & -1 & 3 \\ 3 & -1 & 0 & 0 \end{bmatrix},\ d_1 = \begin{bmatrix} x_{2,3}^2 \\ x_{2,2}^2 \\ 0 \\ -x_{2,2}^2 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} -5 & 2 & -1 \\ 2 & 3 & 2 \\ 1 & -1 & 3 \\ 5 & 0 & -2 \\ -3 & -4 & 1 \\ 5 & 4 & -1 \\ -3 & 4 & 0 \end{bmatrix},\ B_2 = \begin{bmatrix} -5 & 6 & 4 & 0 \\ -4 & -2 & 4 & 5 \\ -4 & 3 & 6 & -1 \\ 0 & 6 & -1 & 4 \\ -4 & -1 & -3 & 3 \\ -1 & 3 & -4 & -2 \\ -1 & 2 & 0 & 0 \end{bmatrix},\ d_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ x_{1,3}^2 \\ 0 \\ -x_{1,2}^2 \\ 0 \end{bmatrix}.$$

There are a total of 1225 $J \in \mathcal{P}$. Five of them contain GNEs. It took around 192.54 seconds to find all GNEs by Algorithm 4.5 and the computational time for each $\mathcal{K}_J$ is between 0.10–2.40 seconds. The first GNE was detected within 116.59 seconds. We found 2 GNEs from 5 $\mathcal{K}_J$'s in total, which are

$$x_1^* = (0.2075, 0.7518, -0.0779), \quad x_2^* = (0.2258, 0.4260, 0.4706);$$
$$x_1^* = (-0.3079, 0.7901, 0.1566), \quad x_2^* = (-0.3011, 0.7604, 0.2406).$$

We remark that these GNEs are also the only KKT points for this problem.

Example 6.7. Consider the convex GNEP with the ith player's optimization

$$\mathrm{F}_i(x_{-i}) : \left\{ \begin{array}{cl} \min\limits_{x_i \in \mathbb{R}^{n_i}} & f_i(x_i, x_{-i}) \\ s.t. & A_i x_i \geq b_i(x_{-i}) := b_i - B_i x_{-i}, \end{array} \right. \tag{6.1}$$

where the vector $b_i(x_{-i})$ has length m_i and the objective f_i is in the form

$$f_i(x_i, x_{-i}) \;=\; c_i^T x + x^T G_i x + (x^{[2]})^T H_i x^{[2]}.$$

In the above, $c_i \in \mathbb{R}^n$, $G_i \in \mathbb{R}^{n \times n}$ is symmetric positive semidefinite and $H_i \in \mathbb{R}^{n \times n}$ is symmetric positive semidefinite with nonnegative entries. For the above choice, f_i is a convex polynomial function in x (see [25, Example 7.1.4]). We use the MATLAB function `unifrnd` to generate random matrices A_i, d_i, and B_i for GNEPs of different sizes. We generate the convex polynomial f_i randomly as $c_i = \texttt{randn}(n, 1)$, $G_i = R_1^T R_1$ with $R_1 = \texttt{randn}(n)$ and $H_i = R_2^T R_2$ with $R_2 = \texttt{rand}(n)$. We randomly generate 10 instances for each case and apply Algorithm 4.5. The computational results are reported in Table 3. For each instance, $\#(\mathcal{K}_J \neq \emptyset)$ counts the number of $\mathcal{K}_J$ that contains at least one KKT point and #GNEs counts the number of all GNEs. The "Avg. Time of Alg. 4.2 for a single $\mathcal{K}_J$" gives the average time (in seconds) taken by Algorithm 4.2.

