AIMS Mathematics, 8(6): 13984–14007.
DOI:10.3934/math.2023715
Received: 07 November 2022
Revised: 04 April 2023
Accepted: 06 April 2023
Published: 14 April 2023

http://www.aimspress.com/journal/Math

Research article

Finding the Nash equilibria of n-person noncooperative games via solving the system of equations

Huimin Li[1,*], **Shuwen Xiang**[2,3,*], **Shunyou Xia**[4] **and Shiguo Huang**[2,5]

[1] School of Mathematics and Statistics, Xuzhou University of Technology, Xuzhou 221018, China

[2] College of Mathematics and Statistics, Guizhou University, Guiyang 550025, China

[3] College of Mathematics and Information Science, Guiyang University, Guiyang 550005, China

[4] School of Mathematics and Big Data, Guizhou Education University, Guiyang 550025, China

[5] Department of Mathematics and Information Science, Zhengzhou University of Light Industry, Zhengzhou 450002, China

* **Correspondence:** Email: lihuimin@xzit.edu.cn, shwxiang@vip.163.com.

Abstract: In this paper, we mainly study the equivalence and computing between Nash equilibria and the solutions to the system of equations. First, we establish a new equivalence theorem between Nash equilibria of n-person noncooperative games and solutions of algebraic equations with parameters, that is, finding a Nash equilibrium point of the game is equivalent to solving a solution of the system of equations, which broadens the methods of finding Nash equilibria and builds a connection between these two types of problems. Second, an adaptive differential evolution algorithm based on cultural algorithm (ADECA) is proposed to compute the system of equations. The ADECA algorithm applies differential evolution (DE) algorithm to the population space of cultural algorithm (CA), and increases the efficiency by adaptively improving the mutation factor and crossover operator of the DE algorithm and applying new mutation operation. Then, the convergence of the ADECA algorithm is proved by using the finite state Markov chain. Finally, the new equivalence of solving Nash equilibria and the practicability and effectiveness of the algorithm proposed in this paper are verified by computing three classic games.

Keywords: n-person noncooperative game; the system of equations; Nash equilibrium; adaptive differential culture algorithm
Mathematics Subject Classification: 68W50, 91A06, 91A10

1. Introduction

Game theory is a mathematical theory that studies interactions between decision makers. The n-person noncooperative game is an important part of game theory [1–3]. As we all know, Von Neumann [4] and Nash [5, 6] established the models of matrix games and n-person finite noncooperative games respectively, and Nash proposed the core equilibrium concept in noncooperative games, called Nash equilibrium, and proved that this kind game has at least one mixed strategies equilibrium point. At present, the n-person noncooperative game and its related researches have been widely used in mathematics, economics, operations research, biology, textile industry and other fields [7–11]. In these researches and applications, the key point is how to find Nash equilibrium points effectively, which largely depends on the good mathematical description of the problem. Therefore, it is very important to find a suitable mathematical model when solving games.

At present, there are many mathematical methods to solve the Nash equilibria, mainly by equating finding Nash equilibria to a fixed point problem, an optimization problem or a tensor complementarity problem. The finding Nash equilibria is equivalent to the fixed point problem, which is mainly handled by the Brouwer fixed point theorem, the Kakutani fixed point theorem and the KyFan point theorem. Many scholars have made a lot of researches in this field [12–15]. Furthermore, the research that finding Nash equilibria is equivalent to the optimization problem is springing up. Because the equilibrium points are natural actions of self interested behavior in competitive situations. In other words, at an equilibrium point, each competitor is maximizing his payoff against all other competitive actions in the situation. Therefore, finding a Nash equilibrium is generally as a nonconvex optimization problem, which has been sufficiently studied in [16–20]. Further, some scholars also have equated the finding Nash equilibria to the tensor complementarity problem [21–24], and the player's payoff function is given by a homogeneous polynomial defined by this player's payoff tensor. In addition to the above usual methods for finding Nash equilibria, in recent years, with the extensive research of multi-agent systems in social science and engineering systems, using distributed algorithms seeking Nash equilibria has become an emerging research topic. This method provides solutions for complex engineering games [25–31], such as generalized games with coupled constraints, aggregative games with non-smooth objective functions and uncertain games, etc.

Although the above research methods and calculation methods have solved a large number of games. With the complexity of real problems and the diversification of games, the above methods have some disadvantages in the computing process, such as high computational complexity, cumbersome derivation process and complex algorithm design. Therefore, we propose a new equivalent method of finding Nash equilibria in this paper, which directly applies the definition of Nash equilibrium to establish an equivalent relationship with the system of equations, and transform the problem of finding the Nash equilibria into solving the system of equations. So that it can be calculated using the developed method of solving the system of equations. Therefore, compared with finding the Nash equilibria equivalent to the fixed point problem and nonconvex optimization problem, the proposed method does not need to define complex set valued mapping, no have tedious theoretical derivation and proof. Compared with find the Nash equilibria equivalent to the tensor complementarity problem, the proposed method does not need to consider the nonlinear complementarity and nonsmooth of the payoff function. This method provides a new idea for find the Nash equilibria, which can be easily understood and applied by readers with weak mathematical foundation. On the one hand, this method

broadens the expression form of Nash equilibria and provides a new method for finding Nash equilibria. It reduces the difficulty of seeking Nash equilibria, and only needs to consider the calculation methods of the equivalent equations. On the other hand, this equivalent method also has certain significance for the realization of Nash equilibrium: since the equations are derived from the definition of Nash equilibrium, each equation essentially represents the rational behavior of the player. In each equation, the probability of each component of the Nash equilibrium is greater than 0 is meaning that the corresponding pure strategy is in the direction of increasing the profit, that is, the direction of Nash equilibrium realization. Therefore, this method has certain research significance.

This paper is organized as follows: In Section 2, we give the basic definitions of the game and an equivalence theorem between the Nash equilibria of n-person noncooperative game and the solutions of equations. In Section 3, we propose the ADECA algorithm and prove its convergence theoretically through the finite state Markov chain. This algorithm applies the DE algorithm to the population space of the CA algorithm, and uses the belief space of the CA to extract hidden evolutionary knowledge and guides the evolution of the population space. Additionally, the DE algorithm is adaptively improved to enhance the performance of the ADECA algorithm. The equivalence theorem of the Nash equilibria and the superiority of the ADECA algorithm proposed in this paper are verified by examples in Section 4.

2. The basic definitions and equivalence theorem of n-person noncooperative game

2.1. The n-person noncooperative game

Definition 2.1. *[32] We consider a* ***n-person noncooperative game*** $\Gamma = \{(N, S^i, P^i, X^i, U^i), i = 1, \ldots, n\}$

where

(1) $N = \{1, \ldots, n\}$ *is the set of players and n is the number of players;*

(2) $S^i = \{s^i_1, \ldots, s^i_{m_i}\}$, $\forall i \in N$ *is the pure strategy set of player i,* m_i *represents the number of strategies available to player i ,* $S = \prod_{i=1}^{n} S^i$*;*

(3) $P^i : S \to \mathbb{R}$, $\forall i \in N$ *represents the payoff function of player i;*

(4) $X^i = \{x^i = (x^i_1, \ldots, x^i_{\mathscr{K}}, \ldots, x^i_{m_i}) : x^i_{\mathscr{K}} \geq 0, \sum_{\mathscr{K}=1}^{m_i} x^i_{\mathscr{K}} = 1\}$, $\forall i \in N$ *is the set of mixed strategies.* $X = \prod_{i=1}^{n} X^i$ *, and each mixed strategy profile meets* $x = (x^1, x^2, \ldots, x^n) \in X$*;*

(5) $U^i : X \to \mathbb{R}$, $\forall i \in N$ *represents the expected payoff function of player i.*

$$u_i(x) = \sum_{\mathscr{K}_1=1}^{m_1} \cdots \sum_{\mathscr{K}_n=1}^{m_n} p_i(s^1_{\mathscr{K}_1}, \ldots, s^n_{\mathscr{K}_n}) \prod_{i=1}^{n} x^i_{\mathscr{K}_i},$$

where $u_i(x)$ *represents the expected payoff value of player i when he chooses a mixed strategy* $x^i = (x^i_1, \ldots, x^i_{m_i}) \in X^i$. $p_i(s^i_{\mathscr{K}_i}, \ldots, s^n_{\mathscr{K}_n})$ *represents the payoff value of player i when each player chooses pure strategy* $s^i_{\mathscr{K}_i} \in S^i, i = 1, \ldots, n$.

Denote by

$$u_i(x \| s^i_{\mathscr{K}_i}) \triangleq u_i(x^1, \ldots, x^{i-1}, s^i_{\mathscr{K}_i}, x^{i+1}, \ldots, x^n) \triangleq$$
$$\sum_{\mathscr{K}_1=1}^{m_1} \cdots \sum_{\mathscr{K}_{i-1}=1}^{m_{i-1}} \sum_{\mathscr{K}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathscr{K}_n=1}^{m_n} p_i(s^1_{\mathscr{K}_1}, \ldots, s^n_{\mathscr{K}_n}) x^1_{\mathscr{K}_1} \ldots x^{i-1}_{\mathscr{K}_{i-1}} x^{i+1}_{\mathscr{K}_{i+1}} \ldots x^n_{\mathscr{K}_n},$$

where $s^i_{\mathscr{K}_i}(1 \le \mathscr{K}_i \le m_i)$ is pure strategy of player i. $(x\|s^i_{\mathscr{K}_i})$ represents $s^i_{\mathscr{K}_i}$ instead of x^i of player i, and the other players do not change their own mixed strategy.

Definition 2.2. *If $x^* = (x^*_1, \ldots, x^*_n)$, such that $u_i(x^*_i, x^*_{i^\wedge}) = \max\limits_{y^i \in X^i} u_i(y^i, x^*_{i^\wedge})$, $\forall i \in N$, then x^* is a **Nash equilibrium point** of n-person noncooperative game, where $i^\wedge = N\backslash\{i\}$, $\forall i \in N$.*

Conclusion 2.1. *[20] Mixed strategy $x^* = (x^*_1, \ldots, x^*_n) \in X$ is a Nash equilibrium point of a game Γ if and only if $u_i(x^*) \ge u_i(x^*\|s^i_{\mathscr{K}_i})$, where, $\mathscr{K}_i = m_1, \ldots, m_n, i = 1, \ldots, n$.*

