J. Nonlinear Funct. Anal. 2022 (2022) 8 https://doi.org/10.23952/jnfa.2022.8

MATHRES

Journal of Nonlinear Functional Analysis

Available online at http://jnfa.mathres.org

IMPROVED DIFFERENTIAL EVOLUTION ALGORITHM FOR SOLVING MULTI-PERSON NONCOOPERATIVE GAME

HUIMIN LI[1,*], SHUWEN XIANG[1,2,*], SHIGUO HUANG[1,3], WENSHENG JIA[1], YANLONG YANG[1]

[1]College of Mathematics and Statistics, Guizhou University, Guiyang 550025, China
[2]College of Mathematics and Information Science, Guiyang University, Guiyang 550005, China
[3]Department of Mathematics and Information Science,
Zhengzhou University of Light Industry, Zhengzhou 450002, China

Abstract. In this paper, the nonzero sum multi-person noncooperative game is considered, and this game can be viewed as a global optimization problem. For solving this problem, an improved differential evolution algorithm is proposed. Our algorithm combines multi-strategy differential operators and crossover operators with a random walk. First, the population is initialized by a set of good points. Second, the single mutation mode is replaced by a multi-mode. Meanwhile, a random walk mechanism is introduced in the crossover process of the differential evolution algorithm. The convergence of the improved algorithm is then proved by using an infinite product method. Finally, the proposed algorithm is illustrated via some numerical examples, and the experimental results show that the proposed new algorithm can solve the multi-person game.

Keywords. Global optimization problem; Improved differential evolution algorithm; Multi-person noncooperative game; Nash equilibrium.

1. INTRODUCTION

It is well known that game theory plays an important role in mathematics, economics, and operations research [1, 16, 17, 18, 21, 26]. Noncooperative game is the classic content of game theory research for n-person noncooperative games. Nash [22, 23] proposed a very important concept of equilibrium, called Nash equilibrium, which is a stable outcome in the sense that a unilateral deviation from a Nash equilibrium point by one of the players does not increase the payoff of that player. However, achieving equilibrium requires players to make predictions according to certain steps in the game, and give a good mathematical description of the problem. Therefore, solving the Nash equilibria can be reduced to a calculation problem. At present,

*Corresponding author.
E-mail address: 13783513360@163.com (Huimin Li), shwxiang@vip.163.com (S. Xiang).
Received October 2, 2021; Accepted November 18, 2021.

using computational methods to seek the Nash equilibrium points of games have been intensively studied, such as the Lemke-Howson algorithm [19], the global Newton algorithm [10], the projection-like method [34], and other traditional numerical methods.

In recent years, many scholars focused on intelligent algorithms for solving games. Basing on the rationality of imitating biological behaviors, such algorithms have the implicit rational characteristics of games. It means that such algorithms can be considered to solve the Nash equilibria. Pavlidis [25], Boryczka [3], Jia [15], Yang [30], and Li [20] used intelligent algorithms to compute the Nash equilibria of corresponding games. But, most of them dealt with zero sum two-person games or nonzero sum two-person games. In addition, some scholars used numerical methods to solve multi-person games [2, 6, 7, 8, 13, 4]. Among them, in [2, 6, 7, 8], the problem of finding the Nash equilibrium points as a nonconvex optimization problem by generalizing Mills theorem, in [4, 13], the multi-person noncooperative game as a tensor complementarity problem are solved by the smoothing type algorithm and the semi-smooth Newton method, respectively. However, less attention has been paid to intelligent algorithms for solving nonzero sum multi-person games. This paper is aim to fulfill this gap.

The paper is organized as follows: In Section 2, we formulate nonzero sum multi-person game model and prove that it is equivalent to a global optimization problem. In Section 3, we propose an improved differential evolution algorithm (IDE) to compute the Nash equilibria of the nonzero sum multi-person games. First, the population is initialized with a good point set to make the initial distribution is global. Second, a mutation operator with multi-mode is introduced into the differential evolution (DE) algorithm, which makes the algorithm have the characteristics of multi-population and multi-mode coordinated evolution. Finally, a random walk mechanism is integrated into the crossover operator to enhance the diversity of the population. The convergence of the IDE is proved theoretically in Section 4. Section 5 is devoted to computational experiments. Section 6 ends this paper.

2. Nonzero Sum Multi-person Game

Consider a multi-person game [31] $\Gamma = \{(N, S_i, U_i, X_i, f_i), i = 1, \ldots, n, n \geq 2\}$, where

(1) $N = \{1, \ldots, n : n \geq 2\}$ is the set of players and n is the number of players;

(2) $S_i = \{s_{i1}, \ldots, s_{im_i}\}$, $\forall i \in N$ is the pure strategy set of player i, m_i represents the number of feasible strategies of player i, $S = \prod_{i=1}^{n} S_i$, and each pure strategy profile meets $(s_{1m_1}, s_{2m_2}, \ldots, s_{ij}, \ldots, s_{nm_n}) \in S, s_{ij} \in S_i$;

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

(4) $X_i = \{x_i = (x_{i1}, \ldots, x_{ik}, \ldots, x_{im_i}) : x_{ik} \geq 0, k = 1, \ldots, m_i, \sum_{k=1}^{m_i} x_{ik} = 1\}$, $\forall i \in N$ is the set of mixed strategies. $X = \prod_{i=1}^{n} X_i$, and each mixed strategy profile meets $(x_1, x_2, \ldots, x_n) \in X$;

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

$$f_i(x_1, \ldots, x_n) = \sum_{k_1=1}^{m_1} \cdots \sum_{k_n=1}^{m_n} U_i(s_{1k_1}, \ldots, s_{nk_n}) \prod_{i=1}^{n} x_{ik_i},$$

where $f_i(x_1, \ldots, x_n)$ represents the expected payoff value of player i when he chooses a mixed strategy $x_i = (x_{i1}, \ldots, x_{im_i}) \in X_i$. $U_i(s_{ik_i}, \ldots, s_{nk_n})$ represents player i obtaining the payoff value when each player chooses pure strategy $s_{ik_i} \in S_i, i = 1, \ldots, n$.

