

Polynomial Time Method for Solving Nash Equilibria of Zero-Sum Games

Yoshihiro Tanaka¹, Mitsuru Togashi²

¹Faculty of Economics and Business, Hokkaido University, Sapporo, Japan

²Graduate School of Accountancy, Waseda University, Tokyo, Japan

Email: tanaka@econ.hokudai.ac.jp, twtthuxhmvjtj@gmail.com

How to cite this paper: Tanaka, Y. and Togashi, M. (2021) Polynomial Time Method for Solving Nash Equilibria of Zero-Sum Games. *American Journal of Computational Mathematics*, 11, 23-30.

<https://doi.org/10.4236/ajcm.2021.111002>

Received: January 15, 2021

Accepted: March 14, 2021

Published: March 17, 2021

Copyright © 2021 by author(s) and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

There are a few studies that focus on solution methods for finding a Nash equilibrium of zero-sum games. We discuss the use of Karmarkar's interior point method to solve the Nash equilibrium problems of a zero-sum game, and prove that it is theoretically a polynomial time algorithm. We implement the Karmarkar method, and a preliminary computational result shows that it performs well for zero-sum games. We also mention an affine scaling method that would help us compute Nash equilibria of general zero-sum games effectively.

Keywords

Zero-Sum Games, Nash Equilibria, Karmarkar's Method, Polynomial Time

1. Introduction

It is well known that John von Neumann the existence of Nash equilibria of zero-sum games in the late 1920s [1].

Dantzig and Thapa [2] mentioned the linear programming formulation of a zero-sum game, and considered reducing linear primal and dual problems to a zero-sum game.

Khachian [3] and Karmarkar's [4] interior-point methods for solving linear programming problems in polynomial time are significant discoveries.

Khachian's ellipsoid method [3] is important in that it is a first polynomial time algorithm, however, computational results with it have been disappointing.

Karmarkar's projective scaling method [4] is considered to be a practical polynomial algorithm. Numerous attempts have been made to refine a variety of interior point methods in the late 1980s [5] [6]. There are few studies concerning computational methods (e.g., [7]) to solve a Nash equilibrium of a zero-sum

game, in spite of those developments in linear programming.

We discuss the use of the Karmarkar’s method to solve a Nash equilibrium of a zero-sum game, which is expressed as linear programming problems derived from the original zero-sum game, and prove that it is without doubt a polynomial time algorithm. We implement the Karmarkar method, and apply it for solving Rock-Paper-Scissors, of which result seems to be promising.

Finally, we also mention an affine scaling method that would help us compute a Nash equilibrium effectively.

2. Formulation

Dantzig and Thapa [2] mentioned that maximin strategy of Player 1 in a zero-sum game can be formulated as the following linear programming problem:

$$\begin{aligned} & \underset{x, \xi}{\text{maximize}} && \xi \\ & \text{subject to} && A^T x \geq \xi e \\ & && \sum_{i=1}^n x_i = 1, x_i \geq 0 \end{aligned} \tag{1}$$

where $\xi \in \mathbb{R}, e = (1, \dots, 1)^T \in \mathbb{R}_+^n, A \in \mathbb{R}^{n \times n}$ is a payoff matrix, and $x \in \mathbb{R}_+^n$ is a mixed strategy vector of Player 1.

Likewise, the minimax strategy of Player 2 can be formulated as the following linear programming problem:

$$\begin{aligned} & \underset{y, \eta}{\text{minimize}} && \eta \\ & \text{subject to} && Ay \leq \eta e \\ & && \sum_{i=1}^n y_i = 1, y_i \geq 0 \end{aligned} \tag{2}$$

where $y \in \mathbb{R}_+^n$ is a mixed strategy of Player 2, which is, in fact, the linear programming dual problem of (1).

The following result which clarifies the relationship between (1) and (2).

