



---

# A NEW QUANTUM PARTICLE SWARM OPTIMIZATION ALGORITHM WITH LOCAL ATTRACTING

*D. Shao, S. Hu, Y. Fei*

---

**Abstract:** This paper proposes a new quantum particle swarm optimization algorithm with local attracting (LAQPSO), which is based on quantum-inspired evolutionary algorithm (QEA) and particle swarm optimization algorithm (PSO). In the proposed LAQPSO, a novel quantum bit expression mechanism called quantum angle is employed to encode the solution onto particle, and a new local attractor is proposed to determine the rotation angle of quantum rotation gate automatically. During the process of seeking the global solution, the magnitude of rotation angle is adjusted by an important parameter called contraction coefficient, which can quantitatively determine the tradeoff between exploration ability and exploitation ability. The simulation results for different contraction coefficients are helpful for selecting the key parameter. A set of benchmark functions are used to evaluate the performance of LAQPSO, QEA and QBPSO, and the results show that the proposed algorithm has a fast convergence rate and can effectively avoid premature convergence.

Key words: *quantum particle swarm optimization, quantum angle, local attractor*

Received: December 13, 2014

DOI: 10.14311/NNW.2016.26.028

Revised and accepted: October 6, 2016

## 1. Introduction

Quantum computing can be viewed as developing algorithms that are derived from the combination of quantum mechanics, computer science and classical information theory [4]. Ever since Benioff [1] presented his quantum Turing machine (QTM) in 1980, the field of quantum computing has begun to be explored. After that, Shor [15] presented a efficient quantum algorithm for factoring integers, and Grover [5] developed a search quantum algorithm, which can provide quadratic speed-up when applied to search problems. All the previous work has proved that many difficult problems in classical computing could be solved efficiently with quantum algorithms.

Quantum computing would be most efficient if run on a quantum computer, as the same do not yet exist, and meantime the quantum simulation on a classical computer has an exponential order overhead. Nevertheless, it is beneficial to introduce

---

Dong Shao – Corresponding author, Shenghai Hu, Yuting Fei, College of Mechanical and Electrical Engineering, Harbin Engineering University, Harbin 150001, PR China, E-mail: dongshao@hrbeu.edu.cn, hushenghai@hrbeu.edu.cn, 370515242@qq.com

the concepts and principles of quantum computing into classical algorithms to offer good performance. Li and Wang [11] proposed a quantum ant colony optimization algorithm, in which the positions of ant were encoded by qubits described on Bloch sphere and the mutation was performed with Hadamard gates. In 2002, Han and Kim [6] proposed a quantum-inspired evolutionary algorithm (QEA), which was inspired by the concepts and principles of quantum computing, such as a quantum bit and superposition of states. Later on, several researches [7, 10, 13] were conducted to improve the QEA to cover its shortages. Meanwhile, [2, 14] applied the QEA directly to solve some practical engineering problems.

Particle swarm optimization algorithm (PSO) was first suggested by Kennedy and Eberhart [9] in 1995, a swarm intelligence algorithm derived from the simulation model of bird flocking. Compared with other evolutionary algorithms, it is easier to implement with relatively less adjusting parameters. In the original version of PSO, a population of candidate solutions called particles are randomly initialized and moved around in the search space to optimize a problem. The velocity and position of each particle at generation  $t$  are determined by its best previous position as  $P_{\text{best},i}^t = [P_{\text{best},i1}^t, P_{\text{best},i2}^t, \dots, P_{\text{best},iD}^t]$  called personal best position and the best previous position of all particles as  $G_{\text{best}}^t = [G_{\text{best},1}^t, G_{\text{best},2}^t, \dots, G_{\text{best},D}^t]$  called global best position. The evolution equations of each particle can be expressed as

$$v_{id}^{t+1} = \omega \cdot v_{id}^t + c_1 \cdot \text{rand}_{i1} \cdot (P_{\text{best},id}^t - x_{id}^t) + c_2 \cdot \text{rand}_{i2} \cdot (G_{\text{best},d}^t - x_{id}^t), \quad (1)$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^t, \quad (2)$$

for  $i = 1, 2, \dots, n$ , where  $n$  is the size of the population, and  $d = 1, 2, \dots, D$ , where  $D$  is the dimension of search space.

PSO has a fast convergence rate and works well for many complex problems. However, global convergence is not guaranteed due to the fixed trajectory and the limited velocity of each particle [20]. To addresses this shortcoming, quantum computing has been introduced into PSO in recent years, and these researches can be classified into two fields: (1) quantum-behaved particle swarm optimization (QPSO), developed by Sun, Feng and Xu [16], originated from a quantum Delta potential well model for PSO, and (2) quantum-inspired particle swarm optimization (QIPSO), which was based on QEA and the evolution equations of PSO were employed to update the particles [8, 12, 21].

In this paper, a new quantum particle swarm optimization algorithm with local attracting (LAQPSO) is proposed based on QEA and PSO. The proposed algorithm employs a novel quantum bit expression mode called quantum angle, and a new local attractor, generated by crossover operation, is proposed to define the rotation angle of quantum rotation gate. The numerical results may provide a reference for selecting parameters to balance the exploration and exploitation abilities. The proposed algorithm is compared with the two well-known algorithms to evaluate its performance.

The remainder of this paper is organized as follows. Section 2 introduces two basic concepts of quantum computing quantum bit and quantum gate, and then explains the basic structure of QEA and an important QIPSO afterwards. The definition of the quantum angle and local attractor as well as the procedure of

LAQPPO is illustrated in Section 3. In order to select the key parameter contraction coefficient and evaluate the performance of proposed algorithm, a set of benchmark functions in the impartial environments are compared and analyzed in Section 4. Some useful conclusions are summarized in Section 5.

## 2. Evolutionary algorithm with quantum computing

### 2.1 Quantum bit

As the elementary unit of classical computation, a classical bit is a system that can exist in two distinct states, which are used to represent binary digit 0 and 1. Similarly, the basic unit of quantum information and quantum computation is a quantum bit, commonly referred to as a Q-bit, which is a two-level quantum system, described by a two-dimensional complex Hilbert space. Just like the states 0 and 1 for a classical bit, a pair of normalized and mutually orthogonal quantum states in the space can be chosen as the computational basis states for a Q-bit:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

where “ $|\ \rangle$ ” is Dirac notation, the standard notation for states in quantum mechanics.

What differs Q-bit and classical bit is that a Q-bit can be in a state other than  $|0\rangle$  and  $|1\rangle$ , which is a linear combinations of the two basic states, often called superposition:

$$|\varphi\rangle = \alpha |0\rangle + \beta |1\rangle,$$

where  $\alpha$  and  $\beta$  are complex numbers, which are referred to as probability amplitudes for quantum states and constrained by the normalization condition [18, 19]:

$$|\alpha|^2 + |\beta|^2 = 1. \quad (3)$$

Therefore, unlike the classical bit, which can only be one of the basic states, a Q-bit's state is a unit vector in a two-dimensional complex vector space, which is parameterized by the continuous variables  $\alpha$  and  $\beta$ . According to our intuition, a system with two states can only be in one state or the other. However, Q-bit allows a continuum of states.

At this stage, a single Q-bit might be used to store an infinite amount of information. Nevertheless, just like a classical bit, a Q-bit must be read to get its value for computing. In order to get this value, we have to design a measuring operation for Q-bit, and the result can only be either 0 or 1. Furthermore, measurement changes the state of a Q-bit, collapsing it from its superposition of  $|0\rangle$  and  $|1\rangle$  to the specific state consistent with the measurement result. In fact, the state  $|\varphi\rangle$  respectively collapses to either  $|0\rangle$  or  $|1\rangle$  according to the probability of either  $|0\rangle$  or  $|1\rangle$  because of measurement. Thus, the outcome of a Q-bit is not deterministic as in classical computing, but rather probabilistic.

