An Improved Quantum Particle Swarm Optimization Algorithm Based on Real Coding Method

Yin Guofu

Weinan Normal University, Weinan, China, 71400, yin_guofu@163.com

Abstract

This paper proposes a novel optimization algorithm combined the mechanism of quantum evolutionary algorithm and real-coding method, called an improved quantum particle swarm optimization algorithm (IQPSO). Like the traditional particle swarm optimization, IQPSO is also characterized by position vector and velocity vector to implement the evolution process. However, the particle of IQPSO is divided into two parts. The first part is real-valued coding; and the rest of it is quantum probability amplitude. Further, IQPSO uses quantum probability amplitude as velocity vector, and a Q-gate is applied to update the quantum probability amplitude. At the same time, a self-adaptable mutation operator is used to improve the diversity of population. To demonstrate the effectiveness of IQPSO, experiments are carried out on the function optimization; and the results show that IQPSO performs well.

Keywords: Quantum Evolution, Particle Swarm Optimization, Real-Coded, Q-Gate, Adaptive Mutation

1. Introduction

Particle Swarm Optimization (PSO), first introduced by Kennedy and Eberhart, is one of the most famous optimization algorithms [1]. Compared with other evolutionary algorithms, the structure of PSO algorithm is simple and it is easy to realize. Nowadays, PSO has applied widely in many fields such as numerical calculation [2], combinatorial optimization [3], data mining [4], classification and clustering [5]. At the same time, PSO algorithm also has some striking shortcomings. Its control on the particle velocity is fixed. There is lack of dynamic adjustment, which makes PSO easy to sink into local optima when it is used to deal with different problems. In addition, the convergence of traditional PSO algorithm is realized in the form of trajectories, and the velocity of particle is limited, so the search space of particle is a limited region in constantly iterative process. And because of this, traditional PSO can not ensure the convergence [6]. To overcome the shortcomings of PSO, in [7], Jun, Feng and Xu updated the evolution rules, they studied the individual particle of a PSO system moving in a quantum multi-dimensional space and establish a quantum Delta potential well model for PSO; further, Liu and Gao gave a quantum PSO [8], this new algorithm can avoid the population fall into the local optimization with adaptive mutation operator.

Additionally, quantum evolutionary algorithm (QEA) is a new optimization algorithm with many features developed on the basis of the existing theory of quantum computing. QEA is more efficient in finding global optimal solution compared with conventional evolution algorithm. Therefore, Li and Li applied directly the mechanism of QEA into the PSO in order to improve the optimization efficiency and accuracy [9]; in [10], Xu et al. proposed an improved QPSO, their proposed QPSO used quantum Hadamard-gate to implement the mutation operation, and proposed an adaptive determination method of the global factors in order to enhance the optimization efficiency of QPSO in [9]. It might also be noted that the above two QPSO algorithms are based on the quantum probability amplitude to encode the particles. The solution space for particles in each dimension is on [-1,1], however, the corresponding variables in actual solution space need to be transferred. This process of transfer will lead to degradation of the population in a certain extent; thereby affect the accuracy of solution.

Considering He and Liang proposed a new QEA [11], they used real-coded schema as the coding method in their algorithm in order to improve the efficiency and precision of calculation. In this paper, the above real-coded schema and the mechanism of QEA will be integrated into PSO to construct the improved quantum particle swarm optimization (IQPSO). Meanwhile, this IQPSO includes self-adaptable mutation strategy in order to improve the diversity and avoid the degradation of population.
The remainder of this paper is organized as following. Section 2 shows a short overview of PSO; section 3 provides an improved quantum particle swarm optimization algorithm; then some experimental results are illustrated and discussed in section 4; finally, the conclusions are made in section 5.

2. Particle Swarm Optimization

The particle swarm concept originated as a simulation of a simplified social system. The original intent was to graphically simulate the graceful but unpredictable choreography of a bird flock. PSO includes a population based on search process where individuals, referred as particles, are grouped into a swarm. As for the optimization problem, every particle represents a candidate solution. When the population is in the searching, each particle can adjust its position within the solution space by using its corresponding experience and that of other particles. Normally it makes use of best position encountered by itself and that of its neighbors to position itself toward an optimal solution. The procedure of classical PSO system with the population size of \( m \) can be described as the following equation: Firstly, the position vector and velocity vector of particle \( i \) in \( n \)-dimensional space are denoted as:

\[
X_i = (x_{i1}, x_{i2}, \ldots, x_{in}) \quad \text{and} \quad V_i = (v_{i1}, v_{i2}, \ldots, v_{in})
\]

