Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 640960, 8 pages
http://dx.doi.org/10.1155/2014/640960

Research Article

A Mixed 0-1 Linear Programming Approach to the Computation of All Pure-Strategy Nash Equilibria of a Finite *n*-Person Game in Normal Form

Zhengtian Wu,[1,2] Chuangyin Dang,[2] Hamid Reza Karimi,[3] Changan Zhu,[1] and Qing Gao[4]

[1] *Department of Precision Machinery and Precision Instrumentation, University of Science and Technology of China, Hefei 230026, China*
[2] *Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong*
[3] *Department of Engineering, Faculty of Engineering and Science, University of Agder, 4898 Grimstad, Norway*
[4] *School of Engineering and Information Technology, University of New South Wales at the Australian Defence Force Academy, Canberra, ACT 2600, Australia*

Correspondence should be addressed to Hamid Reza Karimi; hamid.r.karimi@uia.no

Received 19 January 2014; Accepted 3 March 2014; Published 8 April 2014

Academic Editor: Michael Lütjen

A main concern in applications of game theory is how to effectively select a Nash equilibrium, especially a pure-strategy Nash equilibrium for a finite *n*-person game in normal form. This selection process often requires the computation of all Nash equilibria. It is well known that determining whether a finite game has a pure-strategy Nash equilibrium is an NP-hard problem and it is difficult to solve by naive enumeration algorithms. By exploiting the properties of pure strategy and multilinear terms in the payoff functions, this paper formulates a new mixed 0-1 linear program for computing all pure-strategy Nash equilibria. To our knowledge, it is the first method to formulate a mixed 0-1 linear programming for pure-strategy Nash equilibria and it may work well for similar problems. Numerical results show that the approach is effective and this method can be easily distributed in a distributed way.

1. Introduction

Nash equilibrium, as defined in [7], plays a fundamental role in the development of game theory and its diverse applications in areas such as biology, computer science, economics, and management and social sciences. In these applications, a main concern is how to effectively select a Nash equilibrium, especially a pure-strategy Nash equilibrium for a finite *n*-person game in normal form. To tackle this problem, many contributions have been made in the literature.

In recent years, most related work is focused on finding mixed-strategy Nash equilibrium. In terms of computational complexity, it was shown in [1, 2] that the problem of finding one mixed-strategy Nash equilibrium is PPAD-complete. The first method for computing Nash equilibrium of a two-person game was developed in [3] and was extended to computing Nash equilibria of *n*-person games independently in [4, 5]. The existence of Nash equilibrium was established from an application of Brouwer and Kakutani fixed-point theorems in [6, 7]. To approximate fixed points of continuous mappings, simplicial methods were originated in [8] and substantially developed in the literature such as [9–15]. Simplicial methods were adapted for computing Nash equilibria in [16]. Several more advanced simplicial methods for computing Nash equilibria were presented in [17–19]. A pivoting procedure was proposed in [20, 21] for computing the Nash equilibria of two-person games selected by the linear tracing procedure proposed in [22]. This procedure was extended to *n*-person games in [23]. A survey of the literature on the computation of Nash equilibria before 1996 can be found in [24] and on computing equilibria for two-person games in [25]. By exploiting differentiability of the problem, several methods have been developed for computing Nash equilibria in the literature. A smooth logarithmic tracing procedure with two parameters was proposed in [26]. It was proved in [27] that the procedure converges to a Nash equilibrium as two parameters approach zero one after another. The linear tracing procedure together with well-chosen nonlinear transformations of variables of the system of the optimality conditions leads to a differentiable homotopy method in [28]. As a result of Kohlberg and Mertens' structure theorem [29], the global Newton method in [30] was combined with the homotopy method in [11] to obtain an algorithm for computing Nash equilibrium. A survey on homotopy methods for computing Nash equilibrium is referred to in [31]. Some new related work has been developed in the literature [32–36].

All the algorithms mentioned above can only find a sample Nash equilibrium. Unfortunately, in practice, a game can have many Nash equilibria and which equilibrium should be effectively selected to play is a very challenging issue. The equilibrium selection process often requires the computation of all Nash equilibria of a game. To address this issue, several algorithms have been developed in the literature. A path-following algorithm was described in [37] to compute all Nash equilibria of a bimatrix game. A homotopy algorithm was presented in [38] to find all Nash equilibria of an n-person game. Both of these algorithms are based on systems of polynomial equations. A survey on polyhedral homotopy continuation methods for all Nash equilibria can be found in [39]. In [40], a mixed-integer program (MIP) was formulated to find Nash equilibria of a bimatrix game, which can be employed to compute all Nash equilibria of a bimatrix game. Recently, by enumerating vertices of polyhedrons, two algorithms were introduced in [41] for computing all Nash equilibria of a bimatrix game.

As a special example of the mixed-strategy Nash equilibrium, the pure-strategy Nash equilibrium has gotten more attention recently. The work of [42] shows that determining whether a game has a pure-strategy Nash equilibrium is NP-hard. The work of [2] presents a novel generic mapping between graphical games and Markov random fields so that pure Nash equilibria in the former can be found by statistical inference on the latter. Based on a new description format of game-stimulate-response pair, the work of [43] puts forward a constraints' satisfaction-based algorithm on this data structure to compute pure Nash equilibrium of graphical game. The work of [44] develops an efficient algorithm, which is called Valued Nash Propagation, for computing pure strategy Nash equilibria that satisfy various criteria (such as the utilitarian or Nash-Bernoulli social welfare functions) in games with sparse interaction structure. Some other related work can be found in [2, 45, 46].

As a special case of integer programming, the mixed 0-1 linear programming, in which unknowns are binary, has many applications in operation research. Many classic problems, such as the knapsack problem, facility location problem, production planning problem, and time tabling problem, can be formulated as a mixed 0-1 linear programming. The research of these problems which has a long history and many commercial softwares, most of which are based on branch and bound method, cutting plane method, or branch and cut method, could solve this mixed 0-1 linear programming well.