2.2. Equivalence theorem of the Nash equilibria and solutions of the system of equations

Theorem 2.1. *For any n-person finite noncooperative game, the mixed strategy profile,*

$$x = \left(x^1, x^2, \ldots, x^n\right) \in X, \quad x^i = \left(x^i_1, \ldots, x^i_{\mathfrak{E}_i}, \ldots, x^i_{m_i}\right), \quad \forall i \in N$$

is a Nash equilibrium if and only if there is a set of real numbers

$$\lambda^i_{\mathfrak{E}_i\mathscr{K}_i} \ge 0, \quad i = 1, 2, \ldots, n, \quad \mathfrak{E}_i \ne \mathscr{K}_i, \quad \mathfrak{E}_i, \mathscr{K}_i \in \{1, 2, \ldots, m_i\}$$

such that $\lambda^i_{\mathfrak{E}_i\mathscr{K}_i}$ and $x = \left(x^1, x^2, \ldots, x^n\right)$ satisfy the following equations:

$$\begin{gathered}
\lambda^1_{\mathfrak{E}_1\mathscr{K}_1} + x^1_{\mathfrak{E}_1}\left(\sum_{\mathfrak{E}_2=1}^{m_2}\cdots\sum_{\mathfrak{E}_n=1}^{m_n} p_1\left(s^1_{\mathscr{K}_1}, s^2_{\mathfrak{E}_2}, \ldots, s^n_{\mathfrak{E}_n}\right)\cdot x^2_{\mathfrak{E}_2}\cdot\ldots\cdot x^n_{\mathfrak{E}_n}\right.\\
\left. - \sum_{\mathfrak{E}_2=1}^{m_2}\cdots\sum_{\mathfrak{E}_n=1}^{m_n} p_1\left(s^1_{\mathfrak{E}_1}, s^2_{\mathfrak{E}_2}, \ldots, s^n_{\mathfrak{E}_n}\right)\cdot x^2_{\mathfrak{E}_2}\cdot\ldots\cdot x^n_{\mathfrak{E}_n}\right) = 0 \qquad (2.1)\\
\lambda^1_{\mathfrak{E}_1\mathscr{K}_1} \ge 0, \quad \mathfrak{E}_1 \ne \mathscr{K}_1, \quad \mathfrak{E}_1, \mathscr{K}_1 \in \{1, 2, \ldots, m_1\}\\
\vdots\\
\lambda^i_{\mathfrak{E}_i\mathscr{K}_i} + x^i_{\mathfrak{E}_i}\left(\sum_{\mathfrak{E}_1=1}^{m_1}\cdots\sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}}\sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}}\cdots\sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, \ldots, s^i_{\mathscr{K}_i}, \ldots, s^n_{\mathfrak{E}_n}\right)\cdot x^1_{\mathfrak{E}_1}\cdot\ldots\cdot x^{i-1}_{\mathfrak{E}_{i-1}}\cdot x^{i+1}_{\mathfrak{E}_{i+1}}\cdot\ldots\cdot x^n_{\mathfrak{E}_n}\right.\\
\left. - \sum_{\mathfrak{E}_1=1}^{m_1}\cdots\sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}}\sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}}\cdots\sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, \ldots, s^i_{\mathfrak{E}_i}, \ldots, s^n_{\mathfrak{E}_n}\right)\cdot x^1_{\mathfrak{E}_1}\cdot\ldots\cdot x^{i-1}_{\mathfrak{E}_{i-1}}\cdot x^{i+1}_{\mathfrak{E}_{i+1}}\cdot\ldots\cdot x^n_{\mathfrak{E}_n}\right) = 0\\
\lambda^i_{\mathfrak{E}_i\mathscr{K}_i} \ge 0, \quad \mathfrak{E}_i \ne \mathscr{K}_i, \quad \mathfrak{E}_i, \mathscr{K}_i \in \{1, 2, \ldots, m_i\}\\
\vdots\\
\lambda^n_{\mathfrak{E}_n\mathscr{K}_n} + x^n_{\mathfrak{E}_n}\left(\sum_{\mathfrak{E}_1=1}^{m_1}\cdots\sum_{\mathfrak{E}_{n-1}=1}^{m_{n-1}} p_n\left(s^1_{\mathfrak{E}_1}, \ldots, s^{n-1}_{\mathfrak{E}_{n-1}}, s^n_{\mathscr{K}_n}\right)\cdot x^1_{\mathfrak{E}_1}\cdot\ldots\cdot x^{n-1}_{\mathfrak{E}_{n-1}}\right.\\
\left. - \sum_{\mathfrak{E}_1=1}^{m_1}\cdots\sum_{\mathfrak{E}_{n-1}=1}^{m_{n-1}} p_n\left(s^1_{\mathfrak{E}_1}, \ldots, s^{n-1}_{\mathfrak{E}_{n-1}}, s^n_{\mathfrak{E}_n}\right)\cdot x^1_{\mathfrak{E}_1}\cdot\ldots\cdot x^{n-1}_{\mathfrak{E}_{n-1}}\right) = 0\\
\lambda^n_{\mathfrak{E}_n\mathscr{K}_n} \ge 0, \quad \mathfrak{E}_n \ne \mathscr{K}_n, \quad \mathfrak{E}_n, \mathscr{K}_n \in \{1, 2, \ldots, m_n\}.
\end{gathered}$$

Proof. Necessity: Assume that $x = \left(x^1, x^2, \ldots, x^n\right) \in X, x^i = \left(x_1^i, \ldots, x_{\mathfrak{E}_i}^i, \ldots, x_{m_i}^i\right), \forall i \in N$ is a Nash equilibrium. Then the expected payoff of player i under this equilibrium is

$$\begin{aligned}
u_i(x) &= \sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s_{\mathfrak{E}_1}^1, s_{\mathfrak{E}_2}^2, \ldots, s_{\mathfrak{E}_n}^n\right) \cdot x_{\mathfrak{E}_1}^1 \cdot x_{\mathfrak{E}_2}^2 \cdot \ldots \cdot x_{\mathfrak{E}_n}^n \\
&= \sum_{\mathfrak{E}_i=1}^{m_i} x_{\mathfrak{E}_i}^i \left[\sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s_{\mathfrak{E}_1}^1, \ldots, s_{\mathfrak{E}_i}^i, \ldots, s_{\mathfrak{E}_n}^n\right) \cdot x_{\mathfrak{E}_1}^1 \cdot \ldots \cdot x_{\mathfrak{E}_{i-1}}^{i-1} \cdot x_{\mathfrak{E}_{i+1}}^{i+1} \cdot \ldots \cdot x_{\mathfrak{E}_n}^n\right] \\
&= x_1^i \left[\sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s_{\mathfrak{E}_1}^1, \ldots, s_1^i, \ldots, s_{\mathfrak{E}_n}^n\right) \cdot x_{\mathfrak{E}_1}^1 \cdot \ldots \cdot x_{\mathfrak{E}_{i-1}}^{i-1} \cdot x_{\mathfrak{E}_{i+1}}^{i+1} \cdot \ldots \cdot x_{\mathfrak{E}_n}^n\right] \\
&+ x_2^i \left[\sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s_{\mathfrak{E}_1}^1, \ldots, s_2^i, \ldots, s_{\mathfrak{E}_n}^n\right) \cdot x_{\mathfrak{E}_1}^1 \cdot \ldots \cdot x_{\mathfrak{E}_{i-1}}^{i-1} \cdot x_{\mathfrak{E}_{i+1}}^{i+1} \cdot \ldots \cdot x_{\mathfrak{E}_n}^n\right] \\
&+ \sum_{\mathfrak{E}_i=3}^{m_i} x_{\mathfrak{E}_i}^i \left[\sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s_{\mathfrak{E}_1}^1, \ldots, s_{\mathfrak{E}_i}^i, \ldots, s_{\mathfrak{E}_n}^n\right) \cdot x_{\mathfrak{E}_1}^1 \cdot \ldots \cdot x_{\mathfrak{E}_{i-1}}^{i-1} \cdot x_{\mathfrak{E}_{i+1}}^{i+1} \cdot \ldots \cdot x_{\mathfrak{E}_n}^n\right],
\end{aligned} \tag{2.2}$$

if $x_{\mathfrak{E}_{h_1}}^i, \ldots, x_{\mathfrak{E}_{h_l}}^i > 0, x_{\mathfrak{E}_{h_{l+1}}}^i, \ldots, x_{\mathfrak{E}_{hm_i}}^i = 0$, then for each $\mathfrak{E}_i \in \{\mathfrak{E}_{h_1}, \ldots, \mathfrak{E}_{h_l}\}$, $\mathscr{K}_i \in \{1, \ldots, m_i\}$, and $\mathscr{K}_i \neq \mathfrak{E}_i$, we have

$$\begin{aligned}
&\sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s_{\mathfrak{E}_1}^1, \ldots, s_{\mathfrak{E}_i}^i, \ldots, s_{\mathfrak{E}_n}^n\right) \cdot x_{\mathfrak{E}_1}^1 \cdot \ldots \cdot x_{\mathfrak{E}_{i-1}}^{i-1} \cdot x_{\mathfrak{E}_{i+1}}^{i+1} \cdot \ldots \cdot x_{\mathfrak{E}_n}^n \\
&\geq \sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s_{\mathfrak{E}_1}^1, \ldots, s_{\mathscr{K}_i}^i, \ldots, s_{\mathfrak{E}_n}^n\right) \cdot x_{\mathfrak{E}_1}^1 \cdot \ldots \cdot x_{\mathfrak{E}_{i-1}}^{i-1} \cdot x_{\mathfrak{E}_{i+1}}^{i+1} \cdot \ldots \cdot x_{\mathfrak{E}_n}^n.
\end{aligned} \tag{2.3}$$

(Contradiction): Assuming that the above formula (2.3) does not hold, then there is a certain $\mathfrak{E}_i \in \{\mathfrak{E}_{h_1}, \ldots, \mathfrak{E}_{h_l}\}$, $\mathscr{K}_i \neq \mathfrak{E}_i$, where $\mathscr{K}_i \in \{1, \ldots, m_i\}$, such that

$$\begin{aligned}
&\sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s_{\mathfrak{E}_1}^1, \ldots, s_{\mathfrak{E}_i}^i, \ldots, s_{\mathfrak{E}_n}^n\right) \cdot x_{\mathfrak{E}_1}^1 \cdot \ldots \cdot x_{\mathfrak{E}_{i-1}}^{i-1} \cdot x_{\mathfrak{E}_{i+1}}^{i+1} \cdot \ldots \cdot x_{\mathfrak{E}_n}^n \\
&< \sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s_{\mathfrak{E}_1}^1, \ldots, s_{\mathscr{K}_i}^i, \ldots, s_{\mathfrak{E}_n}^n\right) \cdot x_{\mathfrak{E}_1}^1 \cdot \ldots \cdot x_{\mathfrak{E}_{i-1}}^{i-1} \cdot x_{\mathfrak{E}_{i+1}}^{i+1} \cdot \ldots \cdot x_{\mathfrak{E}_n}^n.
\end{aligned}$$

For the convenience of writing, let $\mathfrak{E}_i = 1$, $\mathscr{K}_i = 2$, that is,

$$\begin{aligned}
&\sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s_{\mathfrak{E}_1}^1, \ldots, s_1^i, \ldots, s_{\mathfrak{E}_n}^n\right) \cdot x_{\mathfrak{E}_1}^1 \cdot \ldots \cdot x_{\mathfrak{E}_{i-1}}^{i-1} \cdot x_{\mathfrak{E}_{i+1}}^{i+1} \cdot \ldots \cdot x_{\mathfrak{E}_n}^n \\
&< \sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s_{\mathfrak{E}_1}^1, \ldots, s_2^i, \ldots, s_{\mathfrak{E}_n}^n\right) \cdot x_{\mathfrak{E}_1}^1 \cdot \ldots \cdot x_{\mathfrak{E}_{i-1}}^{i-1} \cdot x_{\mathfrak{E}_{i+1}}^{i+1} \cdot \ldots \cdot x_{\mathfrak{E}_n}^n.
\end{aligned} \tag{2.4}$$

