Journal of Computers Vol. 31 No. 3, 2020, pp. 58-71
doi:10.3966/199115992020063103005

Hybrid Quantum Particle Swarm Algorithm Based on Lévy Flights

Qi-Wen Zhang*, Song-Qi Hu

School of Computer and Communication, Lanzhou University of Technology University,
Lanzhou, 730050, P. R. China
tdyy2010@126.com, xiaohusq@163.com

Received 4 October 2018; Revised 19 March 2019; Accepted 26 April 2019

Abstract. In order to diversify the particle swarm during the searching process of quantum particle swarm optimization (QPSO) and avoid the algorithm being trapped into premature easily, a hybrid quantum particle swarm optimization algorithm based on Lévy flights is proposed in this paper. The new algorithm effectively takes advantage of quantum computing and Lévy flights. We use the probability amplitude encoding method of the quantum bit to initialize the particle position and combine the potential well particle updating formula with the quantum rotation gate to update the particle swarm, which effectively ameliorates the search process and increases the population diversity. Then the Lévy flights strategy is employed to improve the population variation process and enhance the quality of the solution while preventing the algorithm from falling into the precocious convergence. Compared with other algorithms on benchmark functions, it is shown that the algorithm is effective and feasible.

Keywords: global convergence, Lévy flights, population diversity, quantum computation, quantum particle swarm algorithm

1 Introduction

With the rapid development of quantum technology, quantum computing has drawn much attention of researchers. Since Narayanan et al. [1] proposed a quantum-derived genetic algorithm in 1996, the quantum intelligent algorithm has become an important research field, which combines the search ability of the swarm intelligence algorithm with the computing power of quantum computing and effectively improves the common problems of the swarm intelligence algorithm, including the decline of population diversity in the later search stage, easy to fall into premature convergence, etc. By learning the existing quantum group intelligence algorithms, we find that there are two kinds of quantization methods to improve the swarm intelligence algorithm. One way is to introduce the concept of quantum potential well and put the Schrodinger equation into the population movement formula. Sun et al. [2] proposed a classical quantum particle swarm optimization (QPSO) algorithm by combining the analysis of particle trajectory with quantum mechanism and obtained the quantum potential well centered on a local attractor to represent the particle position. Guo et al. [3] constructed the quantum artificial fish swarm algorithm model by using the delta potential well, the one-dimensional harmonic oscillator and the square potential well. Liu et al. [4] proposed a quantum particle swarm optimization algorithm based on asymmetric potential well to improve the invalid search which is caused by ignoring the definition domain scope of the delta potential well. The other is to introduce the concept of quantum gate and complete population migration through quantum gates. Ma et al. [5] used quantum rotation gate to realize the search mode and tracking mode in the cat swarm algorithm and proposed an improved quantum cat swarm optimization. Li et al. [6] used the qubit to encode individual and proposed a quantum derived cuckoo algorithm. The potential-well improvement method effectively improves the vitality of the algorithm in the search process, and has a positive effect on avoiding the algorithm from falling into local extremum. And the

* Corresponding Author

quantum-gate improvement method enlarges the searching range of the population and improves the diversity of the algorithm under the condition that the number of the population remains unchanged.

Although quantum particle swarm optimization (QPSO) [2] can solve the combinatorial optimization problem and improve the global convergence of particle swarm optimization (PSO) effectively, it is still easy to fall into local extremum in the later period of search process. To this end, the improved algorithm of quantum particle swarm optimization has been enriched continuously, the improvement is roughly divided into the following two trends.

(1) Improve the updating formula, including replacing the distribution of local attractors in the original formula with a new probability distribution [7], coefficient improvement [8], Improved algorithm model [4] and so on.

(2) Developing hybrid algorithms by mixing with other algorithms, e.g. [9] combines niche strategies, [10] uses the variation formula of the vocal velocity in the bat algorithm to improving the control parameters, [11] combines the krill algorithm with the quantum particle swarm algorithm without any additional operators to shares the information position fully.

Based on the above review of the quantization method and the improved trends of quantum particle swarm optimization algorithm, we propose a new algorithm by taking the advantages of the two quantum improvement methods, which is completely different from the general improvement methods. Instead of modifying the update formula of the algorithm, or mixing with other new algorithms, an overall improvement is made. The main contributions of this paper are as follows:

Inspired by these ideas, a hybrid quantum particle swarm optimization algorithm based on Lévy flights is proposed in this paper. The technical achievements of this paper can be summarized as follows.

(1) The concept of quantum gate is combined with QPSO to achieve the quantization of QPSO. Integrate the known update formula of QPSO into the quantum revolving gate to complete the particle update.

(2) In order to improve the diversity of algorithms, mutation process is added after each iteration, and mutation is carried out through NOT-gate.

(3) And we use Lévy flight mechanism to improve the mutation process, because the population renewal process may be destroyed by using NOT-gate mutation.

(4) Experimental results simulated on benchmark functions demonstrate the effectiveness and superiority of the proposed algorithm.

The rest of this paper is organized as follows. Section 2 describes the basic QPSO and reviews the basic theory related to optimization method. Then the detail of the improved algorithm is shown in section 3. Finally, experimental results and conclusions are made in section 4 and section 5, respectively.

2 Preliminaries

2.1 The Basic QPSO

Kennedy [12] found that the particle trajectories conform to the periodic sinusoidal wave and the particle trajectories can be ensured to converge to the local attractor. Sun et al. combined this conclusion with quantum mechanism, and proposed the quantum particle swarm optimization (QPSO) algorithm in 2004. The particles in QPSO converge to the local attractor, while the current position, the individual optimal, the global optimal and the local attractor converge to a point, which ensures the convergence of the algorithm.

In QPSO, the state of the particle is described by the wave function $\Psi(X,t)$. Assuming that the particle in the quantum particle group is not rotated, then the particle state depends on the wave function and is only related to the position. For the wave function does not depend on time, the particle state is consistent with the stationary Schrödinger equation: $-\frac{\hbar^2}{2m}\nabla^2\Psi + U\Psi = E\Psi$.

When the particle space is a delta potential well and the center of the potential well is the local attractor. Bring the formula of delta potential well $V(X) = -\gamma\delta(x-p) = -\gamma\delta(Y)$ into the stationary Schrödinger equation:

$$\frac{\partial^2 \Psi}{\partial Y^2} + \frac{2m}{\hbar^2}[E + \gamma\delta(Y)]\Psi = 0 . \quad \textbf{(1)}$$

In a potential well, the farther the distance between a particle and a potential well p is, the closer the wave function Ψ near to 0, the smaller the probability of the presence of the particle is. The nearer the particle distance to the center of the potential well p, the kinetic energy E will tend to 0, so that the particle cannot escape. In the delta potential well, the wave function can be expressed as in equation (2):

$$\Psi(Y) = \frac{1}{\sqrt{L}} e^{\frac{-|y|}{L}}, L = \frac{\hbar^2}{m\gamma} . \quad \textbf{(2)}$$

Through the Monte Carlo method, we can get the following:

$$X = p \pm \frac{L}{2}\ln(\frac{1}{u}), \quad \textbf{(3)}$$

where $u \sim U(0,1)$, and L is the length of potential well, $L = \left|x_{ij}^t - P_{ij}^t\right|$, P_{ij}^t is optimal position of individual.