This paper studies the computation of all pure-strategy Nash equilibria of a finite n-person game in normal form. To solve this problem, we formulate a mixed 0-1 linear programming by exploiting the properties of pure strategy and multilinear forms in the payoff functions. An enumeration of all the feasible solutions of the mixed 0-1 linear program yields all pure-strategy Nash equilibria. Numerical results show that the method is promising. To our knowledge, it is the first method to formulate a mixed 0-1 linear programming for pure-strategy Nash equilibria and this method may work well for other similar problems. Furthermore, as an advantage of the formulation, one can apply distributed computation to dramatically speed up the computation.

The rest of the paper is organized as follows. Section 2 formulates a mixed 0-1 linear program for computing all pure-strategy equilibria. Section 3 presents numerical results to show that the approach is effective. The paper is concluded with some remarks in Section 4.

2. A Mixed 0-1 Linear Programming Formulation

In this section a mixed 0-1 linear programming will be formulated to all pure-strategy Nash equilibria of a finite game.

Let $N = \{1, 2, \ldots, n\}$ be the set of players. The pure strategy set of player $i \in N$ is denoted by $S^i = \{s^i_j \mid j \in M_i\}$ with $M_i = \{1, 2, \ldots, m_i\}$. Given S^i with $i \in N$, the set of all pure strategy profiles is $S = \prod_{i=1}^n S^i$. We denote the payoff function of player $i \in N$ by $u^i : S \to R$. For $i \in N$, let $S^{-i} = \prod_{k \in N \setminus \{i\}} S^k$. Then, $s = (s^1_{j_1}, s^2_{j_2}, \ldots, s^n_{j_n}) \in S$ can be rewritten as $s = (s^i_{j_i}, s^{-i})$ with $s^{-i} = (s^1_{j_1}, \ldots, s^{i-1}_{j_{i-1}}, s^{i+1}_{j_{i+1}}, \ldots, s^n_{j_n}) \in S^{-i}$. A mixed strategy of player i is a probability distribution on S^i denoted by $x^i = (x^i_1, x^i_2, \ldots, x^i_{m_i})$. Let X^i be the set of all mixed strategies of player i. Then, $X^i = \{x^i = (x^i_1, x^i_2, \ldots, x^i_{m_i}) \in R^{m_i}_+ \mid \sum_{j=1}^{m_i} x^i_j = 1\}$. Thus, for $x^i \in X^i$, the probability assigned to pure strategy $s^i_j \in S^i$ is equal to x^i_j. Given X^i with $i \in N$, the set of all mixed strategy profiles is $X = \prod_{i=1}^n X^i$. For $i \in N$, let $X^{-i} = \prod_{k \in N \setminus \{i\}} X^k$. Then, $x = (x^1, x^2, \ldots, x^n) \in X$ can be rewritten as $x = (x^i, x^{-i})$ with $x^{-i} = (x^1, \ldots, x^{i-1}, x^{i+1}, \ldots, x^n) \in X^{-i}$. If $x \in X$ is played, then the probability that a pure strategy profile $s = (s^1_{j_1}, s^2_{j_2}, \ldots, s^n_{j_n}) \in S$ occurs is $\prod_{i=1}^n x^i_{j_i}$. Therefore, for $x \in X$, the expected payoff of player i is given by $u^i(x) = \sum_{s \in S} u^i(s) \prod_{i=1}^n x^i_{j_i}$. With these notations, a finite n-person game in normal form can be represented as $\Gamma = \langle N, S, \{u^i\}_{i \in N} \rangle$ or $\Gamma = \langle N, X, \{u^i\}_{i \in N} \rangle$.

Definition 1 (Nash, [7]). A mixed strategy profile $x^* \in X$ is a Nash equilibrium of game Γ if $u^i(x^*) \geq u^i(x^i, x^{*-i})$ for all $i \in N$ and $x^i \in X^i$.

With this definition, an application of the optimality condition leads to the fact that x^* is a Nash equilibrium if and only if there are λ^* and μ^* together with x^* satisfying the system of

$$\begin{gathered} u^i\left(s^i_j, x^{-i}\right) + \lambda^i_j - \mu_i = 0, \\ e^{i\top} x^i - 1 = 0, \\ x^i_j \lambda^i_j = 0, \quad x^i_j \geq 0, \\ \lambda^i_j \geq 0, \quad j = 1, 2, \ldots, m_i, \ i = 1, 2, \ldots, n, \end{gathered} \tag{1}$$

where $e^i = (1, 1, \ldots, 1)^\top \in R^{m_i}$.

Lemma 2. *Let β be a given positive number such that*

$$\beta \geq \max_{i \in N} \left\{ \max_{s \in S} u^i(s) - \min_{s \in S} u^i(s) \right\}. \tag{2}$$

Then, (1) *is equivalent to*

$$\begin{gathered} u^i\left(s^i_j, x^{-i}\right) + \lambda^i_j - \mu_i = 0, \\ e^{i\top} x^i - 1 = 0, \\ x^i_j \leq v^i_j, \\ \lambda^i_j \leq \beta\left(1 - v^i_j\right), \quad v^i_j \in \{0, 1\}, \\ x^i_j \geq 0, \qquad \lambda^i_j \geq 0, \\ j = 1, 2, \ldots, m_i, \quad i = 1, 2, \ldots, n. \end{gathered} \tag{3}$$

Proof. Let (x, λ, v) be a solution of the system (3). Suppose that $v^i_j = 0$ in (3). Then, $x^i_j = 0$ and $\lambda^i_j \leq \beta$. Thus, $x^i_j \lambda^i_j = 0$. Suppose that $v^i_j = 1$ in (3). Then, $\lambda^i_j = 0$. Thus, $x^i_j \lambda^i_j = 0$. Therefore, (x, λ) is a solution of the system (1).