Taking $y = \left(x^1, \ldots, x^{i-1}, y^i, x^{i+1}, \ldots, x^n\right), y^i = \left(0, x_1^i + x_2^i, x_3^i, \ldots, x_{\mathfrak{E}_i}^i, \ldots, x_{m_i}^i\right)$, and combining inequality (2.4) with the formula (2.2), we obtain

$$
\begin{aligned}
u_i(y) &= \sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, s^2_{\mathfrak{E}_2}, \ldots, s^n_{\mathfrak{E}_n}\right) \cdot y^1_{\mathfrak{E}_1} \cdot y^2_{\mathfrak{E}_2} \cdot \ldots \cdot y^n_{\mathfrak{E}_n} \\
&= \sum_{\mathfrak{E}_i=1}^{m_i} y^i_{\mathfrak{E}_i} \left[\sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, \ldots, s^i_{\mathfrak{E}_i}, \ldots, s^n_{\mathfrak{E}_n}\right) \cdot x^1_{\mathfrak{E}_1} \cdot \ldots \cdot x^{i-1}_{\mathfrak{E}_{i-1}} \cdot x^{i+1}_{\mathfrak{E}_{i+1}} \cdot \ldots \cdot x^n_{\mathfrak{E}_n} \right] \\
&= \left(x^i_1 + x^i_2\right) \left[\sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, \ldots, s^i_2, \ldots, s^n_{\mathfrak{E}_n}\right) \cdot x^1_{\mathfrak{E}_1} \cdot \ldots \cdot x^{i-1}_{\mathfrak{E}_{i-1}} \cdot x^{i+1}_{\mathfrak{E}_{i+1}} \cdot \ldots \cdot x^n_{\mathfrak{E}_n} \right] \\
&\quad + \sum_{\mathfrak{E}_i=3}^{m_i} x^i_{\mathfrak{E}_i} \left[\sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, \ldots, s^i_{\mathfrak{E}_i}, \ldots, s^n_{\mathfrak{E}_n}\right) \cdot x^1_{\mathfrak{E}_1} \cdot \ldots \cdot x^{i-1}_{\mathfrak{E}_{i-1}} \cdot x^{i+1}_{\mathfrak{E}_{i+1}} \cdot \ldots \cdot x^n_{\mathfrak{E}_n} \right] \\
&= x^i_1 \left[\sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, \ldots, s^i_2, \ldots, s^n_{\mathfrak{E}_n}\right) \cdot x^1_{\mathfrak{E}_1} \cdot \ldots \cdot x^{i-1}_{\mathfrak{E}_{i-1}} \cdot x^{i+1}_{\mathfrak{E}_{i+1}} \cdot \ldots \cdot x^n_{\mathfrak{E}_n} \right] \\
&\quad + x^i_2 \left[\sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, \ldots, s^i_2, \ldots, s^n_{\mathfrak{E}_n}\right) \cdot x^1_{\mathfrak{E}_1} \cdot \ldots \cdot x^{i-1}_{\mathfrak{E}_{i-1}} \cdot x^{i+1}_{\mathfrak{E}_{i+1}} \cdot \ldots \cdot x^n_{\mathfrak{E}_n} \right] \\
&\quad + \sum_{\mathfrak{E}_i=3}^{m_i} x^i_{\mathfrak{E}_i} \left[\sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, \ldots, s^i_{\mathfrak{E}_i}, \ldots, s^n_{\mathfrak{E}_n}\right) \cdot x^1_{\mathfrak{E}_1} \cdot \ldots \cdot x^{i-1}_{\mathfrak{E}_{i-1}} \cdot x^{i+1}_{\mathfrak{E}_{i+1}} \cdot \ldots \cdot x^n_{\mathfrak{E}_n} \right] \\
&> x^i_1 \left[\sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, \ldots, s^i_1, \ldots, s^n_{\mathfrak{E}_n}\right) \cdot x^1_{\mathfrak{E}_1} \cdot \ldots \cdot x^{i-1}_{\mathfrak{E}_{i-1}} \cdot x^{i+1}_{\mathfrak{E}_{i+1}} \cdot \ldots \cdot x^n_{\mathfrak{E}_n} \right] \\
&\quad + x^i_2 \left[\sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, \ldots, s^i_2, \ldots, s^n_{\mathfrak{E}_n}\right) \cdot x^1_{\mathfrak{E}_1} \cdot \ldots \cdot x^{i-1}_{\mathfrak{E}_{i-1}} \cdot x^{i+1}_{\mathfrak{E}_{i+1}} \cdot \ldots \cdot x^n_{\mathfrak{E}_n} \right] \\
&\quad + \sum_{\mathfrak{E}_i=3}^{m_i} x^i_{\mathfrak{E}_i} \left[\sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, \ldots, s^i_{\mathfrak{E}_i}, \ldots, s^n_{\mathfrak{E}_n}\right) \cdot x^1_{\mathfrak{E}_1} \cdot \ldots \cdot x^{i-1}_{\mathfrak{E}_{i-1}} \cdot x^{i+1}_{\mathfrak{E}_{i+1}} \cdot \ldots \cdot x^n_{\mathfrak{E}_n} \right] \\
&= u_i(x),
\end{aligned}
$$

it contradicts that x is a Nash equilibrium. So we get that for each Nash equilibrium $x^i_{\mathfrak{E}_i} > 0$, and $\forall \mathscr{K}_i \in \{1, \ldots, m_i\}$,

$$
\begin{aligned}
&\sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, \ldots, s^i_{\mathfrak{E}_i}, \ldots, s^n_{\mathfrak{E}_n}\right) \cdot x^1_{\mathfrak{E}_1} \cdot \ldots \cdot x^{i-1}_{\mathfrak{E}_{i-1}} \cdot x^{i+1}_{\mathfrak{E}_{i+1}} \cdot \ldots \cdot x^n_{\mathfrak{E}_n} \\
&\qquad \geq \sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, \ldots, s^i_{\mathscr{K}_i}, \ldots, s^n_{\mathfrak{E}_n}\right) \cdot x^1_{\mathfrak{E}_1} \cdot \ldots \cdot x^{i-1}_{\mathfrak{E}_{i-1}} \cdot x^{i+1}_{\mathfrak{E}_{i+1}} \cdot \ldots \cdot x^n_{\mathfrak{E}_n}.
\end{aligned}
$$

In particular, if $\mathfrak{E}_i, \mathscr{K}_i \in \{\mathfrak{E}_{h_1}, \ldots, \mathfrak{E}_{h_l}\}$, $x^i_{\mathfrak{E}_i}, x^i_{\mathscr{K}_i} > 0$, then

$$
\begin{aligned}
&\sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, \ldots, s^i_{\mathfrak{E}_i}, \ldots, s^n_{\mathfrak{E}_n}\right) \cdot x^1_{\mathfrak{E}_1} \cdot \ldots \cdot x^{i-1}_{\mathfrak{E}_{i-1}} \cdot x^{i+1}_{\mathfrak{E}_{i+1}} \cdot \ldots \cdot x^n_{\mathfrak{E}_n} \\
&\qquad = \sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, \ldots, s^i_{\mathscr{K}_i}, \ldots, s^n_{\mathfrak{E}_n}\right) \cdot x^1_{\mathfrak{E}_1} \cdot \ldots \cdot x^{i-1}_{\mathfrak{E}_{i-1}} \cdot x^{i+1}_{\mathfrak{E}_{i+1}} \cdot \ldots \cdot x^n_{\mathfrak{E}_n}.
\end{aligned}
$$

The following proof shows that the Nash equilibrium $x = \left(x^1, x^2, \ldots, x^n\right) \in X$ satisfies the system of equations in Theorem 2.1.

(i) For every $i \in N$, if $x^i_{\mathfrak{E}_i} = 0$, then for each $\mathscr{K}_i \in \{1, \ldots, m_i\}$, $\mathscr{K}_i \neq \mathfrak{E}_i$, and take $\lambda^i_{\mathfrak{E}_i\mathscr{K}_i} = 0$, we have

$$\begin{aligned}\lambda^i_{\mathfrak{E}_i\mathscr{K}_i} + x^i_{\mathfrak{E}_i}\Bigg(&\sum_{\mathfrak{E}_1=1}^{m_1}\cdots\sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}}\sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}}\cdots\sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, \ldots, s^i_{\mathscr{K}_i}, \ldots, s^n_{\mathfrak{E}_n}\right)\cdot x^1_{\mathfrak{E}_1}\cdot\ldots\cdot x^{i-1}_{\mathfrak{E}_{i-1}}\cdot x^{i+1}_{\mathfrak{E}_{i+1}}\cdot\ldots\cdot x^n_{\mathfrak{E}_n}\\ &-\sum_{\mathfrak{E}_1=1}^{m_1}\cdots\sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}}\sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}}\cdots\sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, \ldots, s^i_{\mathfrak{E}_i}, \ldots, s^n_{\mathfrak{E}_n}\right)\cdot x^1_{\mathfrak{E}_1}\cdot\ldots\cdot x^{i-1}_{\mathfrak{E}_{i-1}}\cdot x^{i+1}_{\mathfrak{E}_{i+1}}\cdot\ldots\cdot x^n_{\mathfrak{E}_n}\Bigg) = 0.\end{aligned} \tag{2.5}$$

(ii) If $x^i_{\mathfrak{E}_i} > 0$, then for each $\mathscr{K}_i \in \{1, \ldots, m_i\}$, $\mathscr{K}_i \neq \mathfrak{E}_i$, that is formula (2.3),

$$\begin{aligned}&\sum_{\mathfrak{E}_1=1}^{m_1}\cdots\sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}}\sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}}\cdots\sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, \ldots, s^i_{\mathfrak{E}_i}, \ldots, s^n_{\mathfrak{E}_n}\right)\cdot x^1_{\mathfrak{E}_1}\cdot\ldots\cdot x^{i-1}_{\mathfrak{E}_{i-1}}\cdot x^{i+1}_{\mathfrak{E}_{i+1}}\cdot\ldots\cdot x^n_{\mathfrak{E}_n}\\ \geq&\sum_{\mathfrak{E}_1=1}^{m_1}\cdots\sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}}\sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}}\cdots\sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, \ldots, s^i_{\mathscr{K}_i}, \ldots, s^n_{\mathfrak{E}_n}\right)\cdot x^1_{\mathfrak{E}_1}\cdot\ldots\cdot x^{i-1}_{\mathfrak{E}_{i-1}}\cdot x^{i+1}_{\mathfrak{E}_{i+1}}\cdot\ldots\cdot x^n_{\mathfrak{E}_n}.\end{aligned} \tag{2.6}$$

(ii-1) If the equality sign of formula (2.6) holds, take $\lambda^i_{\mathfrak{E}_i\mathscr{K}_i} = 0$, then formula (2.5) holds.

(ii-2) If the equality sign of formula (2.6) does not hold, that is

$$\begin{aligned}&\sum_{\mathfrak{E}_1=1}^{m_1}\cdots\sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}}\sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}}\cdots\sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, \ldots, s^i_{\mathfrak{E}_i}, \ldots, s^n_{\mathfrak{E}_n}\right)\cdot x^1_{\mathfrak{E}_1}\cdot\ldots\cdot x^{i-1}_{\mathfrak{E}_{i-1}}\cdot x^{i+1}_{\mathfrak{E}_{i+1}}\cdot\ldots\cdot x^n_{\mathfrak{E}_n}\\ >&\sum_{\mathfrak{E}_1=1}^{m_1}\cdots\sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}}\sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}}\cdots\sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, \ldots, s^i_{\mathscr{K}_i}, \ldots, s^n_{\mathfrak{E}_n}\right)\cdot x^1_{\mathfrak{E}_1}\cdot\ldots\cdot x^{i-1}_{\mathfrak{E}_{i-1}}\cdot x^{i+1}_{\mathfrak{E}_{i+1}}\cdot\ldots\cdot x^n_{\mathfrak{E}_n},\end{aligned}$$

then take

$$\begin{aligned}\lambda^i_{\mathfrak{E}_i\mathscr{K}_i} = x^i_{\mathfrak{E}_i}\Bigg(&\sum_{\mathfrak{E}_1=1}^{m_1}\cdots\sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}}\sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}}\cdots\sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, \ldots, s^i_{\mathfrak{E}_i}, \ldots, s^n_{\mathfrak{E}_n}\right)\cdot x^1_{\mathfrak{E}_1}\cdot\ldots\cdot x^{i-1}_{\mathfrak{E}_{i-1}}\cdot x^{i+1}_{\mathfrak{E}_{i+1}}\cdot\ldots\cdot x^n_{\mathfrak{E}_n}\\ &-\sum_{\mathfrak{E}_1=1}^{m_1}\cdots\sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}}\sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}}\cdots\sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, \ldots, s^i_{\mathscr{K}_i}, \ldots, s^n_{\mathfrak{E}_n}\right)\cdot x^1_{\mathfrak{E}_1}\cdot\ldots\cdot x^{i-1}_{\mathfrak{E}_{i-1}}\cdot x^{i+1}_{\mathfrak{E}_{i+1}}\cdot\ldots\cdot x^n_{\mathfrak{E}_n}\Bigg),\end{aligned}$$

therefore

$$\begin{aligned}\lambda^i_{\mathfrak{E}_i\mathscr{K}_i} + x^i_{\mathfrak{E}_i}\Bigg(&\sum_{\mathfrak{E}_1=1}^{m_1}\cdots\sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}}\sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}}\cdots\sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, \ldots, s^i_{\mathscr{K}_i}, \ldots, s^n_{\mathfrak{E}_n}\right)\cdot x^1_{\mathfrak{E}_1}\cdot\ldots\cdot x^{i-1}_{\mathfrak{E}_{i-1}}\cdot x^{i+1}_{\mathfrak{E}_{i+1}}\cdot\ldots\cdot x^n_{\mathfrak{E}_n}\\ &-\sum_{\mathfrak{E}_1=1}^{m_1}\cdots\sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}}\sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}}\cdots\sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s^1_{\mathfrak{E}_1}, \ldots, s^i_{\mathfrak{E}_i}, \ldots, s^n_{\mathfrak{E}_n}\right)\cdot x^1_{\mathfrak{E}_1}\cdot\ldots\cdot x^{i-1}_{\mathfrak{E}_{i-1}}\cdot x^{i+1}_{\mathfrak{E}_{i+1}}\cdot\ldots\cdot x^n_{\mathfrak{E}_n}\Bigg) = 0.\end{aligned}$$

The necessity has been proved. Next, we show the sufficiency of Theorem 2.1.