N	$(n_1, \ldots, n_N)$ $(m_1, \ldots, m_N)$	$\lvert\mathcal{P}\rvert$	$\#(\mathcal{K}_J \neq \emptyset)$	#GNEs	Avg. Time of Alg. 4.2 for a single $\mathcal{K}_J$
2	(2, 2) (4, 4)	36	36, 36, 36, 36, 36, 36, 36, 36, 36, 36	1, 1, 1, 1, 3, 1, 1, 1, 3, 1	0.9, 0.8, 0.7, 0.8, 1.2, 0.8, 0.9, 1.5, 0.9, 0.9
	(3, 3) (5, 5)	100	100, 100, 100, 100, 100, 100, 100, 100, 100, 100	1, 1, 1, 2, 3, 1, 1, 1, 2, 2	3.2, 3.2, 3.6, 4.3, 23.7, 5.4, 6.0, 6.3, 6.6, 7.4
	(3, 4) (6, 6)	300	300, 300, 300, 300, 300, 300, 300, 300, 300, 283	1, 1, 1, 1, 1, 1, 1, 1, 1, 1	11.8, 14.8, 14.7, 16.9, 20.0, 21.7, 12.6, 14.5, 12.6, 8.0
	(3, 3) (6, 6)	400	400, 400, 400, 400, 400, 400, 400, 380, 400, 396	3, 1, 2, 1, 1, 1, 2, 1, 1, 1	7.5, 8.7, 12.0, 8.7, 9.2 10.1, 13.6, 11.6, 12.9, 13.4
3	(2, 3, 3) (4, 4, 4)	96	73, 96, 72, 96, 96, 72, 96, 96, 96, 96	3, 3, 2, 1, 1, 2, 1, 1, 1, 1	25.9, 31.2, 22.9, 23.0, 22.8, 22.9, 30.6, 25.7, 15.7, 26.4
	(2, 2, 3) (4, 4, 4)	144	141, 144, 144, 144, 144, 144, 144, 144, 144, 144	1, 2, 2, 1, 1, 2, 2, 1, 1, 1	18.2, 21.8, 21.0, 20.8, 23.3, 38.8, 38.9, 43.0, 31.5, 7.5
	(2, 2, 2) (4, 4, 4)	216	216, 216, 216, 216, 216, 216, 109, 113, 216, 216	2, 2, 1, 1, 2, 2, 2, 2, 1, 1	4.5, 11.7, 17.2, 16.2, 10.7, 7.7, 5.7, 5.0, 9.0, 9.3
	(2, 2, 3) (4, 4, 5)	360	360, 334, 360, 360, 360 360, 360, 360, 360, 326	2, 3, 2, 2, 1 1, 1, 2, 1, 1	16.6, 14.9, 19.6, 19.6, 20.7 26.8, 24.1, 24.4, 17.8, 25.5
4	(1, 1, 1, 1) (3, 3, 3, 3)	81	81, 81, 81, 81, 81, 81, 81, 81, 81, 81	1, 5, 1, 3, 1 3, 1, 1, 1, 1	2.5, 5.8, 1.9, 9.8, 2.4, 2.7, 3.8, 2.7, 2.5, 3.3
	(1, 2, 2, 2) (4, 4, 4, 4)	864	863, 864, 841, 864, 864, 864, 432, 864, 864, 864	2, 1, 1, 1, 1, 3, 1, 1, 1, 1	9.3, 12.4, 20.0, 25.5, 11.1, 13.0, 18.6, 22.8, 23.2, 17.6
5	(1, 1, 1, 1, 1) (3, 3, 3, 3, 3)	243	81, 243, 243, 243, 243, 243, 243, 243, 243, 243	1, 3, 1, 3, 3, 3, 3, 1, 3, 1	1.2, 2.6, 2.9, 2.9, 3.3, 3.7, 4.3, 5.5, 7.2, 2.5
	(1, 1, 1, 1, 1) (3, 4, 4, 4, 4)	768	195, 768, 768, 768, 768, 768, 768, 768, 768, 192	2, 3, 3, 1, 1, 3, 1, 1, 4, 1	1.3, 4.1, 15.4, 7.6, 8.9, 10.2, 12.1, 16.6, 94.4, 10.1

TABLE 3. Computational results for randomly generated convex GNEPs as in (6.1).

Example 6.8. Consider the 2-player nonconvex GNEP

$$\begin{array}{cl|cl} \min\limits_{x_1 \in \mathbb{R}^4} & 3\|x_1\|^2 + x_{1,2} \cdot \mathbf{1}^T x_1 & \min\limits_{x_2 \in \mathbb{R}^4} & -\|x_2\|^2 - x_{2,3} \cdot \mathbf{1}^T x_2 \\ s.t. & A_1 x_1 \geq b_1(x_2), & s.t. & A_2 x_2 \geq b_2(x_1), \end{array}$$

where

$$A_1 = \begin{bmatrix} 0 & 1 & 1 & 3 \\ -1 & -2 & 3 & 0 \\ 1 & -1 & 1 & 0 \\ 0 & -1 & 2 & 2 \\ -1 & -1 & 1 & 2 \end{bmatrix}, \ A_2 = \begin{bmatrix} 0 & 1 & 1 & 3 \\ -2 & 2 & -1 & 1 \\ 0 & -2 & 1 & -2 \\ 0 & -1 & 0 & 2 \\ 2 & 1 & 0 & -2 \end{bmatrix},$$