Extending one dimension to *m* dimensional Hilbert space, so that each dimension is confined to the delta potential well and updated independently. The iterative formula of the quantum particle swarm is as follows:

$$x_{ij}^{t+1} = p_{ij}^t \pm \left|x_{ij}^t - P_{ij}^t\right| \ln(\frac{1}{u}), \quad \textbf{(4)}$$

where p_{ij}^t is the local attractor, P_{ij}^t is optimal position of individual, $u \sim U(0,1)$. The definition of the local attractor p_{ij}^t is as follows:

$$p_{ij}^t = \varphi_{ij}^t P_{ij}^t + (1 - \varphi_{ij}^t)G_j^t , \quad \textbf{(5)}$$

where G_j^t is the global optimal position, P_{ij}^t is the optimal position of individual and φ_{ij}^t is random numbers uniformly drawn from [0, 1].

In [13], it is proved that the algorithm dose not diverge only when the control parameter $\alpha \le e^{\gamma} \approx 1.781$ through the analysis of individual behavior and parameter selection experiments. So the basic quantum behaved particle swarm optimization still suffers from premature convergence and falls into the local extremum easily. In this paper, an improved hybrid quantum particle swarm optimization algorithm is proposed by combining quantum computing with quantum particle swarm optimization with Lévy flights. The combination of algorithm and quantum computation is helpful to further explore the solution space and diversify the swarm population. And the improvement of variation process by Lévy flights can enhance the mutation effect and improve solution quality.

2.2 Quantum Computation

Quantum intelligence computing makes use of the principles and concepts of quantum theory, and effectively combines the advantages of quantum computing and traditional intelligent algorithms, which opens a new way for the research of intelligent computing. Quantum computing takes advantage of the distinct properties of quantum: superposition, coherence, entanglement and parallelism, and the four hypotheses: state space hypothesis, Schrödinger equation hypothesis, quantum measurement hypothesis and compound system hypothesis, establish the premise of quantum computing.

In classical computation, information is encoded as a bit chain, and the state is represented by "0"or "1". While quantum mechanics uses qubit, and the symbol is "$|\ \rangle$". Single qubit can represent eigenstates $|0\rangle$, $|1\rangle$ and quantum superposition states (neither $|0\rangle$ nor $|1\rangle$). The superposition state of a qubit can be described by two-dimensional Hilbert space: $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$, which means that when the qubit is measured, it will collapse to $|0\rangle$ with the probability of $|\alpha|^2$ and collapse to $|1\rangle$ with the probability of $|\beta|^2$. The normalization condition is $|\alpha|^2 + |\beta|^2 = 1$ and the quantum state can be expressed by $|\Psi\rangle = [\alpha, \beta]^T$.

According to the superposition properties and hypothesis of quantum, quantum computation is realized by the transformation of quantum gates. And on the basic of the normalization condition of probability, the quantum gate transformation matrix must be invertible unitary matrix, which satisfies $U^{+}U = UU^{+}$ (U^{+} represents conjugate transpose of U). The change of qubit is carried out through the single bit quantum rotation gate:

$$\begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix}. \tag{6}$$

The gate has good unitary property, while changing the phase, the length of the qubit will not change.

$$\begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix} = \begin{bmatrix} \cos(\theta + \Delta\theta) \\ \sin(\theta + \Delta\theta) \end{bmatrix}. \tag{7}$$

2.3 Lévy Flights

A Lévy flight is a hot spot in the field of anomalous diffusion, the behavior of using the Lévy flights model includes the propagation of visible light in the chaotic optical medium and the abnormally transport of single molecules in living cells. As a class of stochastic processes with Markov properties characterized by long-range jumps, Lévy flights are of great significance to improve swarm intelligence computation.

The leaping length of Lévy flights satisfies the Lévy distribution which conforms long tail progressive form: $p(x) \sim 1/|x|^{\mu+1}$, $0 < \mu < 2$. The Lévy distribution can be simply defined as:

$$L(s,\gamma,\mu) = \begin{cases} \sqrt{(\frac{\gamma}{2\pi})} \exp[-\frac{\gamma}{2(s-\mu)}] \frac{1}{(s-\mu)^{3/2}} & ,0 < \mu < a < \infty \\ 0 & ,\ otherwise \end{cases},$$

where $\mu > 0$ is a minimum step and γ is a scale parameter.

Lévy flights are defined according to Fourier transform. Because it is hard to solve the inverse of Lévy flights' integral, there is no analytic form except special case. One of the most efficient and straightforward way is to use the so-called Mantegna algorithm. Under the idea of Mantegna algorithm, referencing the definition of Lévy step in pollination algorithm [14], the step length s can be calculated by

$$s = \frac{u}{|v|^{1/\beta}}, \tag{8}$$

where u and v are drawn from normal distributions. That is, $u \sim N(0, \sigma_u^2)$, $v \sim N(0,1)$, where $\sigma_u^2 = \left\{ \frac{\Gamma(1+\beta)\sin(\pi\beta/2)}{\Gamma[(1+\beta)/2]\beta^{2^{(\beta-1)/2}}} \right\}^{\frac{1}{\beta}}$, $\beta = 3/2$, $\Gamma(z)$ is the Gamma function, $\Gamma(z) = (z-1)!$.

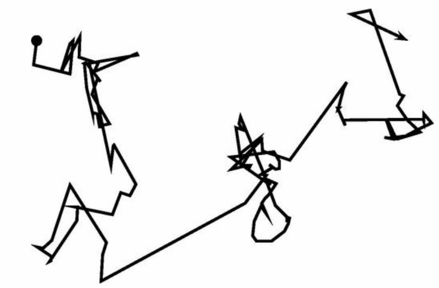

Fig. 1. Lévy flights with β=1 [14]

[15] points out that Lévy flights can maximize the efficiency of resource searches in uncertain environments. The reasons are as follows: (a) The variance of Lévy flights $\sigma^2(t) \sim t^{3-\beta}$, $1 \le \beta \le 2$ increases much faster than the linear relationship of Brownian random walks; (b) From the implementation point of view, the generation of random numbers with Lévy flights consists of two steps: the choice of a random direction and the generation of steps that obey the chosen Lévy distribution. A symmetric Lévy stable distribution can be produced through the Mantegna algorithm, here symmetric means that the steps can be positive and negative. And the power law behavior of Lévy flights over large step (heavy tail) can initiate the exploratory behavior at any stage of convergence, which makes the algorithm escape from local extremum effectively.

3 Hybrid QPSO Algorithm Based on Lévy Flights

3.1 Population Initialization

The position of the particle is expressed by quantum superposition state which is represented by the probability amplitude. Assume the solution space dimension is m, the swarm size is n, and the initialization state of the particle i is formulated as follows:

$$P_i = \begin{bmatrix} \begin{vmatrix} \sin(\theta_{i1}) \\ \cos(\theta_{i1}) \end{vmatrix} \begin{vmatrix} \sin(\theta_{i2}) \\ \cos(\theta_{i2}) \end{vmatrix} \cdots \begin{vmatrix} \sin(\theta_{im}) \\ \cos(\theta_{im}) \end{vmatrix} \end{bmatrix}, \quad (9)$$

where θ_{ij} is qubit phase, $\theta_{ij} = 2\pi \times rand()$, $i = (1,2,3,\ldots,n)$, $j = (1,2,3,\ldots,m)$, normalization condition: $\sin^2\theta + \cos^2\theta = 1$, is workable.