## 2.2 Quantum gates

In classical circuit model, the logic gates perform manipulations of the information, including the identity gate, NOT, AND and OR, etc. Similarly, Some analogous quantum gates have also been defined for Q-bits. Due to the reversibility of quantum computing [18, 19], a quantum gate has the same amount of input Q-bits and output Q-bits. In general, a single-Q-bit gates can be described by a two by two matrices. Affected by a quantum gate, a quantum state  $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$  changes into another quantum state  $|\varphi'\rangle = \alpha'|0\rangle + \beta'|1\rangle$ . According to the normalization condition, the two states should satisfy Eq. (3), and it turn out that the appropriate condition on the matrix representing the gate is that the matrix  $\mathbf{U}$  describing the single-Q-bit gate should be unitary, that is  $\mathbf{U}^*\mathbf{U} = \mathbf{I}$ , where  $\mathbf{U}^*$  is the conjugate transpose of  $\mathbf{U}$ . Three important single-Q-bit gates are detailed below:

$$\text{Hadamard gate: } \mathbf{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix},$$

$$\text{NOT gate: } \mathbf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix},$$

$$\text{Rotation gate: } \mathbf{R} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}.$$

Hadamard gate can create an equal superposition of states, and when applied to  $n$ -Q-bits, it creates a superposition of all  $2^n$  states. The quantum NOT gate takes the state  $\alpha|0\rangle + \beta|1\rangle$  to the corresponding state  $\alpha|1\rangle + \beta|0\rangle$ , in which the role of  $|0\rangle$  and  $|1\rangle$  have been interchanged. The quantum rotation gate can realize a  $\theta$  phase rotation for a single-Q-bit.

## 2.3 Quantum-inspired evolutionary algorithm

The basic structure of QEA was proposed by Han and Kim [6], which was inspired by the concepts and principles of quantum computing. Unlike the classical representations which can be used to encode the solution onto individual in traditional evolutionary algorithms, a Q-bit representation was used in QEA. One Q-bit was defined as  $[\alpha, \beta]^T$ , where  $\alpha$  and  $\beta$  are the probability amplitudes of the corresponding states, and a Q-bit individual is a string of  $m$  Q-bits which is defined as

$$\begin{bmatrix} \alpha_1 & | & \alpha_2 & | & \dots & | & \alpha_m \\ \beta_1 & | & \beta_2 & | & \dots & | & \beta_m \end{bmatrix},$$

where  $|\alpha_j|^2 + |\beta_j|^2 = 1$ ,  $j = 1, 2, \dots, m$ . In a system of  $m$ -Q-bits, all the information of  $2^m$  states can be represented by only one Q-bit individual because the Q-bit representation has the advantage of representing a linear superposition of states. For this reason, evolutionary algorithms with the Q-bit representation has better characteristic of generating diversity in population than other representations in traditional computing. In addition, as  $|\alpha_j|^2$  or  $|\beta_j|^2$  approaches to 0 or 1, the Q-bit individual converges to a determinate state and the property of diversity disappears gradually. This indicates that the Q-bit individual has both the exploration ability and the exploitation ability.

Based on the Q-bit representation, in QEA, a population of Q-bit individuals at generation  $t$  was defined as

$$Q(t) = [q_1^t, q_2^t, \dots, q_n^t],$$

where  $q_i^t$  is a Q-bit individual defined as

$$q_i^t = \left[ \begin{array}{c|c|c|c} \alpha_{i1}^t & \alpha_{i2}^t & \dots & \alpha_{im}^t \\ \beta_{i1}^t & \beta_{i2}^t & \dots & \beta_{im}^t \end{array} \right] = [q_{i1}^t, q_{i2}^t, \dots, q_{im}^t],$$

where  $m$  is the length of the Q-bit individual, and  $i = 1, 2, \dots, n$ . In the iterative process of QEA, the quantum rotation gate was employed to update a Q-bit individual by

$$U(\Delta\theta_{ij}^t) = \begin{bmatrix} \cos \Delta\theta_{ij}^t & -\sin \Delta\theta_{ij}^t \\ \sin \Delta\theta_{ij}^t & \cos \Delta\theta_{ij}^t \end{bmatrix},$$

where  $\Delta\theta_{ij}^t$  is the rotation angle of the  $j$ -th Q-bit of the  $i$ -th Q-bit individual, its value directly affects the speed of convergence and the searching efficiency. As presented in [6], a lookup table (Tab. I) was defined for determining the value of  $\Delta\theta_{ij}^t$  based on  $x_{ij}^t$ ,  $b_{ij}^t$ , and their fitness values  $F(x_{ij}^t)$  and  $F(b_{ij}^t)$ .

$x_{ij}^t$	$b_{ij}^t$	$F(x_{ij}^t) \geq F(b_{ij}^t)$	$\Delta\theta_{ij}^t$
0	0	False	0
0	0	True	0
0	1	False	$0.01\pi$
0	1	True	0
1	0	False	$-0.01\pi$
1	0	True	0
1	1	False	0
1	1	True	0

**Tab. I** Lookup table for  $\Delta\theta_{ij}^t$ ,  $F(x)$  denotes the fitness value of  $x$ .

Jeong et al. [8] proposed a new binary particle swarm optimization inspired by quantum computing (QBPSO), in which the rotation angle can be determined by each particle's own experience and its neighbor's experience as

$$\Delta\theta_{ij}^t = \theta^t \cdot \{\gamma_{1i}^t \cdot P_{\text{best},ij}^t - x_{ij}^t\} + \gamma_{2i}^t \cdot (G_{\text{best},j}^t - x_{ij}^t),$$

where  $\theta^t$  is the magnitude of rotation angle, and  $\gamma_{1i}^t$  and  $\gamma_{2i}^t$  can be obtained by comparing the fitness of current position of particle  $q_i^t$  with those of  $P_{\text{best},i}^t$  and  $G_{\text{best}}^t$  respectively, as follows:

$$\gamma_{1i}^t = \begin{cases} 0 & \text{if } F(x_i^t) \geq F(P_{\text{best},i}^t) \\ 1 & \text{otherwise} \end{cases},$$

$$\gamma_{2i}^t = \begin{cases} 0 & \text{if } F(x_i^t) \geq F(G_{\text{best}}^t) \\ 1 & \text{otherwise} \end{cases}.$$

In QBPSO, the magnitude of rotation angle decreases monotonously from the maximum  $\theta_{\max}$  to the minimum  $\theta_{\min}$  along the iteration:

$$\theta^t = \theta_{\max} - (\theta_{\max} - \theta_{\min}) \cdot \frac{t}{i_{\max}},$$

where  $i_{\max}$  is the number of maximum iterations.