$$b_1(x_2) = \begin{bmatrix} 5 - \|x_2\|^2 \\ \|x_2\|^2 \\ 3x_{2,1}x_{2,2} + 2x_{2,3}x_{2,4} - x_{2,2} \\ 3 + x_{2,1}x_{2,3} \\ 2 + x_{2,2}x_{2,4} \end{bmatrix},$$

$$b_2(x_1) = \begin{bmatrix} 5 - \|x_1\|^2 \\ \|x_1\|^2 \\ 2 - 2x_{1,3} + x_{1,1}x_{1,2} + x_{1,3}x_{1,4} \\ -1 + x_{1,1}x_{1,3} \\ -2 + x_{1,2}x_{1,4} \end{bmatrix}.$$

There are a total of 25 $J \in \mathcal{P}$. It took around 16.05 seconds to find all GNEs by Algorithm 4.5 and the computational time for each $\mathcal{K}_J$ is between $0.13 - 2.80$ seconds. The first GNE was detected within 4.91 seconds, which is

$$x_1^* = (-0.4085, -1.1070, 1.9636, 0.1845), \ x_2^* = (-2.0032, 1.4764, 1.5146, -0.1629).$$

This is the unique GNE. It is contained in $\mathcal{K}_J$ for

$$\begin{aligned} J = & \big(\{1,2,3,4\}, \{2,3,4,5\}\big), \quad \big(\{1,2,3,5\}, \{2,3,4,5\}\big), \\ & \big(\{1,2,4,5\}, \{2,3,4,5\}\big), \quad \text{or} \quad \big(\{2,3,4,5\}, \{2,3,4,5\}\big). \end{aligned}$$

Interestingly, the (x_1^*, x_2^*) is also the unique KKT point.

We also implemented the homotopy method in [22] for finding GNEs of this GNEP. The mixed-volume for the complex KKT system

$$\left\{ \begin{array}{l} \nabla_{x_i} f_i(x) - A_i^T \lambda_i = 0 \ (i = 1, 2), \\ \lambda_i \perp (A_i x_i - b_i(x_{-i})) \ (i = 1, 2) \end{array} \right. \tag{6.2}$$

is 24611. The polyhedral homotopy continuation is implemented in the Julia software `homotopycontinuation.jl` [4], which found 17100 complex roots to (6.2), and 1860 of them are real. After checking the feasibility and the nonnegativity of Lagrange multipliers for each real root, we got the same KKT point (x_1^*, x_2^*), which is verified to be a GNE by solving (4.2). It took around 210.88 seconds for the polyhedral homotopy to solve the complex KKT system and 1.50 seconds to verify the GNE. We also remark that the number of computed complex roots is smaller than the mixed-volume of (6.2), so the homotopy method cannot guarantee the computed ones are the all complex solutions to (6.2). For this reason, it cannot certify uniqueness of the GNE.

Moreover, we tested the Augmented Lagrangian method in [19] and the interior point method in [6] for finding GNEs. The Augmented Lagrangian method cannot find a GNE after 1000 outer iterations because the augmented Lagrangian subproblem cannot be solved accurately. Also, the interior point method failed to find a GNE within 1000 iterations since the Newton directions are usually not descent directions.

7. Conclusion and Discussion

This paper studies GNEPs with quasi-linear constraints and defined by polynomials. We propose a new partial Lagrange multiplier expression approach with KKT conditions. By using partial Lagrange multiplier expressions, we represent KKT sets of such GNEPs by a union of simpler sets with convenient expressions. This helps to relax GNEPs into finite groups of branch polynomial optimization problems. The latter can be solved efficiently by Moment-SOS relaxations. Under some genericity assumptions, we develop algorithms that either find all GNEs or detect their nonexistence. Numerical experiments are given to show the efficiency of our method. There is great potential for our method. It can be interesting future work to apply our method for solving GNEPs arising from machine learning and data science applications.