Denote by

$$f_i(x \shortparallel s_{ik_i}) \triangleq f_i(x_1,\ldots,x_{i-1},s_{ik_i},x_{i+1},\ldots,x_n) \triangleq$$
$$\sum_{k_1=1}^{m_1}\cdots\sum_{k_{i-1}=1}^{m_{i-1}}\sum_{k_{i+1}=1}^{m_{i+1}}\cdots\sum_{k_n=1}^{m_n}U_i(s_{1k_1},\ldots,s_{nk_n})x_{1k_1}\cdots x_{i-1k_{i-1}}x_{i+1k_{i+1}}\cdots x_{nk_n},$$

where $s_{ik_i}(1\le k_i\le m_i)$ is a pure strategy of player i, $(x \shortparallel s_{ik_i})$ represents s_{ik_i} of player i instead of x_i, and the other players do not change their own mixed strategy with x.

If $x^*=(x_1^*,\ldots,x_n^*)\in X$ such that $f_i(x_i^*,x_{i^\wedge}^*)=\max\limits_{u_i\in X_i}f_i(u_i,x_{i^\wedge}^*)$, $\forall i\in N$, then x^* is a Nash equilibrium point of multi-person noncooperative game, where $i^\wedge=N\backslash\{i\}$, $\forall i\in N$.

Conclusion 2.1. A mixed strategy $x^*\in X$ is the Nash equilibrium point of a game Γ if and only if every pure strategy $s_{ik_i}(1\le k_i\le m_i)$ of each player satisfies $f_i(x^*)\ge f_i(x^*\shortparallel s_{ik_i})$.

Theorem 2.2. [20] *A mixed strategy $x^*\in X$ is the Nash equilibrium point of a game Γ if and only if x^* is an optimal solution to the following optimization problem, and the optimal value is 0:*

$$\begin{cases}\min\ f(x)=\sum\limits_{i=1}^{n}\max\limits_{1\le k_i\le m_i}\{f_i(x\shortparallel s_{ik_i})-f_i(x),0\}\\ \sum\limits_{k_i=1}^{m_i}x_{ik_i}=1\\ 0\le x_{ik_i}\le 1\\ i=1,\cdots,n;k_i=1,\cdots,m_i\end{cases}\tag{2.1}$$

Compared with [2, 8] and [4, 13], this transformation is simpler and easier to implement than the nonconvex optimization problem and the tensor complementarity problem, which does not need to calculate the parameters and construct a tensor complementarity problem.

3. The Improved Differential Evolution (IDE) Algorithm

In this section, we outline a novel DE algorithm, IDE algorithm, and explain the steps of the algorithm in details.

3.1. Good Point Set.

The good point set was originally proposed by Loo-Keng Hua et al. [14], and defined as follows:

(1) Let G_s be a unit cube in s-dimensional Euclidean space. Let $x\in G_s$, that is,

$$x=(x_1,x_2,\ldots,x_s)\in G_s, 0\le x_j\le 1, j=1,\ldots,s.$$

(2) Assume that G_s has a set of points $P_n(i)$ with n points.

$$P_n(i)=\{(\{x_1^{(n)}(i)\},\cdots,\{x_j^{(n)}(i)\}\cdots,\{x_s^{(n)}(i)\}),i=1,\cdots,n\},0\le x_j^{(n)}(i)\le 1,j=1,2,\cdots,s,$$

where $\{\cdot\}$ represents the decimal part of the value.

(3) For any given point $r=(r_1,r_2,\cdots,r_s)\in G_s$, let $N_n(r)=N_n(r_1,r_2,\cdots,r_s)$ represents the number of points in $P_n(i)$ that meet the inequality $0\le x_j^{(n)}(i)\le r_j$, $j=1,2,\cdots,s$. The $\varphi(n)=\sup_{r\in G_s}|(N_n(r)/n)-|r||$, where $|r|=r_1r_2\cdot\cdots\cdot r_s$, is called a deviation of the point set $P_n(i)$. If $\forall n$, $\varphi(n)=O(1)$, then $P_n(i)$ is said to be uniformly distributed on G_s and the deviation is $\varphi(n)$.

(4) Let $r \in G_s$, and $P_n(i) = \{(\{r_1 * i\}, \{r_2 * i\}, \cdots, \{r_s * i\}), i = 1, \cdots, n\}$, the deviation $\varphi(n)$ meets $\varphi(n) = C(r,\varepsilon) \cdot n^{-1+\varepsilon}$, where $C(r,\varepsilon)$ is a constant related only to r, ε(ε is an arbitrarily small positive number). Then $P_n(i)$ is called the good point set, and r is called the good point.

In this paper, take

$$r_k = \{2\cos\frac{2\pi k}{p}, 1 \le k \le s\},$$

where p is the smallest prime satisfying $p \ge 2 \cdot s + 3$.

In the following, we generate two distribution figures of 300 populations with the good point set method and the random point method, respectively.