And the best position vector of particle \( i \) and the position vector of the best particle among all the particles in the population are denoted as:

\[
P_i = (p_{i1}, p_{i2}, \ldots, p_{in}) \quad \text{and} \quad P_g = (p_{g1}, p_{g2}, \ldots, p_{gn})
\]

Then the trajectory of the particle is formulated by the following equation:

\[
V_i^{t+1} = w \times V_i^t + c_1 \times \text{rand}() \times (P_i - X_i^t) + c_2 \times \text{rand}() \times (P_g - X_i^t) \quad (1)
\]

\[
X_i^{t+1} = X_i^t + V_i^{t+1} \quad (2)
\]

Where \( i = 1, 2, \ldots, m \), parameter \( w \) is the inertia weight, \( c_1 \) and \( c_2 \) are the acceleration constants. Typically, the particle in the population will update itself according to equation (1) and (2) at each iteration, and then the global optimal solution can be got at last.

3. Improved Quantum Particle Swarm Optimization Algorithm

The traditional PSO uses position vector \( x_i \) and velocity vector \( v_i \) to decide the particle trajectories. However, this means nothing in quantum computing and QEA. QEAs are quite different from traditional EAs in structure of evolution [12]. QEAs are based on the concept and principles of quantum computing such as quantum bits and superposition of states. Generally, Q-bit may be in the “0” state, “1” state, or any other superposition between the two states, and it can be represented as the following.

\[
|\phi\rangle = \alpha |0\rangle + \beta |1\rangle \quad (3)
\]

Where \( \alpha \) and \( \beta \) are complex numbers that specify the corresponding probability amplitudes of the states “0” and “1”. They satisfy the normalization.

\[
|\alpha|^2 + |\beta|^2 = 1 \quad (4)
\]

Meanwhile, He and Liang pointed out that Real-coded QEA is more efficient in finding the global optimal solution and robustness than QEA [11], and because of it, this paper uses the real number coding method in [11] and proposed the following evolutionary strategies.
3.1. Initialize the population

As for the objective function \( f(X) \), \( X = (x_1, x_2, ..., x_n) \), \( n \) is the objective dimensions in the feasible solution space. Let \( x_{j\text{-min}} \leq x_j \leq x_{j\text{-max}} \), \( j = 1, 2, ..., n \). And this paper utilizes the following real number coding method to describe these particles:

\[
p_i = \begin{bmatrix}
  x_{i1} & \cos(\theta_{i1}) & \cos(\theta_{i2}) & \ldots & \cos(\theta_{in}) \\
  \sin(\theta_{i1}) & \sin(\theta_{i2}) & \ldots & \sin(\theta_{in})
\end{bmatrix}
\]

(5)

Where \( i = 1, 2, ..., N \), \( N \) is the population size. \( \theta_j \) is a random number, and it would be assumed to be a velocity vector in this paper. Then the position vector \( x_j \) can update itself with \( \cos(\theta_j) \) and \( \sin(\theta_j) \). When the population is initializing, there are the following process.

\[
x_j = x_{j\text{-min}} + (x_{j\text{-max}} - x_{j\text{-min}}) \times \text{rand()} \quad \text{and} \quad \theta_j = 2\pi \times \text{rand()}
\]

Where \( \text{rand()} \) generates a value between 0 and 1.

3.2. Self-adaptable mutation

In traditional PSO model, there exists the memory information when the particles update themselves; meanwhile, the discrepancy between different particles is getting smaller in later period of evolution. Additionally, because of the clustering of particles in constantly iterative process, PSO has the possibility of degradation. These usually make these particles get into premature convergence. In order to overcome this defect, the IQPSO in this paper uses self-adaptable mutation operator, and it defines a self-adaptable mutation probability, which can create persistent random dithering according to the evolution process. The concrete steps are described below.

Firstly, define the self-adaptable mutation probability \( p_m \), and

\[
p_m = 1 - \frac{2 \times \text{ite}}{\text{iteration}}
\]

(6)

Where \( \text{iteration} \) is the predetermined total number of iteration times, and \( \text{ite} \) means the current iterative number for the population.

Further, as for the \( j \)-th-dimensional space in particle \( i \):

If \( \text{rand()} > p_m \), Then \( \theta_j = \theta_j + 2\pi \times \text{rand()} \).

With the iteration number \( \text{ite} \) rapidly increasing and \( p_m \) decreasing, it can be seen that the probability of event \( \text{rand()} > p_m \) will become larger. Therefore, a dynamic adjusting for \( \theta_j \) is presented, which can improve the diversity of population.