Sufficiency: If there is a set of real numbers $\lambda^i_{\mathfrak{E}_i\mathscr{K}_i} \geq 0, i = 1, 2, \ldots, n, \mathfrak{E}_i \neq \mathscr{K}_i, \mathfrak{E}_i, \mathscr{K}_i \in \{1, 2, \ldots, m_i\}$ and mixed strategy profile $x = \left(x^1, x^2, \ldots, x^n\right) \in X, x^i = \left(x^i_1, \ldots, x^i_{\mathfrak{E}_i}, \ldots, x^i_{m_i}\right), i \in N$, satisfying the

system of equations in Theorem 2.1, The following proof shows that x is a Nash equilibrium. Let $x^i = \left(x_1^i, \ldots, x_{\mathfrak{E}_{i_k}}^i, x_{\mathfrak{E}_{i_{k+1}}}^i, \ldots, x_{m_i}^i\right)$, where $x_1^i, \ldots, x_{\mathfrak{E}_{i_k}}^i > 0, x_{\mathfrak{E}_{i_{k+1}}}^i, \ldots, x_{m_i}^i = 0$. Since x satisfies the system of equations in Theorem 2.1, then for any $\mathfrak{E}_i \in \{1, \ldots, \mathfrak{E}_{i_k}\}$, we get

$$\lambda_{\mathfrak{E}_i\mathscr{K}_i}^i + x_{\mathfrak{E}_i}^i \left(\sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s_{\mathfrak{E}_1}^1, \ldots, s_{\mathscr{K}_i}^i, \ldots, s_{\mathfrak{E}_n}^n\right) \cdot x_{\mathfrak{E}_1}^1 \cdot \ldots \cdot x_{\mathfrak{E}_{i-1}}^{i-1} \cdot x_{\mathfrak{E}_{i+1}}^{i+1} \cdot \ldots \cdot x_{\mathfrak{E}_n}^n \right.$$
$$\left. - \sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s_{\mathfrak{E}_1}^1, \ldots, s_{\mathfrak{E}_i}^i, \ldots, s_{\mathfrak{E}_n}^n\right) \cdot x_{\mathfrak{E}_1}^1 \cdot \ldots \cdot x_{\mathfrak{E}_{i-1}}^{i-1} \cdot x_{\mathfrak{E}_{i+1}}^{i+1} \cdot \ldots \cdot x_{\mathfrak{E}_n}^n \right) = 0.$$

Because $x_1^i, \ldots, x_{\mathfrak{E}_{i_k}}^i > 0$, for $\forall \mathscr{K}_i \in \{1, 2, \ldots, m_i\}$, we have

$$\sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s_{\mathfrak{E}_1}^1, \ldots, s_{\mathfrak{E}_i}^i, \ldots, s_{\mathfrak{E}_n}^n\right) \cdot x_{\mathfrak{E}_1}^1 \cdot \ldots \cdot x_{\mathfrak{E}_{i-1}}^{i-1} \cdot x_{\mathfrak{E}_{i+1}}^{i+1} \cdot \ldots \cdot x_{\mathfrak{E}_n}^n$$
$$\geq \sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s_{\mathfrak{E}_1}^1, \ldots, s_{\mathscr{K}_i}^i, \ldots, s_{\mathfrak{E}_n}^n\right) \cdot x_{\mathfrak{E}_1}^1 \cdot \ldots \cdot x_{\mathfrak{E}_{i-1}}^{i-1} \cdot x_{\mathfrak{E}_{i+1}}^{i+1} \cdot \ldots \cdot x_{\mathfrak{E}_n}^n,$$

and because for any player i, the expected payoff for pure strategy is

$$u_i\left(x \| s_{\mathscr{K}_i}^i\right) = \sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s_{\mathfrak{E}_1}^1, \ldots, s_{\mathscr{K}_i}^i, \ldots, s_{\mathfrak{E}_n}^n\right) \cdot x_{\mathfrak{E}_1}^1 \cdot \ldots \cdot x_{\mathfrak{E}_{i-1}}^{i-1} \cdot x_{\mathfrak{E}_{i+1}}^{i+1} \cdot \ldots \cdot x_{\mathfrak{E}_n}^n,$$

therefore

$$\begin{aligned}
u_i(x) &= \sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s_{\mathfrak{E}_1}^1, \ldots, s_{\mathfrak{E}_i}^i, \ldots, s_{\mathfrak{E}_n}^n\right) \cdot x_{\mathfrak{E}_1}^1 \cdot x_{\mathfrak{E}_2}^2 \cdot \ldots \cdot x_{\mathfrak{E}_n}^n \\
&= \sum_{\mathfrak{E}_i=1}^{m_i} x_{\mathfrak{E}_i}^i \left[\sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s_{\mathfrak{E}_1}^1, \ldots, s_{\mathfrak{E}_i}^i, \ldots, s_{\mathfrak{E}_n}^n\right) \cdot x_{\mathfrak{E}_1}^1 \cdot \ldots \cdot x_{\mathfrak{E}_{i-1}}^{i-1} \cdot x_{\mathfrak{E}_{i+1}}^{i+1} \cdot \ldots \cdot x_{\mathfrak{E}_n}^n \right] \\
&= \sum_{\mathfrak{E}_i=1}^{\mathfrak{E}_{i_k}} x_{\mathfrak{E}_i}^i \left[\sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s_{\mathfrak{E}_1}^1, \ldots, s_{\mathfrak{E}_i}^i, \ldots, s_{\mathfrak{E}_n}^n\right) \cdot x_{\mathfrak{E}_1}^1 \cdot \ldots \cdot x_{\mathfrak{E}_{i-1}}^{i-1} \cdot x_{\mathfrak{E}_{i+1}}^{i+1} \cdot \ldots \cdot x_{\mathfrak{E}_n}^n \right] \\
&\geq \sum_{\mathfrak{E}_i=1}^{\mathfrak{E}_{i_k}} x_{\mathfrak{E}_i}^i \left[\sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s_{\mathfrak{E}_1}^1, \ldots, s_{\mathscr{K}_i}^i, \ldots, s_{\mathfrak{E}_n}^n\right) \cdot x_{\mathfrak{E}_1}^1 \cdot \ldots \cdot x_{\mathfrak{E}_{i-1}}^{i-1} \cdot x_{\mathfrak{E}_{i+1}}^{i+1} \cdot \ldots \cdot x_{\mathfrak{E}_n}^n \right] \\
&= \left[\sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s_{\mathfrak{E}_1}^1, \ldots, s_{\mathscr{K}_i}^i, \ldots, s_{\mathfrak{E}_n}^n\right) \cdot x_{\mathfrak{E}_1}^1 \cdot \ldots \cdot x_{\mathfrak{E}_{i-1}}^{i-1} \cdot x_{\mathfrak{E}_{i+1}}^{i+1} \cdot \ldots \cdot x_{\mathfrak{E}_n}^n \right] \cdot \sum_{\mathfrak{E}_i=1}^{\mathfrak{E}_{i_k}} x_{\mathfrak{E}_i}^i \\
&= \sum_{\mathfrak{E}_1=1}^{m_1} \cdots \sum_{\mathfrak{E}_{i-1}=1}^{m_{i-1}} \sum_{\mathfrak{E}_{i+1}=1}^{m_{i+1}} \cdots \sum_{\mathfrak{E}_n=1}^{m_n} p_i\left(s_{\mathfrak{E}_1}^1, \ldots, s_{\mathscr{K}_i}^i, \ldots, s_{\mathfrak{E}_n}^n\right) \cdot x_{\mathfrak{E}_1}^1 \cdot \ldots \cdot x_{\mathfrak{E}_{i-1}}^{i-1} \cdot x_{\mathfrak{E}_{i+1}}^{i+1} \cdot \ldots \cdot x_{\mathfrak{E}_n}^n \\
&= u_i\left(x \| s_{\mathscr{K}_i}^i\right).
\end{aligned}$$

Judging by Conclusion 2.1 that $x = (x^1, x^2, \ldots, x^n)$ is a Nash equilibrium. □

2.3. Solving the Nash equilibria of bimatrix game based on the system of equations

According to the preparatory knowledge of n-person finite noncooperative game and Theorem 2.1, we can get the equivalence theorem of the bimatrix game.

Corollary 2.1. *Suppose the mixed strategy of the two-player finite noncooperative game is* $x = (x^1, x^2)$, *where* $x^1 = (x_1^1, x_2^1, \ldots, x_{m_1}^1)$, $x^1 = (x_1^2, x_2^2, \ldots, x_{m_2}^2)$, *and* $\sum_{i=1}^{m_1} x_i^1 = 1$, $\sum_{i=1}^{m_2} x_i^2 = 1$. *The payoff matrices are A and B respectively:*

$$A = \begin{bmatrix} a_{11} & \ldots & a_{1m_2} \\ \vdots & \vdots & \vdots \\ a_{m_1 1} & \ldots & a_{m_1 m_2} \end{bmatrix}, \quad B = \begin{bmatrix} b_{11} & \ldots & b_{1m_2} \\ \vdots & \vdots & \vdots \\ b_{m_1 1} & \ldots & b_{m_1 m_2} \end{bmatrix},$$

then, x is a Nash equilibrium if and only if there is a set of real numbers $\lambda^i_{\mathfrak{E}_i \mathscr{K}_i} \geq 0, i = 1, 2, \mathfrak{E}_i \neq \mathscr{K}_i, \mathfrak{E}_i, \mathscr{K}_i \in \{1, 2, \ldots, m_i\}$, *such that* $\lambda^i_{\mathfrak{E}_i \mathscr{K}_i}$ *and* $x = (x^1, x^2)$ *satisfy the following equations:*

$$\begin{cases} \lambda^1_{\mathscr{K}_1 \mathscr{K}_2} + x^1_{\mathscr{K}_1}(A_{\mathscr{K}_2} - A_{\mathscr{K}_1})[x^2]' = 0, \\ \lambda^2_{\mathfrak{E}_1 \mathfrak{E}_2} + x^2_{\mathfrak{E}_1} x^1 (B_{\mathfrak{E}_2} - B_{\mathfrak{E}_1}) = 0, \end{cases} \tag{2.7}$$

where $A_{\mathscr{K}_t}(t = 1, 2)$ *is the* $\mathscr{K}_t$*th row of matrix A,* $B_{\mathfrak{E}_t}(t = 1, 2)$ *is the* $\mathfrak{E}_t$*th column of matrix B,* $x^1 = (x_1^1, \ldots, x_{m_1}^1)$, $x^2 = (x_1^2, \ldots, x_{m_2}^2)$, *and* $\mathscr{K}_1 \neq \mathscr{K}_2 \in \{1, \ldots, m_1\}, \mathfrak{E}_1 \neq \mathfrak{E}_2 \in \{1, \ldots, m_2\}$.