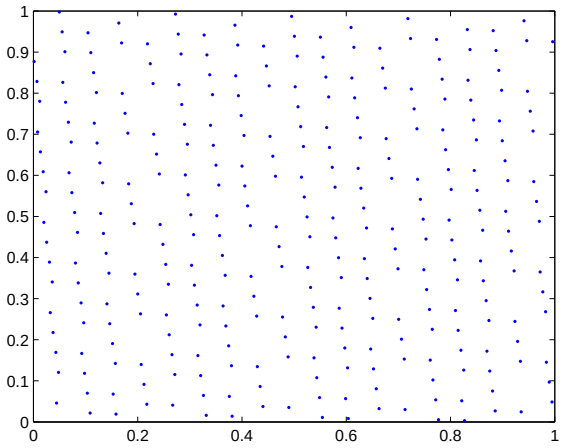

FIGURE 1. Two dimensional initial population generated by good point set.

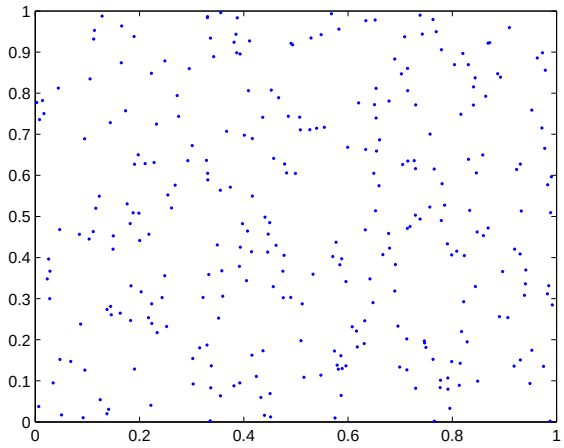

FIGURE 2. Two dimensional initial population generated by random method.

As can be seen from the above figures, the initial population distribution generated by the good point set method is more uniform and global than the random method, with no overlapping points, and has better diversity.

3.2. **Differential Evolution (DE) Algorithm.** The DE algorithm is a new simple and robust evolutionary algorithm, which was first introduced by Storm and Price [28]. Let the population size and space dimension be N and D, respectively, and each individual in the population be $X = (x_{i1}, \ldots, x_{ij}, \ldots, x_{iD})$. In the study of the DE, individuals can be generally produced by the following formula:

$$x_{ij} = rand[0,1] \cdot (x_{j,max} - x_{j,min}) + x_{j,min} \quad i = 1, \ldots, N; j = 1, \ldots, D,$$

where $rand[0,1]$ represents random values in the range [0,1], and $x_{j,max}$ and $x_{j,min}$ respectively are the upper and lower bounds of variable x_j. The basic operations of DE include mutation, crossover, and selection operation.

(1) Mutation

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

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

where x_{r1}, x_{r2} and x_{r3} are three different individuals randomly selected from the parent population X, and $r1 \neq r2 \neq r3 \neq i \in [1, N]$. F is a mutation factor to control the size of difference of two individuals, which is generally between $[0, 2]$, t is the current generation.

(2) Crossover

The purpose of the crossover operation is to improve the diversity of the population through random recombination of the dimensional components of the mutation vector V and the target vector X. The algorithm generates the crossover vector $U = (u_{i1}, \ldots, u_{iD})$ through the following formula:

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

where $rand(j) \in [0, 1]$ is a random value, $CR \in [0, 1]$ is crossover operator, and $rnbr(i) \in \{1, \ldots, D\}$ is a randomly selected integer, which ensures that new individual at least one component value is inherited from the mutation vector.

(3) Selection

In the problem with boundary constraints, it is necessary to ensure that the parameter values of the new individuals are in the feasible region. If the individuals are outside the bounds, the boundary treatment is performed first, which the new individuals beyond the bounds are replaced by the parameter vectors randomly generated in the feasible region. Then the offspring X_i^{t+1} is generated by selecting the new individual and target vector individual according to the following formula:

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

where $f(\cdot)$ is the fitness function. The pseudo code of the standard DE algorithm is shown in Algorithm 1.

According to the operation principle of the DE, it can be seen that DE has the characteristics of free exploration, learning, and inheritance. The free exploration is due to the vectors X_{r1}, X_{r2}, and X_{r3} are randomly selected from X in the mutation operation. The learning and inheritance are because the components of U come from the mutation vector V or at least one dimension generated by vector V according to a certain probability in the crossover operation. Although the DE algorithm has many excellent features, 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 solve the multi-person game better, the DE algorithm is improved.

3.3. **Improved Differential Evolution (IDE) Algorithm and its Implementation.** The above method is the most basic DE operation. In practical applications, the DE algorithm has multiple evolution modes, which are generally expressed in the form of $DE/x/y/z$ [24], where x indicates that the currently variant vector is "random" or "best"; y is the number of difference vectors

Algorithm 1 DE

Input: Parameters N, D, T, F, CR, ε
Output: The best vector (Solution) $\cdots$ Δ

$t \leftarrow 0$ (Initialization)
for $i = 0$ to N **do**
  for $j = 0$ to D **do**
    $x_{i,j}^t = rand[0,1] \cdot (x_{i,j}^U - x_{i,j}^L) + x_{i,j}^L$
  end for
end for
while $|f(\Delta)| \geq \varepsilon$ or $t \leq T$ **do**
  for $i = 1$ to N **do**
    (Mutation)
    for $j = 1$ to D **do**
      $v_{i,j}^t = Mutation\ (x_{i,j}^t)$ (formula (3.1))
    end for
    (Crossover)
    for $j = 1$ to D **do**
      $u_{i,j}^t = Crossover\ (v_{i,j}^t, x_{i,j}^t)$ (formula (3.2))
    end for
    (Selection)
    if $f(u_{i,j}^t) < f(x_{i,j}^t)$ **then**
      $x_{i,j}^t \leftarrow u_{i,j}^t$
    else
      $\Delta \leftarrow x_{i,j}^t$
    end if
  end for
  $t = t + 1$
end while
return *the best vector* Δ

used; z represents the crossover scheme, mainly including binomial and exponential. The above DE operation adopts the $DE/rand/1/bin$, the other modes can be expressed as:

$$DE/best/1/bin:\ v_i^{t+1} = x_{best}^t + F \cdot (x_{r2}^t - x_{r3}^t)\ i = 1, \ldots, N, \tag{3.4}$$

$$DE/rand-to-best/1/bin:\ v_i^{t+1} = x_{i,j}^t + F \cdot (x_{best}^t - x_{i,j}^t) + F \cdot (x_{r1}^t - x_{r2}^t)\ i = 1, \ldots, N, \tag{3.5}$$

where x_{best}^t represents the best individual in the current generation, that is, the optimal position searched by this individual so far [29, 33].