The individual best position of particle i and the global best position are represented by P_{il} and P_g, respectively.

$$P_{il} = (\cos(\theta_{il_1}), \cos(\theta_{il_2}), \cdots \cos(\theta_{il_m})), \quad (10)$$

$$P_g = (\cos(\theta_{g_1}), \cos(\theta_{g_2}), \cdots, \cos(\theta_{g_m})). \quad (11)$$

The encoding mechanism has the following advantages: (a) Each particle occupies two positions, corresponding to the probability amplitude of $|0\rangle$ and $|1\rangle$, respectively, so that the algorithm doubles the search space when the number of particles is constant. (b) The state of the particles can be updated with a phase rotation operation, which can effectively expand the search space and speed up the searching operation. (c) Trigonometric functions are used to represent the probability amplitude of particles, which can efficiently avoid the algorithm from falling into local optimum due to boundary aggregation.

3.2 Solution Space Transformation

The probability amplitude of particles is expressed by trigonometric function, and its value range is [-1, 1]. In order to calculate the location of particles in the solution space, the formula of solution space transformation is given in (12).

$$X_{ij}^{t} = \frac{1}{2}[b(1+\alpha_{ij}^{t}) + a(1-\alpha_{ij}^{t})]. \quad \textbf{(12)}$$

The formula derivation is as follows:

Establish a correspondence $f(\alpha_{ij}^{t})$ from $[-1,1]$ to $[a,b]$,

$f(\alpha_{ij}^{t}) = p\alpha_{ij}^{t} + q$, $f(-1) = a$ when $\alpha_{ij}^{t} = -1$ and $f(1) = b$ when $\alpha_{ij}^{t} = 1$;

Then $q - p = a$, $q + p = b$.

The mapping is as follows:

$$f(\alpha_{ij}^{t}) = \frac{1}{2}[b(1+\alpha_{ij}^{t}) + a(1-\alpha_{ij}^{t})]$$

The one-to-one mapping method from $[-1,1]^{N}$ to $[a,b]^{N}$ can be adapted to the optimization problems of various scale spaces.

3.3 Particle Swarm Updating Rules

The fusion of population update formula of quantum particle swarm optimization algorithm and quantum gate is one of the key research issues in this paper.

In quantum computation, the transformation between quantum states is realized through quantum gates, and the essence of quantum revolving gates is to change the magnitude of the angles. The updating rule of the particle is:

(a) Drawing lessons from (4) to control the quantum rotation gate size:

$$\Delta\theta_{ij}^{t+1} = \pm\alpha\ln(\frac{1}{u})\left|\Delta\theta_{pij}\right|, \Delta\theta_{pij}\begin{cases} 2\pi + \theta_{ij}^{t} - \theta_{pij}^{t} & (\theta_{ij}^{t} - \theta_{pij}^{t} < -\pi) \\ \theta_{ij}^{t} - \theta_{pij}^{t} & (-\pi \le \theta_{ij}^{t} - \theta_{pij}^{t} \le \pi) \\ \theta_{ij}^{t} - \theta_{pij}^{t} - 2\pi & (\theta_{ij}^{t} - \theta_{pij}^{t} > \pi) \end{cases}, \quad \textbf{(13)}$$

where α is a control parameter, $u \sim U(0,1)$ and θ_{pij}^{t} is the corresponding angle of optimal position of individual.

In this paper, the magnitude of the quantum rotation gate is dynamically adjusted in the iterative process. In (13), parameter $\Delta\theta_{ij}^{t+1}$ depends on the size of the $\Delta\theta_{pij}$, that is, the distance between the current position and the optimal position of individual. Large distance can expand the search space and increase the convergence speed, while small distance narrows the search space and the search accuracy will be improved. At the same time, the direction of the rotation angle is determined by u : when $u > 0.5$, the change is counterclockwise ($\Delta\theta_{ij}^{t+1}$ is positive), when $u \le 0.5$, the change is anticlockwise ($\Delta\theta_{ij}^{t+1}$ is negative). This process makes the corners trend to the known individual optimality in the algorithm search process, and increases the diversity of the algorithm due to the randomness of the direction.

(b) Using the quantum rotation gate to update qubit probability amplitude:

$$\begin{bmatrix} \cos(\theta_{ij}^{t+1}) \\ \sin(\theta_{ij}^{t+1}) \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta_{ij}^{t+1}) & -\sin(\Delta\theta_{ij}^{t+1}) \\ \sin(\Delta\theta_{ij}^{t+1}) & \cos(\Delta\theta_{ij}^{t+1}) \end{bmatrix} \begin{bmatrix} \cos(\beta_{ij}^{t}) \\ \sin(\beta_{ij}^{t}) \end{bmatrix} = \begin{bmatrix} \cos(\beta_{ij}^{t} + \Delta\theta_{ij}^{t+1}) \\ \sin(\beta_{ij}^{t} + \Delta\theta_{ij}^{t+1}) \end{bmatrix}, \quad \textbf{(14)}$$

where $\beta_{ij}^{t+1} = \varphi\theta_{pij}^{t} + (1-\varphi)\theta_{g}^{t}$ is the angle of the local attractor.

In this part, the particle trajectory analysis results are introduced to adjust the rotation angle. To some extent, the transformation of the rotation angle is taken as an optimization process. And [16] proved that the convergence point is a weighted average of the personal best position and the global best position. So the calculation formula of β_{ij}^{t} refers to (5), which represents the final point at which particle motion converges.

Firstly, the quantum rotation gate can update the two positions at the same time by changing the quantum phase of the particle, which improves the computational efficiency of the algorithm. The updated position of particle i is as follows:

$$P_{ic}^{t+1} = (\cos(\beta_{i1}^{t} + \Delta\theta_{i1}^{t+1}), \cos(\beta_{i2}^{t} + \Delta\theta_{i2}^{t+1}), \cdots, \cos(\beta_{im}^{t} + \Delta\theta_{im}^{t+1})) , \tag{15}$$

$$P_{is}^{t+1} = (\sin(\beta_{i1}^{t} + \Delta\theta_{i1}^{t+1}), \sin(\beta_{i2}^{t} + \Delta\theta_{i2}^{t+1}), \cdots, \sin(\beta_{im}^{t} + \Delta\theta_{im}^{t+1})) . \tag{16}$$

Secondly, according to the trajectory analysis of the particle swarm algorithm, the particles will eventually converge to the local attractors. We apply the quantum revolving gate to the local attractor phase (determined by the global optimal phase and the individual optimal phase) instead of the phase of the previous iteration to prevent particles from moving too fast to local optimum and enhance algorithm diversity and randomness.