We remark that GNEPs with quasi-linear constraints are typically more difficult than GNEPs with linear constraints. To see this, we compare their algebraic degrees, which count numbers of complex solutions to KKT systems [29]. For the convenience of our discussion, we suppose that for each $i \in [N]$, $f_i(x)$ is a quadratic polynomial in x, A_i is a m_i-by-n_i matrix, and every $b_{i,j}$ ($j \in [m_i]$) is a polynomial in x_{-i} whose degree equals $d_{i,j}$. Without loss of generality, we also assume that $m_i \le n_i$ for each $i \in [N]$, and all constraints are active at every KKT point (otherwise, one may compute the algebraic degree by enumerating all active sets, see [29, Theorem 5.2]). When all $f_i(x)$, A_i and $b_i(x_{-i})$ are generic, the algebraic degree is

$$\prod_{i=1}^{N}\prod_{j=1}^{m_i} \max(1, d_{i,j}).$$

In particular, when the GNEP has only linear constraints (i.e., $d_{i,j} \le 1$ for all i, j), the algebraic degree is equal to 1, which is much less than that for general cases of quasi-linear constraints (i.e., $d_{i,j}$ are greater than 1). For instance, when $N = 2$, $m_1 = m_2 = 4$, and $d_{i,j} = 2$ for all i, j, the algebraic degree for the GNEP with quasi-linear constraints is $2^8 = 256$, under some genericity assumptions. We refer to [29] for more details about algebraic degrees.

Acknowledgements. This project was begun at a SQuaRE at the American Institute for Mathematics. The authors thank AIM for providing a supportive and mathematically rich environment. Jiawang Nie is partially supported by the NSF grant DMS-2110780. Xindong Tang is partially supported by the Hong Kong Research Grants Council HKBU-15303423.

Appendix A.

Example A.1. (FKA3 [12]). Consider the 3-player GNEP

$$\mathrm{F}_i(x_{-i}) : \left\{ \begin{array}{cl} \min\limits_{x_i \in \mathbb{R}^{n_i}} & \frac{1}{2} x_i^T C_i x_i + x_i^T (D_i x_{-i} + t_i) \\ s.t. & A_i x_i \ge b_i(x_{-i}), \end{array} \right. \tag{A.1}$$

where $n_1 = 3$, $n_2 = n_3 = 2$ and

$$C_1 = \begin{bmatrix} 20 & 5 & 3 \\ 5 & 5 & -5 \\ 3 & -5 & 15 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 11 & -1 \\ -1 & 9 \end{bmatrix}, \quad C_3 = \begin{bmatrix} 48 & 39 \\ 39 & 53 \end{bmatrix},$$

$$D_1 = \begin{bmatrix} -6 & 10 & 11 & 20 \\ 10 & -4 & -17 & 9 \\ 15 & 8 & -22 & 21 \end{bmatrix}, \quad D_2 = \begin{bmatrix} 20 & 1 & -3 & 12 & 1 \\ 10 & -4 & 8 & 16 & 21 \end{bmatrix},$$

$$D_3 = \begin{bmatrix} 10 & -2 & 22 & 12 & 16 \\ 9 & 19 & 21 & -4 & 20 \end{bmatrix}, \quad t_1 = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}, \quad t_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad t_3 = \begin{bmatrix} -1 \\ 2 \end{bmatrix},$$

The following are constraints for each player.

$$\begin{aligned} &\text{1st player}: \begin{cases} -10 \cdot \mathbf{1} \le x_1 \le 10 \cdot \mathbf{1},\ x_{1,1} + x_{1,2} + x_{1,3} \le 20, \\ x_{1,1} + x_{1,2} - x_{1,3} \le x_{2,1} - x_{3,2} + 5, \end{cases} \\ &\text{2nd player}: -10 \cdot \mathbf{1} \le x_2 \le 10 \cdot \mathbf{1},\ x_{2,1} - x_{2,2} \le x_{1,2} + x_{1,3} - x_{3,1} + 7, \\ &\text{3rd player}: -10 \cdot \mathbf{1} \le x_3 \le 10 \cdot \mathbf{1},\ x_{3,2} \le x_{1,1} + x_{1,3} - x_{2,1} + 4. \end{aligned}$$

Example A.2. (FKA4 [12]). Consider (A.1) with $N = 3$, $n_1 = 3$, $n_2 = n_3 = 2$,

$$C_1 = \begin{bmatrix} 20 + x_{2,1}^2 & 5 & 3 \\ 5 & 5 + x_{2,2}^2 & -5 \\ 3 & -5 & 15 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 11 + x_{3,1}^2 & -1 \\ -1 & 9 \end{bmatrix},$$

$$C_3 = \begin{bmatrix} 48 & 39 \\ 39 & 53 + x_{1,1}^2 \end{bmatrix},$$

and D_i and t_i are the same as Example A.1. The following are constraints for each player.

