

## Research Article

# A New Algorithm to Solve the Generalized Nash Equilibrium Problem

Luping Liu  and Wensheng Jia 

*College of Mathematics and Statistics, Guizhou University, Guiyang 550025, Guizhou, China*

Correspondence should be addressed to Wensheng Jia; [jws0505@163.com](mailto:jws0505@163.com)

Received 12 March 2020; Accepted 11 June 2020; Published 2 October 2020

Academic Editor: Akemi Gálvez

Copyright © 2020 Luping Liu and Wensheng Jia. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We try a new algorithm to solve the generalized Nash equilibrium problem (GNEP) in the paper. First, the GNEP is turned into the nonlinear complementarity problem by using the Karush–Kuhn–Tucker (KKT) condition. Then, the nonlinear complementarity problem is converted into the nonlinear equation problem by using the complementarity function method. For the nonlinear equation equilibrium problem, we design a coevolutionary immune quantum particle swarm optimization algorithm (CIQPSO) by involving the immune memory function and the antibody density inhibition mechanism into the quantum particle swarm optimization algorithm. Therefore, this algorithm has not only the properties of the immune particle swarm optimization algorithm, but also improves the abilities of iterative optimization and convergence speed. With the probability density selection and quantum uncertainty principle, the convergence of the CIQPSO algorithm is analyzed. Finally, some numerical experiment results indicate that the CIQPSO algorithm is superior to the immune particle swarm algorithm, the Newton method for normalized equilibrium, or the quasivariational inequalities penalty method. Furthermore, this algorithm also has faster convergence and better off-line performance.

## 1. Introduction

In 1952, Debreu [1] came up with the generalized Nash equilibrium problem (GNEP), in which each player's feasible strategy set depends on the strategies of other players. In the GNEP, no player can obtain more benefits through a unilateral change in his strategy. At present, the GNEP has attracted growing concerns, and the GNEP was widely used in many fields such as economics, pricing currency option, evolutionary biology, computer science, artificial intelligence, and autonomous cars [2, 3]. However, Debreu and Arrow had not given an efficient algorithm for the solution of the GNEP, and Roughgarden [4] indicated that the solution of the Nash equilibrium is NP-hard problem. The solution of the GNEP has attracted much attention: Facchinei and Kanzow [5] focused on penalty methods to solve the GNEP; Kunjan and Twinkle [6] showed a modified homotopy algorithm; Aussel [7] developed some characterizations of the GNEP as a quasivariational inequality;

Migot and Cojocaru [8] presented variational inequality methods to track the solution set of the GNEP. Using the regularized indicator Nikaido–Isoda-type function, Lalitha and Dhingra [9] presented two constraint optimization formulas for the GNEP; Dreves [10] considered a linear GNEP and a best-response approach for equilibrium selection in two-player GNEP; Izmailov and Solodov [11] analyzed error bounds and Newton-type methods for the GNEP. The abovementioned approaches for solving the GNEP have the common feature that they convert the GNEP into variational inequalities, quasivariational inequalities, or nonlinear equations under certain assumptions, and then they use the appropriate optimization algorithms to solve them. Inspired by the above work, this paper considers a new algorithm for solving the GNEP. First, the GNEP is transformed into the nonlinear complementarity problem by Karush–Kuhn–Tucker (KKT) condition. Second, the nonlinear complementarity problem is transformed into the nonlinear equation system problem by the complementarity

function method. Finally, we use the CIQPSO algorithm to solve the GNEP by constructing an appropriate fitness function and analyze and compare some numerical examples to show that the algorithm is effective.

In recent years, swarm intelligence algorithm has shown the certain potential possibility for solving NP-hard optimization problems [12, 13], and the swarm intelligence algorithm has many applied in power system management [14], petroleum industry [15], wireless sensor networks [16], and cloud computing multitask scheduling optimization [17] and so forth. It is well known that the GNEP usually has an infinite number of solutions, and the Nash equilibrium set is manifold if the players may share constraints in [18]. Kanzow and Steck [19] proposed an augmented Lagrangian algorithm to solve the GNEP, which they take into account the particular structure of GNEPs with suitable constraint qualifications. Pang and Fukushima [20] applied an augmented Lagrangian-type approach to quasivariational inequalities (QVIs). In [21], an improved version can be found for QVIs. With the development of the computation for Nash equilibrium, many scholars try to use the swarm intelligence algorithm from biological evolutionary process to solve Nash equilibrium problems, such as immune particle swarm algorithm [22] for solving Nash equilibrium of the GNEP, ant colony algorithm [23], and evolutionary algorithm [24] to attain with the Nash equilibrium of dynamic games and evolutionary games [25]. As mentioned above, it has become a new approach to realize Nash equilibrium by using the biological evolution and biological behavior law of swarm intelligence algorithm. Inspired by research works mentioned above, our paper mainly studies swarm intelligence algorithms to solve the GNEP. We presented a co-evolutionary immune quantum particle swarm optimization (CIQPSO) algorithm by involving the immune memory function and the antibody density inhibition mechanism into the quantum particle swarm optimization algorithm and proved the convergence of the CIQPSO algorithm. The CIQPSO algorithm is able to improve global optimum capability. Moreover, the convergence of the CIQPSO algorithm is analyzed by introducing the probability density selection function. Meanwhile, some numerical examples show that the algorithm is effective. This paper is organized as follows. In Section 2, we introduce the model and assumptions of the GNEP. In Section 3, we consider the GNEP is transformed into the nonlinear equation problem through the Karush–Kuhn–Tucker (KKT) condition and the complementarity function method, respectively. In Section 4, we study the design of the CIQPSO algorithm and the convergence of the CIQPSO algorithm. In Section 5, we try to solve the GNEP by using the CIQPSO algorithm. Some numerical examples show that the algorithm is effective.

## 2. Model and Assumptions

In the section, we first introduce the model and some assumptions of the GNEP.