3.3. Updating particles in solution space

Traditional QEA uses a rotation Q-gate \( U(\Delta\theta) \) as a variation operator to drive the individuals toward better solutions. In general, Q-gate can be introduced as follows:

\[
U(\Delta\theta) = \begin{bmatrix}
  \cos(\Delta\theta) & -\sin(\Delta\theta) \\
  \sin(\Delta\theta) & \cos(\Delta\theta)
\end{bmatrix}
\]

(7)

As for the angle vector \( \theta = [\theta_1, \theta_2, ..., \theta_j] \), Q-gate could update the angle parameters as following.
In addition, Eberhart’s research has shown in [13], when \( w > 1 \), the particles are tending towards divergence; meanwhile, when \( w < 0 \), these particles are tending towards rest. If and only if \( w = 0.7298 \), \( c_1 = c_2 = 1.49618 \), these particles have better convergence and higher retrieval efficiency. In this paper, angle parameters will be updated as velocity vector. The process of updating is described here.

\[
\Delta \theta_{ij}^{t+1} = w \times \theta_0 + c_1 \times \text{rand}() \times (P_{ij} - \theta_j^t) + c_2 \times \text{rand}() \times (P_g - \theta_j^t)
\]  

(9)

As shown in formula (9), \( \Delta \theta_{ij}^{t+1} \) is the rotation angle for particle \( i \) in \( j \) th-dimensional variable.

And the corresponding best angle vector of particle \( i \) and the angle vector of the best particle among all the particles in the population are denoted as:

\[
P_i = (P_{i1}, P_{i2}, ..., P_{in}) \quad \text{and} \quad P_g = (P_{g1}, P_{g2}, ..., P_{gn})
\]

In [14], Li and Li gave a method to ensure the direction of quantum rotation angle. Suppose \( w_0 = [\alpha_0, \beta_0]^T \) is the quantum amplitudes of best particle in the population, and \( w_i = [\alpha_i, \beta_i]^T \) is the quantum amplitudes of current particle. Let \( A = (w_0, w_i) \), and then the rotation direction of angle \( \theta \) can be selected by the following rules: when \( |A| \neq 0 \), the direction is \( -\text{sgn}(A) \), \( \text{sgn}(A) \) is a symbolic function. And if \( |A| = 0 \), the direction can take +1 or -1. On the principle that has been enunciated above, this paper defines the direction of angle \( \Delta \theta_{ij} \). Therefore, the angle \( \theta_{ij} \) can be updated as following.

\[
\theta_{ij}^{t+1} = \theta_{ij}^t + (-\text{sgn}(A)) \times \Delta \theta_{ij}^{t+1}
\]  

(10)

Then update real-coded quantum particle.

\[
x_{ij}^{t+1}_{\text{cos}} = x_{ij}^t + x_{ij}^t \times \text{cos}(\theta_{ij}^{t+1}) \quad \text{and} \quad x_{ij}^{t+1}_{\text{sin}} = x_{ij}^t + x_{ij}^t \times \text{sin}(\theta_{ij}^{t+1})
\]

If \( x_{ij}^{t+1}_{\text{cos}} \) and \( x_{ij}^{t+1}_{\text{sin}} \) are out of their domain, they will be adjusted themselves according to following steps in order to let them fall into their given interval.

If \( x_{ij}^{t+1} > x_{j-\text{max}} \), and then \( x_{ij}^{t+1} = x_{j-\text{max}} - (x_{j-\text{max}} - x_{j-\text{min}}) \);

If \( x_{ij}^{t+1} < x_{j-\text{min}} \), and then \( x_{ij}^{t+1} = x_{j-\text{min}} + (x_{j-\text{max}} - x_{j-\text{min}}) \);

As two independent candidate solutions, \( x_{ij}^{t+1}_{\text{cos}} \) and \( x_{ij}^{t+1}_{\text{sin}} \) contain three kinds information. They are parent information, the best information of current particle, and the information of best particle in the population respectively. At the same time, they form tow different evolution branches in the solution space which also support the following result to get \( x_{ij}^{t+1} \).

\[
x_{ij}^{t+1} = \text{rand}() \times x_{ij}^{t+1}_{\text{cos}} + (1 - \text{rand}()) \times x_{ij}^{t+1}_{\text{sin}}
\]  

(11)

Where \( \text{rand}() \) gives the probability that \( x_{ij}^{t+1} \) will be constructed from \( x_{ij}^{t+1}_{\text{cos}} \) or \( x_{ij}^{t+1}_{\text{sin}} \). By using the random coefficients \( \text{rand}() \), the population can maintain its diversity effectively in the course of evolution.

3.4. The Process for Implementing IQPSO

In this paper, the IQPSO consists of the following steps.