Theorem 1. *There exist solutions $\{\bar{x}, \bar{\xi}\}$ and $\{\bar{y}, \bar{\eta}\}$ for (1) and (2). Moreover,*

$$\bar{\xi} = \bar{\eta}.$$

Proof We let $\xi = \xi_1 - \xi_2, \xi_1 \geq 0, \xi_2 \geq 0$, and rewrite (1) as

$$\begin{aligned} & \underset{x, \xi_1, \xi_2}{\text{minimize}} && -(\xi_1 - \xi_2) \\ & \text{subject to} && A^T x - (\xi_1 - \xi_2)e \geq 0 \\ & && e^T x \geq 1 \\ & && -e^T x \geq -1 \\ & && x_i \geq 0, i = 1, \dots, n, \xi_1, \xi_2 \geq 0 \end{aligned} \tag{3}$$

The dual linear program of (1) is

$$\begin{aligned} & \underset{y, \eta_1, \eta_2}{\text{maximize}} && \eta_1 - \eta_2 \\ & \text{subject to} && Ay + (\eta_1 - \eta_2)e \leq 0 \\ & && -e^T y \leq -1 \\ & && e^T y \leq 1 \\ & && y_i \geq 0, i = 1, \dots, n, \eta_1, \eta_2 \geq 0 \end{aligned} \tag{4}$$

Note that (4) is equivalent to (2) by letting $\eta = -(\eta_1 - \eta_2)$.

There exists a feasible point for (3), such as $x' = \frac{1}{n}e$, $\xi'_1 = \min_j \sum_i a_{ij}$, $\xi'_2 = 0$. We also observe that (3) has a finite solution, as $(\xi_1 - \xi_2)$ should be bounded because $x \in \Delta^{n-1}$, where Δ^{n-1} is an $n-1$ simplex. Therefore, $-\bar{\xi} = -(\bar{\xi}_1 - \bar{\xi}_2) = \bar{\eta}_1 - \bar{\eta}_2 = -\bar{\eta}$ because the strong duality theorem [8] holds. \square

The above result shows that (2) is the linear programming dual problem of (1).

Next, we establish the validity of the above formulation. Let $S_x = \{s_1, \dots, s_n\}$ and $S_y = \{s_1, \dots, s_n\}$ be sets of pure strategies of Players 1 and 2, respectively. Let $E(x, y)$ denote the expected payoff value for Player 1 when Player 1 takes x and Player 2 takes y as a mixed strategy, that is,

$$\begin{aligned} E(x, y) &= \sum_{i=1}^n x_i E(s_i, y) \\ &= \sum_{i=1}^n x_i \left(\sum_{j=1}^n y_j E(s_i, s_j) \right) \\ &= \sum_{i=1}^n x_i \left(\sum_{j=1}^n y_j a_{ij} \right) = x^T A y \end{aligned} \tag{5}$$

Theorem 2.

$$E(x, \bar{y}) \leq E(\bar{x}, \bar{y}) \leq E(\bar{x}, y). \tag{6}$$

Proof Because \bar{x} and \bar{y} are the solutions for (1) and (2), respectively,

$$A\bar{y} \leq \alpha e \leq A^T \bar{x},$$

where $\alpha = \bar{\xi} = \bar{\eta}$ from Theorem 1.

Therefore,

$$x^T A\bar{y} \leq \alpha \text{ and } \alpha \leq y^T A^T \bar{x} = \bar{x}^T A y$$

namely,

$$E(x, \bar{y}) \leq E(\bar{x}, \bar{y}) \leq E(\bar{x}, y). \square$$

The result from Theorem 2 is known as the minimax theorem [1]; however, the direct derivation above would be simple and easy to comprehend.

3. Karmarkar Method

We can use the simplex method to find a Nash equilibrium from the results of Section 2.

However, the curse of dimensionality may occur if the running time required to solve a linear problem using the simplex method may increase rapidly as the number of variables increases.

Therefore, we employ a variation of Karmarkar’s method [9] which belongs to an interior point method and assures the polynomial time convergence property.

The Karmarkar method [9] we adopt deals with the following canonical form:

$$\begin{aligned} \text{minimize } & z = c^T x \\ \text{subject to } & Ax = 0 \\ & a^T x = 1 \\ & x \geq 0 \end{aligned} \tag{7}$$

where $c, x \in \mathbb{R}^n$, $a = \left(\overbrace{1, \dots, 1}^n, 0, \dots, 0 \right)^T \in \mathbb{R}_+^n$, and $A \in \mathbb{R}^{n \times n}$.