Let  $N = \{1, 2, \dots, n\}$  denote the index set of all players,  $x = (x^1, \dots, x^n)^T \in \mathbb{R}^n$  ( $n = \sum_{v=1}^n n_v$ ) denote the decision variables of all players, and  $x^v = (x^{v,1}, x^{v,2}, \dots,$

$x^{v,n_v}) \in \mathbb{R}^{n_v}$  ( $v \in N$ ) indicate the control variable of each player  $v$ .  $x^{-v}$  denotes all other players decision variables apart from the player  $v$ , and we write that  $x = (x^v, x^{-v})$ . Assume that the  $v$ th player's strategy set is  $x^v \in X^v(x^{-v})$ , which relies on the strategy of all other players.  $X^v(x^{-v})$  is also called the  $v$ th player's feasible set, which is clearly defined by equality constraints or inequality constraints. Each player  $v$  has a *payoff function* or *utility function*  $\phi^v(x^v, x^{-v}): \mathbb{R}^n \rightarrow \mathbb{R}$ , where  $\phi^v$  is related to both the player's decision variable  $x^v$  and other players' decision variables  $x^{-v}$ . In general, for each player  $v \in N$  ( $\forall v = 1, 2, \dots, n$ ), some constraint functions are given by

$$\begin{aligned} X^v(x^{-v}) = \{ & x^v \mid H^v(x^v) \leq 0, \\ & S^v(x^v, x^{-v}) \leq 0\}, \end{aligned} \quad (1)$$

where  $H^v(x^v)$  is the self-constraints of all players, which only relies on the player's decision variable.  $S^v(x^v, x^{-v})$  indicates all players' common constraints, which relies on all players' decision variables. The dimensions of  $h^v$  and  $g^v$  are denoted as  $l_v$  with  $l = l_1 + \dots + l_n$  and  $m_v$  with  $m = m_1 + \dots + m_n$ , respectively.

If  $x^* = (x^{*,1}, \dots, x^{*,n}) \in X^1(x^{-1}) \times \dots \times X^n(x^{-n})$  is the Nash equilibrium for the GNEP, then for any  $v \in N$ ,  $x^{*,v} \in X^v(x^{*,-v})$ , the equation is satisfied with the following optimization problem:

$$\phi^v(x^{*,v}, x^{*,-v}) = \min_{x^v} \phi^v(x^v, x^{*,-v}), \text{ s.t. } x^v \in X^v(x^{*,-v}). \quad (2)$$

From now on, we assume that the following conditions are satisfied.

*Assumption 1* (smoothness assumption and convexity assumption)

- The  $C^1$ -functions  $\phi^v: \mathbb{R}^{n_v} \rightarrow \mathbb{R}$ ,  $H^v: \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{l_v}$ , and  $S^v: \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{m_v}$  are twice differential with local Lipschitz continuous second-order derivable
- For each player  $v \in N$ , given any  $x^{-v}$ , the objective function  $\phi^v(x^v, x^{-v})$  is convex with respect to  $x^v$ , and constraint functions  $H^v: \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{l_v}$  and  $S^v: \mathbb{R}^{n_v} \rightarrow \mathbb{R}^{m_v}$  are convex

## 3. The GNEP, Nonlinear Complementarity Problem, and Nonlinear Equation Problem

*3.1. The GNEP Is Converted into Nonlinear Complementarity Problem.* First the GNEP is transformed to nonlinear complementarity problem by using the Karush–Kuhn–Tucker (KKT) condition. Therefore, if  $x^*$  satisfies the appropriate constraint qualification, then there are multiplier vectors  $\lambda^{*,v} \in \mathbb{R}^{l_v}$  and  $\mu^{*,v} \in \mathbb{R}^{m_v}$  to make  $(x^{*,v}, \lambda^{*,v}, \mu^{*,v})$  meet the following KKT system [26]:

$$\begin{aligned} \nabla_{x^v} L_v(x^v, x^{*,-v}, \lambda^v, \mu^v) &= 0, \\ 0 \leq \lambda^v \perp -H^v(x^v) &\geq 0, \\ 0 \leq \mu^v \perp -S^v(x^v, x^{*,-v}) &\geq 0, \end{aligned} \quad (3)$$

where  $L_v: \mathbb{R}^n \times \mathbb{R}^{l_v} \times \mathbb{R}^{m_v} \rightarrow \mathbb{R}$  represents the Lagrangian equation of system (3), which is equivalent to

$$L_v(x, \lambda^v, \mu^v) = \phi^v(x) + \lambda^v H^v(x^v)^T + \mu^v S^v(x)^T, \quad (4)$$

where T means transpose. Thus, we know that any solution in system (3) is subject to the solution of equation (2). For the KKT system of all players, if  $x^*$  is the GNEP's solution, then there are multiplier vectors  $\lambda^* \in \mathbb{R}^{l_v}$  and  $\mu^* \in \mathbb{R}^{m_v}$  to make  $(x^*, \lambda^*, \mu^*)$  meet the following system:

$$\begin{aligned} L(x, \lambda, \mu) &= 0, \\ 0 \leq \lambda \perp -H(x) &\geq 0, \\ 0 \leq \mu \perp -\mathbb{S}(x) &\geq 0, \end{aligned} \quad (5)$$

where

$$\begin{aligned} \mu &= \begin{bmatrix} \mu^1 \\ \vdots \\ \mu^n \end{bmatrix}, \\ \lambda &= \begin{bmatrix} \lambda^1 \\ \vdots \\ \lambda^n \end{bmatrix}, \\ H(x) &= \begin{bmatrix} H^1(x^1) \\ \vdots \\ H^n(x^n) \end{bmatrix}, \\ \mathbb{S}(x) &= \begin{bmatrix} S^1(x) \\ \vdots \\ S^n(x) \end{bmatrix}, \\ L(x, \lambda, \mu) &= \begin{bmatrix} \nabla_{x^1} L_1(x, \lambda^1, \mu^1) \\ \vdots \\ \nabla_{x^n} L_n(x, \lambda^n, \mu^n) \end{bmatrix}. \end{aligned} \quad (6)$$

Under the above assumptions, system (5) can be considered as the GNEP's first-order necessary condition. If the GNEP further satisfies Assumption 1 (b), then for any  $x^{-v}$ , formula (2) is transformed into a convex optimization problem, and system (5) becomes the GNEP's sufficient condition. Let the GNEP satisfy Assumption 1, if  $(x^*, \lambda^*, \mu^*)$  meets system (5), then  $x^*$  is the GNEP's Nash equilibrium.