Procedure IQPSO

Begin

\( t \leftarrow 0 \)
Step 1: Let $N$ be the population size, each particle is corresponding to the dimensions of search space $D$, the parameters related with the IQPSO are expressed as following: $\theta_0 = 0.01\pi$, $w = 0.7298$, $c_1 = c_2 = 1.49618$, and then define a variable invalid $= 0$ that means accumulated invalid evolution number. Subsequently, initialize the population;

Step 2: Evaluate each particle in the population, and store the global optimal solution as $G_{best}$.

While (not termination-condition) do

Begin

$t \leftarrow t + 1$

Step 3: Implement the operation of self-adaptable mutation;

Step 4: Calculating these particles in solution space, and if $G_{best}(t)$ is better than $G_{best}(t-1)$, then updating $G_{best}$; else invalid $= invalid + 1$;

Step 5: As for each particle, find its corresponding best solution as $P_{best}$;

Step 6: if invalid $\leq 5$, update the quantum probability amplitude, and calculate these particles in solution space; else if invalid $> 5$, update the quantum probability amplitude with the following mutation operator: $\theta^{t+1}_j = \theta^t_j + 2 \times \pi \times \text{rand}()$, then calculate these particles in solution space, and let invalid $= 0$.

End

End

4. Experiments and Results

In order to testify the effectiveness of IQPSO, simulation is done based on the following four typical optimization functions in this section [8]. At the same time, a comparison is made with the existing algorithm QPSO [9]. The population size of these two algorithms is 20, the number of iteration is 100, and each of them runs 100 times.

Function 1: $\min f_1(x) = \sum_{i=1}^{n} [x_i^2 - 10 \cos(2\pi x_i) + 10]$. Where $x_i \in [-5.12, 5.12]$; $n = 20$, and its optimal value is 0.

Function 2: $\min f_2(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$. Where $x_i \in [-600, 600]$, $n = 20$, and its optimal value is 0.

Function 3: $\min f_3(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i|$. Where $x \in [-10, 10]$, $n = 20$, and its optimal value is 0.

Function 4: $\max f_4(x, y) = 0.5 - \frac{\sin^2 \sqrt{x^2 + y^2} - 0.5}{[1 + 0.001(x^2 + y^2)]^{0.5}}$. Where $x, y \in [-100, 100]$, function 4 has a lot of local maxima, but there is only one global maximum which value is 1. Therefore, function 4 is easy to fall into local maximum point. When the optimization result is greater than 0.995, and then the algorithm is considered to be convergence. The convergence curves for these optimization functions are described in the following figures.
Figure 1. Comparison of convergence curves on function 1

Figure 2. Comparison of convergence curves on function 2

Figure 3. Comparison of convergence curves on function 3
Regarding the convergence speed, the above diagrams illustrate IQPSO is faster than QPSO, and the optimization performance of the presented IQPSO is more efficient in finding global optimal solution through compared with the existing QPSO. In order to make a further analysis between IQPSO and QPSO, the statistical analysis about the simulation results of the four functions is made in table 1.

Table 1. Comparison of statistical results for the four numerical optimization functions

<table>
<thead>
<tr>
<th>Names</th>
<th>Mean</th>
<th>Optimal Value</th>
<th>Convergence Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function 1</td>
<td>IQPSO</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>QPSO</td>
<td>1.4130e-3</td>
<td>0</td>
</tr>
<tr>
<td>Function 2</td>
<td>IQPSO</td>
<td>1.5455e-12</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>QPSO</td>
<td>3.6688e-5</td>
<td>0</td>
</tr>
<tr>
<td>Function 3</td>
<td>IQPSO</td>
<td>2.1697e-8</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>QPSO</td>
<td>7.2100e-3</td>
<td>0</td>
</tr>
<tr>
<td>Function 4</td>
<td>IQPSO</td>
<td>9.9877e-1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>QPSO</td>
<td>9.9987e-1</td>
<td>1</td>
</tr>
</tbody>
</table>

Based on the foregoing analysis, it is clear that the IQPSO is better than QPSO in both the accuracy and efficiency of numerical simulating for these functions.

5. Conclusions

This paper proposed an improved quantum particle swarm optimization algorithm (IQPSO). The particle includes two branches in IQPSO. Both of them can act on each other according to the new evolution mechanism. Meanwhile, a self adaptable mutation operator is used to improve the diversity of population in order to complete the evolution process efficiently and spontaneously. The experimental results demonstrated the applicability and effectiveness of IQPSO to the complicated optimization function problems.

6. Acknowledgements

The author is grateful to a referee and the anonymous reviewers for their careful reading of this paper and for their very helpful comments and suggestions. This work has been supported by the Natural Science Foundation of Shaanxi Province (2011JM8020, Research on Key Techniques of Multi-node Image Co-processing of Wireless Multi-Media Sensor Networks).
7. References