If $c^T \bar{x} = 0$, where \bar{x} is the solution of (7), we refer to (7) as the *canonical form*.

It should be noted that the original Karmarkar's method [4] utilizes $e = (1, \dots, 1)^T \in \mathbb{R}_+^n$ instead of a . In this study, we utilize a because we can formulate the part of components of x for mixed strategies as $a^T x = 1$ in (14) and (15) later.

The key projective transformation sends x to

$$\tilde{x} = \frac{X^{-1}x}{e^T X^{-1}x}$$

where $X = \text{diag}(x_k)$. We notice that $\tilde{x} \in \Delta^{n-1}$, where Δ^{n-1} is $(n-1)$ -simplex. The corresponding inverse transformation is

$$x = \frac{X\tilde{x}}{a^T X\tilde{x}}$$

As seen in [9], the problem in (7) is transformed as

$$\begin{aligned} &\text{minimize } \tilde{c}^T \tilde{x} \\ &\text{subject to } \tilde{A}\tilde{x} = 0 \\ &\quad e^T \tilde{x} = 1 \\ &\quad \tilde{x} \geq 0 \end{aligned} \tag{8}$$

where $\tilde{c} = Xc$ and $\tilde{A} = AX$. We denote the constraint matrix using

$$B = \begin{pmatrix} \tilde{A} \\ e^T \end{pmatrix},$$

and the corresponding orthogonal projection matrix is $P_B = I - B^T(BB^T)^{-1}B$.

The projected steepest-descent direction is

$$\Delta\tilde{x} = -P_B \tilde{c} = -P_B Xc = -\left(\tilde{P} - \frac{1}{n} ee^T \right) Xc = -\tilde{P}Xc + \frac{c^T x^k}{n} e \tag{9}$$

At each step, the next point \tilde{x}^{k+1} is given by

$$\tilde{x}^{k+1} = \frac{e}{n} + \alpha \Delta\tilde{x}, \quad \alpha > 0,$$

more precisely,

$$\tilde{x}^{k+1} = \frac{e}{n} + \frac{1}{\sqrt{n(n+1)} \|\Delta\tilde{x}\|} \cdot \Delta\tilde{x}, \tag{10}$$

so that \tilde{x}^{k+1} will be an inner point in Δ^{n-1} .

The next point x^{k+1} in x -space is obtained by

$$x^{k+1} = \frac{X\tilde{x}^{k+1}}{a^T X\tilde{x}^{k+1}}. \tag{11}$$

Finally, we define the optimality criterion as

$$c^T x^k < 2^{-q} c^T x^0, \tag{12}$$

where 2^{-q} is a prescribed precision.

Next, we describe the Karmarkar method as follows:

Algorithm 1.

Input: $A \in \mathbb{Z}^{n \times n}$, $c \in \mathbb{Z}^n$, a feasible point x^0 for (7)

Output: x^* such that $c^T x^* = 0$, $Ax^* = 0$, $a^T x^* = 1$, $x^* \geq 0$.

1: Compute $\Delta \tilde{x}$ by (9).

2: Determine \tilde{x}^{k+1} by (10).

3: Obtain x^{k+1} using (11). Check the optimality criterion using (12).

4: $k \leftarrow k + 1$. Go to Step 1.

Karmarkar defined a potential function as follows:

$$f(x) = n \ln c^T x - \sum_{j=1}^n \ln x_j = \sum_{j=1}^n \ln \left(\frac{c^T x}{x_j} \right). \tag{13}$$

He proved that Algorithm 1 generates, $\{x^k\}$, which reduces $f(x^k)$ by $\gamma > 0$ at each iteration.

In fact, the following result follows for original Karmarkar’s algorithm [4].

Theorem 3. *Let $\{x^k\}$ be the sequence generated by Karmarkar’s algorithm [4] applied to (3). Then, (3) can be solved in polynomial time.*

Proof We can transform (3) to original Karmarkar’s original canonical form [4] in polynomial time in the manner of [4]. Then the result follows since the original canonical form can be solved in at most $(n(q + \log_2 n))$ iterations from Theorem 1 [4]. \square

We can establish the following result.