$$\begin{aligned} &\text{1st player}: \begin{cases} \mathbf{1} \le x_1 \le 10 \cdot \mathbf{1},\ x_{1,1} + x_{1,2} + x_{1,3} \le 20, \\ x_{1,1} + x_{1,2} - x_{1,3} \le x_{2,1} - x_{3,2} + 5, \end{cases} \\ &\text{2nd player}: \mathbf{1} \le x_2 \le 10 \cdot \mathbf{1},\ x_{2,1} - x_{2,2} \le x_{1,2} + x_{1,3} - x_{3,1} + 7, \\ &\text{3rd player}: \mathbf{1} \le x_3 \le 10 \cdot \mathbf{1},\ x_{3,2} \le x_{1,1} + x_{1,3} - x_{2,1} + 4. \end{aligned}$$

Example A.3. (FKA5 [12]). Consider (A.1) with $N = 3$, $n_1 = 3$, $n_2 = n_3 = 2$,

$$C_1 = \begin{bmatrix} 20 & 6 & 0 \\ 6 & 6 & -1 \\ 0 & -1 & 8 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 11 & 1 \\ 1 & 7 \end{bmatrix}, \quad C_3 = \begin{bmatrix} 28 & 14 \\ 14 & 29 \end{bmatrix},$$

$$D_1 = \begin{bmatrix} -1 & -2 & -4 & -3 \\ 0 & -3 & 0 & -4 \\ 0 & 1 & 9 & 6 \end{bmatrix}, \quad D_2 = \begin{bmatrix} -1 & 0 & 0 & -7 & 4 \\ -2 & -3 & 1 & 4 & 11 \end{bmatrix},$$

$$D_3 = \begin{bmatrix} -4 & 0 & 9 & -7 & 4 \\ -3 & -4 & 6 & 4 & 11 \end{bmatrix}, \quad t_1 = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}, \quad t_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad t_3 = \begin{bmatrix} -1 \\ 2 \end{bmatrix},$$

The following are constraints for each player.

$$\begin{aligned} &\text{1st player}: \begin{cases} 0 \le x_1 \le 10 \cdot \mathbf{1},\ x_{1,1} + x_{1,2} + x_{1,3} \le 20, \\ x_{1,1} + x_{1,2} - x_{1,3} \le x_{2,1} - x_{3,2} + 5, \end{cases} \\ &\text{2nd player}: 0 \le x_2 \le 10 \cdot \mathbf{1},\ x_{2,1} - x_{2,2} \le x_{1,2} + x_{1,3} - x_{3,1} + 7, \\ &\text{3rd player}: 0 \le x_3 \le 10 \cdot \mathbf{1},\ x_{3,2} \le x_{1,1} + x_{1,3} - x_{2,1} + 4. \end{aligned}$$

Example A.4. (FKA8 [10, 12]). Consider the 3-player GNEP

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

The original problem has infinitely many KKT points, so we added extra constraints to the first and third players' optimization so that the KKT set is finite.

Example A.5. (FKA12 [12]). Consider the duopoly model with 2-players:

$$\mathrm{F}_i(x_{-i}) : \left\{ \begin{array}{cl} \min\limits_{x_i \in \mathbb{R}^1} & x_i(x_1 + x_2 - 16) \\ s.t. & -10 \le x_i \le 10. \end{array} \right.$$

Example A.6. (NT59 [31]). Consider the environmental pollution control problem for $N = 3$ countries:

$$\mathrm{F}_i(x_{-i}) : \left\{ \begin{array}{cl} \min\limits_{x_i \in \mathbb{R}^2} & -x_{i,1}\left(b_i - \frac{1}{2}x_{i,1}\right) + \frac{x_{i,2}^2}{2} + d_i(x_{i,1} - \gamma_i x_{i,2}) + \sum\limits_{j \ne i} c_{i,j} x_{i,2} x_{j,1} \\ s.t. & x_{i,2} \ge 0,\ x_{i,1} \le b_i,\ 0 \le x_{i,1} - \gamma_i x_{i,2} \le E_i, \end{array} \right.$$

where parameters are set as

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

Example A.7. (NT510 [31]). Consider the electricity market problem in [12] with $N = 3$ generating companies.