For  $\mathbb{F}: \mathbb{R}^n \rightarrow \mathbb{R}^n$ ,  $\mathbb{F}(x) = \text{vec}\{\nabla_{x^v} \phi^v(x)\}_{v=1}^n$ , system (5) can be written as follows:

$$\begin{aligned} \mathbb{F}(x) + \lambda^v \sum_{v=1}^n \nabla_{x^v} H^v(x^v) + \mu^v \sum_{v=1}^n \nabla_x S^v(x^v, x^{-v}) &= 0, \\ 0 \leq \lambda^v \perp -H^v(x^v) &\geq 0, \\ 0 \leq \mu^v \perp -\mathbb{S}(x) &\geq 0. \end{aligned} \quad (7)$$

System (7) is a nonlinear complementarity problem, so the GNEP is converted into a nonlinear complementarity

problem by using the KKT condition of the convex optimization problem.

**3.2. The Nonlinear Complementarity Problem Is Converted into the Nonlinear Equation Problem.** The key point in this paper is to transform system (7) into nonlinear equations by using the complement function method. If the formula

$$\psi(A, B) = 0 \iff A \geq 0, B \geq 0, A \cdot B = 0, \quad (8)$$

holds, then the function  $\psi: \mathbb{R}^2 \rightarrow \mathbb{R}$  is called a complementarity function. This paper uses the Fischer–Burmeister (FB) function, that is,

$$\psi_{\text{FB}}(A, B) = A + B - \sqrt{A^2 + B^2}. \quad (9)$$

Then, system (7) is equivalent to the following formula:

$$\Psi_{\text{FB}}(x, \lambda, \mu) = \begin{bmatrix} L(x, \lambda, \mu) \\ \psi_{\text{FB}}(-H_1^1(x^1), \lambda_1^1) \\ \vdots \\ \psi_{\text{FB}}(-H_v^i(x^i), \lambda_v^i) \\ \vdots \\ \psi_{\text{FB}}(-H_n^n(x^n), \lambda_n^n) \\ \psi_{\text{FB}}(-S^1(x), \mu^1) \\ \vdots \\ \psi_{\text{FB}}(-S^{m_n}(x), \mu^{m_n}) \end{bmatrix}. \quad (10)$$

Accordingly, system (7) is equivalent to  $\Psi_{\text{FB}}(x, \lambda, \mu) = 0$ . Furthermore, the GNEP is transformed into the nonlinear equation problem by complementarity function. Finally, we design the CIQPSO algorithm to solve the GNEP by constructing a suitable fitness function in this paper.

In the CIQPSO algorithm, the fitness function is designed as follows:

$$F(x) = \|\Psi_{\text{FB}}(x, \lambda, \mu)\|^2. \quad (11)$$

Obviously, we can get  $F(x) = 0$  from  $\Psi_{\text{FB}}(x, \lambda, \mu) = 0$ , which is equivalent to solve the minimum value of  $\min_{x \in X^v(x^{-v})} F(x)$  in the feasible constraint region, that is, the smaller the particle fitness function value, the more particles can adapt to the environment e.

## 4. The Design of the CIQPSO Algorithm

**4.1. Quantum Particle Swarm Optimization (QPSO) Algorithm.** Kennedy and Eberhart [27] first proposed the idea of the particle swarm optimization (PSO) algorithm in 1995. The PSO algorithm has the properties of iterative optimization and biological intelligence. However, Bergh [28] proved that the PSO algorithm does not guarantee to global optimal or local optimal. In 2004, Sun et al. [29] proposed the quantum-behaved particle swarm optimization (QPSO) algorithm from the perspective of quantum mechanics. In the QPSO algorithm with the population size  $M$  and the space dimension  $N$ , the current position vector of particle  $i$  is  $X_i = (X_i^1, X_i^2, \dots, X_i^N)^T$ . Each particle's position

is affected by its local best-known position  $P_i(t) = [P_i^1(t), P_i^2(t), \dots, P_i^N(t)]^T$  and it also guided toward the global best-known position  $P_g(t) = [P_g^1(t), P_g^2(t), \dots, P_g^N(t)]^T$  in the feasible strategy search space. At the  $t$  step, each particle's individual best position  $P_i$  and all particles' global best position  $P_g$  are updated using the following formulas:

$$P_i(t+1) = \begin{cases} X_i(t+1), & \text{if } F[X_i(t+1)] \leq F[P_i(t)], \\ P_i(t), & \text{if } F[X_i(t+1)] \geq F[P_i(t)], \end{cases}$$

$$P_g(t+1) = \arg \min_{P_i} F(P_i(t+1)), \quad 1 \leq i \leq M. \quad (12)$$

The particle's trajectory analyses indicated that each particle  $i$  must converge to its local attractor  $\mathcal{B}_i(t) = [\mathcal{B}_i^1(t), \mathcal{B}_i^2(t), \dots, \mathcal{B}_i^N(t)]^T$ , which is randomly generated between  $P_i(t)$  and  $P_g(t)$ . The coordinate can be calculated by the following equation:

$$\mathcal{B}_i^j(t) = \varphi_i^j(t) \cdot P_i^j(t) + (1 - \varphi_i^j(t)) \cdot P_g^j(t), \quad j = 1, 2, \dots, N, \quad (13)$$

where  $\varphi_i^j(t) \sim U(0, 1)$ , and the function  $U(0, 1)$  denotes random numbers evenly distributed between 0 and 1.