Theorem 4. *A Nash equilibrium of zero-sum games can be found in polynomial time.*

Proof Nash equilibria of zero-sum games can be formulated as (1) and (2), even if the value of the game is not necessarily 0. We can rewrite (1) and (2) as (3) and (4) respectively. (3) (and (4)) can be solved in polynomial time from Theorem 3. \square

We should observe that the transformation in the above proof is used only for the proof itself. When $\bar{\xi} \neq 0$, we can employ the dual problem scheme [9, Chapter 17.5] to update an estimated lower bound to, z^* , which is used to transform (3) into the canonical form (7).

4. Computational Results

In this section, we employ the Karmarkar method described in the previous section.

The zero-sum game (1) and (2) can be rewritten as:

$$\begin{aligned} & \underset{x, \xi, s}{\text{minimize}} && -(\xi_1 - \xi_2) \\ & \text{subject to} && -A^T x + (\xi_1 - \xi_2)e + (s_1, \dots, s_n)^T = 0 \\ & && e^T x = 1 \\ & && x_i \geq 0, i = 1, \dots, n, \xi_1, \xi_2 \geq 0, s_i \geq 0, i = 1, \dots, n \end{aligned} \tag{14}$$

and

$$\begin{aligned}
 & \underset{y, \eta, t}{\text{maximize}} && \eta_1 - \eta_2 \\
 & \text{subject to} && Ay + (\eta_1 - \eta_2)e + (t_1, \dots, t_n)^T = 0 \\
 & && e^T y = 1 \\
 & && y_i \geq 0, i = 1, \dots, n, \quad \eta_1, \eta_2 \geq 0, \quad t_i \geq 0, i = 1, \dots, n
 \end{aligned} \tag{15}$$

where $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{n \times n}$, $e = (1, \dots, 1)^T \in \mathbb{R}^n$, $y \in \mathbb{R}^n$.

These problems belong to the canonical form (7), which can be solved by Algorithm 1.

Rock-Paper-Scissors is used as a test problem of zero-sum games. Its payoff matrix for Player 1 (gain) and Player 2 (loss) is shown in **Table 1**.

The Nash equilibrium of Rock-Paper-Scissors is

$$\{\bar{x}, \bar{y}\} = \left\{ \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3} \right), \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3} \right) \right\}$$

and the value of the game is 0.

It should be noted that (14) and (15) are exactly the same because $A^T = -A$ in this case. Thus, we have only the results of (14) to demonstrate, namely, the problem is:

$$\begin{aligned}
 & \underset{x, \xi, s}{\text{minimize}} && -(\xi_1 - \xi_2) \\
 & \text{subject to} && -x_2 + x_3 + \xi_1 - \xi_2 + s_1 = 0 \\
 & && x_1 - x_3 + \xi_1 - \xi_2 + s_2 = 0 \\
 & && -x_1 + x_2 + \xi_1 - \xi_2 + s_3 = 0 \\
 & && x_1 + x_2 + x_3 = 1 \\
 & && x_1, x_2, x_3 \geq 0, \quad \xi_1, \xi_2 \geq 0, \quad s_1, s_2, s_3 \geq 0
 \end{aligned} \tag{16}$$

The Karmarkar method was coded in Python, and run on a Windows PC (Core i7 8550U, RAM 16GB).

Table 2 summarizes the sequence generated by the Karmarkar method applied to (16) from $x^0 = (0.8, 0.1, 0.1)$, $\xi_1^0 = 0.1$, $\xi_2^0 = 0.9$, $s^0 = (0.8, 0.1, 1.5)$. We can see that the objective value decreases by half in each iteration, which with (12) validates Theorem 3.

Figure 1 shows the sequences generated by the Karmarkar method from three different starting points

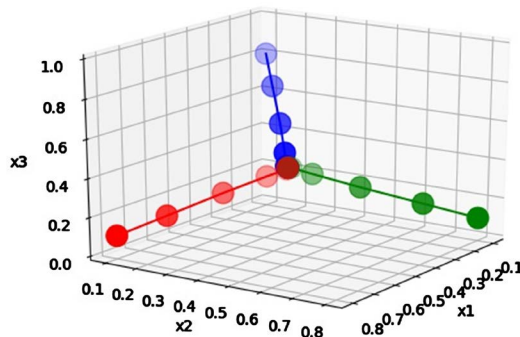


Figure 1. Sequences for different starting points.