For the convenience of writing, formula (2.1) can be simplified as:

$$\begin{cases} f_1(\lambda_1, x_1, \ldots, x_{\mathscr{K}}) = 0, \\ f_2(\lambda_2, x_1, \ldots, x_{\mathscr{K}}) = 0, \\ \vdots \\ f_m(\lambda_m, x_1, \ldots, x_{\mathscr{K}}) = 0, \end{cases} \tag{2.8}$$

where $m = \sum_{i=1}^{n} m_i(m_i - 1)$, $\mathscr{K} = \sum_{i=1}^{n} m_i$, $\lambda_i \geq 0, i = 1, 2, \ldots, m$, and $0 \leq x_i \leq 1, i = 1, 2, \ldots, \mathscr{K}$, $\sum_{i=1}^{m_1} x_i = 1, \sum_{i=m_1+1}^{m_1+m_2} x_i = 1, \ldots, \sum_{i=m_{n-1}+1}^{m_{n-1}+m_n} x_i = 1$.

According to the equivalent methods of solving the equations, the solution of the Eq (2.8) is usually equivalent to the following optimization problem, and the optimal value is 0:

$$min\ F(\lambda_1, \ldots, \lambda_m, x_1, \ldots, x_{\mathscr{K}}) = \sum_{i=1}^{m} |f_i(\lambda_i, x_1, \ldots, x_{\mathscr{K}})|. \tag{2.9}$$

An algorithm for finding the Nash equilibria based on the system of equations is given below.

3. Adaptive differential evolution algorithm based on cultural algorithm (ADECA)

3.1. The cultural algorithm

In 1994, Reynolds [33] simulated the evolution of human society and proposed a bionic intelligent computing method with a bilevel evolutionary mechanism, cultural algorithm (CA). This algorithm is composed of independently evolved population space and belief space. The population space simulates

the evolution process of individuals according to certain behavioral criteria from the microscopic perspective, while the belief space simulates the evolution process of culture formation, transmission and comparison from the macroscopic perspective. Population space and belief space are two relatively independent evolutionary processes, and the two spaces are connected according to special protocols (acceptance function and influence function). Those interactions are depicted in Figure 1.

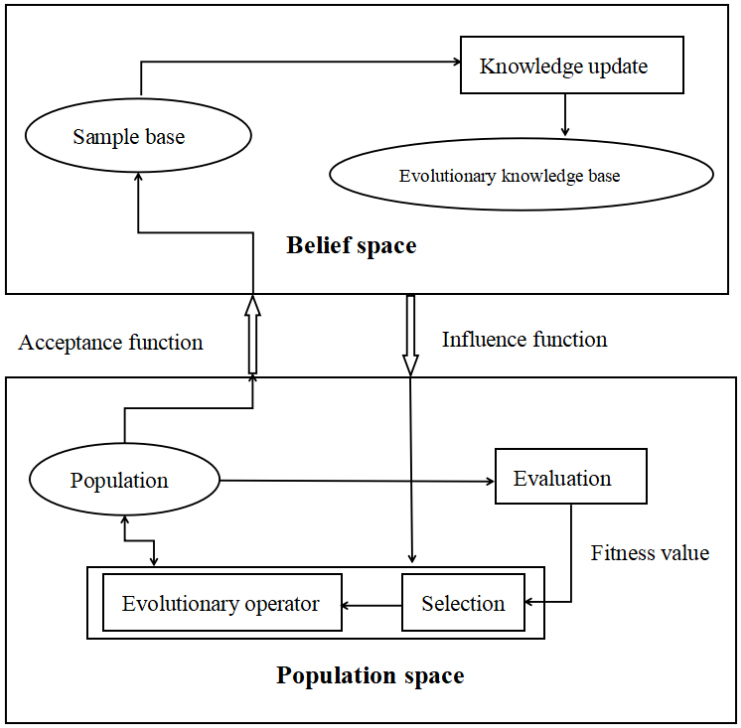

Figure 1. Cultural algorithm framework.

The population space is used to implement any population evolutionary algorithm. On the one hand, it evaluates the fitness value of individuals, and implements evolutionary operations such as selection, crossover and mutation for population; On the other hand, excellent individuals are provided to the belief space as samples. The belief space selects the sample individuals from the evaluated populations in the population space through the acceptance function. Under the function of knowledge updating, the hidden information carried by the sample individuals is processed and summarized, described and stored in the form of knowledge. Finally, all kinds of knowledge react on the population space through the influence function to guide the evolution of the population space, so as to accelerate the evolutionary convergence and improve the adaptability of the algorithm. The details on the interaction and influence between belief space and population space will be given below. (The following discussions are based on the assumption of solving the minimum optimization problem).

3.2. The population space

The CA algorithm provides a computational model of a multi evolutionary processes. Therefore, from the perspective of computational model, an evolutionary algorithm that meets the requirements of CA algorithm can be regarded as an evolutionary process of population space. In this paper, the proposed algorithm uses DE algorithm with simple operation in the population space of CA algorithm.

3.2.1. Basic DE algorithm

The DE algorithm is a new simple and robust evolutionary algorithm. It is first introduced by Storm and Price [34]. There are three main operations: mutation, crossover and selection operation.

(1) Mutation

The mutation operation is mainly executed to distinguish DE from other evolutionary algorithms. The mutation individual $V = (v_{i1}, \ldots, v_{ij}, \ldots, v_{iD}), i = 1, \ldots, N, j = 1, \ldots, D$ is generated by the following equation:

$$v_i^{t+1} = x_{r1}^t + F \cdot (x_{r2}^t - x_{r3}^t), \tag{3.1}$$

where N and D denote population size and space dimension respectively, $r1$, $r2$ and $r3$ are randomly generated integers within $[1, N]$, and $r1 \neq r2 \neq r3 \neq i$, F is a constriction factor to control the size of difference of two individuals and t is the current generation.

(2) Crossover

We use the crossover between the parent and offspring with the given probability for generating new individual $U = (u_{i1}, \ldots, u_{ij}, \ldots, u_{iD})$:

$$u_{ij}^{t+1} = \begin{cases} v_{ij}^{t+1}, & if\ (rand(j) \leq CR)\ or\ (j = rnbr(i)), \\ x_{ij}, & otherwise, \end{cases} \tag{3.2}$$

where $rand(j) \in [0, 1]$ is random values, $CR \in [0, 1]$ is crossover operator, $rnbr(i)$ is a randomly selected integer on $[1, D]$, which ensure that a new individual gets at least one component value from the mutation vector.

(3) Selection

The offspring X_{ij}^{t+1} is generated by selecting the individual and parent according to the following formula:

$$X_{ij}^{t+1} = \begin{cases} U_{ij}^{t+1}, & if\ (f(U_{ij}^{t+1}) < f(U_{ij}^t)), \\ X_{ij}^t, & otherwise, \end{cases} \tag{3.3}$$

where $f(\cdot)$ is the fitness function.

Although the DE algorithm is widely used in optimization problems, with the increase of the complexity of solving problems, the DE algorithm also has some disadvantages, such as slow convergence, low accuracy and weak stability. Therefore, in order to better apply to CA algorithm, the DE algorithm is improved.

3.2.2. Adaptive differential evolution (ADE) algorithm

Many scholars have verified that the performance of the DE algorithm mainly depends on the selection of mutation operation and related parameters. Furthermore, the mutation factor F and crossover operator CR directly affect the searching capability and solving efficiency of the DE algorithm [35]. According to formula (3.1), the mutation factor F is used to control the influence of the difference vector on the mutation individual v_i. When F is large, the difference vector $(x_{r2} - x_{r3})$ has a great influence on v_i and causes a large disturbance, which is beneficial for maintaining population diversity and global search of the algorithm, but at the same time it will reduce the convergence speed; On the contrary, when F is small, the disturbance is small, that is, the diversity of the population is small, which is conducive to local search of the algorithm and makes the convergence speed

faster. However, the algorithm is easy to fall into the local optimal and the "premature". According to formula (3.2), when CR is larger, the more contribution v_i makes to u_i, which is beneficial to expanding new spaces and accelerating the convergence of the algorithm. However, in the later period, the mutation individuals tend to be the same, which is detrimental for maintaining population diversity, making the algorithm prone to "premature" phenomenon; Conversely, CR is smaller, the more contribution x_i makes, which is beneficial for maintaining population diversity. However, in this way, the ability of the algorithm to explore new spaces is weakened, and the convergence speed is relatively slow.

In order to make the algorithm have better global search capability and convergence speed, adaptive mutation and crossover operator are adopted:

$$\begin{cases} \lambda = e^{1-\frac{T}{T+1-t}}, \\ F = F_0 \cdot 2^{\lambda}, \\ CR = CR_0 \cdot 2^{\lambda}, \end{cases} \tag{3.4}$$

where t is the current iterate time, T is the maximum number of iteration, F_0 and CR_0 respectively are the initial mutation factor and crossover operator. The adaptive mutation factor and crossover operator are proposed, which can adaptively determine the mutation rate and crossover rate according to the search progress of the algorithm. According to formula (3.4), since $t \in [1, T]$, then $\lambda \in [e^{1-T}, 1] \in (0, 1]$, so, $F = F_0 \cdot 2^{\lambda} \in (F_0, 2F_0]$, that is, $2F_0$ is taken when $t = 1$, which has a larger value to avoid the algorithm "premature"; With the progress of the algorithm, the value of F gradually decreases, until later close to F_0, which retains good information, avoids the destruction of the optimal solution, and increases the possibility of searching for the global optimal solution. The $CR = CR_0 \cdot 2^{\lambda} \in (CR_0, 2CR_0]$ is similar: CR has a large value in the early stage, which increases the exploration ability of the algorithm and avoids the algorithm falling into local optimal. In the later stage, the crossover rate is gradually reduced to retain good individuals. Therefore, the probability of finding the global optimal solution is increased.

There are two main ways of traditional mutation operations in the DE algorithm [36]:

$$(i)\ \ DE/rand/1/bin : v_i^{t+1} = x_{r1}^t + F \cdot (x_{r2}^t - x_{r3}^t),$$
$$(ii)\ \ DE/best/1/bin : v_i^{t+1} = x_{best}^t + F \cdot (x_{r2}^t - x_{r3}^t),$$

where x_{best}^t represents the best individual in the current generation, that is, the optimal position searched by this individual so far. The first mutation method has a strong global search capability but slow convergence speed. The second method has a fast convergence speed, but it is easy to fall into local optimum values. In order to overcome these shortcomings, a new mutation operation is proposed,

$$v_i^{t+1} = \lambda x_{r1}^t + (1 - \lambda) x_{best}^t + F \cdot (x_{r2}^t - x_{r3}^t). \tag{3.5}$$

In the population space of this paper, in order to jump out of the local optimum and accelerate the convergence of the algorithm. Formulas (3.4) and (3.5) will be used to improve the DE algorithm.

3.3. The belief space

Belief space adopts the $\langle S, N \rangle$ structure proposed in the literature [33]. Where $S = \{s_1^t, s_2^t, \ldots, s_i^t, \ldots, s_m^t\}$ is the set of optimal individuals, s_i^t represents the ith optimal individual in the

tth generation, and m represents the size of the set. $N = \langle X_1, X_2, \ldots, X_j, \ldots, X_n \rangle$ is the normative knowledge, representing the feasible search area of the problems, where $X_j = \langle I_j, L_j, U_j \rangle$ and n is the number of variables. $I_j = [l_j, u_j] = \{x | \ l_j \leq x_j \leq u_j\}$, l_j and u_j respectively are the upper and lower bound of variable x_j; L_j represents the fitness value corresponding to the variable l_j; U_j represents the fitness value of the variable u_j.

In order to simplify the operation process of the algorithm, only the current optimal individual s^t is used to update the S in belief space, and update it according to the following formula:

$$s^t = \begin{cases} x^t_{best}, & f(x^t_{best}) < f(s^t), \\ s^t, & otherwise, \end{cases} \tag{3.6}$$

where x^t_{best} is the optimal individual of the tth generation.