$\text{mbest}(t)$  denotes the average best position and is written by  $E(t)$ , that is,

$$E(t) = [E^1(t), E^2(t), \dots, E^N(t)]^T = \frac{1}{M} \sum_{i=1}^M P_i(t) = \left[ \frac{1}{M} \sum_{i=1}^M P_i^1(t), \frac{1}{M} \sum_{i=1}^M P_i^2(t), \dots, \frac{1}{M} \sum_{i=1}^M P_i^N(t) \right]^T. \quad (14)$$

In the quantum time-space model, the quantum state of a particle is represented by a wave function. From the view of classical dynamics, when particles move in the attractive potential field whose center is point  $\mathcal{B}_i^j(t)$ , they must be in a bound state to avoid explosion and ensure convergence. In quantum bound state, the particle can appear anywhere with a certain probability as long as the probability density of the particle tends to be 0 when the particle distance from the center tends to be  $\infty$ . In quantum space, particle's position is updated with the Monte Carlo method, and the updated formula is as follows:

$$X_i^j(t+1) = \mathcal{B}_i^j + \text{rand}(t) \cdot \gamma \cdot |E^j(t) - X_i^j(t)| \cdot \ln\left(\frac{1}{\mu_i^j(t)}\right), \quad (15)$$

where  $\mu_i^j(t) \sim U(0, 1)$ .  $\gamma$  denotes contraction-expansion coefficient.  $\gamma$  can adopt fixed value or linear reduction (see [30]). In this paper, the updated formula of  $\gamma$  is as follows:

$$\gamma = \left( \gamma_1 - \gamma_2 \times \frac{K_{\max} - t}{K_{\max}} \right) + \gamma_2, \quad (16)$$

where  $\gamma_1$  and  $\gamma_2$  are the maximum and the minimum contraction-expansion coefficients, respectively.  $K_{\max}$  and  $t$  are the maximum and the current iteration number, respectively.

**4.2. The Design of the CIQPSO Algorithm.** The CIQPSO algorithm is proposed by combining the QPSO algorithm and the immune PSO (IPSO) algorithm. The CIQPSO algorithm is to introduce immune memory function and the antibody density inhibition mechanism into the QPSO algorithm. The IPSO has the properties of antigen recognition, immune memory function, and the antibody density inhibition mechanism. Immune memory cells are obtained by preserving  $P_g$  of each generation, and the similar memory cells are eliminated. The CIQPSO algorithm shares information

and is coevolutionary among all particles. The objective function and constraints are regarded as antigens, and the GNEP's solution is considered as an antibody (also called the particle). The particle continuously adjusts its current position to receive the global optimal solution by the particle's current optimal position information and all particles' optimal position information. The QPSO algorithm is easier to trap in the local optimal solution when particles are excessively centralized and avoid losing those particles that are of poor fitness but have a preferable evolutionary trend. Thus, we introduce the antibody's probability density selection formula to keep the diversity of the particle in this paper. Antibodies  $G + M$  consist of nonempty set  $X$ , and the distance of the antibody  $F(x_i)$  is calculated by the following equation:

$$d(x_i) = \sum_{j=1}^{G+M} |F(x_i) - F(x_j)|. \quad (17)$$

In [31], we can define the density of particle  $i$  as follows:

$$D(x_i) = \frac{1}{\sum_{j=1}^{G+M} |F(x_i) - F(x_j)|}, \quad i = 1, 2, \dots, G + M. \quad (18)$$

Therefore, we can obtain the probability density selection function from

$$\mathcal{P}(x_i) = \frac{1/D(x_i)}{\sum_{i=1}^{G+M} 1/D(x_i)} = \frac{\sum_{j=1}^{G+M} |F(x_i) - F(x_j)|}{\sum_{i=1}^{G+M} \sum_{j=1}^{G+M} |F(x_i) - F(x_j)|}, \quad (19)$$

where  $x_i$  and  $F(x_i)$  are the optimum position and fitness function value of the particle  $i$ , respectively. Adding new member  $G$  principally maintains the dynamical balance of swarm and plays the role to adjust swarm diversity. In particular, when the evolution swarm appears the bad diversity and the weak global search ability, the random

generation of the self-antibody allows the swarm shift to the region with better solution.

4.3. *Arithmetic Flow of the CIQPSO Algorithm.* The design of the CIQPSO algorithm implementation steps is as follows:

Step 1: parameter initialization.  $K_{\max}$  is the maximum iteration number, precision is  $\varepsilon$ , population size is  $M$ , and contraction-expansion coefficient is  $\gamma$ ; randomly generate particles' initial position  $x_i (i = 1, 2, \dots, M)$ .

Step 2: compute the fitness value of each particle, if  $F(P_i) > F(x_i)$ , then  $P_i = x_i$ , otherwise  $x_i = P_i$ . Each particle's best current position is  $P_i$  and all particles' best current position is  $P_g$ .

Step 3: the average best position mbest is calculated by formula (14).

Step 4: use formula (15) to update the position of particle, and  $P_g$  of each iteration is taken as memory cell particle to be placed into a memory library.

Step 5: compute the fitness value of the updated particle, if  $F(P_i) > F(P_g)$ , then  $P_i = P_g$ , otherwise  $P_g = P_i$ . If stop conditions are satisfied, go to Step 10, and otherwise, go to Step 6.

Step 6: randomly generate particles  $G$  and particles' initial position  $x_i (i = 1, 2, \dots, G)$ .

Step 7: we choose particles  $M$  from particles  $G + M$  by the probability density selection formula (19).

Step 8: we select particles from the memory library to replace particles of poor fitness in the population and the immune system generates a new generation  $p_1$  to perform the next iteration.

Step 9: use formula (15) to calculate new positions of particles, and return to Step 2.

Step 10: stopping condition. Whether the maximum iteration number or precision meets termination condition? If yes, stop evolution; otherwise return to Step 2.

The CIQPSO algorithm implementation process is as shown in Figure 1.

4.4. *The Performance Evaluation of the CIQPSO Algorithm.* The CIQPSO algorithm is a new type of bioevolving swarm intelligence algorithm, which is relevant to the evolution rules of genetic algorithm (GA), for instance, natural selection and survival of the fittest. Therefore, we use the quantitative analysis method [32] to test the convergence of the algorithm by using the off-line performance of the algorithm.