$$\mathrm{F}_i(x_{-i}) : \left\{ \begin{array}{cl} \min\limits_{x_i \in \mathbb{R}^i} & \sum\limits_{j=1}^{i} \left(\frac{1}{2} c_{i,j} x_{i,j}^2 - d_{i,j} x_{i,j}\right) - (10 - \mathbf{1}^T x)\mathbf{1}^T x_i \\ s.t. & 0 \le x_{i,j} \le E_{i,j}\ (\forall j \in [i]), \end{array} \right.$$

where parameters are set as

$$\begin{array}{llllll} c_{1,1} = 0.4, & c_{2,1} = 0.35, & c_{2,2} = 0.35, & c_{3,1} = 0.46, & c_{3,2} = 0.5, & c_{3,3} = 0.5, \\ d_{1,1} = 2, & d_{2,1} = 1.25, & d_{2,2} = 1, & d_{3,1} = 2.25, & d_{3,2} = 3, & d_{3,3} = 3, \\ E_{1,1} = 2, & E_{2,1} = 2.5, & E_{2,2} = 0.67, & E_{3,1} = 1.2, & E_{3,2} = 1.8, & E_{3,3} = 1.6. \end{array}$$

Example A.8. (NTGS53 [32]). Consider the 2-player GNEP

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

The original problem has infinitely many KKT points, so we added extra constraints to each players' optimization so that the KKT set is finite.

Example A.9. (NTGS54 [32]). Consider the 2-player GNEP

$$\begin{array}{cl|cl} \min\limits_{x_1 \in \mathbb{R}^2} & -2x_{1,2}^2 + x_{2,1}x_{1,2} + x_{1,1}x_{2,1} & \min\limits_{x_2 \in \mathbb{R}^2} & \|x_2\|^2 - 2x_{2,2} \cdot \mathbf{1}^T x_1 \\ s.t. & \mathbf{1}^T x = 1,\ x_{1,1} \ge 0.1,\ x_{1,2} \ge 0.1, & s.t. & \mathbf{1}^T x = 1,\ x_{2,1} \ge 0.1,\ x_{2,2} \ge 0.1, \\ & x_1 \ge x_2, & & x_2 \ge x_1,\ 0.1 \ge x_{2,1}. \end{array}$$

The original problem has infinitely many KKT points, so we added extra constraints to each players' optimization so that the KKT set is finite.

Example A.10. (FR33 [15]). Consider the 2-player GNEP

$$\mathrm{F}_i(x_{-i}) : \left\{ \begin{array}{cl} \min\limits_{x_i \in \mathbb{R}^2} & -2x_{i,1} - (2i-1)x_{i,2} \\ s.t. & x_i \in X_i(x_{-i}), \end{array} \right.$$

where constraining sets

$$X_1(x_2) = \left\{ x_1 \in \mathbb{R}^2 \;\middle|\; \begin{array}{l} 0 \le x_{1,1} \le 5,\ 0 \le x_{1,2} \le 2.5,\ x_{1,1} + 2x_{1,2} \le 5, \\ 4x_{1,1} + x_{1,2} - \frac{16}{3} x_{2,1} - \frac{1}{3} x_{2,2} \le 0. \end{array} \right\},$$

$$X_2(x_1) = \left\{ x_2 \in \mathbb{R}^2 \;\middle|\; \begin{array}{l} 0 \le x_{2,1} \le 1.5,\ 0 \le x_{2,2} \le 6,\ 4x_{2,1} + x_{2,2} \le 6, \\ 15x_{1,1} - 10x_{1,2} + x_{2,1} + 2x_{2,2} \le 0. \end{array} \right\}.$$

Example A.11. (SAG41 [38]). Consider the 2-player NEP

$$\mathrm{F}_i(x_{-i}) : \begin{cases} \min\limits_{x_i \in \mathbb{R}^2} & 4x_{i,1}^2 + (-1)^{i+1} 2x_{1,1}x_{2,1} - \alpha_i x_{i,1} + \beta_i x_{i,2} \\ s.t. & x_{i,1} - x_{i,2} \le 0,\ 0 \le x_{i,1} \le 1,\ 0 \le x_{i,2} \le 1, \end{cases}$$

where $\alpha_1 = 10, \alpha_2 = 8, \beta_1 = 5$, and $\beta_2 = 7$.