Let (x, λ) be a solution of the system (1). Since $\sum_{j=1}^{m_i} x^i_j = 1$, there must exist some $k \in M_i$ such that $x^i_k > 0$. From $x^i_k \lambda^i_k = 0$, we get that $\lambda^i_k = 0$. Thus, $\mu_i = u^i(s^i_k, x^{-i}) \leq \max_{s \in S} u^i(s)$. Therefore, for any $j \in M_i$, we obtain from the system (1) that

$$\lambda^i_j = \mu_i - u^i\left(s^i_j, x^{-i}\right) \leq \max_{s \in S} u^i(s) - \min_{s \in S} u^i(s) \leq \beta. \tag{4}$$

Suppose that $x^i_j = 0$. Let $v^i_j = 0$. Thus,

$$\begin{gathered} x^i_j = 0 \leq v^i_j, \\ \lambda^i_j \leq \beta\left(1 - v^i_j\right) = \beta. \end{gathered} \tag{5}$$

Suppose that $x^i_j > 0$. Let $v^i_j = 1$. Thus,

$$\begin{gathered} x^i_j \leq v^i_j = 1, \\ \lambda^i_j = 0 \leq \beta\left(1 - v^i_j\right) = 0. \end{gathered} \tag{6}$$

Therefore, (x, λ, v) is a solution of the system (3). This completes the proof. □

This lemma implies that finding a pure-strategy Nash equilibrium is equivalent to finding a solution to the system of

$$\begin{gathered} u^i\left(s^i_j, x^{-i}\right) + \lambda^i_j - \mu_i = 0, \\ e^{i\top} x^i - 1 = 0, \\ \lambda^i_j \leq \beta\left(1 - x^i_j\right), \quad x^i_j \in \{0, 1\}, \\ \lambda^i_j \geq 0, \quad j = 1, 2, \ldots, m_i, \ i = 1, 2, \ldots, n. \end{gathered} \tag{7}$$

For any $s^i_j \in S^i$, one can obtain from $u^i(x)$ that

$$u^i\left(s^i_j, x^{-i}\right) = \sum_{s^{-i} = \left(s^1_{j_1}, \ldots, s^{i-1}_{j_{i-1}}, s^{i+1}_{j_{i+1}}, \ldots, s^n_{j_n}\right) \in S^{-i}} u^i\left(s^i_j, s^{-i}\right) \prod_{k \neq i} x^k_{j_k}. \tag{8}$$

Let $y(s^{-i}) = \prod_{k \neq i} x^k_{j_k}$ for $s^{-i} = (s^1_{j_1}, \ldots, s^{i-1}_{j_{i-1}}, s^{i+1}_{j_{i+1}}, \ldots, s^n_{j_n}) \in S^{-i}$. Then,

$$u^i\left(s^i_j, x^{-i}\right) = \sum_{s^{-i} = \left(s^1_{j_1}, \ldots, s^{i-1}_{j_{i-1}}, s^{i+1}_{j_{i+1}}, \ldots, s^n_{j_n}\right) \in S^{-i}} u^i\left(s^i_j, s^{-i}\right) y\left(s^{-i}\right). \tag{9}$$

Substituting this into (7) yields

$$\begin{gathered} \sum_{s^{-i} \in S^{-i}} u^i\left(s^i_j, s^{-i}\right) y\left(s^{-i}\right) + \lambda^i_j - \mu_i = 0, \\ e^{i\top} x^i - 1 = 0, \\ \lambda^i_j \leq \beta\left(1 - x^i_j\right), \quad x^i_j \in \{0, 1\}, \\ \lambda^i_j \geq 0, \quad j = 1, 2, \ldots, m_i, \ i = 1, 2, \ldots, n, \\ y\left(s^{-i}\right) = \prod_{k \neq i} x^k_{j_k}, \quad s^{-i} \in S^{-i}, \ i = 1, 2, \ldots, n. \end{gathered} \tag{10}$$

Observe that $y(s^{-i}) \in \{0, 1\}$ since $x^i_j \in \{0, 1\}$. This property leads to the following result.

Lemma 3. *For $i \in N$ and $s^{-i} = (s^1_{j_1}, \ldots, s^{i-1}_{j_{i-1}}, s^{i+1}_{j_{i+1}}, \ldots, s^n_{j_n}) \in S^{-i}$, the system of*

$$\begin{gathered} y\left(s^{-i}\right) = \prod_{k \neq i} x^k_{j_k}, \\ x^k_{j_k} \in \{0, 1\}, \quad k = 1, 2, \ldots, n, \ k \neq i, \end{gathered} \tag{11}$$

is equivalent to the system of

$$\begin{gathered} y\left(s^{-i}\right) \geq \sum_{h \neq i} x^h_{j_h} - (n - 2), \\ y\left(s^{-i}\right) \leq x^k_{j_k}, \quad 0 \leq y\left(s^{-i}\right), \\ x^k_{j_k} \in \{0, 1\}, \quad k = 1, 2, \ldots, n, \ k \neq i. \end{gathered} \tag{12}$$

Proof. Let (y, x) be a solution of the system (12). Suppose that $x^k_{j_k} = 1$, for all $k \neq i$. Then,

$$\begin{gathered} y\left(s^{-i}\right) \geq \sum_{h \neq i} x^h_{j_h} - (n-2) = (n-1) - (n-2) = 1, \\ y\left(s^{-i}\right) \leq x^k_{j_k} = 1, \quad k \neq i. \end{gathered} \tag{13}$$

Thus, $y(s^{-i}) = \prod_{k \neq i} x^k_{j_k} = 1$.

Suppose that $x^k_{j_k} = 0, \exists k \neq i$. Then,

$$y\left(s^{-i}\right) \leq x^k_{j_k} = 0, \quad \exists k \neq i, \; 0 \leq y\left(s^{-i}\right). \tag{14}$$

Thus, $y(s^{-i}) = \prod_{k \neq i} x^k_{j_k} = 0$. Therefore, (y, x) is a solution of the system (11).