*Definition 1.* Off-line performance  $s_e^*$ :  $X \rightarrow \mathbb{R}$  is defined as follows:

$$s_e^*(x) = \frac{1}{K} \sum_{t=1}^T F_e^*(x), \quad (20)$$

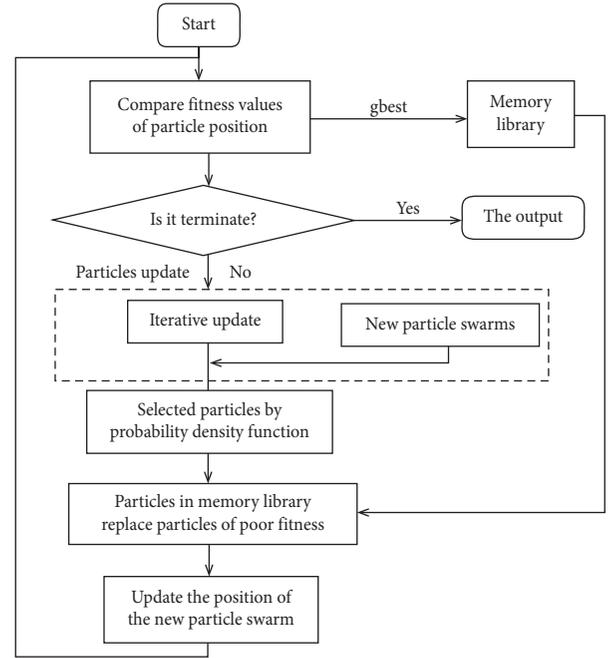


FIGURE 1: The flow diagram of the CIQPSO algorithm.

where  $F_e^*(x) = \min\{F_e(1), F_e(2), \dots, F_e(N)\}$ . In the algorithm iteration process, off-line performance is the cumulative average of the best fitness values of each generation.

#### 4.5. The Convergence of the CIQPSO Algorithm

**Theorem 1** (see [26]). *In  $N$ -dimensional space, the necessary and sufficient condition for particle  $i$ 's position  $X_i(t) = [X_i^1(t), X_i^2(t), \dots, X_i^N(t)]^T (i = 1, 2, \dots, M)$  to converge to its attractor  $\mathcal{B}_i(t) = [\mathcal{B}_i^1(t), \mathcal{B}_i^2(t), \dots, \mathcal{B}_i^N(t)]^T$  with probability is that every 1-dimensional coordinate  $X_i^j(t) (j = 1, 2, \dots, N)$  converges to  $\mathcal{B}_i^j$  with probability.*

*Proof.* Necessity: since  $X_i(t) \xrightarrow{P} \mathcal{B}_i(t), \forall \varepsilon > 0$ ,

$$\lim_{t \rightarrow \infty} \mathcal{P}(|X_i(t) - \mathcal{B}_i(t)| \geq \varepsilon) = 0. \quad (21)$$

(1) When  $|X_i(t) - P_i| < \varepsilon, \forall j \in \{1, 2, \dots, N\}$ ,

$$|X_i^j(t) - \mathcal{B}_i^j(t)| < \varepsilon, \quad (22)$$

is true, then  $\lim_{t \rightarrow \infty} \mathcal{P}(|X_i^j(t) - \mathcal{B}_i^j(t)| \geq \varepsilon) = 0$  is established.

(2) When  $|X_i(t) - \mathcal{B}_i(t)| \geq \varepsilon$ , so  $\exists j \in \{1, 2, \dots, N\}$  s.t.  $|X_i^j(t) - \mathcal{B}_i^j(t)| \geq \varepsilon$ , and the inclusion relation of the following connection is true:

$$\{|X_i(t) - \mathcal{B}_i(t)| \geq \varepsilon\} \supseteq \{|X_i^j(t) - \mathcal{B}_i^j(t)| \geq \varepsilon\}, \quad (23)$$

thereby

$$\mathcal{P}\{|X_i(t) - \mathcal{B}_i(t)| \geq \varepsilon\} \supseteq \mathcal{P}\{|X_i^j(t) - \mathcal{B}_i^j(t)| \geq \varepsilon\}. \quad (24)$$

We take the limit on both sides of the above equation altogether:

$$0 = \lim_{t \rightarrow \infty} \mathcal{P}\{|X_i(t) - \mathcal{B}_i(t)| \geq \varepsilon\} \supseteq \lim_{t \rightarrow \infty} \mathcal{P}\{|X_i^j(t) - \mathcal{B}_i^j(t)| \geq \varepsilon\}. \quad (25)$$

We obtain

$$\lim_{t \rightarrow \infty} \mathcal{P}\{|X_i^j(t) - \mathcal{B}_i^j(t)| \geq \varepsilon\} = 0, \quad (26)$$

which is  $X_i^j(t) \xrightarrow{p} \mathcal{B}_i^j(t)$ .

Sufficiency: let  $X_i^j(t) \xrightarrow{p} \mathcal{B}_i^j(t)$ , then for each  $j \in \{1, 2, \dots, N\}$ ,  $\forall \varepsilon > 0$ , we have

$$\lim_{t \rightarrow \infty} \mathcal{P}\left\{|X_i^j(t) - \mathcal{B}_i^j(t)| \geq \frac{\varepsilon}{\sqrt{N}}\right\} = 0. \quad (27)$$

If  $|X_i^j(t) - \mathcal{B}_i^j(t)| \geq \varepsilon/\sqrt{N}$ , we obtain

$$|X_i(t) - \mathcal{B}_i(t)| = \left(\sum_{j=1}^N |X_i^j(t) - \mathcal{B}_i^j(t)|^2\right)^{1/2} \geq \left[N\left(\frac{\varepsilon}{\sqrt{N}}\right)^2\right]^{1/2} = \varepsilon, \quad (28)$$

and  $\{|X_i^j(t) - \mathcal{B}_i^j(t)| \geq \varepsilon/\sqrt{N}\} \supseteq \{|X_i(t) - \mathcal{B}_i(t)| \geq \varepsilon\}$ .  
Therefore,

$$\mathcal{P}\left\{|X_i^j(t) - \mathcal{B}_i^j(t)| \geq \frac{\varepsilon}{\sqrt{N}}\right\} \supseteq \mathcal{P}\{|X_i(t) - \mathcal{B}_i(t)| \geq \varepsilon\}. \quad (29)$$