The control parameter α changes dynamically with a linear reduction strategy based on the number of iterations. According to [17], most functions can get a better optimization result in the range of $[0.5,1]$.

$$\alpha = 0.5 + \frac{(1-0.5)(t_{\max} - t)}{t_{\max}} , \tag{17}$$

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

3.4 Population Variation Based on Lévy Flights

The process of population variation usually uses NOT gate to realize inversion between quantum states. As a single quantum logic gate, NOT gate can be applied to single quantum: $N|0\rangle = |1\rangle$, $N|1\rangle = |0\rangle$. To prevent population from falling into local optimum, the operation of NOT gate mutation is as follows:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \cos(\theta_{ij}) \\ \sin(\theta_{ij}) \end{bmatrix} = \begin{bmatrix} \sin(\theta_{ij}) \\ \cos(\theta_{ij}) \end{bmatrix} = \begin{bmatrix} \cos(\frac{\pi}{2} - \theta_{ij}) \\ \sin(\frac{\pi}{2} - \theta_{ij}) \end{bmatrix} .$$

The use of the NOT gate can be regarded as the subtraction of the phase and $\pi/2$, which is equivalent to the exchange of two probability amplitudes, that is to say, two positions can be changed at the same time by one mutation. The NOT gate is easy to destroy the optimization process and lose the optimal solution while it is improving the effect of the population variation. For this reason, Lévy flights are adopted to improve the mutation process and the angle size is as follows:

$$\Delta\theta_{ij} = \gamma \cdot s_j \times \frac{\pi}{2}, \gamma = 0.01 , \tag{18}$$

where s_j is controlled by (8) and γ according to [18].

According to the mutation probability P_m, one or more variant objects are randomly generated, and the variation process can be expressed as:

$$\begin{bmatrix} \cos(\theta_{ij}) \\ \sin(\theta_{ij}) \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta_{ij}) & -\sin(\Delta\theta_{ij}) \\ \sin(\Delta\theta_{ij}) & \cos(\Delta\theta_{ij}) \end{bmatrix} \begin{bmatrix} \cos(\theta_{ij}) \\ \sin(\theta_{ij}) \end{bmatrix} = \begin{bmatrix} \cos(\theta_{ij} + \Delta\theta_{ij}) \\ \sin(\theta_{ij} + \Delta\theta_{ij}) \end{bmatrix} . \tag{19}$$

There are two advantages of the improvement. On the one hand, the process of variation is independent and not affected by factors such as local optimality. It can effectively break through precocious convergence and improve operational efficiency and particle diversity. On the other hand, Lévy flights have a walking pattern, which alternates short distance walks with small step and occasionally long distance walks with large step. This approach allows larger jumps to separate aggregation generated by smaller jumps, which increases the diversity of the population, expands the search range and plays a positive role in jumping out of local extremum. And smaller jumps will not damage the optimization process too much while increasing population diversity.

3.5 Framework of the Proposed Algorithm

Step1: Initialize the swarm population according to (9), the optimal position of individual and the global optimal position. Setting the maximum number of iterations, the solution space range [a, b] and other parameters.

Step2: Carry out the space transformation according to the (12).

Step3: Calculate the fitness value of each particle.

Step4: If the particle's current position is better than the individual's historical optimal position, update P_{il} ; if it is better than the currently global optimal value, update P_g .

Step5: Update the control parameters according to (17) and the particle state using (13) and (14).

Step6: In order to ensure the diversity of particles and prevent falling into local optimum, mutation is carried out according to a certain probability $P_m = 0.05$, which is consistent with [19]. The rotation angle provided by (18) and using the (19) to complete the population variation.

Step7: Check the stop condition, if satisfied, output the result, otherwise, return to Step 2 to continue.

The hybrid quantum particle swarm algorithm combines quantum computing with quantum particle swarm optimization. It has the following advantages. (a) The potential well transformation formula is incorporated into the revolving door as a new search strategy, replacing the original mechanism with a probability distribution mechanism, enhancing the ability of global search, and improving the accuracy and speed of the algorithm convergence. (b) Using the parallel nature formed by the superposition and entanglement of quantum, the search scope of the variable is extended by the encoding method, and the search process is accelerated. (c) In a quantum computer, the classical calculation of a quantum state is equivalent to the calculation of each superposition component of the quantum state simultaneously. Calculations in this algorithm are realized by the unitary transformation, which conforms to the Schrödinger equation hypothesis in the quantum theoretical hypothesis. It is equivalent to the simultaneous completion of several classical computations and superpose them by a certain probability, which speed the calculation. Using Lévy flights to improve variation process can help the algorithm to increase the diversity of the population and avoid falling into the local optimum. At the same time, the smaller rotation amplitude can reduce the damage to the potential optimal solution during the mutation process.

4 Experimental Setting and Results

In order to verify the performace of the proposed algorithm, 12 benchmark functions provided by CEC2005 (http://www.ntu.edu.sg/home/epnsugan/), as shown in Table 1 are tested in the experiments. The swarm size is set to $n = 50$, the number of iterations $G = 1000$. In the case of the dimension $m = 2,10,20$, the algorithms are independently run on the benchmark function 25 times. The best value, the worst value, the average value and the variance are recorded as the comparative evaluation indexes. Table 2 shows the comparison results of hybrid quantum particle swarm optimization algorithm using quantum NOT gate (QPSOIII) and hybrid quantum particle swarm optimization algorithm based on Lévy flights (QPSOIII-L). The experimental results of the proposed algorithms (QPSOIII-L), basic QPSO (QPSOI) and the algorithm in [19] (QPSOII) are given in Table 3. In Table 2 and Table 3, the smaller the average value is, the better the search effect is. If the average value is the same, the smaller the variance is, the better the search effect is. The optimal case is represented by bold.

Table 2 shows the result of the hybrid quantum particle swarm optimization algorithm using the NOT gate (QPSOIII) and the improved gate with Lévy flights (QPSOIII -L) during the mutation process. The algorithm QPSOIII -L can get a better solution in each dimension of the function except the function F_3 , F_5 , F_8 , F_{10} , so the algorithm works well in most cases. In F_3 , the QPSOIII-L shows better results than QPSOIII in dimension 10 and dimension 20, and in F_{10} , the QPSOIII-L shows better results than QPSOIII in dimension 20. So to some extent, the QPSOIII-L will fully exploit its advantages as the problem dimension increases. As for F_5 and F_8 , the results of QPSOIII-L are inferior to QPSOIII. It can be seen that the improvement of Lévy flights is not applicable to all situations, and its optimization ability is limited.