Example A.12. (DSM31 [8]). Consider the 2-player GNEP

$$\mathrm{F}_i(x_{-i}) : \begin{cases} \min\limits_{x_i \in \mathbb{R}^2} & x_{i,2} \\ s.t. & x_{1,1} + x_{2,1} \le 1,\ x_{i,1} \ge 0, \\ & \alpha_{i,1}x_{1,1} + \alpha_{i,2}x_{1,2} + \alpha_{i,3}x_{2,1} + \alpha_{i,4}x_{2,2} \ge 0, \\ & \beta_{i,1}x_{1,1} + \beta_{i,2}x_{1,2} + \beta_{i,3}x_{2,1} + \beta_{i,4}x_{2,2} \ge 0, \end{cases}$$

where parameters are set as

$$\begin{array}{llllll} \alpha_{1,1} = -1, & \alpha_{1,2} = 1, & \alpha_{1,3} = 2, & \alpha_{1,4} = 0, & \beta_{1,1} = 1, & \beta_{1,2} = 1, \\ \beta_{1,3} = 1, & \beta_{1,4} = 0, & \alpha_{2,1} = 1, & \alpha_{2,2} = 0, & \alpha_{2,3} = -1, & \alpha_{2,4} = 1, \\ \beta_{2,1} = -1, & \beta_{2,2} = 0, & \beta_{2,3} = 1, & \beta_{2,4} = 1. & & \end{array}$$

References

[1] M. ApS, *The MOSEK optimization Toolbox for MATLAB manual. Version 9.0.*, 2019.

[2] Q. Ba and J.-S. Pang, *Exact Penalization of Generalized Nash Equilibrium Problems*, Operations Research, 70 (2022), pp. 1448–1464.

[3] E. Börgens and C. Kanzow, *ADMM-Type Methods for Generalized Nash Equilibrium Problems in Hilbert Spaces*, SIAM Journal on Optimization, 31 (2021), pp. 377–403.

[4] P. Breiding and S. Timme, *HomotopyContinuation.jl: A package for homotopy continuation in Julia*, In: International Congress on Mathematical Software, pp. 458–465, Springer, 2018.

[5] Y. Cui and J.-S. Pang, *Modern Nonconvex Nondifferentiable Optimization*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2021.

[6] A. Dreves, F. Facchinei, C. Kanzow, and S. Sagratella, *On the solution of the KKT conditions of generalized Nash equilibrium problems*, SIAM Journal on Optimization, 21 (2011), pp. 1082–1108.

[7] A. Dreves, C. Kanzow, and O. Stein, *Nonsmooth optimization reformulations of player convex generalized Nash equilibrium problems*, Journal of Global Optimization, 53 (2012), pp. 587–614.

[8] A. Dreves and N. Sudermann-Merx, *Solving linear generalized Nash equilibrium problems numerically*, Optimization Methods and Software, 31 (5), 1036–1063, 2016.

[9] F. Facchinei, A. Fischer, and V. Piccialli, *On generalized Nash games and variational inequalities*, Operations Research Letters, 35 (2007), pp. 159–164.

[10] ———, *Generalized Nash equilibrium problems and Newton methods*, Mathematical Programming, 117 (2009), pp. 163–194.

[11] F. Facchinei and C. Kanzow, *Generalized Nash Equilibrium Problems*, Annals of Operations Research, 175 (2010), pp. 177–211.

[12] ———, *Penalty Methods for the Solution of Generalized Nash Equilibrium Problems*, SIAM Journal on Optimization, 20 (2010), pp. 2228–2253.

[13] F. Facchinei and J. Pang, *Nash equilibria: the variational approach*, in Convex Optimization in Signal Processing and Communications, D. P. Palomar and Y. C. Eldar, eds., Cambridge University Press, 2010, pp. 443–493.

[14] F. Facchinei, V. Piccialli, and M. Sciandrone, *Decomposition algorithms for generalized potential games*, Computational Optimization and Applications, 50 (2011), pp. 237–262.

[15] Z. Feinstein and B. Rudloff, *Characterizing and Computing the Set of Nash Equilibria via Vector Optimization*, Operations Research, `doi.org/10.1287/opre.2023.2457`.

[16] D. Henrion, M. Korda, and J. B. Lasserre, *The Moment-SOS Hierarchy*, WORLD SCIENTIFIC (EUROPE), 2020.

[17] D. Henrion and J.-B. Lasserre, *Detecting Global Optimality and Extracting Solutions in GloptiPoly*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 293–310.

[18] D. Henrion, J.-B. Lasserre, and J. Löfberg, *GloptiPoly 3: moments, optimization and semidefinite programming*, Optimization Methods & Software, 24 (2009), pp. 761–779.

[19] C. Kanzow and D. Steck, *Augmented Lagrangian Methods for the Solution of Generalized Nash Equilibrium Problems*, SIAM Journal on Optimization, 26 (2016), pp. 2034–2058.