### 3. The QPSO improvement based on local attracting

#### 3.1 Quantum angle

In order to introduce the concepts of quantum computing into PSO more conveniently, we employ a quantum angle representation to encode the potential solution onto particle. According to the normalization condition (Eq. (3)), a quantum angle can be defined as

$$q_{ij}^t = \begin{bmatrix} \alpha_{ij}^t \\ \beta_{ij}^t \end{bmatrix} \rightarrow \theta_{ij}^t : \begin{cases} |q_{ij}^t\rangle = \cos \theta_{ij}^t |0\rangle + \sin \theta_{ij}^t |1\rangle \\ \theta_{ij}^t = \arctan \frac{\alpha_{ij}^t}{\beta_{ij}^t} \end{cases} .$$

Thus, a Q-bit can be represented by a quantum angle. Then the population of Q-bit particles can be interpreted in the form of quantum angles:

$$\begin{aligned} q_i^t &= [q_{i1}^t, q_{i2}^t, \dots, q_{im}^t] \\ &\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ \theta_i^t &= [\theta_{i1}^t, \theta_{i2}^t, \dots, \theta_{im}^t] \\ \\ Q(t) &= [q_1^t, q_2^t, \dots, q_n^t] \\ &\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ \theta(t) &= [\theta_1^t, \theta_2^t, \dots, \theta_n^t] \end{aligned} .$$

The quantum rotation gate can also be replaced by a quantum angle operator:

$$\begin{aligned} \begin{bmatrix} \alpha_{ij}^{t+1} \\ \beta_{ij}^{t+1} \end{bmatrix} &= \begin{bmatrix} \cos \Delta\theta_{ij}^t & -\sin \Delta\theta_{ij}^t \\ \sin \Delta\theta_{ij}^t & \cos \Delta\theta_{ij}^t \end{bmatrix} \begin{bmatrix} \alpha_{ij}^t \\ \beta_{ij}^t \end{bmatrix} \\ &\downarrow \\ \theta_{ij}^{t+1} &= \Delta\theta_{ij}^t + \theta_{ij}^t \end{aligned} .$$

In fact, a quantum angle means an angle vector in a two-dimensional complex vector space, corresponding to a Q-bit's unit vector, as shown in Fig. 1.

#### 3.2 Local attractor

In PSO, the particles fly stochastically in the search space with a velocity constantly updated by the experience of each particle and the experience of the particle's neighbors or the whole swarm. Eqs. (1) and (2) show that the state of the particle can be depicted by its position vector and velocity vector, both determine the trajectory of the particle.

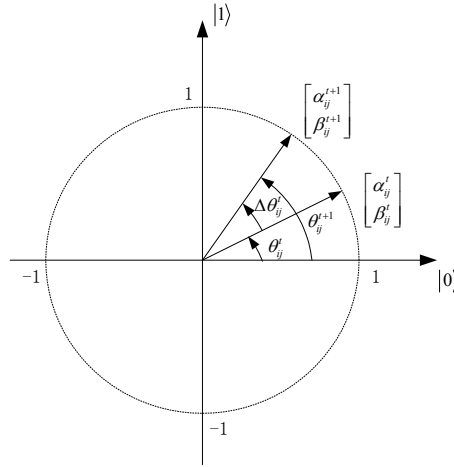


Fig. 1 Quantum angle.

Trajectory analyses in [3] determined the fact that the convergence of PSO algorithm is achieved only if each particle in the PSO system converges to its local attractor  $P_i^t = [P_{i1}^t, P_{i2}^t, \dots, P_{im}^t]$  as it moves around in the space, of which the coordinates are

$$P_{ij}^t = \varphi_{ij}^t \cdot P_{\text{best},ij}^t + (1 - \varphi_{ij}^t) \cdot G_{\text{best},j}^t, \quad (4)$$

where  $\varphi_{ij}^t$  is a random number distributed uniformly on  $[0, 1]$ .

In the convergence procedure of PSO, the  $i$ -th particle moves around and careers toward point  $P_i^t$  with its velocity declining. That is, the  $i$ th particle is flying in an attraction potential field centering around point  $P_i^t$  to guarantee convergence. Nevertheless, the local attractor calculated by Eq. (4) is considered in the real search space, and it does not exist in the discrete binary version of PSO. To generate the local attractor represented by binary coding, Sun et al. [17] suggested a crossover operation like that used in Genetic Algorithm (GA). The point  $P_i^t$  is obtained by randomly selection from two offspring generated by exerting crossover on the two parents,  $P_{\text{best},i}^t$  and  $G_{\text{best}}^t$ . The crossover operation can be classified universally as one-point and multi-point crossover. In the proposed algorithm, the local attractor with binary coding is generated according to the following equation:

$$P_{ij}^t = \tilde{\varphi}_{ij}^t \cdot P_{\text{best},ij}^t + (1 - \tilde{\varphi}_{ij}^t) \cdot G_{\text{best},j}^t, \quad (5)$$

where  $\tilde{\varphi}_{ij}^t$  is a random integer distributed uniformly on  $[0, 1]$ . Obviously, the position of the point  $P_i^t$  resulted from the above approach lies between  $P_{\text{best},i}^t$  and  $G_{\text{best}}^t$  in binary space. Therefore, the particle's converging to the point  $P_i^t$  reduces the diversity of the population, correspondingly increasing to the local search of particle.

### 3.3 Procedure of LAQPSO

Before describing the procedure of the proposed LAQPSO, some important definitions and modifications are explained below.

In the proposed algorithm, we use the quantum angle to encode Q-bit and regard all Q-bit particles in the population as a quantum angle swarm. Just like other evolutionary algorithms, the first step is initialization. Each quantum angle  $\theta_{ij}^0$ ,  $j = 1, 2, \dots, m$ , of all particles  $\theta_i^0$ ,  $i = 1, 2, \dots, n$ , in the population  $\theta(t)$  are initialized with  $\pi/4$ . In essence, it is equivalent to initializing each probability amplitude with  $1/\sqrt{2}$  in QEA. It means that one particle represents the linear superposition off all possible states with the same probability  $1/\sqrt{2^m}$ .

For a Q-bit, we can read to get its value by observing its state, which is either 0 or 1 according to the probability of either  $|\alpha|^2$  or  $|\beta|^2$  respectively. Hence, a quantum angle can be observed as following:

$$x_{ij}^t = \begin{cases} 0 & \text{rand} < |\cos \theta_{ij}^t|^2 \\ 1 & \text{otherwise} \end{cases}, \quad (6)$$

where “rand” is the uniformly distributed random number between  $[0, 1]$ .

Through this operation, the  $i$ th particle  $\theta_i^t$  transforms into a binary solution in  $x_i^t = [x_{i1}^t, x_{i2}^t, \dots, x_{im}^t]$ , which is a binary string of length  $m$ . Then we can evaluate the fitness value of each particle and find the personal best position and the global best position.

In the evolution process of the proposed LAQPSO, we update quantum angle by a quantum rotation gate, where the rotation angle is determined by the local attractor.

1) To determine the direction of rotation angle, we define the direction through comparing the local attractor and the particles of current swarm as follows:

$$\text{Dir}(\theta_{ij}^t) = P_{ij}^t - x_{ij}^t. \quad (7)$$

2) To determine the magnitude of rotation angle, we first give a definition on the quantum angles corresponding to the local attractor  $P_i^t$ , that is, the local attractor can be obtained with 100% probability by observing a pre-defined quantum angle individual  $\tilde{\theta}_i^t = [\tilde{\theta}_{i1}^t, \tilde{\theta}_{i2}^t, \dots, \tilde{\theta}_{im}^t]$ . If the value range of  $\tilde{\theta}_{ij}^t$  is set as  $[0, \pi/2]$ , according to the observing operation (Eq. (6)) we can derive that when  $P_{ij}^t$  is 1 or 0, the corresponding quantum angle should be  $\pi/2$  or 0 respectively. In summary, the quantum angle individual  $\tilde{\theta}_i^t$  can be defined as

$$\tilde{\theta}_{ij}^t = \frac{\pi}{2} P_{ij}^t. \quad (8)$$

Then we use  $\tilde{\theta}_{ij}^t$  and the current quantum angle  $\theta_{ij}^t$  to determine the magnitude of rotation angle as

$$|\Delta\theta_{ij}^t| = a \cdot \left| \tilde{\theta}_{ij}^t - \theta_{ij}^t \right| \cdot \text{rand}, \quad (9)$$

where  $a$  is an important parameter called contraction coefficient which can adjust the magnitude of rotation angle. Therefore, we can obtain the rotation angle formulated by Eqs. (7)–(9) of the quantum rotation gate, which makes the local attractor  $P_i^t$  produce an attraction potential field to guarantee the bound state of the particle. In general, the particle with bound state can be in any position of the feasible solution space with a certain probability, thus, the swarm can search in the whole space. Besides, the proposed approach has a better characteristic of search



efficiency than other approaches, due to its capability of automatic adaption to the figures and features of the search space.