Let (y, x) be a solution of the system (11). Suppose that $x^k_{j_k} = 1$, for all $k \neq i$. Then, $y(s^{-i}) = \prod_{k \neq i} x^k_{j_k} = 1$. Thus,

$$\begin{gathered} \sum_{h \neq i} x^h_{j_h} - (n-2) = (n-1) - (n-2) = 1 \leq y\left(s^{-i}\right) = 1, \\ y\left(s^{-i}\right) = 1 \leq x^k_{j_k} = 1, \quad k \neq i, \; 0 \leq y\left(s^{-i}\right). \end{gathered} \tag{15}$$

Suppose that $x^k_{j_k} = 0, \exists k \neq i$. Then, $y(s^{-i}) = \prod_{k \neq i} x^k_{j_k} = 0$. Thus,

$$\begin{gathered} \sum_{h \neq i} x^h_{j_h} - (n-2) \leq (n-2) - (n-2) = y\left(s^{-i}\right) = 0, \\ y\left(s^{-i}\right) = 0 \leq x^k_{j_k}, \quad k \neq i, \; 0 \leq y\left(s^{-i}\right). \end{gathered} \tag{16}$$

Therefore, (y, x) is a solution of the system (12). This completes the proof. □

Replacing $y(s^{-i}) = \prod_{k \neq i} x^k_{j_k}$ of the system (10) with the system (12), we obtain a mixed 0-1 linear program, which is as follows:

$$\begin{gathered} \sum_{s^{-i} \in S^{-i}} u^i\left(s^i_j, s^{-i}\right) y\left(s^{-i}\right) + \lambda^i_j - \mu_i = 0, \\ e^{i\top} x^i - 1 = 0, \\ \lambda^i_j \leq \beta\left(1 - x^i_j\right), \\ x^i_j \in \{0, 1\}, \\ \lambda^i_j \geq 0, \\ j = 1, 2, \ldots, m_i, \quad i = 1, 2, \ldots, n, \\ y\left(s^{-i}\right) \geq \sum_{h \neq i} x^h_{j_h} - (n-2), \\ y\left(s^{-i}\right) \leq x^k_{j_k}, \quad k \neq i, \\ 0 \leq y\left(s^{-i}\right), \\ s^{-i} \in S^{-i}, \quad i = 1, 2, \ldots, n. \end{gathered} \tag{17}$$

As a corollary of Lemma 3, we obtain the following result.

Corollary 4. *The system* (17) *is equivalent to the system* (7).

The above idea is illustrated in the following example.

Example 5. Consider a three-player game $\Gamma = (N, S, \{u^i\}_{i \in N})$, where $N = \{1, 2, 3\}$, $S^i = \{s^i_1, s^i_2\}$, $i \in N$, and $\{u^i\}_{i \in N}$ are given by

$$\begin{array}{ccc|cc} & s^2_1 & s^2_2 & s^2_1 & s^2_2 \\ s^1_1 & (1,1,1) & (1,0,1) & (0,1,0) & (1,0,0) \\ s^1_2 & (1,1,1) & (1,0,1) & (0,1,0) & (0,0,0) \\ & \multicolumn{2}{c|}{s^3_1} & \multicolumn{2}{c}{s^3_2} \end{array} \tag{18}$$

Let $\beta = 1$. The mixed 0-1 linear program for this game is given by

$$\begin{gathered} y\left(s^2_1, s^3_1\right) + y\left(s^2_2, s^3_1\right) + y\left(s^2_2, s^3_2\right) + \lambda^1_1 - \mu_1 = 0, \\ y\left(s^2_1, s^3_1\right) + y\left(s^2_2, s^3_1\right) + \lambda^1_2 - \mu_1 = 0, \\ x^1_1 + x^1_2 = 1, \\ y\left(s^1_1, s^3_1\right) + y\left(s^1_2, s^3_1\right) + y\left(s^1_1, s^3_2\right) + y\left(s^1_2, s^3_2\right) + \lambda^2_1 - \mu_2 = 0, \\ \lambda^2_2 - \mu_2 = 0, \\ x^2_1 + x^2_2 = 1, \\ y\left(s^1_1, s^2_1\right) + y\left(s^1_2, s^2_1\right) + y\left(s^1_1, s^2_2\right) + y\left(s^1_2, s^2_2\right) + \lambda^3_1 - \mu_3 = 0, \\ \lambda^3_2 - \mu_3 = 0, \\ x^3_1 + x^3_2 = 1, \\ \lambda^i_j \leq 1 - x^i_j, \quad x^i_j \in \{0, 1\}, \; \lambda^i_j \geq 0, \\ j = 1, 2, \quad i = 1, 2, 3, \\ y\left(s^2_1, s^3_1\right) \geq x^2_1 + x^3_1 - 1, \qquad y\left(s^2_1, s^3_1\right) \leq x^2_1, \\ y\left(s^2_1, s^3_1\right) \leq x^3_1, \quad y\left(s^2_1, s^3_1\right) \geq 0, \\ y\left(s^2_1, s^3_2\right) \geq x^2_1 + x^3_2 - 1, \qquad y\left(s^2_1, s^3_2\right) \leq x^2_1, \\ y\left(s^2_1, s^3_2\right) \leq x^3_2, \quad y\left(s^2_1, s^3_2\right) \geq 0, \\ y\left(s^2_2, s^3_1\right) \geq x^2_2 + x^3_1 - 1, \qquad y\left(s^2_2, s^3_1\right) \leq x^2_2, \\ y\left(s^2_2, s^3_1\right) \leq x^3_1, \quad y\left(s^2_2, s^3_1\right) \geq 0, \\ y\left(s^2_2, s^3_2\right) \geq x^2_2 + x^3_2 - 1, \qquad y\left(s^2_2, s^3_2\right) \leq x^2_2, \end{gathered}$$