Table 1. Benchmark functions

	Expression	Range	f_bias
F_1	$F_1(x)=\sum_{i=1}^{D} z_i^2 + f\_bias, z = x-o, x=[x_1,x_2,\cdots,x_D]$	[-100,100]	-450
F_2	$F_2(x)=\sum_{i=1}^{D}(\sum_{j=1}^{i} z_j)^2 + f\_bias, z = x-o, x=[x_1,x_2,\cdots,x_D]$	[-100,100]	-450
F_3	$F_3(x)=\sum_{i=1}^{D}(10^6)^{\frac{i-1}{D-1}} z_i^2 + f\_bias, z=(x-o)*M, x=[x_1,x_2,\cdots,x_D]$	[-100,100]	-450
F_4	$F_4(x)=(\sum_{i=1}^{D}(\sum_{j=1}^{i} z_j^2))*(1+0.4\lvert N(0,1)\rvert)+f\_bias, z=x-o, x=[x_1,x_2,\cdots,x_D]$	[-100,100]	-450
F_5	$F_5(x)=\max\{\lvert A_i x - B\rvert\}+f\_bias, i=1,2,...,D, x=[x_1,x_2,\cdots,x_D]$	[-100,100]	-310
F_6	$F_6(x)=\sum_{i=1}^{D-1}(100(z_i^2-z_{i+1})^2+(z_i-1)^2)+f\_bias, z=x-o, x=[x_1,x_2,\cdots,x_D]$	[-100,100]	390
F_7	$F_7(x)=\sum_{i=1}^{D}\frac{z_i^2}{4000}-\prod_{i=1}^{D}\cos(\frac{z_i}{\sqrt{i}})+1+f\_bias, z=(x-o)*M, x=[x_1,x_2,\cdots,x_D]$	[-600,600]	-180
F_8	$F_8(x)=-20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} z_i^2})-\exp(\frac{1}{D}\sum_{i=2}^{D}\cos(2\pi z_i))+20+e+f\_bias,$ $z=(x-o)*M, x=[x_1,x_2,\cdots,x_D]$	[-32,32]	-140
F_9	$F_9(x)=\sum_{i=1}^{D}(z_i^2-10\cos(2\pi z_i)+10)+f\_bias, z=(x-o), x=[x_1,x_2,\cdots,x_D]$	[-5,5]	-330
F_{10}	$F_{10}(x)=\sum_{i=1}^{D}(z_i^2-10\cos(2\pi z_i)+10)+f\_bias, z=(x-o)*M, x=[x_1,x_2,\cdots,x_D]$	[-5,5]	-330
F_{11}	$F_{11}(x)=\sum_{i=1}^{D}(\sum_{k=0}^{k_{max}}[a^k\cos(2\pi b^k(z_i+0.5))])-D\sum_{k=0}^{k_{max}}[a^k\cos(2\pi b^k\cdot 0.5)]$ $+f\_bias, a=0.5, b=3, k_{max}=20, z=(x-o)*M, x=[x_1,x_2,\cdots,x_D]$	[-0.5,0.5]	90
F_{12}	$F_{12}(x)=\sum_{i=1}^{D}(A_i-B_i(x))^2+f\_bias, x=[x_1,x_2,\cdots,x_D],$ $A_i=\sum_{j=1}^{D}(a_{ij}\sin\alpha_j+b_{ij}\cos\alpha_j), B_i(x)=\sum_{j=1}^{D}(a_{ij}\sin x_j+b_{ij}\cos x_j)$	[-100,100]	-460

Table 2. Comparison result of QPSOIII, QPSOIII-L

	m	QPSOIII		QPSOIII-L			m	QPSOIII		QPSOIII-L	
		Mean	Var.	Mean	Var.			Mean	Var.	Mean	Var.
	2	**-4.50E+02**	**0.00E+00**	**-4.50E+02**	**0.00E+00**		2	-1.80E+02	1.75E-04	**-1.80E+02**	**1.29E-05**
F_1	10	-4.50E+02	9.48E-06	**-4.50E+02**	**4.87E-25**	F_7	10	-1.76E+02	1.09E+01	**-1.78E+02**	**9.51E-01**
	20	-4.49E+02	1.43E+01	**-4.50E+02**	**2.87E-11**		20	-8.75E+01	5.58E+03	**-1.17E+02**	**2.32E+03**
	2	-4.50E+02	1.35E-19	**-4.50E+02**	**0.00E+00**		2	-1.40E+02	8.25E-04	**-1.40E+02**	**2.21E-05**
F_2	10	-4.20E+02	5.81E+03	**-4.50E+02**	**1.82E-01**	F_8	10	-1.20E+02	8.01E-03	**-1.20E+02**	**6.72E-03**
	20	3.96E+03	7.07E+06	**9.88E+00**	**6.43E+04**		20	**-1.19E+02**	**3.13E-03**	-1.19E+02	4.19E-03
	2	**-4.43E+02**	**1.55E+02**	-4.20E+02	5.30E+03		2	-3.30E+02	6.60E-27	**-3.30E+02**	**0.00E+00**
F_3	10	2.36E+06	2.02E+12	**1.00E+06**	**4.76E+11**	F_9	10	-3.23E+02	6.22E+01	**-3.26E+02**	**4.13E+00**
	20	1.71E+08	3.17E+15	**1.31E+08**	**2.59E+15**		20	-3.07E+02	8.04E+01	**-3.12E+02**	**8.34E+01**
	2	**-4.50E+02**	**0.00E+00**	**-4.50E+02**	**0.00E+00**		2	**-3.30E+02**	**2.99E-14**	-3.30E+02	7.59E-02
F_4	10	-2.97E+02	2.01E+04	**-3.95E+02**	**1.59E+04**	F_{10}	10	**-3.02E+02**	**1.19E+02**	-2.97E+02	1.47E+02
	20	8.14E+03	2.42E+07	**5.57E+03**	**1.16E+07**		20	-1.80E+02	5.37E+02	**-1.89E+02**	**3.59E+02**
	2	-3.10E+02	1.19E-24	**-3.10E+02**	**0.00E+00**		2	9.00E+01	1.76E-07	**9.00E+01**	**0.00E+00**
F_5	10	**-2.08E+02**	**1.64E+04**	-1.61E+02	9.90E+04	F_{11}	10	9.72E+01	8.39E-01	**9.61E+01**	**1.66E+00**
	20	**5.46E+03**	**3.37E+06**	5.87E+03	3.96E+06		20	1.14E+02	8.67E-01	**1.14E+02**	**5.17E-01**
	2	3.90E+02	1.33E-05	**3.90E+02**	**1.85E-08**		2	-4.60E+02	9.91E-05	**-4.60E+02**	**1.74E-12**
F_6	10	1.11E+03	1.94E+06	**6.89E+02**	**3.31E+05**	F_{12}	10	7.59E+03	2.42E+07	**3.79E+03**	**2.33E+07**
	20	1.70E+03	4.88E+06	**7.37E+02**	**4.39E+05**		20	1.51E+05	2.90E+09	**6.95E+04**	**9.28E+08**