With the above definitions and modifications, the Quantum Particle Swarm Optimization with Local Attracting (LAQPSO) algorithm can be described as the following procedure:

---

**Algorithm 1** LAQPSO algorithm.

---

Initialize a quantum angle swarm:  $\theta(0) = [\theta_1^0, \theta_2^0, \dots, \theta_n^0]$ .

**repeat**

Determine the binary swarm by observing the states of  $\theta(t)$  using Eq. (6):

$X(t) = [x_1^t, x_2^t, \dots, x_n^t]$ .

Evaluate the fitness values of all particles:  $F(x_i^t)$ .

**if**  $F(x_i^t) < F(P_{\text{best},i}^{t-1})$  **then**

Update the personal best position:  $P_{\text{best},i}^t = x_i^t$ .

**end if**

**if**  $F(x_i^t) < F(G_{\text{best}}^{t-1})$  **then**

Update the global best position:  $G_{\text{best}}^t = x_i^t$ .

**end if**

Generate the local attractor according to Eq. (5):  $P(t) = [P_1^t, P_2^t, \dots, P_n^t]$ .

Determine the direction and magnitude of rotation angle, then evaluate the rotation angle  $\Delta\theta(t) = [\Delta\theta_1^t, \Delta\theta_2^t, \dots, \Delta\theta_n^t]$  by the following equation:

$$\Delta\theta_{ij}^t = \text{Dir}(\Delta\theta_{ij}^t) \cdot |\Delta\theta_{ij}^t|.$$

Update the quantum angle swarm:  $\theta(t+1) = \theta(t) + \Delta\theta(t)$ .

**until** a stop criterion is satisfied or a pre-specified number of iterations are completed.

Output the best individual and optimal solution.

---

## 4. Optimization experiments

### 4.1 Test functions

In order to evaluate the performance of LAQPSO, we select six unconstrained minimization benchmark functions, which are considered in the real search spaces. Mathematical models of the functions are given in Tabs. II and III. The functions in Tab. II are unimodal functions with variable dimensions, for whose the convergence rate of the search algorithm is more important than the final results. Three functions of Tab. III are multimodal high-dimension functions, where finding an optimal solution is important. These functions have many local optima and the number of local optima increases exponentially as the dimension increases, so they are difficult to optimize.

In Tabs. II and III,  $D$  is the dimension of the function,  $Range$  is the admissible range of the variable and  $Optimal$  is the optimum value of the function. The special properties of the benchmark functions taken in this paper are described as following:

Test function	$D$	Range	Optimal
$F_1 = \sum_{i=1}^D x_i^2$	30/50	[-100, 100]	0
$F_2 = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$	30/50	[-30, 30]	0
$F_3 = \sum_{i=1}^D ix_i^4 + \text{rand}$	30/50	[-1.28, 1.28]	0

**Tab. II** Unimodal test functions.

Test function	$D$	Range	Optimal
$F_4 = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30/50	[-5.12, 5.12]	0
$F_5 = \sum_{i=1}^D 418.9829 - x_i \sin(\sqrt{ x_i })$	30/50	[-500, 500]	0
$F_6 = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}}) + 1$	30/50	[-600, 600]	0

**Tab. III** Multimodal test functions.

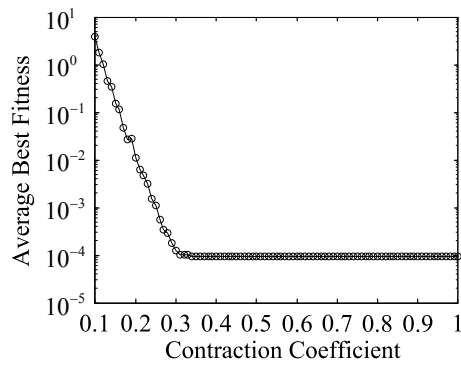
- $F_1$  is the Sphere function. It is a continuous and strictly convex function, which has  $D$  local minima except for the global one.
- $F_2$  is the Rosenbrock function, also known as the Valley or Banana function. It is a popular optimization problem, and the global minimum is hidden inside a narrow, parabolic valley. However, although this valley is easy to find, convergence to the minimum is difficult.
- $F_3$  is the forth function of De Jong. It is a continuous, convex, unimodal quartic function with noise.
- $F_4$  is the Rastrigin function. It is highly multimodal, but locations of minima are regularly distributed.
- $F_5$  is the Schwefel function. It is a complex function with many local minima.
- $F_6$  is the Griewank function. It has many widespread local minima, which are regularly distributed.

In this paper, 15 bits are used to represent the continuous variable of the functions, thus the length of the binary string onto each particle is  $m = 15D$ . Each optimization experiment is run independently 50 times.

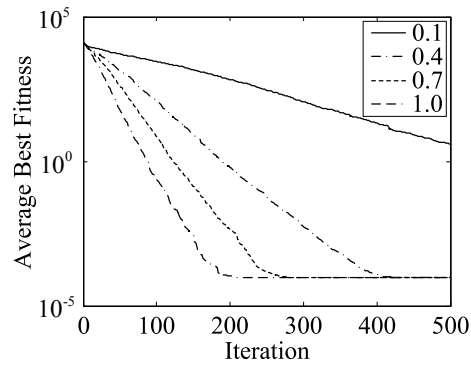
## 4.2 The comparison by different $a$

In general, the final results of the optimization algorithm are greatly dependent on the dimension of search space ( $D$ ), the number of all particles ( $n$ ) and the

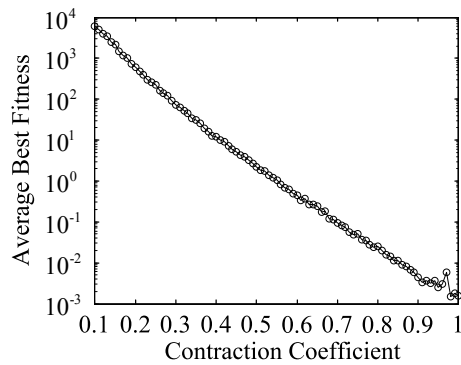
complexity of the objective function, etc. In addition, in the proposed LAQPSO, the magnitude of rotation angle could be affected by contraction coefficient  $a$ , and rotation angle directly determines the exploration and exploitation abilities of the algorithm. In order to study the influence of the contraction coefficient  $a$ , a sensitivity analysis of LAQPSO algorithm is mainly conducted to investigate the relationship between  $D$ ,  $n$  and  $a$ .



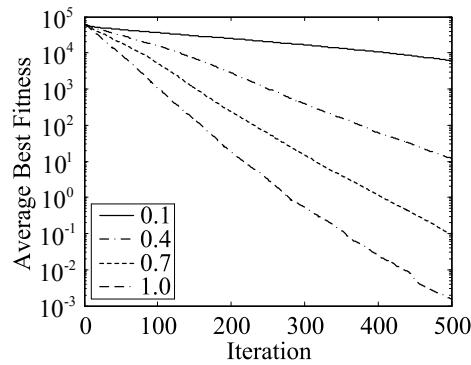
**Fig. 2** The contraction coefficient  $a$  on  $F_1$  ( $D = 10$ ).