For both sides of the above equation, take the limit simultaneously:

$$\begin{aligned} 0 &= \lim_{t \rightarrow \infty} \mathcal{P}\left\{|X_i^j(t) - \mathcal{B}_i^j(t)| \geq \frac{\varepsilon}{\sqrt{N}}\right\} \\ &\supseteq \lim_{t \rightarrow \infty} \mathcal{P}\{|X_i(t) - \mathcal{B}_i(t)| \geq \varepsilon\}. \end{aligned} \quad (30)$$

We obtain

$$\lim_{t \rightarrow \infty} \mathcal{P}\{|X_i(t) - \mathcal{B}_i(t)| \geq \varepsilon\} = 0, \quad (31)$$

which is  $X_i(t) \xrightarrow{p} \mathcal{B}_i(t)$ .

From Theorem 1, the convergence of the single particle's position satisfying equation (15) can be summarized as the convergence of the particle in 1-dimensional space. If Theorem 1 is satisfied, then it means the convergence of the CIQPSO algorithm. This also shows that the necessary and sufficient condition that the particle's position converges to the attractor with probability is  $\lim_{t \rightarrow \infty} [\text{rand}(t)\gamma \cdot |E^j(t) - X_i^j(t)|] = 0$ .  $\square$

## 5. Numerical Experiments

For the sake of testing the performance of the CIQPSO algorithm, we set the parameters as follows: population size is  $M = 20$  and  $G = 10$  and the maximum number of iterations is  $K_{\max} = 300$ . The fitness function precision of both

Examples 1 and 2 is  $\varepsilon = 10^{-3}$ , and the fitness function precision of Example 3 is  $\varepsilon = 10^{-6}$ .

*Example 1.* Considering the GNEP from [33], the player's decision variables are  $x_1$  and  $x_2$  and the *payoff function* of the players is given by

$$\begin{aligned} \min_{x_1} x_1^2 + \frac{8}{3}x_1x_2 - 34x_1, \\ \min_{x_2} x_2^2 + \frac{5}{4}x_1x_2 - \frac{98}{4}x_2, \end{aligned} \quad (32)$$

$$\text{s.t. } x_1 + x_2 \leq 15, 0 \leq x_1, x_2 \leq 10.$$

The off-line performance and numerical results are shown in Table 1 and Figure 2, respectively.

We can derive the number of average iterations of five calculations is 94 from Table 1, and we can get the GNEP approximate solution of Example 1 is  $(5, 9)^T$ . Compared with the projection-like methods [33], the CIQPSO algorithm not only decreases iterative number, but also the initial point can be randomly generated. The calculation process does not depend on the selection of the initial points. However, when the initial point in [33] is selected as  $(10, 0)$ , the Nash equilibrium point cannot be calculated. In this paper, the average number of iterations is superior to 110 in [22], which shows that the algorithm is more time-saving than immune particle swarm optimization algorithm. In addition, we can know that the CIQPSO algorithm converges faster than the IPSO algorithm [22] from Figure 2, which means the CIQPSO algorithm has better convergence performance.

*Example 2.* Considering the GNEP from [26] and this generalized game has infinite solutions given by  $(\alpha, 1 - \alpha)$ ,  $\forall \alpha \in [1/2, 1]$ , the player's decision variables are  $x_1$  and  $x_2$ , and the *payoff function* of the players is given by

$$\begin{aligned} \min_{x_1} (x_1 - 1)^2, \\ \min_{x_2} \left(x_2 - \frac{1}{2}\right)^2, \end{aligned} \quad (33)$$

$$\text{s.t. } x_1 + x_2 \leq 1, x_1, x_2 \geq 0.$$

The off-line performance and numerical results are shown in Table 2 and Figure 3, respectively.

We can derive the number of average iterations of five calculations is 35 from Table 2, and we can obtain the GNEP approximate solution of Example 2 is  $(0.75, 0.25)^T$ . In [26], Facchinei discussed the normalized equilibrium of Newton methods, but the existence of the GNEP solution can be lost. However, the CIQPSO algorithm can guarantee that some poorly fitness solutions not be lost by probability density selection function. Furthermore, the CIQPSO algorithm is faster than the IPSO algorithm in Figure 3, which indicates the fitness function value is smaller in the generalized Nash equilibrium point, and the convergence performance of the CIQPSO algorithm is effective.

TABLE 1: The calculation results of Example 1.

Times	Iteration numbers	Generalized Nash equilibrium solution	Fitness function value
1	122	$x_1 = 4.9623, x_2 = 9.0514$	$5.6e - 03$
2	110	$x_1 = 5.0012, x_2 = 8.9834$	$4.2e - 03$
3	43	$x_1 = 5.0115, x_2 = 8.9922$	$1.2e - 03$
4	88	$x_1 = 5.0244, x_2 = 8.9838$	$2.9e - 03$
5	108	$x_1 = 4.9431, x_2 = 9.0987$	$6.8e - 03$

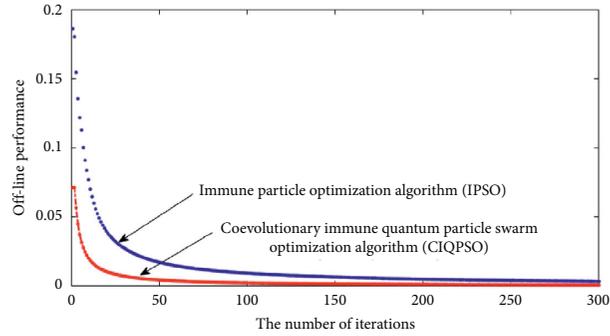


FIGURE 2: Off-line performance of the CIQPSO algorithm.