Table 3. Comparison result of QPSOI, QPSOII, QPSOIII-L

	m	**QPSOⅠ**				**QPSOⅡ**				**QPSOⅢ-L**			
		Worst	Best	Mean	Var.	Worst	Best	Mean	Var.	Worst	Best	Mean	Var.
F_1	2	-4.50E+02	-4.50E+02	**-4.50E+02**	**0.00E+00**	-4.50E+02	-4.50E+02	-4.50E+02	8.02E-12	-4.50E+02	-4.50E+02	**-4.50E+02**	**0.00E+00**
	10	-4.50E+02	-4.50E+02	**-4.50E+02**	**0.00E+00**	-1.67E+02	-4.48E+02	-4.06E+02	4.10E+03	-4.50E+02	-4.50E+02	-4.50E+02	4.87E-25
	20	6.96E+02	-1.93E+02	1.66E+02	5.44E+04	4.82E+03	-2.04E+02	1.73E+03	1.76E+06	-4.50E+02	-4.50E+02	**-4.50E+02**	**2.87E-11**
F_2	2	-4.50E+02	-4.50E+02	**-4.50E+02**	**0.00E+00**	-4.50E+02	-4.50E+02	-4.50E+02	6.38E-10	-4.50E+02	-4.50E+02	**-4.50E+02**	**0.00E+00**
	10	-4.48E+02	-4.50E+02	-4.50E+02	2.13E-01	2.38E+02	-3.92E+02	-1.77E+02	2.95E+04	-4.48E+02	-4.50E+02	**-4.50E+02**	**1.82E-01**
	20	7.01E+03	1.23E+03	3.59E+03	2.33E+06	3.02E+04	4.78E+03	1.45E+04	2.68E+07	5.06E+02	-3.23E+02	**9.88E+00**	**6.43E+04**
F_3	2	9.93E+02	-4.50E+02	-3.01E+02	1.15E+05	-2.90E+02	-4.50E+02	**-4.25E+02**	**1.27E+03**	-1.38E+02	-4.50E+02	-4.20E+02	5.30E+03
	10	2.83E+07	1.40E+06	1.22E+07	6.58E+13	1.68E+07	3.72E+05	4.61E+06	1.34E+13	2.87E+06	1.44E+05	**1.00E+06**	**4.76E+11**
	20	1.15E+09	1.58E+08	5.62E+08	8.64E+16	2.66E+08	8.91E+07	1.81E+08	2.03E+15	2.97E+08	6.40E+07	**1.31E+08**	**2.59E+15**
F_4	2	-4.50E+02	-4.50E+02	**-4.50E+02**	**0.00E+00**	-4.50E+02	-4.50E+02	-4.50E+02	2.16E-09	-4.50E+02	-4.50E+02	**-4.50E+02**	**0.00E+00**
	10	-2.62E+02	-4.49E+02	**-4.19E+02**	**1.99E+03**	1.57E+03	-4.00E+02	-2.46E+01	1.54E+05	1.84E+02	-4.50E+02	-3.95E+02	1.59E+04
	20	2.37E+04	4.73E+03	1.17E+04	2.15E+07	2.92E+04	9.78E+03	1.85E+04	3.19E+07	1.19E+04	7.19E+02	**5.57E+03**	**1.16E+07**
F_5	2	-3.10E+02	-3.10E+02	**-3.10E+02**	**0.00E+00**	-3.10E+02	-3.10E+02	-3.10E+02	6.54E-11	-3.10E+02	-3.10E+02	**-3.10E+02**	**0.00E+00**
	10	2.78E+03	1.97E+01	1.66E+03	6.34E+05	6.48E+02	-2.56E+02	8.79E+00	7.78E+04	8.04E+02	-3.09E+02	**-1.61E+02**	**9.90E+04**
	20	1.53E+04	6.90E+03	1.21E+04	4.73E+06	9.47E+03	2.59E+03	**5.63E+03**	**3.03E+06**	1.03E+04	2.58E+03	5.87E+03	3.96E+06
F_6	2	3.90E+02	3.90E+02	**3.90E+02**	**1.35E-28**	3.92E+02	3.90E+02	3.90E+02	2.49E-01	3.90E+02	3.90E+02	3.90E+02	1.85E-08
	10	1.24E+05	3.98E+02	1.21E+04	7.71E+08	1.04E+06	2.13E+03	1.62E+05	7.04E+10	2.92E+03	3.90E+02	**6.89E+02**	**3.31E+05**
	20	2.44E+08	1.93E+07	9.46E+07	4.31E+15	3.12E+08	3.22E+06	1.00E+08	9.08E+15	3.20E+03	3.97E+02	**7.37E+02**	**4.39E+05**
F_7	2	-1.80E+02	-1.80E+02	-1.80E+02	2.43E-05	-1.80E+02	-1.80E+02	-1.80E+02	2.97E-04	-1.80E+02	-1.80E+02	**-1.80E+02**	**1.29E-05**
	10	-1.78E+02	-1.80E+02	**-1.79E+02**	**2.13E-01**	1.08E+03	1.08E+03	1.08E+03	4.18E-06	-1.76E+02	-1.80E+02	-1.78E+02	9.51E-01
	20	5.66E+02	-2.49E+01	2.07E+02	2.00E+04	6.80E+02	4.57E+02	5.85E+02	4.20E+03	2.48E+01	-1.67E+02	**-1.17E+02**	**2.32E+03**
F_8	2	-1.20E+02	-1.35E+02	-1.22E+02	1.87E+01	-1.37E+02	-1.40E+02	-1.40E+02	3.97E-01	-1.40E+02	-1.40E+02	**-1.40E+02**	**2.21E-05**
	10	-1.20E+02	-1.20E+02	-1.20E+02	5.35E-03	-1.19E+02	-1.20E+02	-1.20E+02	6.93E-03	-1.20E+02	-1.20E+02	**-1.20E+02**	**6.72E-03**
	20	-1.19E+02	-1.19E+02	-1.19E+02	4.22E-03	-1.19E+02	-1.19E+02	-1.19E+02	3.46E-03	-1.19E+02	-1.19E+02	**-1.19E+02**	**4.19E-03**
F_9	2	-3.30E+02	-3.30E+02	**-3.30E+02**	**0.00E+00**	-3.30E+02	-3.30E+02	-3.30E+02	1.42E-08	-3.30E+02	-3.30E+02	**-3.30E+02**	**0.00E+00**
	10	-3.09E+02	-3.22E+02	-3.15E+02	1.20E+01	-2.78E+02	-2.81E+02	-2.80E+02	8.90E-01	-3.22E+02	-3.29E+02	**-3.26E+02**	**4.13E+00**
	20	-1.78E+02	-2.56E+02	-2.14E+02	4.83E+02	-1.85E+02	-2.80E+02	-2.38E+02	5.49E+02	-2.84E+02	-3.22E+02	**-3.12E+02**	**8.34E+01**
F_{10}	2	-3.29E+02	-3.30E+02	-3.30E+02	3.96E-02	-3.30E+02	-3.30E+02	**-3.30E+02**	**5.14E-05**	-3.29E+02	-3.30E+02	-3.30E+02	7.59E-02
	10	-2.79E+02	-3.19E+02	**-3.02E+02**	**1.05E+02**	-2.16E+02	-2.38E+02	-2.31E+02	2.08E+01	-2.67E+02	-3.18E+02	-2.97E+02	1.47E+02
	20	1.08E+01	-1.31E+02	-7.86E+01	1.39E+03	-1.24E+02	-2.07E+02	-1.56E+02	3.35E+02	-1.38E+02	-2.26E+02	**-1.89E+02**	**3.59E+02**
F_{11}	2	9.03E+01	9.01E+01	9.02E+01	5.35E-03	9.01E+01	9.00E+01	9.00E+01	5.69E-04	9.00E+01	9.00E+01	**9.00E+01**	**0.00E+00**
	10	1.00E+02	9.81E+01	9.92E+01	3.62E-01	9.98E+01	9.61E+01	9.82E+01	1.01E+00	9.82E+01	9.37E+01	**9.61E+01**	**1.66E+00**
	20	1.16E+02	1.13E+02	1.15E+02	8.49E-01	1.16E+02	1.12E+02	1.14E+02	5.71E-01	1.15E+02	1.12E+02	**1.14E+02**	**5.17E-01**
F_{12}	2	-4.58E+02	-4.60E+02	-4.60E+02	1.37E-01	-4.59E+02	-4.60E+02	-4.60E+02	2.29E-02	-4.60E+02	-4.60E+02	**-4.60E+02**	**1.74E-12**
	10	4.84E+04	6.12E+03	3.34E+04	1.04E+08	3.86E+04	1.08E+04	2.28E+04	4.65E+07	1.84E+04	-2.95E+02	**3.79E+03**	**2.33E+07**
	20	3.99E+05	2.23E+05	3.20E+05	3.24E+09	3.72E+05	2.07E+05	3.02E+05	1.67E+09	1.18E+05	2.23E+04	**6.95E+04**	**9.28E+08**