**Fig. 3** The performance of several contraction coefficients on  $F_1$  ( $D = 10$ ).

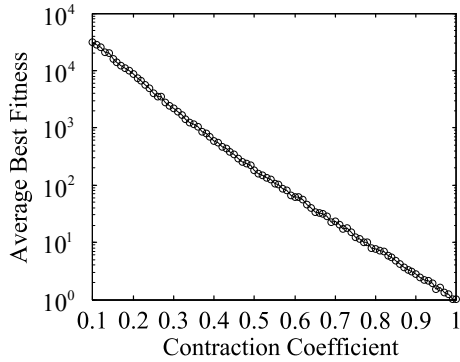


**Fig. 4** The contraction coefficient  $a$  on  $F_1$  ( $D = 30$ ).

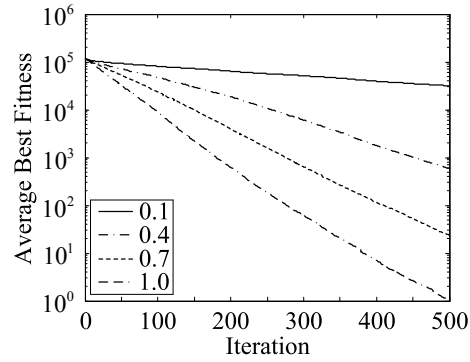


**Fig. 5** The performance of several contraction coefficients on  $F_1$  ( $D = 30$ ).

For function  $F_1$ , the population size is set to 40, and the number of maximum iterations is set at 500. When the dimension of search space increases from 10 to 50, we compare and analyze the contraction coefficient  $a$ . The average results of 50 independent trials are shown in Figs. 2, 4 and 6. The results show that the best contraction coefficient  $a$  does not change as the dimension of search space increases from 10 to 50, and the larger the contraction coefficient  $a$  is, the better results will be got. The convergence progress of several contraction coefficients on  $F_1$  are illustrated in Figs. 3, 5 and 7. The results in these figures show that large  $a$  increases the speed of converging into the global optimum. For the unimodal function, the

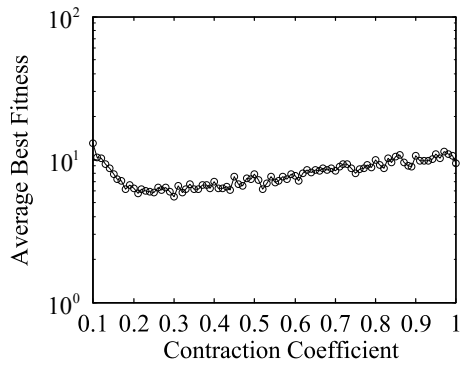


**Fig. 6** The contraction coefficient  $a$  on  $F_1$  ( $D = 50$ ).

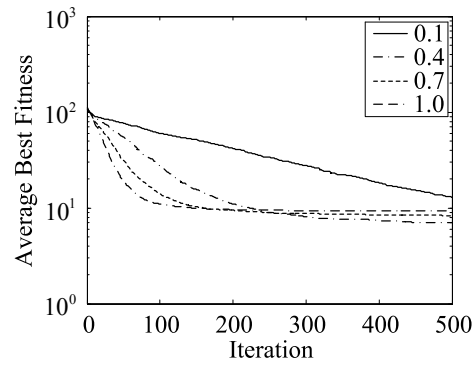


**Fig. 7** The performance of several contraction coefficients on  $F_1$  ( $D = 50$ ).

fast convergence speed may yield good results since it has few local optimums except for the global one. Therefore, it is better to select large contraction coefficient  $a$  to get the good results.

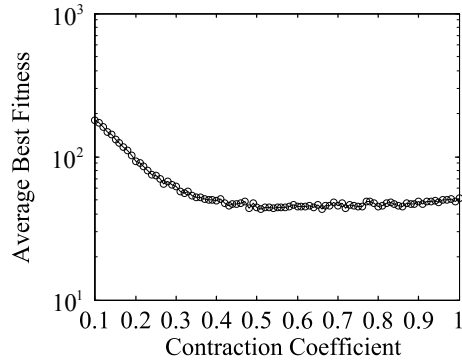


**Fig. 8** The contraction coefficient  $a$  on  $F_4$  ( $D = 10$ ).

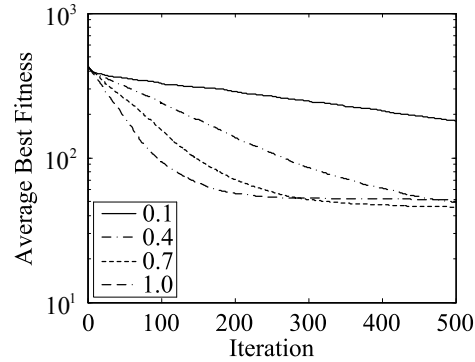


**Fig. 9** The performance of several contraction coefficients on  $F_4$  ( $D = 10$ ).

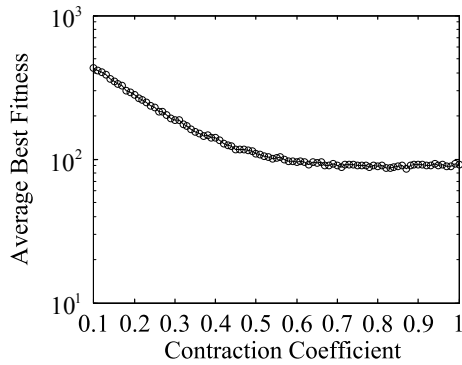
For function  $F_4$ , the population size and the number of maximum iterations are set to 40 and 500. The contraction coefficients  $a$  are compared in Figs. 8, 10 and 12 as the dimension of search space increases from 10 to 50, and Figs. 9, 11 and 13 show the convergence progress of several contraction coefficients on  $F_4$ . The results in Figs. 8, 10 and 12 show that the best contraction coefficient  $a$  increases with the dimension of search space increasing, when the contraction coefficient  $a$  increases from 0.1 to 1, the solution decreases first and then increases. Due to large number of local optimums in the landscape of multimodal high-dimension function, it requires strong exploration ability of the particles to acquire a good solution. When the dimension of search space is relatively small, there will be less local optimum, and the large contraction coefficient  $a$  easily gives rise to premature convergence because of the powerful exploitation ability of the particles, so the contraction coefficient  $a$



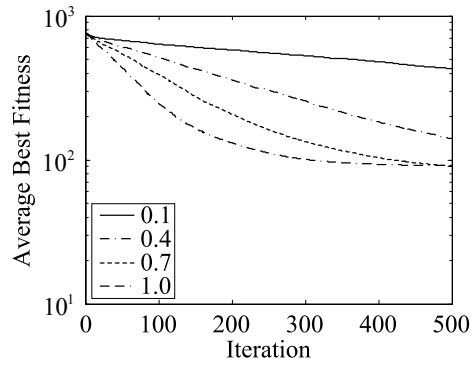
**Fig. 10** The contraction coefficient  $a$  on  $F_4$  ( $D = 30$ ).



**Fig. 11** The performance of several contraction coefficients on  $F_4$  ( $D = 30$ ).



**Fig. 12** The contraction coefficient  $a$  on  $F_4$  ( $D = 50$ ).



**Fig. 13** The performance of several contraction coefficients on  $F_4$  ( $D = 50$ ).

should better be selected to be small value to avoid premature convergence to get good solution. However, if the contraction coefficient is too small, a bad solution will be obtained due to the weak convergence. With the dimension of search space increasing, there will be more local optimum, and then the proposed algorithm not only needs the powerful exploration ability of the particles to avoid premature convergence, but also need many epochs to converge into the suboptimal or global optimum, therefore it is better to select a large contraction coefficient  $a$ .