Table 1. Payoff matrix of rock-paper-scissors for player 1 (P1) and player 2 (P2).

P2 \ P1	Rock	Paper	Scissors
Rock	0	-1	1
Paper	1	0	-1
Scissors	-1	1	0

Table 2. Details for $x^0 = (0.8, 0.1, 0.1)$.

iteration	x_1	x_2	x_3	objective value
0	0.8	0.1	0.1	0.8
1	0.66527	0.16691	0.16782	0.64421
2	0.51157	0.24279	0.24565	0.43445
3	0.39292	0.30417	0.30291	0.23486
4	0.33965	0.33292	0.32742	0.10883
5	0.33317	0.33338	0.33345	0.04942
6	0.33335	0.33333	0.33332	0.02240
7	0.33333	0.33333	0.33333	0.01014
8	0.33333	0.33333	0.33333	4.59E-3

$$x^0 = (0.8, 0.1, 0.1), (0.1, 0.8, 0.1) \text{ and } (0.1, 0.1, 0.8).$$

It seems that each sequence tends linearly to the Nash equilibrium from an arbitrary interior point.

Table 2 and **Figure 1** show that the Karmarkar method is effective and efficient even for small-size problems.

5. Concluding Remarks

Linear programming formulations can represent the minimax principle for zero-sum games.

We have proved that the Karmarkar's algorithm solves the linear programming problems in polynomial time.

The Karmarkar method performs well when the value of the game is 0. If the value of the game is not 0, we can utilize the dual problem scheme ([9], Chapter 17.5).

We can resort to an affine scaling algorithm [10] because, either (1) or (2) is the linear programming standard form generated by introducing slack variables, regardless of whether or not the value of the game is 0. The affine scaling algorithm for solving equilibria of zero-sum games would be practically effective; however, there is no proof that it has a polynomial time convergence property [11].

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] von Neumann, J. (1928) Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, **100**, 295-320. <https://doi.org/10.1007/BF01448847>
- [2] Dantzig, G.B. and Thapa, M.N. (1997) Linear Programming 1: Introduction. Springer-Verlag, New York, 166.
- [3] Khachiyan, L.G. (1979) A Polynomial Algorithm in Linear Programming. *Doklady Akademii Nauk*, **244**, 1093-1096.
- [4] Karmarkar, N. (1984) A New Polynomial Time Algorithm for Linear Programming. *Combinatorica*, **4**, 373-395. <https://doi.org/10.1007/BF02579150>
- [5] Renegar, J. (1988) A Polynomial-Time Algorithm, Based on Newton's Method, for Linear Programming. *Mathematical Programming*, **40**, 59-93. <https://doi.org/10.1007/BF01580724>
- [6] Vanderbei, R.J., Meketon, M.S. and Freedman, B.A. (1986) A Modification of Karmarkar's Linear Programming Algorithm. *Algorithmica*, **1**, 395-407. <https://doi.org/10.1007/BF01840454>
- [7] Daskalakis, C., Deckelbaum, A. and Kim, A. (2015) Near-Optimal No-Regret Algorithms for Zero-Sum Games. *Games and Economic Behavior*, **100**, 327-348. <https://doi.org/10.1016/j.geb.2014.01.003>
- [8] Nocedal, J. and Wright, S.J. (2006) Numerical Optimization. 2nd Edition, Springer, Berlin.
- [9] Nash, S.G. and Sofer, A. (1996) Linear and Nonlinear Programming. McGraw-Hill, New York.
- [10] Dikin, I.I. (1967) Iterative Solution of the Problems of Linear and Quadratic Programming. *Soviet Mathematics Doklady*, **8**, 674-675.
- [11] Tsuchiya, T. and Muramatsu, M. (1995) Global Convergence of a Long-Step Affine Scaling Algorithm for Degenerate Linear Programming Problems. *SIAM Journal on Optimization*, **5**, 525-551. <https://doi.org/10.1137/0805027>