With the process of evolution, the search space gradually gathers in the dominant region. Therefore, when the individuals exceed the current search space, the normative knowledge N is updated according to the following formula:

$$l_j^{t+1} = \begin{cases} x^t_{ij}, & x^t_{ij} < l^t_j \ or \ f(x^t_{ij}) < L^t_j, \\ l^t_j, & otherwise, \end{cases} \tag{3.7}$$

$$L_j^{t+1} = \begin{cases} f(x^t_j), & x^t_{ij} < l^t_j \ or \ f(x^t_{ij}) < L^t_j, \\ L^t_j, & otherwise, \end{cases} \tag{3.8}$$

$$u_j^{t+1} = \begin{cases} x^t_{ij}, & x^t_{ij} > u^t_j \ or \ f(x^t_{ij}) < U^t_j, \\ u^t_j, & otherwise, \end{cases} \tag{3.9}$$

$$U_j^{t+1} = \begin{cases} f(x^t_j), & x^t_{ij} > u^t_j \ or \ f(x^t_{ij}) < U^t_j, \\ U^t_j, & otherwise. \end{cases} \tag{3.10}$$

Formulas (3.7) and (3.9) are the updating process of upper and lower bound of variable x_j, formulas (3.8) and (3.10) are the updating process of fitness value of corresponding variable x_j.

3.4. Acceptance function and influence function

Acceptance function and influence function are the links between belief space and population space. The acceptance function selects better individuals from the population space and submits them to the belief space for knowledge updating, and the core of its research is to select better individuals. In the early stage of evolution, the better individuals have more valuable information, so more individuals can be selected for the belief space. In the later period of evolution, the algorithm gradually converges, and the dominant individuals and their implied information are similar. In order to maintain the diversity of knowledge, the number of accepting individuals should be reduced. Therefore, the dynamic acceptance function proposed by Saleem [37] is used, the number of accepted individuals decreases while the number of generations increases. We get the number of accepted individuals with the following expression:

$$accept() = P\% \cdot N + (P\% \cdot N)/t, \tag{3.11}$$

where N is the population size, t is the current iterate moment, in this paper, we take $P\% = 20\%$, and the number of an interval to execute the accept function is usually preset (according to the maximum number of iterations, we take the number of the interval as 5).

The main purpose of the influence function is to guide the population evolution by using knowledge of the belief space. According to the ADE model (Formula (3.5), (3.2), (3.3)), if the parent individuals are within the interval given by the normative knowledge, the particles will be updated according to the ADE algorithm. If it is outside the given interval, the offspring generation rule (formula (3.3)) becomes the following form:

$$X_{ij}^{t+1} = \begin{cases} rand(0,1) \cdot (u_j - l_j) + l_j, & if\ x_{ij}^t < l_j\ or\ x_{ij}^t > u_j, \\ X_{ij}^t, & otherwise. \end{cases} \tag{3.12}$$

3.5. The ADECA algorithm design and experimental processes

The pseudo code of the ADECA algorithm is shown in Algorithm 1, and the specific implementation processes are described in detail as follows:

Step 1. Set the parameters of the ADECA algorithm, such as N, D, CR_0, F_0, l, u, P, T and accuracy ε.

Step 2. Initializing population space and belief space. Randomly generate N initial individuals $P(0)$ in population space, and each individual satisfies $\sum_{\mathscr{K}_i=1}^{m_i} x^i_{\mathscr{K}_i} = 1, x^i_{\mathscr{K}_i} \geq 0, x^i_{\mathscr{K}_i} \in X^i, i = 1, \ldots, N; \mathscr{K}_i = 1, \ldots, m_i$. The search range defined by the corresponding normative knowledge in the belief space and as the whole search space.

Step 3. Using the acceptance function (formula (3.11)) to accept better individuals and update the belief space.

Step 4. Calculating the fitness function value $f(\cdot)$ of each individual in population $P(t)$ and determining the x^t_{pbest}.

Step 5. The next generation population $P(t + 1)$ is generated by mutation operation (formula (3.5)), crossover operation (formula (3.2)), selection operation (formula (3.3)).

Step 6. Updating the normative knowledge of belief space by formulas (3.7), (3.8), (3.9), (3.10).

Step 7. Determining whether to end according to the accuracy and the maximum number of iteration, and output the optimal value, otherwise, turn to step 3.

Algorithm 1 ADECA

Input: Parameters N, D, CR_0, F_0, l, u, P, T, ε

Output: The best vector (Solution) ... Δ

$t \leftarrow 0$ (Initializing population space and belief space)
$x_{ij}^t = rand(0,1) \cdot (u_j - l_j) + l_j$
$L = min(accept\ f(\cdot))$
$U = max(accept\ f(\cdot))$
$l = min(accept\ x_{ij})$
$u = max(accept\ x_{ij})$
while $|f(\Delta)| \geq \varepsilon$ or $t \leq T$ **do**
  for $i = 1$ to N **do**
    (Mutation and Crossover)
    for $j = 1$ to D **do**
      $v_{ij}^t = Mutation\ (x_{ij}^t)$ (formula (3.5))
      $u_{ij}^t = Crossover\ (v_{ij}^t, x_{ij}^t)$
    end for
    (Selection)
    if $f(u_{ij}^t) < f(x_{ij}^t)$ **then**
      $x_{ij}^t \leftarrow u_{ij}^t$
    else $\{f(x_{ij}^t) < f(\Delta)\}$
      $\Delta \leftarrow x_{ij}^t$
    end if
  end for
  (Updating the normative knowledge)
  if $x_{ij}^t < l_j^t$ or $f(x_{ij}^t) < L_j^t$) **then**
    $l_j^t \leftarrow x_{ij}^t$
    $L_j^t \leftarrow f(x_{ij}^t)$
  else
    $l_j^t \leftarrow l_j^t$
    $L_j^t \leftarrow L_j^t$
  end if
  $t = t + 1$
end while

3.6. Convergence analysis of the ADECA algorithm

As a random optimization algorithm based on population, the evolution process of the adaptive differential evolution algorithm in the population space can be described by a random process. For the convenience of discussion, considering mutation and crossover operations as a difference operator (DO), the selection operation is regarded as a selection operator (SO). The following uses the finite state Markov chain to analyze the search process of the ADECA algorithm, and use the axiomatic model [38] to study the probability distribution of the adaptive difference population in the decision space. Then, it is proved that the ADECA algorithm weakly converges to the global optimal solution in probability.

Definition 3.1. *The DO operation randomly selects three different individuals from the population to execute formula (3.5) to generate the mutation vector V, and then performs crossover operation with the original target individual X according to formula (3.2) to generate the middle vector U. This process can be regarded as* ***a random mapping*** $\psi_D : \Omega \times S \to S^2$*, that is*

$$\psi_D(\omega, X_i) = \psi_D(\langle X, V \rangle) = U, \tag{3.13}$$

where Ω *is a nonempty abstract set, its element* ω *is a basic event and the* S *is the solution space.*

Definition 3.2. *The SO operation is a process of selecting the individual with better fitness from the middle vector* U_i *and the target vector* X_i *according to the greedy selection method, which is recorded as* ***a mapping*** $\psi_S : S^2 \to S$*,*

$$\psi_S(\langle X_i, U_i \rangle) = \min\{f(X_i), f(U_i)\}. \tag{3.14}$$

According to the above two definitions, one iteration of ADECA's population space can be seen as a mapping $\psi = (\psi_S \circ \psi_D) : \Omega \times S \to S$.

Definition 3.3. *[39] Let $X = (X_n, n = 0, 1, 2, \cdots)$ is a random process, and its state space S is a finite set. If X has Markov property defined by the following formula, that is, for any $n \geq 0$ and any state $i_0, i_1, \cdots, i_{n+1} \in S$, and $P\{X_0 = i_0, X_1 = i_1, \cdots, X_n = i_n\} > 0$, if*

$$P\{X_{n+1} = i_{n+1} | X_0 = i_0, X_1 = i_1, \cdots, X_n = i_n\} = P\{X_{n+1} = i_{n+1} | X_n = i_n\} \tag{3.15}$$

*is established, X is called **finite Markov chain**.*

Here, S represents the range of random variables $X_n, n = 0, 1, 2, \cdots$, that is, the set of all possible values. According to the above definition, the finite Markov chain is expressed as: in a finite state, the state of the next moment is only related to the state of the current moment.

In the ADECA algorithm, X is the population space, S is the solution space, X_i and X_j are the ith and jth generation of evolutionary individuals, respectively. Knowledge-guided differential operation and selection operation can be regarded as a random transformation from state $X_i \in S$ to state $X_j \in S$. In the process of state transition, state $X(t + 1)$ is only related to state $X(t)$, therefore, according to the definition of finite Markov chain, the evolution process of ADECA algorithm is a finite Markov chain. Assume that the state transition matrix of the difference operation guided by knowledge is P_D, and the state transition matrix of the selection operation is P_S. Since the knowledge that guides the difference operation and selection operation in each generation contains different information, the state transition matrix $P(t) = P_D(t)P_S(t)$ is different, that is, P is related to the evolutionary generation t. Therefore, the evolutionary process of ADECA algorithm is a non-homogeneous Markov chain (process). The following uses the axiomatic model to analyze the random search process of the ADECA algorithm.

Definition 3.4. *[40] Random variable sequence $\{X(t)\}$ **weakly converges to the global optimal solution set** S^* in probability, if it satisfies*

$$\lim_{t \to \infty} P\{X(t) \cap S^* \neq \emptyset\} = 1, \tag{3.16}$$

where $S^* = \{X | \forall Y \in S, f(X) \leq f(Y)\}$ is the global optimal solution set of the fitness function $f(\cdot)$.

Definition 3.5. *[40] If for any $X \in B, Y \notin B$, there is $f(X) < f(Y)$, then $B \subset S$ is called the **satisfactory solution set**.*

According to the definition, the fitness value of each individual in the satisfactory solution set is smaller than the fitness value of the individual outside the satisfactory solution set. Obviously, the global optimal solution set S^* is the satisfactory solution set, and is the minimum satisfactory solution set.

Theorem 3.1. *[40] If for any satisfactory solution set $B \subset S$, there is*

$$P\{X(t + 1) \cap B = \emptyset | X(t) = X\} \leq a(t), X \cap B \neq \emptyset, \tag{3.17}$$

$$P\{X(t + 1) \cap B = \emptyset | X(t) = X\} \leq b(t), X \cap B = \emptyset, \tag{3.18}$$

and satisfy: (1)$\sum_{t=1}^{\infty}(1 - b(t)) = \infty$,(2)$\frac{a(t)}{1-b(t)} \to 0$, then $\{X(t)\}$ weakly converges to the global optimal solution set S^* in probability.

Corollary 3.1. *[40] If the transition probability of* $\{X(t)\}$ *satisfies the condition:*

$$P\{X(t+1) \cap B \neq \emptyset | X(t) = X\} = \begin{cases} 1, & X \cap B \neq \emptyset, \\ \geq \delta, & X \cap B = \emptyset. \end{cases} \tag{3.19}$$

Then $\{X(t)\}$ *weakly converges to the global optimal solution set* S^* *in probability.*

Theorem 3.2. *The ADECA algorithm weakly converges to the global optimal solution set* S^* *in probability.*

Proof. In the iteration of the ADECA algorithm, assuming the current state is $X(t) = X_i$, after the differential operation and selection operation, the next generation state is $X(t+1) = X_j$.

In the ADECA algorithm, the normative knowledge K_{nor} represents the change of the feasible search space. With the evolution, the search range should be concentrated in the dominant area. In each iteration, the current optimal individual recorded by the normative knowledge is retained in the next generation population, and the optimal individual will no longer participate in the next difference operation and selection operation.

(1) When $X_i \cap B \neq \emptyset$, it means that the current optimal individual must be included in the satisfactory set B. From the optimal retention strategy, it can be seen that no matter which generation t is, there is

$$P\{X(t+1) \cap B \neq \emptyset | X(t) = X_i\} = 1, \quad X_i \cap B \neq \emptyset. \tag{3.20}$$

(2) When $X_i \cap B = \emptyset$, it indicates that the current population $X(t) = X_i$ does not contain individuals of satisfactory set B. Then we consider each population X_j after X_i, if $X_j \cap B = \emptyset$, it is obviously meaningless. Therefore, the next generation population $X_j \cap B \neq \emptyset$ is further discussed below.