Figs. 14–16 show the average final results for different contraction coefficient  $a$  on  $F_2$  and  $F_5$  with the number of all particles increasing from 20 to 80. In those tests, the dimension of search space is set to 50, and the number of maximum iterations is set to 500. From the results we can see that, when the number of all particles increases, the contraction coefficient  $a$  does not greatly change, only getting better solution.

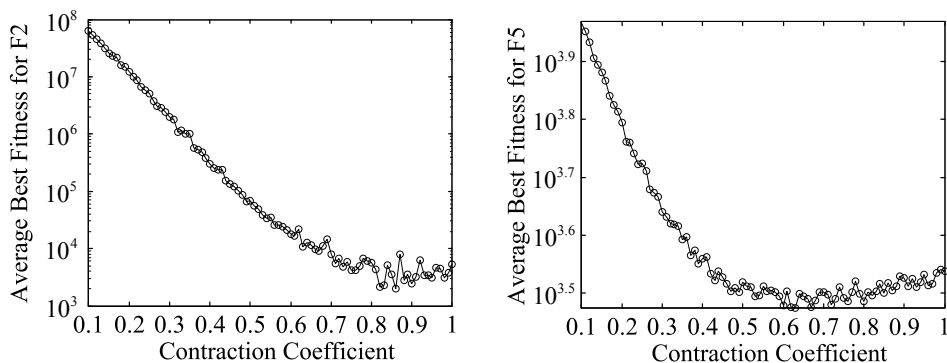


Fig. 14 The contraction coefficient  $a$  on  $F_2$  and  $F_5$  ( $n = 20$ ).

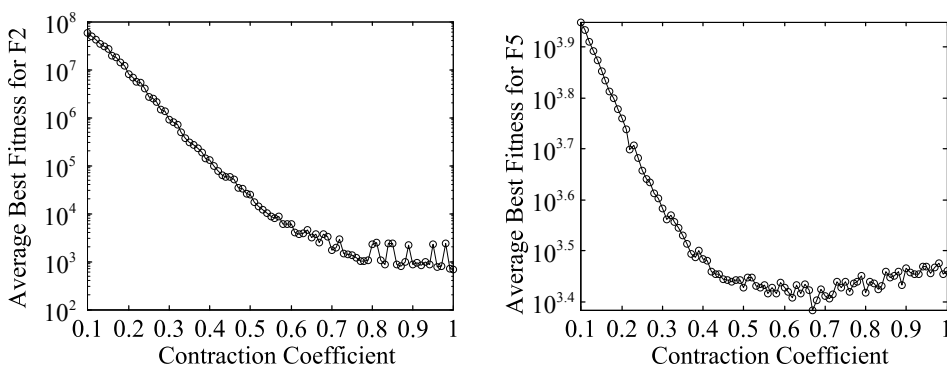


Fig. 15 The contraction coefficient  $a$  on  $F_2$  and  $F_5$  ( $n = 40$ ).

### 4.3 The comparison by different algorithms

In this subsection, some comparisons of LAQPSO, the standard QEA and QBPSO algorithms are performed on the unimodal and multimodal benchmark functions shown in Tabs. II and III to evaluate the efficiency of the proposed algorithm. The maximum iteration is 1000 for the functions, the number of all particles is set to 40. and the dimension of the functions is 30 and 50. For the unimodal functions in Tab. II, the contraction coefficient  $a$  in LAQPSO is fixed to 1. For the multimodal functions in Tab. III, the contraction coefficient  $a$  in LAQPSO is set to 0.5 and 0.65 while the dimension of those functions is 30 and 50 respectively. Besides, the absolute value of  $\Delta\theta_{ij}^t$  used in QEA and QBPSO is set as  $0.01\pi$ .

We record the best fitness value when the algorithm terminates at each run. The average best fitness (Avg. BF), the standard deviation (St. D), the maximum and minimum of the best fitness (Max. BF and Min. BF) in the last iteration are used to evaluate the quality of the solutions produced by algorithms. These statistical results are listed on Tabs. IV and V.

The results in Tab. IV show that LAQPSO performs the best and converges the fastest for the unimodal functions in Tab. II. For those functions, the results of QEA and QBPSO are far from those of LAQPSO, and especially for the function  $F_1$ ,

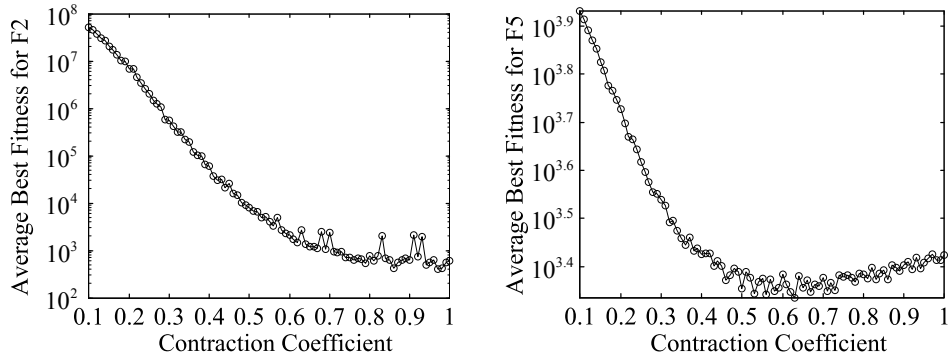


Fig. 16 The contraction coefficient  $a$  on  $F_2$  and  $F_5$  ( $n = 80$ ).

Function	$D$	Algorithm	Avg. BF	St. D	Max. BF	Min. BF
$F_1$	30	LAQPSO	<b>2.79E-04</b>	<b>1.10E-19</b>	<b>2.79E-04</b>	<b>2.79E-04</b>
		QEA	1.74E+01	9.33E+00	4.79E+01	3.14E+00
		QBPSO	2.03E+01	6.30E+00	4.00E+01	7.33E+00
	50	LAQPSO	<b>5.58E-04</b>	<b>1.67E-04</b>	<b>1.51E-03</b>	<b>4.66E-04</b>
		QEA	6.07E+02	2.23E+02	1.37E+03	2.63E+02
		QBPSO	4.35E+02	1.20E+02	6.66E+02	2.45E+02
$F_2$	30	LAQPSO	<b>5.93E+02</b>	<b>7.90E+02</b>	<b>3.51E+03</b>	<b>2.58E+01</b>
		QEA	1.79E+03	1.18E+03	4.87E+03	2.66E+02
		QBPSO	1.08E+03	8.09E+02	3.85E+03	2.08E+02
	50	LAQPSO	<b>4.96E+03</b>	<b>1.71E+04</b>	<b>7.48E+04</b>	<b>4.86E+01</b>
		QEA	1.15E+05	7.00E+04	3.33E+05	2.04E+04
		QBPSO	4.53E+04	2.65E+04	1.47E+05	1.54E+04
$F_3$	30	LAQPSO	<b>4.77E-02</b>	<b>1.86E-02</b>	<b>9.33E-02</b>	<b>1.65E-02</b>
		QEA	9.81E-02	4.96E-02	3.24E-01	3.55E-02
		QBPSO	6.46E-02	2.27E-02	1.13E-01	2.15E-02
	50	LAQPSO	<b>1.16E-01</b>	<b>4.54E-02</b>	<b>2.42E-01</b>	<b>4.24E-02</b>
		QEA	5.53E-01	1.93E-01	1.12E+00	2.25E-01
		QBPSO	3.85E-01	1.32E-01	7.70E-01	1.57E-01

Tab. IV Minimization results of the unimodal functions in Tab. II. Avg. BF denotes the average best fitness, St. D is the standard deviation, Max. BF and Min. BF stand for the maximum and minimum of the best fitness, respectively.

the maximum best fitness of LAQPSO is smaller than the minimum best fitness of QEA and QBPSO. This indicates that for this parameters configuration, QEA and QBPSO with a small rotation angle cannot converge into the suboptimal or global optimum because of the low speed of converging. In regard to LAQPSO, though the optimized contraction coefficient  $a$  is selected for  $F_1$ , the obtained results for  $F_2$  and  $F_3$  are also better. This confirms LAQPSO has the potential to achieve better results when solving unimodal functions.