TABLE 2: The calculation results of Example 2.

Times	Iteration numbers	Generalized Nash equilibrium solution	Fitness function value
1	16	$x_1 = 0.7537, x_2 = 0.2507$	$7.5324e - 03$
2	30	$x_1 = 0.7562, x_2 = 0.2498$	$3.1302e - 03$
3	65	$x_1 = 0.7512, x_2 = 0.2500$	$1.7588e - 03$
4	42	$x_1 = 0.7591, x_2 = 0.2512$	$2.7906e - 03$
5	22	$x_1 = 0.7506, x_2 = 0.2510$	$1.0214e - 03$

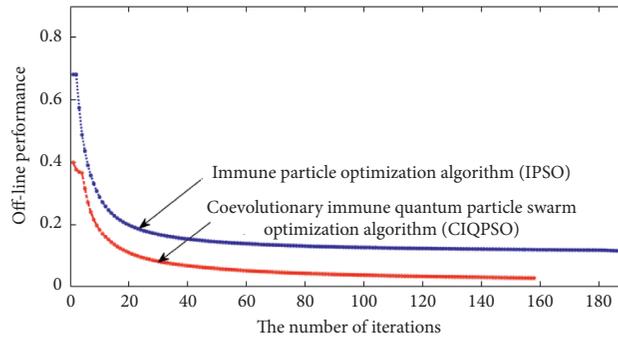


FIGURE 3: Off-line performance of the CIQPSO algorithm.

*Example 3.* Considering the GNEP from [34], the player's decision variables are  $x_1$  and  $x_2$ , and the *payoff function* of the players is given by

$$\begin{aligned}
 & \min_{x_1} x_1^2 - 4x_1 + 3, \\
 & \min_{x_2} x_2^2 - 4x_2 + 4, \\
 & \text{s.t. } x_1 + x_2 \leq 1, \quad -x_1 + 2x_2 \leq 2, \quad x_1, x_2 \geq 0.
 \end{aligned} \tag{34}$$

We use the CIQPSO algorithm to solve this problem, and the numerical results are shown in Table 3.

We can know that the number of average iterations is 2 from Table 3, and we can get the GNEP approximate solution of Example 3 is  $(1.60, 1.80)^T$ . Under the same precision, this paper can find the approximate Nash equilibrium solution only in 2 iterations which is better than the quasi-variational inequalities penalty method in [34], in that the number of iterations is less, the calculation speed is faster,

TABLE 3: The calculation results of Example 3.

Times	Iteration numbers	Generalized Nash equilibrium solution	Fitness function value
1	1	$x_1 = 1.6000, x_2 = 1.8000$	$0.1456e - 06$
2	2	$x_1 = 1.5999, x_2 = 1.8002$	$1.1103e - 06$
3	2	$x_1 = 1.6000, x_2 = 1.8000$	$0.0429e - 06$
4	2	$x_1 = 1.6000, x_2 = 1.7999$	$1.7806e - 06$
5	3	$x_1 = 1.6101, x_2 = 1.7989$	$1.0214e - 07$

and it does not rely on the selection of the initial point of the algorithm. Thus, the example further illustrates the effectiveness of using swarm intelligence algorithm to solve the GNEP.

## 6. Conclusions

By some transformations of GNEP and the construction of appropriate fitness functions, we try to solve approximation solutions of the GNEP by using the CIQPSO algorithm. Some numerical examples illustrate that the algorithm is effective. The GNEP is transformed into the nonlinear equation problem in Section 3. The CIQPSO algorithm is designed and the algorithm is applied in solving the GNEP. What is more, with probability density selection function and quantum uncertainty principle, the convergence of the CIQPSO algorithm is demonstrated. Some numerical examples are given in this paper, Examples 1 and 2 prove that the CIQPSO algorithm is better than the IPSO algorithm, and its convergence speed is faster and not relies on the selection of the initial point. Example 2 proves that the CIQPSO algorithm is superior to the traditional Newton method for normalized equilibria, and the CIQPSO algorithm can guarantee that some poorly fitness solutions not be lost by probability density selection function. Example 3 proves that the CIQPSO algorithm is superior to the quasivariational inequalities penalty method, and the calculation is faster and more time-saving under the same precision. In summary, we know that the CIQPSO algorithm is superior to the immune particle swarm algorithm, the Newton method for normalized equilibria, or quasivariational inequalities penalty method, which means the algorithm has faster convergence and better off-line performance. More specifically, the algorithm did not require highly smoothing of fitness function and the choice of the initial point, and it is also demonstrated that the CIQPSO algorithm is more effective in solving the GNEP. So far, it is not much known about the swarm intelligence algorithm to solve the GNEP, but this theme is well worth further study.

## Data Availability

All the data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 11561013) and Guizhou Science Foundation (Grant no. [2020]1Y284).