When the state X_j satisfies $X_j \cap B \neq \emptyset$, that is, X_j contains the better individuals of the satisfactory set B. The transition probability generated by the difference operation under the guidance of normative knowledge is

$$\begin{aligned} p_{ij}^D &= P_D\{X(t+1) = X_j | X(t) = X_i\} \\ &= \prod_{k=1}^{n} P_D\{(X_k(t+1) = x_{jk} | X_k(t) = x_{ik}) | (Nor_k)\} P(Nor_k) \\ &= \prod_{k=1}^{n} P_D\{(X_k(t+1) = x_{jk} | X_k(t) = x_{ik}) | (Nor_k)\} (accept(Nor)), \end{aligned} \tag{3.21}$$

where Nor_k represents that the kth individual is affected by the normative knowledge, and $P(Nor_k)$ is the probability of the next generation of differential operation, $accept(Nor)$ represents the population acceptance proportion under the guidance of normative knowledge.

According to the meaning of the normative knowledge, each dimension of the individual after the difference operation does not exceed the feasible region. Therefore

$$P_D\{(X_k(t+1) = x_{jk} | X_k(t) = x_{ik}) | (Nor_k)\} > 0, \tag{3.22}$$

and because of $accpet(Nor) > 0$, so $p_{ij}^D > 0$.

Considering that the selection mechanism of survival of the fittest is used in the ADECA algorithm to obtain the next generation population, the transition probability $p_{ij}^S > 0$, so we can get

$$P\{X(t+1) \cap B \neq \emptyset | X(t) = X_i\} = p_{ij}^D p_{ij}^S > 0, \quad X_i \cap B = \emptyset. \tag{3.23}$$

It can be seen from Eqs (3.20) and (3.23) that the transition probability of the population evolution process of ADECA algorithm meets the condition of Corollary 3.1. It can be obtained that the state vector $\{X(t)\}$ of the ADECA algorithm weakly converges to the global optimal solution set S^* in probability. □

4. Experimental design and results

In this section, some matrix games are calculated to demonstrate the effectiveness of the proposed method. Taking the classic prisoner's dilemma game, the husband and wife game and the rock-paper-scissors game as examples. The algorithm ADECA, DECA and DE are used to find the Nash equilibria based on the system of equations, and the simulation comparison figures of them are given. The parameters of the algorithm are set as: $N = 100$, $T = 300$, $F_0 = 0.4$, $CR_0 = 0.9$, $P = 0.2$, $\varepsilon = 10^{-8}$, etc. The following is the process of computing these games:

Example 4.1. *Prisoner's dilemma game* $\Gamma_1(S^1, S^2, X^1, X^2, A, B)$, *where* $S^1 = \{s_1^1, s_2^1\} = \{confess, not\ confess\}$, $S^2 = \{s_1^2, s_2^2\} = \{confess, not\ confess\}$, $X^1 = \{x_1^1, x_2^1\}$, $X^2 = \{x_1^2, x_2^2\}$, *and* x_1^1, x_1^2 *respectively represent the probability of player 1, 2 choosing to confess,* x_2^1, x_2^2 *respectively represent the probability of player 1, 2 choosing not to confess, A and B are payoff matrices of the player in this game:*

$$A = \begin{bmatrix} -5 & 0 \\ -8 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} -5 & -8 \\ 0 & -1 \end{bmatrix}.$$

According to formula (2.7), the above game can be transformed into the following equations:

$$\begin{cases} \lambda_{12}^1 + x_1^1(-3x_1^2 - x_2^2) = 0, \\ \lambda_{21}^1 + x_2^1(3x_1^2 + x_2^2) = 0, \\ \lambda_{12}^2 + x_1^2(-3x_1^1 - x_2^1) = 0, \\ \lambda_{21}^2 + x_2^2(3x_1^1 + x_2^1) = 0. \end{cases} \tag{4.1}$$

Denote each equation of the above equations by f_i, $i = 1, 2, 3, 4$. *According to the formula (2.9), the Eq (4.1) is equivalent to the following optimization problem:*

$$\begin{aligned} min\ F = |f_1(\lambda_{12}^1, x_1^1, x_1^2, x_2^2)| + |f_2(\lambda_{21}^1, x_2^1, x_1^2, x_2^2)| + |f_3(\lambda_{12}^2, x_1^2, x_1^1, x_2^1)| + |f_4(\lambda_{21}^2, x_2^2, x_1^1, x_2^1)|, \\ x_1^1 + x_2^1 = 1, \\ x_1^2 + x_2^2 = 1. \end{aligned} \tag{4.2}$$

Making use of the ADECA algorithm, the DECA algorithm and the DE algorithm, we show that the solution of this problem is $x^1 = \{1, 0\}, x^2 = \{1, 0\}, \lambda_{12}^1 = \lambda_{12}^2 = 3, \lambda_{21}^1 = \lambda_{21}^2 = 0$, *that is, the prisoner's dilemma game has only one pure strategy Nash equilibrium* $(confess, confess)$.

Example 4.2. *Husband and wife game* $\Gamma_2(S^1, S^2, X^1, X^2, A, B)$, *where* $S^1 = \{s_1^1, s_2^1\} = \{football\ match, fashion\ show\}$, $S^2 = \{s_1^2, s_2^2\} = \{football\ match, fashion\ show\}$, $X^1 = \{x_1^1, x_2^1\}$, $X^2 = \{x_1^2, x_2^2\}$, *and A and B are the payoff matrices of player 1, 2 respectively, as shown below:*

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}.$$

According to the above payoff matrices and the calculation steps of Example 1, it is found that this problem has two pure strategy Nash equilibria and one mixed strategy Nash equilibrium: $x^1 = (1, 0), x^2 = (1, 0)$; $x^1 = (0, 1), x^2 = (0, 1)$; $x^1 = (1/3, 2/3), x^2 = (2/3, 1/3)$.

Example 4.3. *Rock-paper-scissors game* $\Gamma_3(S^1, S^2, X^1, X^2, A, B)$, *where* $S^1 = \{s_1^1, s_2^1, s_3^1\} = \{rock, paper, scissors\}$, $S^2 = \{s_1^2, s_2^2, s_3^2\} = \{rock, paper, scissors\}$, $X^1 = \{x_1^1, x_2^1, x_3^1\}$, $X^2 = \{x_1^2, x_2^2, x_3^2\}$ *and A, B are the payoff matrices of player 1, 2 respectively, as shown below:*

$$A = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix}.$$

Calculating this example by using the ADECA, DECA and DE algorithms, we find there is only one mixed strategy Nash equilibrium: $x^1 = (1/3, 1/3, 1/3), x^2 = (1/3, 1/3, 1/3)$.

The following Figures 2–4 are comparison figures of solving the above games with the ADECA, DECA and DE algorithms.

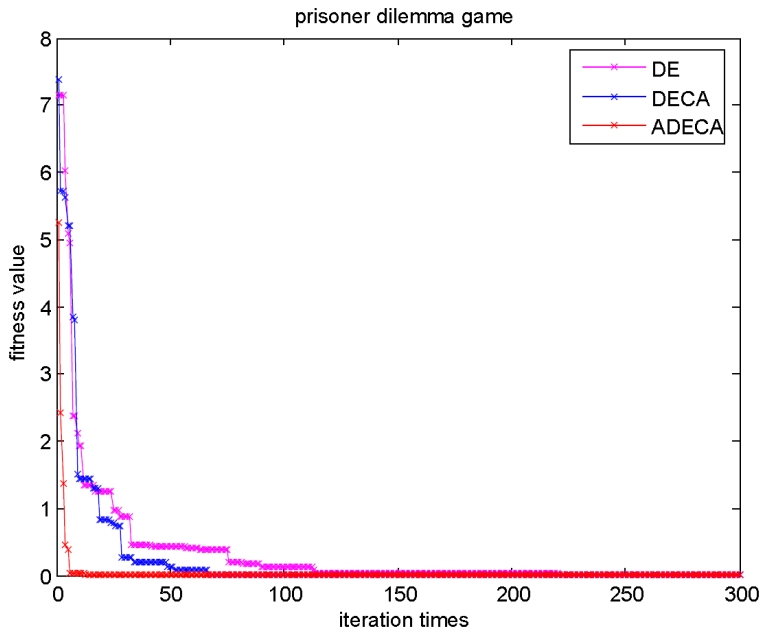

Figure 2. Comparison of solving the prisoner's dilemma game.

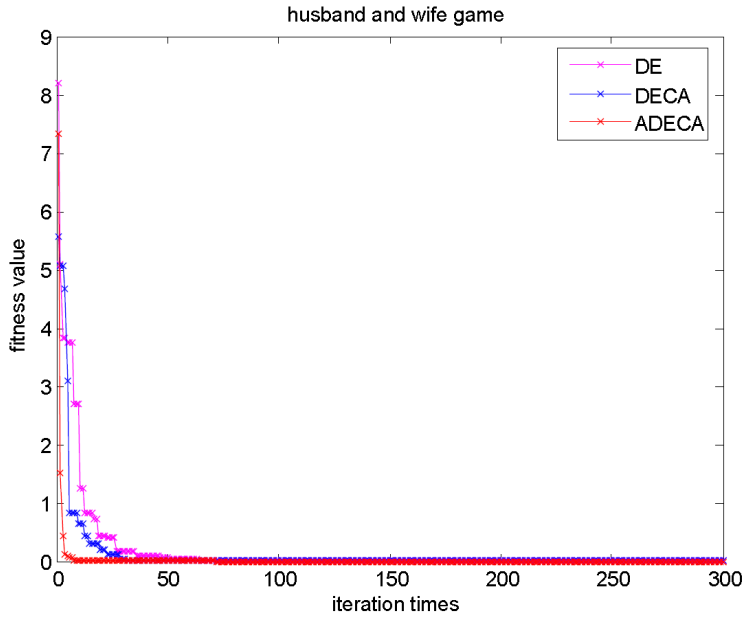

Figure 3. Comparison of solving the husband and wife game.

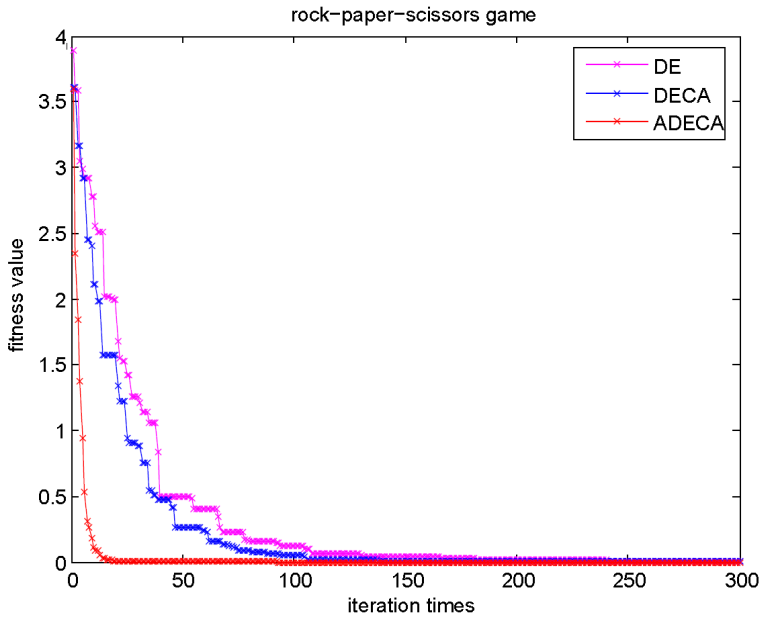

Figure 4. Comparison of solving the rock-paper-scissors game.