[20] J. B. Lasserre, *An Introduction to Polynomial and Semi-Algebraic Optimization*, Cambridge Texts in Applied Mathematics, Cambridge University Press, 2015.

[21] M. Laurent, *Sums of squares, moment matrices and optimization over polynomials*, in Emerging Applications of Algebraic Geometry, M. Putinar and S. Sullivant, eds., The IMA volumes in mathematics and its applications, New York, NY, 2009, Springer New York, pp. 157–270.

[22] K. Lee and X. Tang, *On the Polyhedral Homotopy Method for Solving Generalized Nash Equilibrium Problems of Polynomials*, Journal of Scientific Computing, 95 (2023), p. 13.

[23] J. Lofberg, *Yalmip : a toolbox for modeling and optimization in matlab*, in 2004 IEEE International Conference on Robotics and Automation, 2004, pp. 284–289.

[24] J. Nie, *Certifying convergence of Lasserre's hierarchy via flat truncation*, Mathematical Programming, 142 (2013), pp. 485–510.

[25] ———, *Moment and Polynomial Optimization*, SIAM, Philadelphia, 2023.

[26] ———, *Polynomial optimization with real varieties*, SIAM Journal on Optimization, 23 (2013), pp. 1634–1646.

[27] ———, *Optimality conditions and finite convergence of Lasserre's hierarchy*, Mathematical Programming, 146 (2014), pp. 97–121.

[28] ———, *Tight relaxations for polynomial optimization and Lagrange multiplier expressions*, Mathematical Programming, 178 (2019), pp. 1–37.

[29] J. Nie, K. Ranestad, and X. Tang, *Algebraic degrees of generalized nash equilibrium problems*, Preprint (2022). `arXiv:2208.00357`

[30] J. Nie and X. Tang, *Convex generalized Nash equilibrium problems and polynomial optimization*, Mathematical Programming, 198 (2023), pp. 1485–1518.

[31] ———, *Nash equilibrium problems of polynomials*, Mathematics of Operations Research, 49.2 (2023), pp. 1065–1090.

[32] J. Nie, X. Tang, and L. Xu, *The Gauss–Seidel method for generalized Nash equilibrium problems of polynomials*, Computational Optimization and Applications, 78 (2021), pp. 529–557.

[33] J. Nie, X. Tang, and S. Zhong, *Rational generalized Nash equilibrium problems*, SIAM Journal on Optimization, 33 (2023), pp. 1587 - 1620.

[34] J. Nie, J. J. Ye, and S. Zhong, *PLMEs and disjunctive decompositions for bilevel optimization*, Preprint (2023). `arXiv:2304.00695`

[35] J.-S. Pang and M. Fukushima, *Quasi-variational inequalities, generalized Nash equilibria, and multi-leader-follower games*, Computational Management Science, 2 (2005), pp. 21–56.

[36] J.-S. Pang and G. Scutari, *Nonconvex Games with Side Constraints*, SIAM Journal on Optimization, 21 (2011), pp. 1491–1522.

[37] M. Putinar, *Positive polynomials on compact semi-algebraic sets*, Indiana University Mathematics Journal, 42 (1993), pp. 969–984.

[38] S. Sagratella. *On generalized Nash equilibrium problems with linear coupling constraints and mixed-integer variables*, Optimization, 68 (2019), pp. 197–226.

[39] D. A. Schiro, J.-S. Pang, and U. V. Shanbhag, *On the solution of affine generalized Nash equilibrium problems with shared constraints by Lemke's method*, Mathematical Programming, 142 (2013), pp. 1–46.

[40] R. Schneider, *Convex Bodies: The Brunn–Minkowski Theory*, Encyclopedia of Mathematics and its Applications, Cambridge University Press, 2013.

[41] A. von Heusinger, C. Kanzow, and M. Fukushima, *Newton's method for computing a normalized equilibrium in the generalized Nash game through fixed point formulation*, Mathematical Programming, 132 (2012), pp. 99–123.

Jiyoung Choi, Jiawang Nie, Department of Mathematics, University of California San Diego, 9500 Gilman Drive, La Jolla, CA, USA, 92093.
Email address: jichoi@ucsd.edu, njw@math.ucsd.edu

Xindong Tang, Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Kowloon, Hong Kong.
Email address: xdtang@hkbu.edu.hk

Suhan Zhong, Department of Mathematics, Texas A&M University, College Station, TX, USA, 77843.
Email address: suzhong@tamu.edu