Function	$D$	Algorithm	Avg. BF	St. D	Max. BF	Min. BF
$F_4$	30	LAQPSO	<b>3.72E+01</b>	<b>8.27E+00</b>	<b>5.31E+01</b>	<b>2.09E+01</b>
		QEA	5.53E+01	9.92E+00	8.52E+01	3.23E+01
		QBPSO	5.22E+01	1.17E+01	8.26E+01	2.76E+01
	50	LAQPSO	<b>8.22E+01</b>	<b>1.18E+01</b>	<b>1.10E+02</b>	<b>5.66E+01</b>
		QEA	1.17E+02	1.54E+01	1.49E+02	8.87E+01
		QBPSO	1.32E+02	1.61E+01	1.71E+02	8.84E+01
$F_5$	30	LAQPSO	<b>1.05E+03</b>	<b>2.87E+02</b>	<b>1.79E+03</b>	<b>4.36E+02</b>
		QEA	1.59E+03	3.54E+02	2.48E+03	9.22E+02
		QBPSO	1.31E+03	3.51E+02	2.19E+03	7.06E+02
	50	LAQPSO	<b>2.56E+03</b>	<b>4.86E+02</b>	<b>3.88E+03</b>	<b>1.74E+03</b>
		QEA	3.67E+03	5.58E+02	4.99E+03	2.54E+03
		QBPSO	3.34E+03	5.37E+02	4.56E+03	2.40E+03
$F_6$	30	LAQPSO	<b>5.03E-02</b>	<b>8.82E-02</b>	<b>3.89E-01</b>	<b>6.72E-04</b>
		QEA	1.20E+00	8.92E-02	1.46E+00	1.08E+00
		QBPSO	1.18E+00	5.84E-02	1.36E+00	1.07E+00
	50	LAQPSO	<b>9.71E-02</b>	<b>1.04E-01</b>	<b>3.85E-01</b>	<b>1.37E-02</b>
		QEA	6.46E+00	2.01E+00	1.33E+01	3.37E+00
		QBPSO	4.91E+00	1.08E+00	7.00E+00	3.21E+00

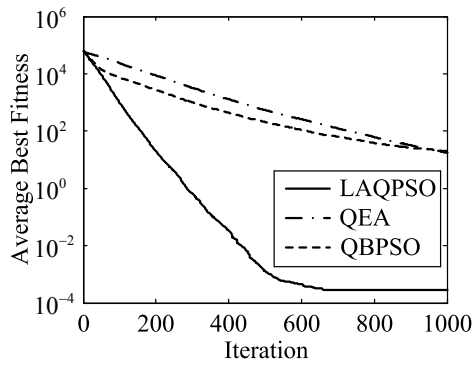
**Tab. V** Minimization results of the multimodal functions in Tab. III. Avg. BF denotes the average best fitness, St. D is the standard deviation, Max. BF and Min. BF stand for the maximum and minimum of the best fitness, respectively.

Tab. V shows the results of multimodal functions in Tab. III. The results show that LAQPSO offer the best results for the functions, while QEA and QBPSO cannot tune themselves and are trapped in local minima. The results of QBPSO is better than those of QEA except for  $F_4$ . In addition, the LAQPSO results show the small standard deviations in the results for  $F_4$  and  $F_5$ . Once again, for this parameters configuration, LAQPSO shows up greater adaptability and better performance among the algorithms.

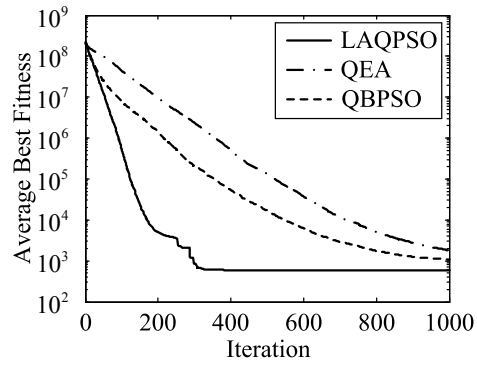
The progress of the average best fitness for function  $F_1$ ,  $F_2$  and  $F_3$  with  $D = 30$  are illustrated in Figs. 17–19 respectively. The results in these figures show that LAQPSO with a unified contraction coefficient  $a$  converges to the best solution of the unimodal functions with a relatively fast speed, but QEA and QBPSO cannot provide a good solution because of the decreasing search speed. Additionally, the convergence rate of QBPSO is faster than that of QEA at the first stage, but later on, QEA catches up QBPSO and even generates a better solution.

Figs. 20–22 show the progress of the average best fitness for the multimodal functions  $F_4$ ,  $F_5$  and  $F_6$  with  $D = 30$ . The results show that QBPSO converges rapidly than LAQPSO and QEA at the early stage of running, but falls into a local minimum for the functions because of premature convergence. The convergence rate of QEA is always slowest, and the premature convergence cannot avoided either. In contrast, though LAPSO does not converges fastest at the early stage, it tunes itself to jump out of local minima to provide the best solution. Fig. 22 obviously shows that LAQPSO has fallen into a local minimum for function at the

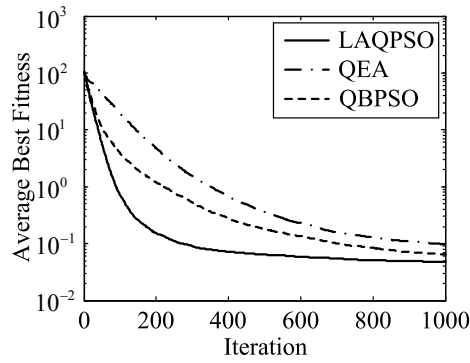




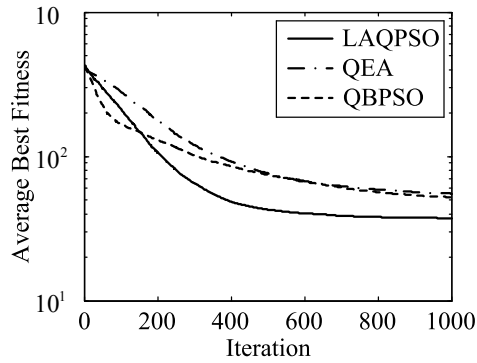
**Fig. 17** Convergence performance of LAQPSO, QEA and QBPSO on  $F_1$  with  $D = 30$ .



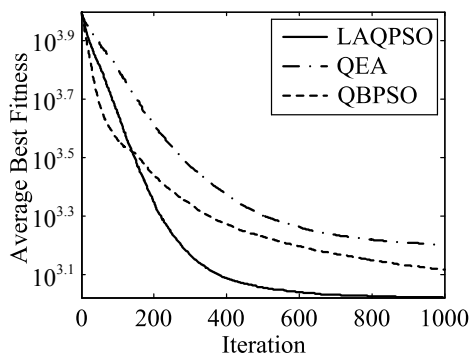
**Fig. 18** Convergence performance of LAQPSO, QEA and QBPSO on  $F_2$  with  $D = 30$ .



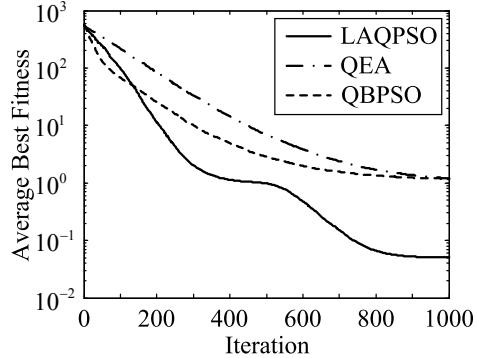
**Fig. 19** Convergence performance of LAQPSO, QEA and QBPSO on  $F_3$  with  $D = 30$ .