In the basic DE, there are three types of evolution strategies $DE/rand/1/bin$, $DE/best/1/bin$, and $DE/rand-to-best/1/bin$. The IDE integrates these three evolution strategies, that is, one of them is selected during each iteration. Therefore, the IDE with multiple evolution strategies not only has all the characteristics of DE, but also has the characteristics of multi-group and multi-strategy coordination evolution [11]. Moreover, in order to make the DE algorithm have better global search capability and convergence speed, an adaptive mutation operator is adopted:

$$F = F_0 \cdot 2^{\lambda},$$
$$\lambda = e^{1 - T/(T+1-t)}.$$

In order to further enhance the population diversity of the DE algorithm and improve the ability of the algorithm to jump out of local optimum [32], the random walk mechanism is integrated into the crossover operation, and its expression is:

$$u_{ij}^{t+1} = \begin{cases} v_{ij}^{t+1},\ (rand(j) \leq CR) or (j = rnbr(i)), \\ rand(L_j, U_j),\ rand(0,1) \leq RW, \\ x_{ij}^{t+1},\ otherwise, \end{cases} \tag{3.6}$$

where $RW = 0.1 - 0.099 \cdot t/T$, and $L_j, U_j, j = 1, \cdots, D$ are the lower and upper bounds of the jth dimension parameter and $rand(L_j, U_j)$ is a random number subject to uniform distribution between L_j and U_j.

The pseudo code of the IDE algorithm is shown in Algorithm 2, and the specific steps are described in detail as follows:

Step 1. Set the parameters of the IDE, such as N, D, CR, F_0, x_{min}, x_{max}, T, ε.

Step 2. Randomly generate N initial populations $P(0)$ by using a good point set, and satisfy $\sum_{k_i=1}^{m_i} x_{ik_i} = 1, x_{ik_i} \geq 0, x_{ik_i} \in X_i, i = 1, \ldots, N; k_i = 1, \ldots, m_i$.

Step 3. Calculate the fitness function value $f(x)$ of each individual in population $P(t)$ and determine the x^t_{pbest}.

Step 4. The next generation population $P_1(t)$ is generated by selecting mutation of formulas (3.1), (3.4), and (3.5) in turn .

Step 5. The population $P_2(t)$ is generated by the crossover of formula (3.6).

Step 6. According to formula (3.3), selecting of population $P(t)$ and $P_2(t)$ is to generate offspring population $P(t+1)$ and the fitness function value of population $P(t+1)$ is calculated.

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

Algorithm 2 IDE

Input: Parameters $N, D, CR, F_0, x_{min}, x_{max}, T, \varepsilon$

Output: The best vector (Solution) $\cdots \Delta$

$t \leftarrow 1$ (Initialization with good point set)
for $i = 0$ to N **do**
  for $j = 0$ to D **do**
    $x^t_{i,j} = rand(0,1) \cdot (x_{j,max} - x_{j,min}) + x_{j,min}$
  end for
end for
while $|f(\Delta)| \geq \varepsilon$ or $t \leq T$ **do**
  for $i = 1$ to N **do**
    (Multi-mode Mutation)
    for $j = 1$ to D **do**
      $v^t_{i,j} = Mutation(x^t_{i,j})$ (formulas (3.1), (3.4), (3.5))
      $(mod1\ v^t_{i,j}, or\ mod2\ v^t_{i,j}, or\ mod3\ v^t_{i,j})$
    end for
    (Crossover with random walk)
    Let $RW = 0.1 - 0.099 \cdot t/T$
    for $j = 1$ to D **do**
      if $rand(j) \leq CR) or (j = rnbr(i)$ **then**
        $u^t_{i,j} \leftarrow v^t_{i,j}$
      else $\{rand(0,1) \leq RW\}$
        $rand(L_j, U_j)$ (formula (3.6))
      else
        $u^t_{i,j} \leftarrow x^t_{i,j}$
      end if
    end for
    (Selection)
    if $f(u^t_{i,j}) < f(x^t_{i,j})$ **then**
      $x^t_{i,j} \leftarrow u^t_{i,j}$
    else $\{f(x^t_{i,j}) < f(\Delta)\}$
      $\Delta \leftarrow x^t_{i,j}$
    end if
  end for
  $t = t + 1$
end while
return *the best vector* Δ

Comparing the implementation process of the two algorithms, we can see that only one mutation model of the IDE algorithm is executed in each iteration, and the crossover operation is

one more judgment than the DE algorithm in each iteration. These improved operations do not add additional calculations, so the time complexity of the IDE is the same as the DE algorithm, which provides a guarantee for the performance of the IDE algorithm proposed in this paper. In the next section, we strictly prove another important performance of the algorithm, global convergence.