$$\begin{gathered}
y\left(s_2^2, s_2^3\right) \leq x_2^3, \quad y\left(s_2^2, s_2^3\right) \geq 0, \\
y\left(s_1^1, s_1^3\right) \geq x_1^1 + x_1^3 - 1, \qquad y\left(s_1^1, s_1^3\right) \leq x_1^1, \\
y\left(s_1^1, s_1^3\right) \leq x_1^3, \quad y\left(s_1^1, s_1^3\right) \geq 0, \\
y\left(s_2^1, s_1^3\right) \geq x_2^1 + x_1^3 - 1, \qquad y\left(s_2^1, s_1^3\right) \leq x_2^1, \\
y\left(s_2^1, s_1^3\right) \leq x_1^3, \quad y\left(s_2^1, s_1^3\right) \geq 0, \\
y\left(s_1^1, s_2^3\right) \geq x_1^1 + x_2^3 - 1, \qquad y\left(s_1^1, s_2^3\right) \leq x_1^1, \\
y\left(s_1^1, s_2^3\right) \leq x_2^3, \quad y\left(s_1^1, s_2^3\right) \geq 0, \\
y\left(s_2^1, s_2^3\right) \geq x_2^1 + x_2^3 - 1, \qquad y\left(s_2^1, s_2^3\right) \leq x_2^1, \\
y\left(s_2^1, s_2^3\right) \leq x_2^3, \quad y\left(s_2^1, s_2^3\right) \geq 0, \\
y\left(s_1^1, s_1^2\right) \geq x_1^1 + x_1^2 - 1, \qquad y\left(s_1^1, s_1^2\right) \leq x_1^1, \\
y\left(s_1^1, s_1^2\right) \leq x_1^2, \quad y\left(s_1^1, s_1^2\right) \geq 0, \\
y\left(s_1^1, s_2^2\right) \geq x_1^1 + x_2^2 - 1, \qquad y\left(s_1^1, s_2^2\right) \leq x_1^1, \\
y\left(s_1^1, s_2^2\right) \leq x_2^2, \quad y\left(s_1^1, s_2^2\right) \geq 0, \\
y\left(s_2^1, s_1^2\right) \geq x_2^1 + x_1^2 - 1, \qquad y\left(s_2^1, s_1^2\right) \leq x_2^1, \\
y\left(s_2^1, s_1^2\right) \leq x_1^2, \quad y\left(s_2^1, s_1^2\right) \geq 0, \\
y\left(s_2^1, s_2^2\right) \geq x_2^1 + x_2^2 - 1, \qquad y\left(s_2^1, s_2^2\right) \leq x_2^1, \\
y\left(s_2^1, s_2^2\right) \leq x_2^2, \quad y\left(s_2^1, s_2^2\right) \geq 0.
\end{gathered} \tag{19}$$

3. Numerical Results

In this section, some numerical results are presented. Under the software Microsoft Visual Studio 2008 and ILOG CPLEX, we use C++ to call ILOG CPLEX API functions which are based on the branch and cut method to solve the mixed 0-1 linear program (17). The MIP (mixed integer programming) search method, which is a dynamic search strategy or branch and cut strategy in ILOG CPLEX, is determined by the ILOG CPLEX automatically. All other parameters are also set automatically by ILOG CPLEX itself.

We have run our code on a workstation of Lenovo ThinkStation D20 4155-BM4 with 16 processors and 16 G RAM. In the presentation of numerical results, some symbols are explained as follows:

Num$N$: the number of players;

Num$S$: the number of strategies for each player;

NumEquilibra: the number of pure-strategy Nash equilibria for the instance;

Time: the total computational time to solve the problem.

Table 1: The payoff function $u^i(s_j^i, s^{-i})$ is generated from 0 to 10 randomly.

Prob.	Num$N$	Num$S$	NumEquilibra	Time (s)
1	3	4	1	0.53
2	3	4	2	0.62
3	3	4	2	0.56
4	3	5	3	0.56
5	3	6	2	0.41
6	3	6	1	0.66
7	3	7	5	0.59
8	3	8	6	0.58
9	3	8	3	0.89
10	3	8	3	0.81
11	3	10	2	1.00
12	3	11	3	1.47
13	3	12	4	1.81
14	3	14	4	4.91
15	3	14	8	3.81
16	3	15	6	5.87
17	3	19	13	27.86
18	3	19	3	33.40
19	3	21	13	47.28
20	3	23	10	102.51
21	3	24	22	122.94
22	3	25	21	239.20
23	3	26	13	278.50
24	3	26	15	301.58
25	3	28	21	419.97
26	3	33	22	1870.24
27	3	35	20	2082.34
28	3	40	24	4599.72
29	3	44	20	3033.29
30	3	45	26	16992.10

Example 1. Consider a three-player game $\Gamma = (N, S, \{u^i\}_{i \in N})$, where $N = \{1, 2, 3\}$. The number of strategies for each player, Num$S$, is generated from 3 to 45 randomly. The $\{u^i\}_{i \in N}$ are generated randomly too. There are three different ranges for $\{u^i\}_{i \in N}$ in this example, which are from 0 to 10, from 0 to 50, and from 0 to 100.

Let $\beta = 1000$; 30 instances have been solved for each range of $\{u^i\}_{i \in N}$. Numerical results for each range are presented in Tables 1, 2, and 3, respectively.

Example 2. Consider a five-player game $\Gamma = (N, S, \{u^i\}_{i \in N})$, where $N = \{1, 2, 3, 4, 5\}$. The number of strategies for each player, Num$S$, is generated from 3 to 7 randomly. The $\{u^i\}_{i \in N}$ are generated randomly too. There are three different ranges for $\{u^i\}_{i \in N}$ in this example, which are from 0 to 10, from 0 to 50, and from 0 to 100.

Table 2: The payoff function $u^i(s_j^i, s^{-i})$ is generated from 0 to 50 randomly.

Prob.	Num$N$	Num$S$	NumEquilibra	Time (s)
1	3	3	4	0.46
2	3	3	0	0.50
3	3	3	1	0.40
4	3	4	0	0.34
5	3	4	2	0.52
6	3	6	0	0.45
7	3	7	1	0.70
8	3	7	2	0.82
9	3	7	1	0.57
10	3	8	0	0.73
11	3	9	0	0.66
12	3	10	0	0.93
13	3	11	1	1.64
14	3	16	0	13.26
15	3	19	0	33.22
16	3	20	3	40.67
17	3	21	2	55.10
18	3	22	4	74.20
19	3	24	0	195.78
20	3	24	2	142.14
21	3	27	3	507.44
22	3	29	2	1178.18
23	3	30	1	1287.87
24	3	30	1	1667.70
25	3	32	1	1638.94
26	3	33	2	1761.86
27	3	34	0	3125.08
28	3	35	5	3446.67
29	3	44	3	36601.30
30	3	45	5	43583.30

Table 3: The payoff function $u^i(s_j^i, s^{-i})$ is generated from 0 to 100 randomly.