In addition to numerical comparison, the convergence of each algorithm in specific functions is illustrated in this paper. Fig. 2 and Fig. 3 show the optimization of the two algorithms on F_4 with dimension 10 and F_{12} with dimension 20, respectively. According to the convergence curve, Fig. 2 shows that QPSOIII-L not only finds better solution than QPSOIII, but also has better search speed QPSOIII. Fig.3 shows that QPSOIII-L has better ability than QPSOIII to break away from local extremum in the later search period. The convergence of QPSOIII-L is significantly better than QPSOIII. Therefore, the integration of Lévy flights not only maintains the original fast optimization speed, but also has better performance in the later stage. QPSOIII-L can continuously approach the optimal value and better jump out of the local extremum.

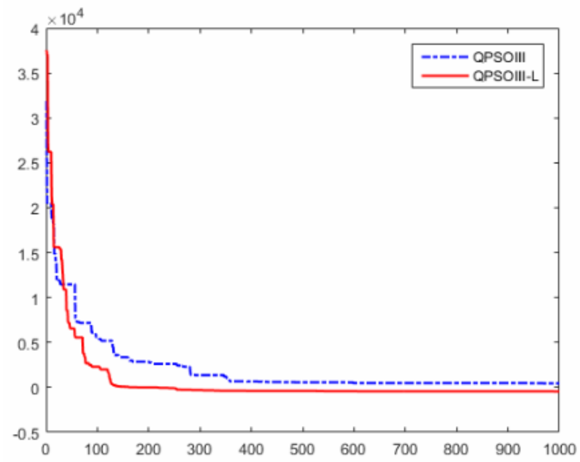

Fig. 2. F_4 ($m = 10$, iteration number $G = 1000$)

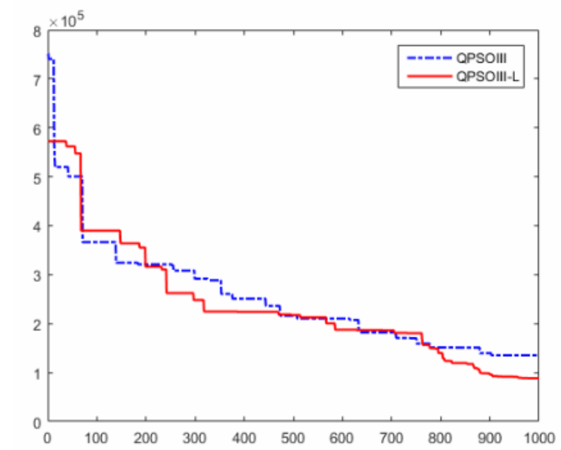

Fig. 3. F_{12} ($m = 20$, iteration number $G = 1000$)

From Table 3, we can see that QPSOIII-L deals with unimodal functions and multimodal functions well, especially in multimodal functions and high-dimensional cases. Comparing with QPSOI and QPSOII, the results achieved by QPSOIII-L are significantly better than the former ones. In F_3 and F_{10}, the new algorithm cannot get the best result at the beginning, but the effect will get better as the dimension increases. And in F_5, the bad results show that the improvement of the algorithm has certain limitations. For this kind of function, whose value is transformed smoothly and the extremum is at the boundary has room for improvement.

Fig. 4 to Fig. 6 shows the optimization effects of each algorithm in the different dimensions of the F_1. Fig. 7 to Fig. 9 shows the optimization effects of each algorithm in the 20 dimension of the F_2, F_7 and F_{10}. Through comparison, it can be found that QPSOIII-L can converge under a smaller number of iterations and obtain better results in different dimensions. In particular, QPSOIII-L has good performance on multimodal functions. And in the calculation of the large dimensions, QPSOIII-L has a good convergence speed in the early period, and has the ability to jump out of the local optimum in the later period of the search, the results of QPSOIII-L show a marked improvement compared with the results obtained by QPSO I and QPSO II.

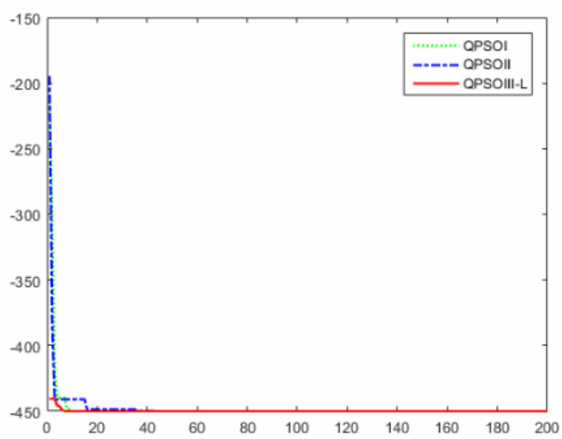

Fig. 4. F_1 ($m = 2$, iteration number $G = 200$)

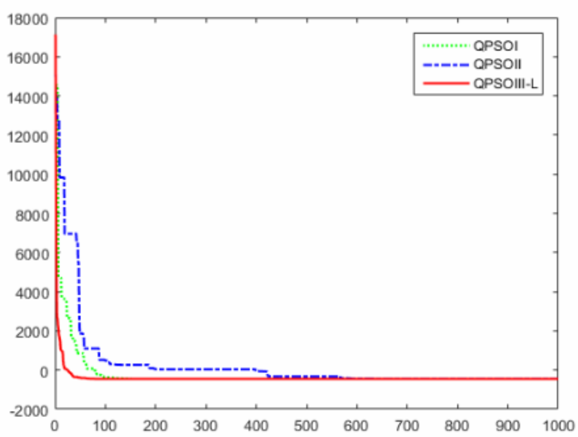

Fig. 5. F_1 ($m = 10$, iteration number $G = 1000$)

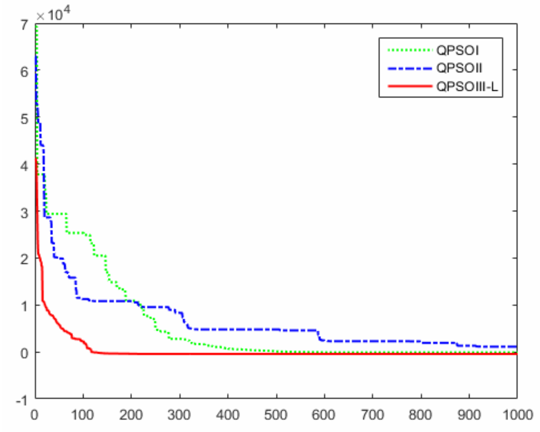

Fig. 6. F_1 ($m = 20$, iteration number $G = 1000$)