4. Convergence Analysis of the IDE Algorithm

Here, we list some classic results on the convergence of the basic DE. In [9], Ghosh *et al.* proved the local convergence of the basic DE on a class of special function based on the Lyapunov's stability. In [11], He *et al.* established a random functional model of the DE and used the random contraction mapping theorem to prove that the basic DE gradually converges to the random fixed point. In [27], the absorption state of the Markov chain was used to prove that the basic DE cannot ensure global convergence in probability. However, there are few studies on the convergence of improved differential evolution algorithm. In this section, we use the infinite products to prove the global convergence of the IDE algorithm proposed in this paper.

Property 4.1. (The property of the infinite product) [5] If the series $\sum_{i=1}^{+\infty}\ell(i)$ $(0<\ell(i)<1)$ is discrete, then $\prod_{i=1}^{+\infty}(1-\ell(i))=0$.

Definition 4.2. Let $\{X(t),t=0,1,\cdots\}$ be a population sequence generated by using IDE to solve the optimization problem (2.1). Then IDE converges to the global optimum if and only if

$$\lim_{t\to\infty} p\{X(t)\cap S_\delta^*\neq\emptyset\}=1,$$

where δ is a small positive real number, $S_\delta^*=\{x|\ |f(x)-f(x^*)|<\delta\}$ is the expanded optimal solution set, and x^* is an optimum solution to optimization problem (2.1).

Theorem 4.3. [12] *Let $\{X(t),t=0,1,\cdots\}$ be a population sequence of DE with a greedy selection operator. In the t_kth target population $X(t_k)$, it corresponds to the trial individual $U(t_k)$, and there is at least one individual $u\in U(t_k)$, which meets that*

$$p\{u\in S_\delta^*\}\geq\ell(t_k)>0, \tag{4.1}$$

and the series $\sum_{i=1}^{+\infty}\ell(t_k)$ diverges. Then, DE is global convergent in probability, where $\{t_k,k=1,2,\cdots\}$ denotes any subsequence of natural number set, $p\{u\in S_\delta^*\}$ denotes the probability that u belongs to the optimal solution set S_δ^*, and $\ell(t_k)$ is a small positive value depending on t_k.

The following shows that the IDE algorithm proposed in this paper is globally convergent.

Conclusion 4.4. The IDE algorithm converges globally in probability.

Proof. From Theorem 4.3, it needs to prove that the IDE algorithm satisfies the following two characteristics.

(1) The IDE algorithm can remain the optimal solution of the current population enter into the next generation.

The IDE algorithm uses the parent-child competition selection operation of the DE, so the optimal solution in the population is greedily remain to the next generation.

(2) The probability of trial individuals in each generation entering into the global optimal solution set S_δ^* is large enough.

The IDE's reproduce operators include the mutation operator of multiple evolution strategies and crossover operator with random walk operator. During each iteration, the IDE selects one of the mutation models. Moreover, in the crossover operation with the random walk, the probability of trial vector u entering the S_δ^* is equal to the sum of the three parts: the probability of v entering S_δ^*, the probability of individual generated by random walk operator entering S_δ^*, and the probability of x entering S_δ^*. Therefore, under the crossover operation with the random walk, the probability of trial vector u entering S_δ^* is greater than the probability of v plus x entering S_δ^*, that is, the probability of basic DE trial vector u entering into S_δ^*. So the probability

$$p\{u_{IDE} \in S_\delta^*\} \geq p\{u_{DE} \in S_\delta^*\} = (1 - RW^D) \cdot \frac{\mu(S_\delta^*)}{\mu(\psi)} > 0,$$

where $\mu(\cdot)$ represents the measure of a set, and ψ is solution space. Let $0 < \ell(t) = (1 - RW^D) \cdot \frac{\mu(S_\delta^*)}{\mu(\psi)} < 1$. Obviously, the general term of the series is not limited to 0. Then, the infinite series $\sum_{i=1}^{\infty} \ell(t)$ diverges. So, according to Theorem 4.3, we can conclude the desired conclusion immediately. □

5. Experimental Design and Results

In this section, the proposed IDE algorithm is applied to multi-person noncooperative games. For the three-person noncooperation game, only the $DE/rand/1/bin$ mutation operator can be used to solve. The other two examples are performed by using the multi-strategy mutation operator. The parameters of the algorithm are set as: $N = 30$, $T = 150$, $CR = 0.2$, $\varepsilon = 10^{-5}$, etc. Eventually, in all the cases, the Nash equilibrium points are found successfully. These examples are as follows:

Example 5.1. [8] Assume that the pure strategy sets of player 1, 2, and 3 respectively are $A = (a_1, a_2)$, $B = (b_1, b_2)$, and $C = (c_1, c_2)$, and mixed strategies are $x_1 = (x_{11}, x_{12})$, $x_2 = (x_{21}, x_{22})$, and $x_3 = (x_{31}, x_{32})$. Their strategy profiles and corresponding payoffs are shown in Table 1:

Table 1. The three-player game $\Gamma_1(A, B, C, f_1, f_2, f_3)$.

Strategy profiles	Payoff vector	Strategy profiles	Payoff vector
(a_1, b_1, c_1)	(5,2,2)	(a_2, b_1, c_1)	(0,3,-2)
(a_1, b_1, c_2)	(3,4,0)	(a_2, b_1, c_2)	(8,5,6)
(a_1, b_2, c_1)	(6,-1,-4)	(a_2, b_2, c_1)	(2,4,8)
(a_1, b_2, c_2)	(7,0,-1)	(a_2, b_2, c_2)	(1,9,9)