Prob.	Num$N$	Num$S$	NumEquilibra	Time (s)
1	3	3	0	0.41
2	3	3	2	0.45
3	3	4	2	0.51
4	3	5	0	0.49
5	3	5	2	0.61
6	3	7	0	0.56
7	3	7	2	0.41
8	3	8	0	0.61
9	3	8	2	0.89
10	3	8	2	0.68
11	3	10	1	1.11
12	3	12	1	4.00
13	3	13	1	2.64
14	3	13	2	2.64
15	3	16	0	14.25
16	3	19	4	25.70
17	3	22	4	55.94
18	3	24	2	132.87
19	3	24	1	142.96
20	3	25	2	246.76
21	3	26	2	499.40
22	3	28	0	804.76
23	3	28	1	671.67
24	3	30	2	1382.03
25	3	32	1	2821.16
26	3	34	2	3325.67
27	3	37	1	7088.06
28	3	38	2	9395.13
29	3	41	2	20441.10
30	3	43	0	29204.70

Table 4: The payoff function $u^i(s_j^i, s^{-i})$ is generated from 0 to 10 randomly.

Prob.	Num$N$	Num$S$	NumEquilibra	Time (s)
1	5	3	4	1.09
2	5	3	3	1.18
3	5	3	2	1.19
4	5	4	3	6.85
5	5	4	4	5.26
6	5	5	2	98.95
7	5	5	3	152.76
8	5	5	1	122.88
9	5	6	4	2092.13
10	5	6	2	1693.61

Let $\beta = 1000$; 10 instances have been solved for each range of $\{u^i\}_{i\in N}$. Numerical results for each range are presented in Tables 4, 5, and 6, respectively.

From Tables 1, 2, and 3, one can see that this mixed 0-1 linear programming method can find all pure-strategy Nash equilibria of three-person game effectively. For example, in Table 1 the problem 30 with 45 strategies can be solved in 16992.10 seconds and the number of equilibria is 26. It has been shown that every n-person game can be reduced to a three-person game in polynomial time in [47]. However, some five-person games are still solved by this method in our examples in order to test the effectiveness of this mixed 0-1 linear programming approach. The numerical results for five-person can be obtained from Tables 4, 5, and 6. All the numerical results show that this new method can handle all pure-strategy Nash equilibria of n-person successfully. However, the computation time increases quickly with the increasing of the problems' dimension. To tackle this problem, the distributed computation, which is a salient feature of the integer programming, will be considered in our next work. Besides, some similar problems with multilinear terms could be solved by this method too.

Table 5: The payoff function $u^i(s^i_j, s^{-i})$ is generated from 0 to 50 randomly.

Prob.	Num$N$	Num$S$	NumEquilibra	Time (s)
1	5	3	3	1.20
2	5	3	1	0.99
3	5	3	1	1.71
4	5	4	1	5.69
5	5	4	1	5.62
6	5	4	1	11.90
7	5	4	3	6.77
8	5	5	1	1996.60
9	5	6	2	2031.19
10	5	7	3	18094.80

Table 6: The payoff function $u^i(s^i_j, s^{-i})$ is generated from 0 to 100 randomly.

Prob.	Num$N$	Num$S$	NumEquilibra	Time (s)
1	5	3	0	0.80
2	5	4	0	10.98
3	5	4	0	11.11
4	5	4	1	6.93
5	5	4	0	11.07
6	5	5	1	99.30
7	5	5	0	138.33
8	5	6	1	2408.44
9	5	6	2	1960.49
10	5	7	3	22237.60

4. Conclusions

In this paper, a mixed 0-1 linear programming approach has been proposed to find all pure-strategy Nash equilibria of a finite n-person game in normal form. This method is based on the properties of pure strategy and multilinear terms in the payoff functions. Some numerical results have been given too. The results are promising. However, there is still some interesting future work to be done in the next step. For one thing, as a nice feature of integer programming, the computation will be implemented in a distributed way, which can significantly speed up the efficiency. For another, more research about the properties of strategy and multilinear terms has to be done in order to compute more general Nash equilibria, mixed-strategy Nash equilibria.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was partially supported by the RCN (Research Council of Norway) DECOFF project number 225231/O70 and by GRF: CityU 112910 of Hong Kong SAR Government.

References

[1] X. Chen and X. Deng, “Settling the complexity of two-player Nash equilibrium,” in *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS ’06)*, pp. 261–270, Berkeley, Calif, USA, October 2006.

[2] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, “The complexity of computing a Nash equilibrium,” in *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC ’06)*, pp. 71–78, ACM, New York, NY, USA, 2006.

[3] C. E. Lemke and J. T. Howson, “Equilibrium points of bimatrix games,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 12, pp. 413–423, 1964.

[4] J. Rosenmüller, “On a generalization of the Lemke-Howson algorithm to noncooperative n-person games,” *SIAM Journal on Applied Mathematics*, vol. 21, pp. 73–79, 1971.

[5] R. Wilson, “Computing equilibria of n-person games,” *SIAM Journal on Applied Mathematics*, vol. 21, pp. 80–87, 1971.

[6] J. F. Nash, “Equilibrium points in n-person games,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 36, pp. 48–49, 1950.

[7] J. F. Nash, “Non-cooperative games,” *Annals of Mathematics*, vol. 54, no. 2, pp. 286–295, 1951.

[8] H. Scarf, “The approximation of fixed points of a continuous mapping,” *SIAM Journal on Applied Mathematics*, vol. 15, pp. 1328–1343, 1967.

[9] E. L. Allgower and K. Georg, “Piecewise linear methods for nonlinear equations and optimization,” *Journal of Computational and Applied Mathematics*, vol. 124, no. 1-2, pp. 245–261, 2000.

[10] C. Y. Dang, “The D1-triangulation of Rn for simplicial algorithms for computing solutions of nonlinear equations,” *Mathematics of Operations Research*, vol. 16, no. 1, pp. 148–161, 1991.

[11] B. C. Eaves, “Homotopies for computation of fixed points,” *Mathematical Programming*, vol. 3, pp. 1–22, 1972.