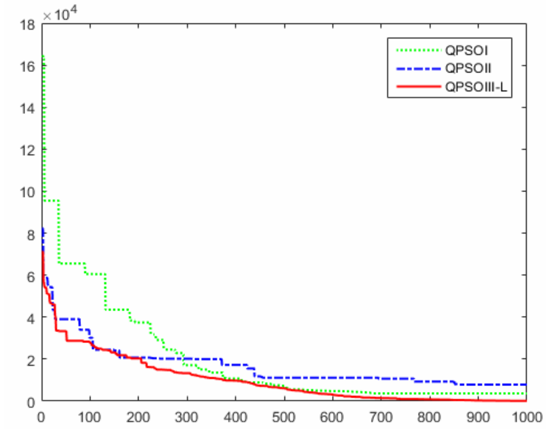

Fig. 7. F_2 ($m = 20$, iteration number $G = 1000$)

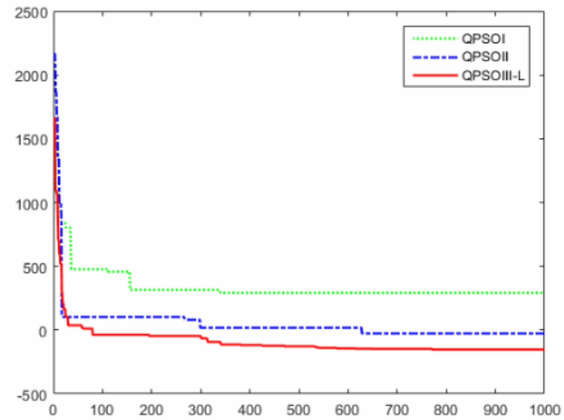

Fig. 8. F_7 ($m = 20$, iteration number $G = 1000$)

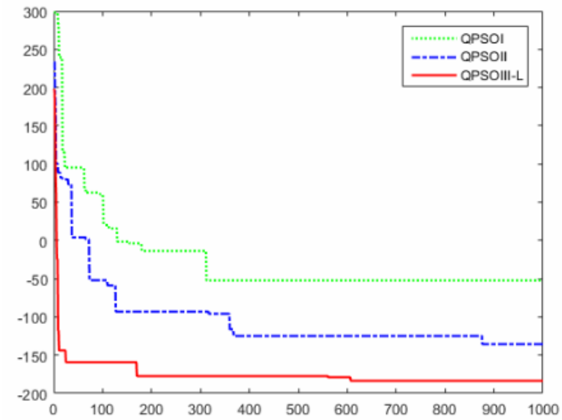

Fig. 9. F_{10} ($m = 20$, iteration number $G = 1000$)

5 Conclusions

In order to enhance the diversity of the QPSO algorithm and avoid the premature convergence of the algorithm, this paper combines quantum computing with quantum particle swarm optimization (QPSO), and presents a hybrid quantum particle swarm optimization (QPSO) algorithm based on Lévy flights. On the one hand, the algorithm uses probabilistic amplitude encoding method and the rotation gate strategy to expand the search space of the particles, which improves the diversity of the quantum particle swarm optimization algorithm and avoids the particles falling into local extremum. At the same time, the search speed of the algorithm is improved at the early stage. On the other hand, the variation process based on Lévy flights retains the better population structure while improving the population diversity.

Through the above experiments, it can be seen that the hybrid quantum particle swarm algorithm based on Lévy flights has better searching ability for multimodal and high-dimensional functions. However, the improved algorithm performs not well enough in individual functions and we will make further study on the setting strategies of mutation probability, step size factor γ and even the trigonometric function representation of the particle position in the future work.

Acknowledgements

This work is partially supported by the National Natural Science Foundation of China (61862041). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation

References

[1] A. Narayanan, M. Moore, Quantum-inspired genetic algorithms, in: Proc. IEEE International Conference on Evolutionary Computation, 1996.

[2] J. Sun, B. Feng, W.B. Xu, Particle swarm optimization with particles having quantum behavior, in: Proc. Congress on Evolutionary Computation, 2004.

[3] Y. Guo, X. Wu, W. Huang, H. Wang, Adaptive minimum entropy blind equalization algorithm for quantum artificial fish swarm optimization. <https://www.researchgate.net/publication/300934838_Adaptive_minimum_entropy_blind_equalization_algorithm_based_on_quantum_artificial_fish_swarm_optimization>, 2016.

[4] K.C. Liu, Y.G. He, L.Y. Huang, et al, Quantum Particle Swarm Optimization Based on Asymmetric Quantum Potential Well and Application, Power System Technology 4(2)(2016) 363-368.

[5] B.X. Ma, C.M. Ye, Research of flow-shop scheduling problem based on quantum cat swarm optimization, Comput. Eng. Appl., 2015.

[6] P.C. Li, S.Y. Yang, X.D. Li et al., Quantum derived cuckoo search algorithm, Computer Syst. Appl. 26(9)(2017) 122-127.

[7] J. Sun, W. Fang, V. Palade, X. Wu, W. Xu, Quantum-behaved particle swarm optimization with Gaussian distributed local attractor point, Applied Mathematics & Computation 218(7)(2011) 3763-3775.

[8] G.S. Yang, X.Y. Wang, X.D. Li, Setting value optimization method in integration for relay protection based on improved quantum particle swarm optimization algorithm, in: Proc. IOP Conference Series Earth and Environmental Science, 2018.

[9] Z.P. Li, W.Z. Li, Y. Jiang et al., Quantum particle swarm optimization using niche and reverse learning strategies, Journal of Chongqing University of Technology: Natural Science, 2018.

[10] K. Zhou, Y. Wang, Z.C. Ji, Hybrid Quantum Particle Swarm Optimization for Mould Shop Scheduling, Journal of Systems Simulation 28(6)(2016) 1247-1254.

[11] G.G. Wang, A.H. Gandomi, A.H. Alavi, S. Deb, A hybrid method based on krill herd and quantum-behaved particle swarm optimization, Neural Computing and Applications 27(4)(2016) 989-1006.

[12] M. Clerc, J. Kennedy, The particle swarm-explosion, stability and convergence in a multidimensional complex space, IEEE Transactions on Evolutionary Computation 6(2)(2002) 58-73.

[13] J. Sun, W. Fang, X. Wu, V. Palade, W. Xu, Quantum-behaved particle swarm optimization: Analysis of individual particle behavior and parameter selection, Evolutionary Computation 20(3)(2012) 349-393. doi:10.1162/EVCO_a_00049

[14] X. S. Yang, Nature-Inspired Optimization Algorithms, Elsevier Science Publishers, B. V, 2014.

[15] M. Gutowski, Lévy flights as an underlying mechanism for global optimization algorithms. <https://arxiv.org/abs/math-ph/0106003>, 2001.

[16] F.V.D. Bergh, A.P. Engelbrecht, A study of particle swarm optimization particle trajectories, Information Sciences 176(8)(2006) 937-971.

[17] W. Fang, J. Sun, Z.-P. Xie, W.-B. Xu, Convergence analysis of quantum-behaved particle swarm optimization algorithm and study on its control parameter, Acta Physica Sinica 59(6)(2010) 3686-3694.

[18] X.S. Yang, Multiobjective cuckoo search for design optimization, Computers & Operations Research 40(6)(2013) 1616-1624.

[19] S.Y. Li, P.C. Li, Quantum Computation and Quantum Optimization Algorithm, Harbin Inst. Technol. Press, Harbin, 2009.