## References

- [1] G. Debreu, "A social equilibrium existence theorem," *Proceedings of the National Academy of Sciences*, vol. 38, no. 10, pp. 886–893, 1952.
- [2] G. Scutari, F. Facchinei, J.-S. Pang, and D. P. Palomar, "Real and complex monotone communication games," *IEEE Transactions on Information Theory*, vol. 60, no. 7, pp. 4197–4231, 2014.
- [3] A. Dreves and M. Gerdtts, "A generalized Nash equilibrium approach for optimal control problems of autonomous cars," *Optimal Control Applications and Methods*, vol. 39, no. 1, pp. 326–342, 2018.
- [4] T. Roughgarden, "Computing equilibria: a computational complexity perspective," *Economic Theory*, vol. 42, no. 1, pp. 193–236, 2010.
- [5] F. Facchinei and C. Kanzow, "Penalty methods for the solution of generalized Nash equilibrium problems," *SIAM Journal on Optimization*, vol. 20, no. 5, pp. 2228–2253, 2010.
- [6] K. J. Shah and T. Singh, "The modified homotopy algorithm for dispersion phenomena," *International Journal of Applied and Computational Mathematics*, vol. 3, no. 1, pp. 785–799, 2017.
- [7] D. Aussel, A. Sultana, and V. Vetrivel, "On the existence of projected solutions of quasi-variational inequalities and generalized Nash equilibrium problems," *Journal of Optimization Theory and Applications*, vol. 170, no. 3, pp. 818–837, 2016.
- [8] T. Migot and M.-G. Cojocaru, "A parametrized variational inequality approach to track the solution set of a generalized Nash equilibrium problem," *European Journal of Operational Research*, vol. 283, no. 3, pp. 1136–1147, 2020.
- [9] C. S. Lalitha and M. Dhingra, "Optimization reformulations of the generalized Nash equilibrium problem using regularized indicator Nikaido and Isoda function," *Journal of Global Optimization*, vol. 57, no. 57, pp. 843–861, 2013.
- [10] A. Dreves, "A best-response approach for equilibrium selection in two-player generalized Nash equilibrium problems," *Optimization*, vol. 68, no. 12, pp. 2269–2295, 2019.
- [11] A. F. Izmailov and M. V. Solodov, "On error bounds and Newton-type methods for generalized Nash equilibrium problems," *Computational Optimization and Applications*, vol. 59, no. 1-2, pp. 201–218, 2014.
- [12] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, Springer-Verlag, Berlin, Germany, 2nd edition, 2015.
- [13] E. Wild, *A Study of Heuristic Approaches for Solving Generalized Nash Equilibrium Problems and Related games*, University of Guelph, Guelph, Canada, 2017.

- [14] G. S. Jasmine and P. Vijayakumar, "Congestion management in deregulated power system using heuristic search algorithms incorporating wireless technology," *Wireless Personal Communications*, vol. 94, pp. 2665–2680, 2017.
- [15] H. Rahmanifard and T. Plaksina, "Application of artificial intelligence techniques in the petroleum industry: a review," *Artificial Intelligence Review*, vol. 52, no. 4, pp. 2295–2318, 2019.
- [16] S. P. Maruthi and T. Panigrahi, "Robust mixed source localization in WSN using swarm intelligence algorithms," *Digital Signal Processing*, vol. 98, Article ID 102651, 2020.
- [17] A. Gawanmeh, S. Parvin, and A. Alwadi, "A genetic algorithmic method for scheduling optimization in cloud computing services," *Arabian Journal for Science and Engineering*, vol. 43, no. 12, pp. 6709–6718, 2018.
- [18] D. Dorsch, H. T. Jongen, and V. Shikhman, "On structure and computation of generalized Nash equilibria," *SIAM Journal on Optimization*, vol. 23, no. 1, pp. 452–474, 2013.
- [19] C. Kanzow and D. Steck, "Augmented Lagrangian methods for the solution of generalized Nash equilibrium problems," *Siam Journal on Optimization*, vol. 26, no. 4, pp. 2034–2058, 2016.
- [20] J.-S. Pang and M. Fukushima, "Quasi-variational inequalities, generalized Nash equilibria, and multi-leader-follower games," *Computational Management Science*, vol. 2, no. 1, pp. 21–56, 2005.
- [21] C. Kanzow, "On the multiplier-penalty-approach for quasi-variational inequalities," *Mathematical Programming*, vol. 160, no. 1-2, pp. 33–63, 2016.
- [22] W. S. Jia, S. W. Xiang, J. F. Yang, and J. H. He, "Solving generalized Nash equilibrium problem based on immune particle swarm," *Application Research of Computers*, vol. 30, no. 9, pp. 2637–2640, 2013.
- [23] Y. Wang, W. G. Zhang, L. Fu et al., "Nash equilibrium strategies approach for aerial combat based on elite re-election particle swarm optimization," *Control Theory and Applications*, vol. 32, no. 7, pp. 857–865, 2015.
- [24] A. Rezaee, "The Nash equilibrium point of dynamic games using evolutionary algorithms in linear dynamics and quadratic system," *Automatic Control and Computer Sciences*, vol. 52, no. 2, pp. 109–117, 2018.
- [25] Z. Wang, M. Jusup, L. Shi, J.-H. Lee, Y. Iwasa, and S. Boccaletti, "Exploiting a cognitive bias promotes cooperation in social dilemma experiments," *Nature Communications*, vol. 9, no. 1, 2018.
- [26] F. Facchinei, A. Fisher, and V. Piccialli, "Generalized Nash equilibrium problems and Newton methods," *Mathematical Programming*, vol. 117, no. 1, pp. 163–194, 2009.
- [27] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, Piscataway, NJ, USA, 1995.
- [28] F. V. D. Bergh, *An Analysis of Particle Swarm Optimizers*, University of Pretoria, Pretoria, South Africa, 2001.
- [29] J. Sun, B. Feng, and W. B. Xu, "Particle swarm optimization with particles having quantum behavior," in *Proceedings of the 2004 Congress on Evolutionary Computation*, Portland, OR, USA, June 2004.
- [30] J. Sun, W. Fang, X. Wu, V. Palade, and W. Xu, "Quantum-behaved particle swarm optimization: analysis of individual particle behavior and parameter selection," *Evolutionary Computation*, vol. 20, no. 3, pp. 349–393, 2012.
- [31] G. Lu, D. Tan, and M. Zhao, "Improvement on regulating definition of antibody density of immune algorithm," in *Proceedings of the 9th International Conference on Neural Information Processing*, Singapore, November 2002.
- [32] K. A. Dejong, *Analysis of the Behaviour of a Class of Genetic Adaptive system*, University Michigan, Ann Arbor, MI, USA, 1975.
- [33] J. Z. Zhang, B. Qu, and N. H. Xiu, "Some projection-like methods for the generalized Nash equilibria," *Computational Optimization and Applications*, vol. 45, no. 1, pp. 89–109, 2010.
- [34] W. J. Shi, G. H. Chen, and Z. B. Zhu, "Using quasi-variational inequalities penalty method to solve the generalized Nash equilibrium problems," *China Computer and Communication*, vol. 19, no. 10, pp. 49–51, 2017.