According to Table 1, problem (2.1) can be written as:

$$\begin{aligned} \min \quad & f(x) = max(h_{11} - f_1, h_{12} - f_1, 0) + max(h_{21} - f_2, h_{22} - f_2, 0) \\ & + max(h_{31} - f_3, h_{32} - f_3, 0). \\ & \begin{cases} x_{11} + x_{12} = 1, x_{21} + x_{22} = 1, x_{31} + x_{32} = 1, \\ 0 \leq x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32} \leq 1, \end{cases} \end{aligned} \tag{5.1}$$

where

$$
\begin{aligned}
&f_1 = 5x_{11}x_{21}x_{31} + (3x_{11} + 8x_{12})x_{21}x_{32} + (6x_{11} + 2x_{12})x_{22}x_{31} + (7x_11 + x_12)x_{22}x_{32},\\
&h_{11} = f_1(x \shortparallel s_{11}) = 5x_{21}x_{31} + 3x_{21}x_{32} + 6x_{22}x_{31} + 7x_{22}x_{32},\\
&h_{12} = f_1(x \shortparallel s_{12}) = 8x_{21}x_{32} + 2x_{22}x_{31} + x_{22}x_{32},\\
&f_2 = (2x_{21} - x_{22})x_{11}x_{31} + 4x_{21}x_{11}x_{32} + (3x_{21} + 4x_{22})x_{12}x_{31} + (5x_{21} + 9x_{22})x_{12}x_{32},\\
&h_{21} = f_1(x \shortparallel s_{21}) = 2x_{11}x_{31} + 4x_{11}x_{32} + 3x_{12}x_{31} + 5x_{12}x_{32},\\
&h_{22} = f_1(x \shortparallel s_{22}) = -x_{11}x_{31} + 4x_{12}x_{31} + 9x_{12}x_{32},\\
&f_3 = 2x_{31}x_{11}x_{21} + (-4x_{31} - x_{32})x_{11}x_{22} + (-2x_{31} + 6x_{32})x_{12}x_{21} + (8x_{31} + 9x_{32})x_{12}x_{22},\\
&h_{31} = f_1(x \shortparallel s_{31}) = 2x_{11}x_{21} - 4x_{11}x_{22} - 2x_{12}x_{21} + 8x_{12}x_{22},\\
&h_{32} = f_1(x \shortparallel s_{32}) = -x_{11}x_{22} + 6x_{12}x_{21} + 9x_{12}x_{22}.
\end{aligned}
$$

The Nash equilibrium points of this problem are:
(1). $x_1^* = (1,0)$, $x_2^* = (1,0)$, $x_3^* = (1,0)$, $f(x^*) = 0$.
(2). $x_1^* = (0.5,0.5)$, $x_2^* = (0.5455,0.4545)$, $x_3^* = (0,1)$, $f(x^*) = 0$.
(3). $x_1^* = (0.8,0.2)$, $x_2^* = (1,0)$, $x_3^* = (0.5,0.5)$, $f(x^*) = 0$.

Example 5.2. [2] Assume that the strategy sets of player 1, 2, 3,*and*4 respectively are $A = (a_1,a_2)$, $B = (b_1,b_2)$, $C = (c_1,c_2)$, and $D = (d_1,d_2)$, and mixed strategies are $\{x_{ik_i} | \, i = 1,\cdots,4; k_i = 1,2\}$. Their strategy profiles and corresponding payoffs are shown in Table 2:

TABLE 2. The four-player game $\Gamma_2(A,B,C,D,f_1,f_2,f_3,f_4)$.

Strategy profiles	Payoff vector	Strategy profiles	Payoff vector
(a_1,b_1,c_1,d_1)	(2,0,1,1)	(a_2,b_1,c_1,d_1)	(-1,3,1,1)
(a_1,b_1,c_1,d_2)	(-1,2,1,1)	(a_2,b_1,c_1,d_2)	(0,2,2,2)
(a_1,b_1,c_2,d_1)	(-1,1,2,2)	(a_2,b_1,c_2,d_1)	(1,2,3,1)
(a_1,b_1,c_2,d_2)	(-4,3,1,1)	(a_2,b_1,c_2,d_2)	(-3,-4,1,1)
(a_1,b_2,c_1,d_1)	(3,-1,3,3)	(a_2,b_2,c_1,d_1)	(-2,-1,0,0)
(a_1,b_2,c_1,d_2)	(0,1,2,2)	(a_2,b_2,c_1,d_2)	(0,3,2,2)
(a_1,b_2,c_2,d_1)	(2,1,-3,-3)	(a_2,b_2,c_2,d_1)	(4,-1,0,0)
(a_1,b_2,c_2,d_2)	(2,0,-1,-1)	(a_2,b_2,c_2,d_2)	(0,3,40,40)

According to the data in Table 2 and the solution steps of Example 5.1. By many iterations, solutions to this problem are not unique and satisfied:

$$f(x^*) = 0,\ x_1^* = (0,1),\ x_2^* = (1,0),\ x_3^* = (0,1),\ x_4^* = (t,1-t),\ \forall t \in [0.5,1].$$

Then, there is a special single point pure strategy Nash equilibrium:

$$f(x^*) = 0,\ x_1^* = (0,1),\ x_2^* = (1,0),\ x_3^* = (0,1),\ x_4^* = (1,0).$$

Example 5.3. Assume that the strategy sets of player $1,2,3,4$, and 5, respectively, are $A = (a_1,a_2)$, $B = (b_1,b_2)$, $C = (c_1,c_2)$, $D = (d_1,d_2)$, and $E = (e_1,e_2)$, and mixed strategies are

$\{x_{ik_i}|\ i = 1,\cdots,5; k_i = 1,2\}$. Their strategy profiles and corresponding payoffs are shown in Table 3:

TABLE 3. The five-player game $\Gamma_3(A,B,C,D,E,f_1,f_2,f_3,f_4,f_5)$.

Strategy profiles	Payoff vector	Strategy profiles	Payoff vector
(a_1,b_1,c_1,d_1,e_1)	(1,0,0,0,1)	(a_2,b_1,c_1,d_1,e_1)	(3,0,1,1,1)
(a_1,b_1,c_1,d_1,e_2)	(-2,1,0,0,2)	(a_2,b_1,c_1,d_1,e_2)	(-1,2,1,2,2)
(a_1,b_1,c_1,d_2,e_1)	(-2,0,1,1,1)	(a_2,b_1,c_1,d_2,e_1)	(-1,1,2,2,1)
(a_1,b_1,c_1,d_2,e_2)	(-5,2,0,0,2)	(a_2,b_1,c_1,d_2,e_2)	(-3,3,1,2,2)
(a_1,b_1,c_2,d_1,e_1)	(2,-2,2,2,1)	(a_2,b_1,c_2,d_1,e_1)	(3,0,4,3,1)
(a_1,b_1,c_2,d_1,e_2)	(-1,0,1,1,3)	(a_2,b_1,c_2,d_1,e_2)	(0,1,3,4,3)
(a_1,b_1,c_2,d_2,e_1)	(1,0,-4,-4,1)	(a_2,b_1,c_2,d_2,e_1)	(2,2,-3,-3,1)
(a_1,b_1,c_2,d_2,e_2)	(1,-1,-2,-2,0)	(a_2,b_1,c_2,d_2,e_2)	(2,0,0,-2,0)
(a_1,b_2,c_1,d_1,e_1)	(-2,2,0,0,1)	(a_2,b_2,c_1,d_1,e_1)	(0,3,1,1,1)
(a_1,b_2,c_1,d_1,e_2)	(-1,1,1,1,1)	(a_2,b_2,c_1,d_1,e_2)	(0,3,3,2,1)
(a_1,b_2,c_1,d_2,e_1)	(0,1,2,0,2)	(a_2,b_2,c_1,d_2,e_1)	(1,2,4,2,2)
(a_1,b_2,c_1,d_2,e_2)	(-4,-5,0,0,1)	(a_2,b_2,c_1,d_2,e_2)	(-3,-3,2,1,1)
(a_1,b_2,c_2,d_1,e_1)	(-3,-2,-1,-1,1)	(a_2,b_2,c_2,d_1,e_1)	(-2,-1,1,0,1)
(a_1,b_2,c_2,d_1,e_2)	(-1,2,1,1,2)	(a_2,b_2,c_2,d_1,e_2)	(0,4,2,3,2)
(a_1,b_2,c_2,d_2,e_1)	(3,-2,-1,-1,2)	(a_2,b_2,c_2,d_2,e_1)	(4,-1,1,1,2)
(a_1,b_2,c_2,d_2,e_2)	(-1,2,3,3,1)	(a_2,b_2,c_2,d_2,e_2)	(1,4,4,4,1)

According to the data in Table 3 and the solution steps of Example 5.1. Solutions to this problem are also not unique. It consists of two sets and a pure strategy Nash equilibrium such as:
(1). $f(x^*) = 0, x_1^* = (0,1), x_2^* = (0,1), x_3^* = (1,0), x_4^* = (1,0), x_5^* = (u,1-u), \forall u \in [0,0.5]$.
(2). $f(x^*) = 0, x_1^* = (0,1), x_2^* = (1,0), x_3^* = (1,0), x_4^* = (v,1-v), x_5^* = (0,1), \forall v \in [0,0.3]$.
(3). $f(x^*) = 0, x_1^* = (0,1), x_2^* = (0,1), x_3^* = (1,0), x_4^* = (0,1), x_5^* = (1,0)$.

For solutions (1) and (2), when $u = 0$ and $v = 0$, there are two pure strategy Nash equilibria, namely:

$$x_1^* = (0,1), x_2^* = (0,1), x_3^* = (1,0), x_4^* = (1,0), x_5^* = (0,1),$$
$$x_1^* = (0,1), x_2^* = (1,0), x_3^* = (1,0), x_4^* = (0,1), x_5^* = (0,1).$$

By calculating the above examples, it is found that the simpler equivalent model adopted in this paper can also solve the multi-person game well. This equivalent form is that finding the Nash equilibria is equivalent to solving the optimal solutions of a constrained optimization problem. So, it can be easily implemented by using the IDE algorithm proposed in this paper.

6. Conclusions

In this paper, we study the nonzero sum multi-person game and propose a new intelligent algorithm to solve the game, which fills the gap in the intelligent algorithm for solving the multi-person game. Because there are many players and complex strategy choices in the multi-person game, it is necessary to find a more suitable mathematical model to describe the game. First, a theorem shows that finding the Nash equilibria is equivalent to solving a constrained optimization problem. Comparing this equivalent transformation with the already existing references, it is found that the equivalent form applied in this paper is simpler and easier to calculate. Second, in order to solve this optimization problem better, the IDE algorithm is proposed. Under the condition that the computational complexity of the algorithm is unchanged, the multi-strategy mutation model is carried out on the differential operator of the algorithm, and the random walk mechanism is introduced into the crossover operator to improve the performance of the algorithm. Then the global convergence of the IDE algorithm is proved by the infinite product, which provides theoretical support for solving the following global optimization problems. Finally, numerical experiments verify the effectiveness of the IDE in solving multi-person games. In real life, people not only consider one objective when making decisions, but often consider multiple objectives. So, it is of interest to study the issue of multi-player game under the multi-objective, and develop other algorithms and compare them with the IDE algorithm in the future.

Acknowledgements
This study was supported by the National Natural Science Foundation of China (Grant No. [71961003], [12061020]), the Qian Jiaohe YJSCXJH ([2019]029), and the Scientific Research Foundation of Guizhou University ([2019]49).

References

[1] C. Arumugasamy, G. Sengodan, Linear complementarity problems and bi-linear games, Appl. Math. 65 (2020) 665-675.
[2] S. Batbileg, N. Tungalag, A. Anikin, A. Gornov, E. Finkelstein, A global optimization algorithm for solving a four-person game, Optim. Lett. 13 (2017) 587-596.
[3] U. Boryczka, P. Juszczuk, Differential evolution as a new method of computing Nash equilibria, Transactions on Computational Collective Intelligence IX, pp. 192-216, Springer, Berlin, Heidelberg, 2013.
[4] C.Y. Chen, L.P. Zhang, Finding Nash equilibrium for a class of multi-person noncooperative games via solving tensor complementarity problem, Appl. Numer. Math. 145 (2019) 458-468.
[5] C.Z. Chen, F.L. Jin, X.Y. Zhu, G.Z. Ouyang, Mathematics Analysis, Higher Education Press, Beijing, 2000.
[6] R. Enkhbat, S. Batbileg, A. Anikin, N. Tungalag, A. Gornov, A note on solving 5-person game, Adv. Modeling Optim. 19 (2017) 227-232.
[7] R. Enkhbat, S. Batbileg, N. Tungalag, A. Anikin, A. Gornov, A computational method for solving n-person game, The Bulletin of Irkutsk State University, Series "Mathematics". 20 (2017) 109-121.
[8] R. Enkhbat, N. Tungalag, A. Gornov, A. Anikin, The curvilinear search algorithm for solving three-person game, pp. 574-583, Proceedings of DOOR 2016, CEUR-WS, 2016.
[9] S. Ghosh, S. Das, A. V. Vasilakos, K. Suresh, On convergence of differential evolution over a class of continuous functions with unique global optimum, IEEE Tran. Sys. Man Cybernetics Part B Cybernetics 42 (2012) 107-124.
[10] S. Govindan, R. Wilson, A global newton method to compute Nash equilibria, J. Economic Theory 110 (2003) 65-86.
[11] Y.C. He, X.Z. Wang, K.Q. Liu, Y.Q. Wang, Convergent analysis and algorithmic improvement of differential evolution, J. Software 21 (2010) 875-885.

[12] Z.B. Hu, S.W. Xiong, Q.H. Su, X.W. Zhang, Sufficient conditions for global convergence of differential evolution algorithm, J. Appl. Math. 2013 (2013) 1-14.
[13] Z.H. Huang, L.Q. Qi, Formulating an n-person noncooperative game as a tensor complementarity problem, Comput. Optim. Appl. 66 (2017) 557-576.
[14] L.G. Hua, Y. Wang, Application of Number Theory in Modern Analysis, Science Press, Beijing, 1978.
[15] W.S. Jia, S.W. Xiang, J.F. Yang, W.S. Hu, Solving Nash equilibrium for n-person non-cooperative game based on immune particle swarm algorithm, Appl. Res. Comput. 29 (2012) 28-31.
[16] I.V. Konnov, Equilibrium formulations of relative optimization problems, Math. Meth. Oper. Res. 90 (2019) 137-152.
[17] G.M. Korres, A. Kokkinou, Political decision in a game theory approach, European Socio-Economic Integration 28 (2012) 51-61.
[18] T. Kunieda, K. Nishimura, Finance and economic growth in a dynamic game, Dyn. Games Appl. 8 (2018) 588-600.
[19] C.E. Lemke, J.T.J. Howson, Equilibrium points of bimatrix games, J. Soc. Ind. Appl. Math. 12 (1964) 413-423.
[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 Math. 6 (2021) 1309-1323.
[21] C. Margaria, Learning and payoff externalities in an investment game, Games Econom. Behav. 119 (2020) 234-250.
[22] J.F. Nash, Non-cooperative games, Ann. Math. 54 (1951) 286-295.
[23] J.F. Nash, Equilibrium points in n-person games, Proc. Nat. Acad. Sci. 36 (1950) 48-49.
[24] N. Noman, H. Iba, Enhancing differential evolution performance with local search for high dimensional function optimization, Proceedings of the 7th annual conference on Genetic and evolutionary computation, pp. 967-974, 2005.
[25] N.G. Pavlidis, K.E. Parsopoulos, M.N. Vrahatis, Computing Nash equilibria through computational intelligence methods, J. Comput. Appl. Math. 175 (2005) 113-136.
[26] B. Panicucci, M. Pappalardo, M. Passacantando, On solving generalized Nash equilibrium problems via optimization, Optim. Lett. 3 (2009) 419-435.
[27] C.F. Sun, Differential Evolution Algorithm and Its Application on the Optimal Scheduling of Electrical Power System, Ph.D thesis, Huazhong University of Science and Technology, Wuhan, 2010.
[28] R. Storn, K. Price, Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces, J. Global Optim. 23 (1995).
[29] Y.Q. Wu, Y.G. Wu, X.L. Liu, Couple-based particle swarm optimization for short-term hydrothermal scheduling, Appl. Soft Comput. 74 (2019) 440-450.
[30] Y.l. Yang, S.W. Xiang, S.Y. Xia, W.S. Jia, Solving Nash equilibrium of non-cooperative game based on fireworks algorithm, Comput. Appl. Softw. 35 (2018) 215-218.
[31] J. Yu, Selection of Game Theory, Science Press, Beijing, 2014.
[32] Z.H. Zhan, J. Zhang, Enhance differential evolution with random walk, Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation, 1513-1514, 2012.
[33] J.Q. Zhang, A.C. Sanderson, JADE: Adaptive differential evolution with optional external archive, IEEE Trans. Evolutionary Comput. 13 (2009) 945-958.
[34] J. Zhang, B. Qu, N. Xiu, Some projection-like methods for the generalized Nash equilibria, Comput. Optim. Appl. 45 (2008) 89-109.