[12] C. B. Garcia and W. I. Zangwill, *Pathways to Solutions, Fixed Points, and Equilibria*, Prentice Hall, Englewood Cliffs, NJ, USA, 1981.

[13] M. Kojima and Y. Yamamoto, “A unified approach to the implementation of several restart fixed point algorithms and a new variable dimension algorithm,” *Mathematical Programming*, vol. 28, no. 3, pp. 288–328, 1984.

[14] G. van der Laan and A. J. J. Talman, “A restart algorithm for computing fixed points without an extra dimension,” *Mathematical Programming*, vol. 17, no. 1, pp. 74–84, 1979.

[15] M. J. Todd, *The Computation of Fixed Points and Applications*, vol. 1 of *Lecture Notes in Economics and Mathematical Systems*, Springer, Berlin, Germany, 1976.

[16] C. B. Garcia, C. E. Lemke, and H. Luethi, “Simplicial approximation of an equilibrium point of noncooperative n-person games,” *Mathematical Programming*, vol. 4, pp. 227–260, 1973.

[17] T. M. Doup and A. J. J. Talman, “A new simplicial variable dimension algorithm to find equilibria on the product space of unit simplices,” *Mathematical Programming*, vol. 37, no. 3, pp. 319–355, 1987.

[18] T. M. Doup and A. J. J. Talman, “A continuous deformation algorithm on the product space of unit simplices,” *Mathematics of Operations Research*, vol. 12, no. 3, pp. 485–521, 1987.

[19] G. van der Laan, A. J. J. Talman, and L. van der Heyden, “Simplicial variable dimension algorithms for solving the nonlinear complementarity problem on a product of unit simplices using a general labelling,” *Mathematics of Operations Research*, vol. 12, no. 3, pp. 377–397, 1987.

[20] A. H. van den Elzen and A. J. J. Talman, "A procedure for finding Nash equilibria in bi-matrix games," *Zeitschrift für Operations Research: Mathematical Methods of Operations Research*, vol. 35, no. 1, pp. 27–43, 1991.

[21] A. van den Elzen and D. Talman, "An algorithmic approach toward the tracing procedure for bi-matrix games," *Games and Economic Behavior*, vol. 28, no. 1, pp. 130–145, 1999.

[22] J. C. Harsanyi, "The tracing procedure: a Bayesian approach to defining a solution for *n*-person noncooperative games," *International Journal of Game Theory*, vol. 4, no. 1-2, pp. 61–94, 1975.

[23] P. J.-J. Herings and A. van den Elzen, "Computation of the Nash equilibrium selected by the tracing procedure in *n*-person games," *Games and Economic Behavior*, vol. 38, no. 1, pp. 89–117, 2002.

[24] R. D. McKelvey and A. McLennan, "Computation of equilibria in finite games," in *Handbook of Computational Economics*, vol. 1, pp. 87–142, North-Holland, Amsterdam, The Netherlands, 1996.

[25] B. Von Stengel, "Computing equilibria for two-person games," in *Handbook of Game Theory with Economic Applications*, vol. 3, pp. 1723–1759, 2002.

[26] J. C. Harsanyi and R. Selten, *A General Theory of Equilibrium Selection in Games*, vol. 1, MIT Press, Cambridge, Mass, USA, 1988.

[27] S. H. Schanuel, L. K. Simon, and W. R. Zame, "The algebraic geometry of games and the tracing procedure," *Game Equilibrium Models*, vol. 2, pp. 9–43, 1991.

[28] P. J.-J. Herings and R. J. A. P. Peeters, "A differentiable homotopy to compute Nash equilibria of *n*-person games," *Economic Theory*, vol. 18, no. 1, pp. 159–185, 2001.

[29] S. Govindan and R. Wilson, "A global Newton method to compute Nash equilibria," *Journal of Economic Theory*, vol. 110, no. 1, pp. 65–86, 2003.

[30] S. Smale, "A convergent process of price adjustment and global Newton methods," *Journal of Mathematical Economics*, vol. 3, no. 2, pp. 107–120, 1976.

[31] P. J.-J. Herings and R. J. A. P. Peeters, "Homotopy methods to compute equilibria in game theory," *Economic Theory*, vol. 42, no. 1, pp. 119–156, 2010.

[32] S. Yin, S. X. Ding, A. Haghani, H. Hao, and P. Zhang, "A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark tennessee eastman process," *Journal of Process Control*, vol. 22, no. 9, pp. 1567–1581, 2012.

[33] S. Yin, S. X. Ding, A. H. A. Sari, and H. Hao, "Data-driven monitoring for stochastic systems and its application on batch process," *International Journal of Systems Science*, vol. 44, no. 7, pp. 1366–1376, 2013.

[34] H. Zhang, Y. Shi, and J. Wang, "On energy-to-peak filtering for nonuniformly sampled nonlinear systems: a Markovian jump system approach," *IEEE Transactions on Fuzzy Systems*, vol. 22, no. 1, pp. 212–222, 2014.

[35] S. Yin, H. Luo, and S. Ding, "Real-time implementation of fault-tolerant control systems with performance optimization," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 5, pp. 2402–2411, 2013.

[36] H. Zhang, X. Zhang, and J. Wang, "Robust gain-scheduling energy-to-peak control of vehicle lateral dynamics stabilisation," *Vehicle System Dynamics*, 2014.

[37] M. M. Kostreva and L. A. Kinard, "A differentiable homotopy approach for solving polynomial optimization problems and noncooperative games," *Computers & Mathematics with Applications*, vol. 21, no. 6-7, pp. 135–143, 1991.

[38] P. J.-J. Herings and R. J. A. P. Peeters, "A globally convergent algorithm to compute all Nash equilibria for *n*-person games," *Annals of Operations Research*, vol. 137, no. 1, pp. 349–368, 2005.

[39] R. S. Datta, "Finding all Nash equilibria of a finite game using polynomial algebra," *Economic Theory*, vol. 42, no. 1, pp. 55–96, 2010.