**Fig. 20** Convergence performance of LAQPSO, QEA and QBPSO on  $F_4$  with  $D = 30$ .



**Fig. 21** Convergence performance of LAQPSO, QEA and QBPSO on  $F_5$  with  $D = 30$ .



**Fig. 22** Convergence performance of LAQPSO, QEA and QBPSO on  $F_6$  with  $D = 30$ .

beginning of evolutionary process, but then jumps out of it and launches global search. It shows that LAQPSO can avoid premature convergence more effectively.

## 5. Conclusions

This paper proposes a novel QPSO evolved on a new local attractor. The proposed LAQPSO introduces a convenient quantum bit expression mechanism called quantum angle to represent the particle, and employs the local attractor to determine the direction and magnitude of rotation angle in quantum rotation gate. In LAQPSO, the rotation angle can be adapted to the search progress automatically, and thereby balances between exploration ability and exploitation ability. It is shown by comparing the present results with these obtained by QEA and QBPSO algorithms that, LAQPSO is an effective approach that has the potential to achieve better solutions in most cases. For exploiting more potential of LAQPSO, a comprehensive parameter study regarding the relationship between the contraction coefficient  $a$  and many other factors is performed. Through the obtained results, it is concluded that

1. when the proposed LAQPSO is used to deal with the objectives having few local optimum, assigning a larger value to the contraction coefficient  $a$  is the effective way to enhance the exploitation ability. On the contrary, if many local optimums appear in the objective, it is suggested to balance the tradeoff between exploitation ability and exploration ability by reducing the contraction coefficient properly;
2. in the case of the low dimension optimization problem, the smaller contraction coefficient  $a$  can produce good results. With the dimension increases, the contraction coefficient should be increased to extend the diversity of the LAQPSO algorithm;
3. owing to the adopting of the new local attractor, the proposed LAQPSO can adapt itself to effectively avoid premature convergence and provides good results.

This work can provide some novel ideas for improving both the particle swarm optimization algorithm and quantum-inspired evolutionary algorithm. Based on the new local attractor of a basic element, the other optimization algorithms can also be improved, and their performance can be thus further investigated.

## Acknowledgement

This work is supported by the National Natural Science Foundation of China (Grant No. 51175099).

## References

- [1] BENIOFF P. The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines. *Journal of Statistical Physics*. 1980, 22(5), pp. 563–591, doi: 10.1007/bf01011339.

- [2] CHUNG C.Y., YU H., WONG K.P. An advanced quantum-inspired evolutionary algorithm for unit commitment. *IEEE Transactions on Power Systems*. 2011, 26(2), pp. 847–854, doi: 10.1109/TPWRS.2010.2059716.
- [3] CLERC M., KENNEDY J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*. 2002, 6(1), pp. 58–73, doi: 10.1109/4235.985692.
- [4] GEPP A., STOCKS P. A review of procedures to evolve quantum algorithms. *Genetic programming and evolvable machines*. 2009, 10(2), pp. 181–228, doi: 10.1007/s10710-009-9080-7.
- [5] GROVER L.K. A fast quantum mechanical algorithm for database search. In: *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, Philadelphia, USA. Pennsylvania: ACM, 1996, pp. 212–219, doi: 10.1145/237814.237866.
- [6] HAN K.H., KIM J.H. Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE Transactions on Evolutionary Computation*. 2002, 6(6), pp. 580–593, doi: 10.1109/tevc.2002.804320.
- [7] HAN K.H., KIM J.H. Quantum-inspired evolutionary algorithms with a new termination criterion,  $H^2$  gate, and two-phase scheme. *IEEE Transactions on Evolutionary Computation*. 2004, 8(2), pp. 156–169, doi: 10.1109/TEVC.2004.823467.
- [8] JEONG Y.W., PARK J.B., JANG S.H., LEE K.Y. A new quantum-inspired binary PSO: application to unit commitment problems for power systems. *IEEE Transactions on Power Systems*. 2010, 25(3), pp. 1486–1495, doi: 10.1109/TPWRS.2010.2042472.
- [9] KENNEDY J., EBERHART R. Particle swarm optimization. In: *Proceedings of 1995 IEEE International Conference on Neural Networks*, Perth, Australia. WA: IEEE, 1995, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.
- [10] LI P., LI S. Quantum-inspired evolutionary algorithm for continuous space optimization based on Bloch coordinates of qubits. *Neurocomputing*. 2008, 72(1), pp. 581–591, doi: 10.1016/j.neucom.2007.11.017.
- [11] LI P., WANG H. Quantum ant colony optimization algorithm based on bloch spherical search. *Neural Network World*. 2012, 22(4), pp. 325–341, doi: 10.14311/NNW.2012.22.019.
- [12] MENG K., WANG H.G., DONG Z.Y., WONG K.P. Quantum-inspired particle swarm optimization for valve-point economic load dispatch. *IEEE Transactions on Power Systems*. 2010, 25(1), pp. 215–222, doi: 10.1109/TPWRS.2009.2030359.
- [13] NETO J.X.V., BERNERT D.L.A., COELHO L.S. Improved quantum-inspired evolutionary algorithm with diversity information applied to economic dispatch problem with prohibited operating zones. *Energy Conversion and Management*. 2011, 52(1), pp. 8–14, doi: 10.1016/j.enconman.2010.05.023.
- [14] NICOLAU A.S., SCHIRRU R., ALVARENGA de MOURA MENESES A. Quantum evolutionary algorithm applied to transient identification of a nuclear power plant. *Progress in Nuclear Energy*. 2011, 53(1), pp. 86–91, doi: 10.1016/j.pnucene.2010.08.004.
- [15] SHOR P.W. Algorithms for quantum computation: discrete logarithms and factoring. In: *Proceedings of the 35th Annual Symposium on Foundations of Computer Science (FOCS 1994)*, Santa Fe, USA. NM: IEEE, 1994, pp. 124–134, doi: 10.1109/SFCS.1994.365700.
- [16] SUN J., FENG B., XU W. Particle swarm optimization with particles having quantum behavior. In: *Congress on Evolutionary Computation*, Portland, USA. Oregon: IEEE, 2004, pp. 325–331, doi: 10.1109/CEC.2004.1330875.
- [17] SUN J., XU W., FANG W., CHAI, Z. Quantum-behaved particle swarm optimization with binary encoding. *Adaptive and Natural Computing Algorithms, LNCS*. Berlin: Springer, 2007, pp. 376–385, doi: 10.1007/978-3-540-71618-1\_42.
- [18] SVÍTEK M. Wave probabilistic models. *Neural network World*. 2007, 17(5), pp. 469–481.
- [19] SVÍTEK M. Wave probabilities and quantum entanglement. *Neural Network World*. 2008, 18(5), pp. 401–406

- [20] van den BERGH F., ENGELBRECHT A.P. A new locally convergent particle swarm optimizer. In: *Proceedings of the IEEE international conference on systems, man, and cybernetics, Yasmine Hammamet, Tunisia*: IEEE, 2002, pp. 6–9, doi: 10.1109/ICSMC.2002.1176018.
- [21] WANG Y., FENG X.Y., HUANG Y.X., PU D.B., ZHOU W.G., LIANG Y.C., ZHOU C.G. A novel quantum swarm evolutionary algorithm and its applications. *Neurocomputing*. 2007, 70(4), pp. 633–640, doi: 10.1016/j.neucom.2006.10.001.