It can be clearly seen from Figures 2–4 that the ADECA algorithm has the best performance in the calculation process, followed by DECA algorithm, and DE algorithm is the worst, especially Figure 4 is the most obvious, where the ADECA algorithm finds the global optimal solution in the first 25 iterations, while the DECA and DE algorithms reach the optimal solution after 200 iterations. On the one hand, we found that the DECA algorithm and the DE algorithm have similar performance. This seems rational because the DECA algorithm only adds one more belief space in the implementation

process, which increases the operation of extracting individual hidden information, so it is only a little faster than DE algorithm in convergence speed. On the other hand, with the increase of the number of iterations, the diversity of individuals decreases and the algorithm easily falls into a local optimum. Therefore, neither the DE algorithm nor the DECA algorithm avoids this result. However, the adaptive parameters and improved mutation operations of the ADECA algorithm have played a role, which not only maintaining the diversity of the population in the early stage of iteration, but also avoiding the destruction of the optimal solution in the later stage of iteration, and speeding up the search for the global optimal solution.

By analyzing the results obtained above, it can be seen that the ADECA algorithm proposed in this paper can well find the equilibrium solutions of matrix games based on the system of equations. Further, comparing this algorithm with the DECA and DE algorithms, we find that this algorithm has a faster convergence speed and avoids falling into the local optimum. It provides a reference for future studies on related issues.

5. Conclusions

In this paper, we study the Nash equilibria based on the system of equations. First, we give the equivalence theorem between the Nash equilibria and the solutions of the system of equations, that is, the Nash equilibria of the n-person finite noncooperative game are equivalent to the solutions of the system of algebraic equations with a series of relaxation parameters. This method is a new method for solving Nash equilibria except that the finding Nash equilibria is equivalent to solving optimization problems and fixed point problems. It not only builds a bridge between these two types of problems so that people can use the theories and methods of studying one problem to study another problem, broadens the methods for finding the Nash equilibria and lays a mathematical foundation for finding the Nash equilibria based on equations in the future. However, the equivalence form is also derived from the definition of Nash equilibrium, which implies the rational behavior of players. Second, we provide an equivalent form of solving matrix games and propose the ADECA algorithm. This algorithm uses a simple DE algorithm as the population space of CA algorithm, and in order to improve the convergence ability of the algorithm and global search capability, the mutation factor and crossover operator of the DE algorithm are adaptively improved, and a more globally convergent mutation operation is also proposed. Then, its convergence is proved theoretically by using finite Markov chain. Finally, the superiority of the ADECA algorithm is verified through three classic matrix game examples, and we also show that this algorithm can be used to find the Nash equilibria based on the equations, which lays the foundation for future research.

Although the method proposed in this paper can solve n-person noncooperative games theoretically, and provides specific steps for solving matrix games. It has certain research significance. However, when calculating specific multiplayer games or more complex games, the fitness function becomes increasingly complex due to the increase of the number of participants and the number of strategies, which may lead to the result falling into the local optimal solution and ending the computing in advance. Therefore, in order to overcome the limitations of the algorithm in solving multiplayer games or more complex games, the algorithm needs to be further improved in the future.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (Grant Nos. [71961003] and [12061020]), the Scientific Research Foundation of Guizhou University (Grant No. [2019]49).

Conflict of interest

The authors declare no potential conflict of interests.

References

1. M. J. Beckmann, Spatial oligopoly as a noncooperative game, *Int. J. Game Theory*, **2** (1973), 263–268. https://doi.org/10.1007/BF01737575
2. A. Muthoo, M. J. Osborne, A. Rubinstein, A course in game theory, *Economica*, **63** (1996), 164–165. https://doi.org/10.2307/2554642
3. E. Braggion, N. Gatti, R. Lucchetti, T. Sandholm, B. von Stengel, Strong Nash equilibria and mixed strategies, *Int. J. Game Theory*, **49** (2020), 699–710. https://doi.org/10.1007/s00182-020-00723-3
4. J. V. Neumann, O. Morgenstern, *The theory of games and economic behaviour*, Princeton: Princeton University Press, 1944.
5. J. F. Nash, Equilibrium points in n-person games, *Proc. Natl. Acad. Sci. USA*, **36** (1950), 48–49. https://doi.org/10.1073/pnas.36.1.48
6. J. F. Nash, Non-cooperative games, *Ann. Math.*, **54** (1951), 286–295. https://doi.org/10.2307/1969529
7. I. M. Bomze, Non-cooperative two-person games in biology: a classification, *Int. J. Game Theory*, **15** (1986), 31–57. https://doi.org/10.1007/BF01769275
8. S. Govindan, R. Wilson, A global newton method to compute Nash equilibria, *J. Econ. Theory*, **110** (2003), 65–86. https://doi.org/10.1016/S0022-0531(03)00005-X
9. C. E. Lemke, J. T. Howson Jr., Equilibrium points of bimatrix games, *Journal of the Society for Industrial and Applied Mathematics*, **12** (1964), 413–423. https://doi.org/10.1137/0112033
10. T. Kunieda, K. Nishimura, Finance and economic growth in a dynamic game, *Dyn. Games Appl.*, **8** (2018), 588–600. https://doi.org/10.1007/s13235-018-0249-7
11. X. Wang, K. L. Teo, Generalized Nash equilibrium problem over a fuzzy strategy set, *Fuzzy Set. Syst.*, **434** (2022), 172–184. https://doi.org/10.1016/j.fss.2021.06.006
12. S. W. Xiang, S. Y. Xia, J. H. He, Y. L. Yang, C. W. Liu, Stability of fixed points of set-valued mappings and strategic stability of Nash equilibria, *J. Nonlinear Sci. Appl.*, **10** (2017), 3599–3611. https://doi.org/10.22436/jnsa.010.07.20
13. S. W. Xiang, D. J. Zhang, R. Li, Y. L. Yang, Strongly essential set of KyFan's points and the stability of Nash equilibrium, *J. Math. Anal. Appl.*, **459** (2018), 839–851. https://doi.org/10.1016/j.jmaa.2017.11.009

14. J. P. Torres-Martínez, Fixed points as Nash equilibria, *Fixed Point Theory Appl.*, **2006** (2006), 36135. https://doi.org/10.1155/FPTA/2006/36135

15. M. Schaefer, D. Štefankovič, Fixed points, Nash equilibria, and the existential theory of the reals, *Theory Comput. Syst.*, **60** (2015), 172–193. https://doi.org/10.1007/s00224-015-9662-0

16. H. Mills, Equilibrium points in finite games, *Journal of the Society for Industrial and Applied Mathematics*, **8** (1960), 397–402. https://doi.org/10.1137/0108026

17. N. G. Pavlidis, K. E. Parsopoulos, M. N. Vrahatis, Computing Nash equilibria through computational intelligence methods, *J. Comput. Appl. Math.*, **175** (2005), 113–136. https://doi.org/10.1016/j.cam.2004.06.005

18. A. S. Strekalovskii, R. Enkhbat, Polymatrix games and optimization problems, *Autom. Remote Control*, **75** (2014), 632–645. https://doi.org/10.1134/S0005117914040043

19. E. Rentsen, N. Tungalag, A. Gornov, A. Anikin, The curvilinear search algorithm for solving three-person game, In: *Supplementary Proceedings of the 9th International Conference on Discrete Optimization and Operations Research and Scientific School (DOOR 2016)*, Vladivostok, Russia, September 19–23, 2016, 574–583.

20. H. M. Li, S. W. Xiang, Y. L. Yang, C. W. Liu, Differential evolution particle swarm optimization algorithm based on good point set for computing nash equilibrium of finite noncooperative game, *AIMS Mathematics*, **6** (2021), 1309–1323. https://doi.org/10.3934/math.2021081

21. J. T. Howson Jr., Equilibria of polymatrix games, *Manage. Sci.*, **18** (1972), 312–318. https://doi.org/10.1287/mnsc.18.5.312

22. J. Y. Han, N. H. Xiu, H. D. Qi, *Nonlinear complementarity theory and algorithm*, (Chinese), Shanghai: Shanghai Scienceand Technology Press, 2006.

23. Z. H. Huang, L. Qi, Formulating an n-person noncooperative game as a tensor complementarity problem, *Comput. Optim. Appl.*, **66** (2017), 557–576. https://doi.org/10.1007/s10589-016-9872-7

24. C. Chen, L. Zhang, Finding nash equilibrium for a class of multi-person noncooperative games via solving tensor complementarity problem, *Appl. Numer. Math.*, **145** (2019), 458–468. https://doi.org/10.1016/j.apnum.2019.05.013

25. F. Salehisadaghiani, L. Pavel, Distributed nash equilibrium seeking: a gossip-based algorithm, *Automatica*, **72** (2016), 209–216. https://doi.org/10.1016/j.automatica.2016.06.004

26. G. Chen, Y. Ming, Y. Hong, P. Yi, Distributed algorithm for ε-generalized Nash equilibria with uncertain coupled constraints, *Automatica*, **123** (2021), 109313. https://doi.org/10.1016/j.automatica.2020.109313

27. Y. Zou, B. Huang, Z. Meng, W. Ren, Continuous-time distributed nash equilibrium seeking algorithms for non-cooperative constrained games, *Automatica*, **127** (2021), 109535. https://doi.org/10.1016/j.automatica.2021.109535

28. C. Cenedese, G. Belgioioso, S. Grammatico, M. Cao, An asynchronous distributed and scalable generalized nash equilibrium seeking algorithm for strongly monotone games, *Eur. J. Control*, **58** (2021), 143–151. https://doi.org/10.1016/j.ejcon.2020.08.006

29. S. Liang, P. Yi, Y. Hong, Distributed Nash equilibrium seeking of a class of aggregative games, In: *2017 13th IEEE International Conference on Control & Automation (ICCA)*, Ohrid, Macedonia, 2017, 58–63. https://doi.org/10.1109/ICCA.2017.8003035

30. M. Ye, G. Hu, Distributed Nash equilibrium seeking by a consensus based approach, *IEEE Trans. Automat. Contr.*, **62** (2017), 4811–4818. https://doi.org/10.1109/TAC.2017.2688452

31. X. Wang, A computational approach to dynamic generalized Nash equilibrium problem with time delay, *Commun. Nonlinear Sci. Numer. Simulat.*, **117** (2023), 106954. https://doi.org/10.1016/j.cnsns.2022.106954

32. J. Yu, *Selection of game theory*, (Chinese), Beijing: Science Press, 2014.

33. R. G. Reynolds, W. Sverdlik, Problem solving using cultural algorithms, In: *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, Orlando, FL, USA, 1994, 645–650. https://doi.org/10.1109/ICEC.1994.349983

34. R. Storn, K. Price, Minimizing the real functions of the icec'96 contest by differential evolution, In: *Proceedings of IEEE international conference on evolutionary computation*, Nagoya, Japan, 1996, 842–844. https://doi.org/10.1109/ICEC.1996.542711

35. K. Price, P. M. Storn, J. A. Lampinen, *Differential evolution: a practical approach to global optimization*, Berlin, Heidelberg: Springer, 2005. https://doi.org/10.1007/3-540-31306-0

36. N. Noman, H. Iba, Enhancing differential evolution performance with local search for high dimensional function optimization, In: *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, 2005, 967–974. https://doi.org/10.1145/1068009.1068174

37. S. M. Saleem, Knowledge-based solution to dynamic optimization problems using cultural algorithms, PhD thesis, Wayne State University, Detroit, Michigan, 2001.

38. Z. B. Xu, Z. P. Chen, X. S. Zhang, Theoretical development on genetic algorithms: a review, (Chinese), *Advances in Mathematics*, **29** (2000), 97–114. https://doi.org/10.3969/j.issn.1000-0917.2000.02.001

39. B. Zhang, J. X. Zhang, *Applied stochastic processes*, (Chinese), Beijing: Tsinghua University Press, 2006.

40. W. X. Zhang, Y. Liang, *Mathematical foundation of geneticalgorithms*, (Chinese), Xi'an: Xi'an Jiaotong University Press, 2001.

AIMS Press