[40] T. Sandholm, A. Gilpin, and V. Conitzer, "Mixed-integer programming methods for finding Nash equilibria," in *Proceedings of the 20th National Conference on Artificial Intelligence*, pp. 495–501, July 2005.

[41] D. Avis, G. D. Rosenberg, R. Savani, and B. von Stengel, "Enumeration of Nash equilibria for two-player games," *Economic Theory*, vol. 42, no. 1, pp. 9–37, 2010.

[42] G. Gottlob, G. Greco, and F. Scarcello, "Pure Nash equilibria: hard and easy games," in *Proceedings of the 9th Conference on Theoretical Aspects of Rationality and Knowledge*, pp. 215–230, ACM, 2003.

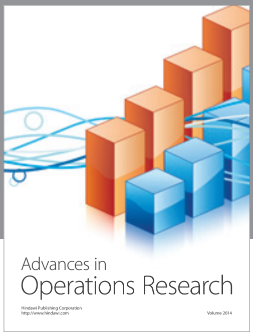
[43] M. Jiang, "Finding pure Nash equilibrium of graphical game via constraints satisfaction approach," in *Combinatorics, Algorithms, Probabilistic and Experimental Methodologies*, pp. 483–494, Springer, New York, NY, USA, 2007.

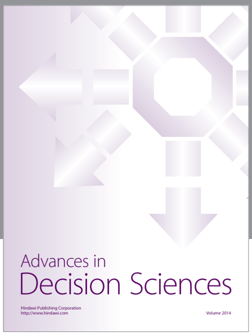
[44] A. C. Chapman, A. Farinelli, E. M. de Cote, A. Rogers, and N. R. Jennings, "A distributed algorithm for optimising over pure strategy Nash equilibria," in *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pp. 749–755, Atlanta, Ga, USA, July 2010.

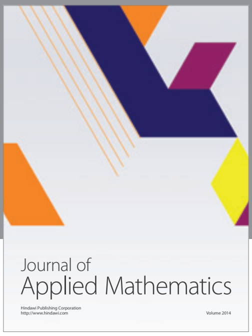
[45] G. Gottlob, G. Greco, and T. Mancini, "Complexity of pure equilibria in bayesian games," in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1294–1299, 2007.

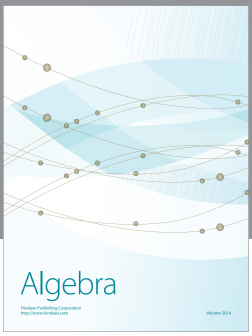
[46] A. C. Chapman, *Control of large distributed systems using games with pure strategy Nash equilibria [Ph.D. thesis]*, University of Southampton, 2009.

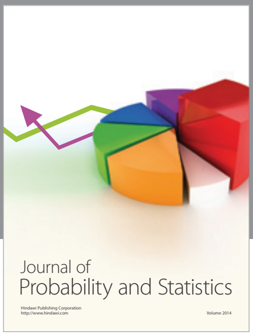
[47] V. Bubelis, "On equilibria in finite games," *International Journal of Game Theory*, vol. 8, no. 2, pp. 65–79, 1979.


Advances in
Operations Research

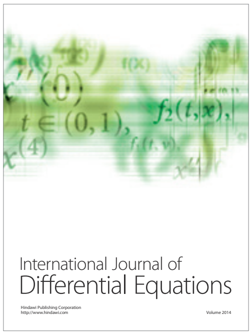

Advances in
Decision Sciences

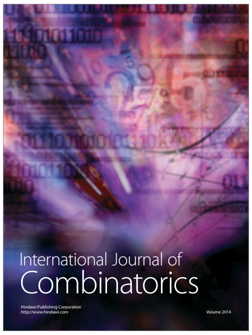

Journal of
Applied Mathematics

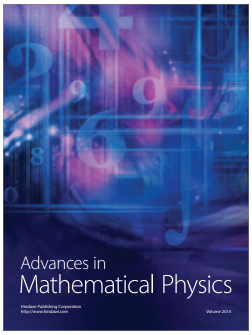

Algebra

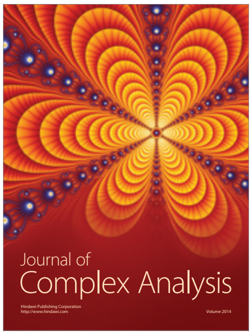

Journal of
Probability and Statistics

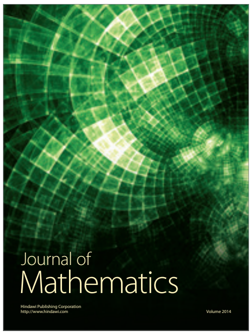
The Scientific
World Journal

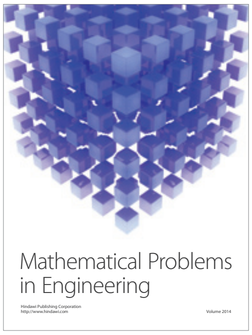

International Journal of
Differential Equations

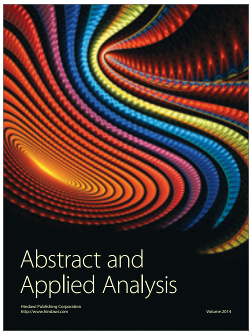

International Journal of
Combinatorics

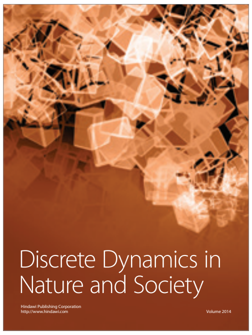

Advances in
Mathematical Physics

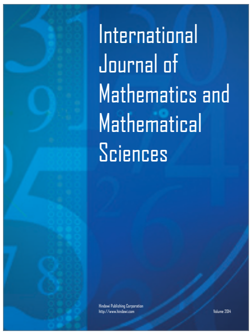

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

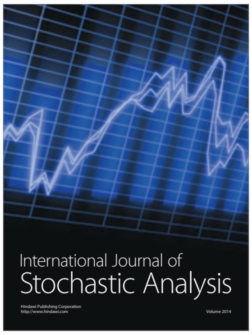

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization