Improved Quantum Particle Swarm Optimization by Bloch Sphere

Yu Du[1], Haibin Duan[1,2], Renjie Liao[1], and Xihua Li[1]

[1] National Key Laboratory of Science and Technology on Holistic Control, School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China
[2] Provincial Key Laboratory for Information Processing Technology, Suzhou University, Suzhou 215006, China
duyu1962@sina.com, hbduan@buaa.edu.cn, lrjconan@gmail.com, maillixihua@qq.com

Abstract. Quantum Particle Swarm Optimization (QPSO) is a global convergence guaranteed search method which introduces the Quantum theory into the basic Particle Swarm Optimization (PSO). QPSO performs better than normal PSO on several benchmark problems. However, QPSO's quantum bit(Qubit) is still in Hilbert space's unit circle with only one variable, so the quantum properties have been undermined to a large extent. In this paper, the Bloch Sphere encoding mechanism is adopted into QPSO, which can vividly describe the dynamic behavior of the quantum. In this way, the diversity of the swarm can be increased, and the local minima can be effectively avoided. The proposed algorithm, named Bloch QPSO (BQPSO), is tested with PID controller parameters optimization problem. Experimental results demonstrate that BQPSO has both stronger global search capability and faster convergence speed, and it is feasible and effective in solving some complex optimization problems.

Keywords: Quantum Particle Swarm Optimization (QPSO), Bloch Sphere, Bloch QPSO(BQPSO), global search.

1 Introduction

Particle Swarm Optimization (PSO)[1], is a population based stochastic optimization technique proposed by Kennedy and Eberhart. As an emerging intelligent technology, PSO proves to be comparable in performance with other evolutionary algorithms such as Simulated Annealing (SA) and Genetic Algorithm (GA)[2]-[4]. However, as demonstrated by F. Van Den Bergh[5], the particle in PSO is restricted to a finite sampling space for each iteration of the swarm. This restriction weakens the global search capability and its optimization efficiency and may lead to premature convergence.

To overcome the above shortcomings of the PSO, a Quantum Particle Swarm Optimization (QPSO)[6], which takes into account the global optimization capability and accelerated calculation characteristic of quantum computing, was proposed recently. However, QPSO uses qubits that are in the Hilbert space's unit circle with only one variable[7], therefore the quantum properties have been undermined to a large extent.

Y. Tan, Y. Shi, and K.C. Tan (Eds.): ICSI 2010, Part I, LNCS 6145, pp. 135–143, 2010.

In order to improve the global search capability, a novel Bloch Sphere encoding mechanism was proposed in this paper.

2 Basic QPSO

In order to improve search ability and optimization efficiency and to avoid premature convergence for particle swarm optimization, a novel Quantum Particle Swarm Optimization for continuous space optimization is proposed. The positions of particles are encoded by the probability amplitudes of qubits, and the movements of particles are performed by quantum rotation gates, which achieve particles searching. The mutations of particles are performed by quantum non-gate to increase particles diversity. As each qubit contains two probability amplitudes, and each particle occupies two positions in space, therefore it accelerates the searching process.

Similar to PSO, QPSO is also a probabilistic search algorithm. A qubit position vector as a string of n qubits can be defined as follows:

$$P_i = \begin{bmatrix} \cos(\theta_{i1}) & \cos(\theta_{i2}) & \cdots & \cos(\theta_{in}) \\ \sin(\theta_{i1}) & \sin(\theta_{i2}) & \cdots & \sin(\theta_{in}) \end{bmatrix}$$

where $\theta_{ij} = 2\pi \times rand$; $rand$ is the random number between 0 and 1, $i = 1,2,\cdots,m;\ j = 1,2,3,\cdots,n$, m is the population size and n is the space dimension.

This shows that each particle of the populations occupies the following two positions, corresponding to the probability amplitudes of state '0' and '1'.

$$P_{ic} = (\cos(\theta_{i1}), \cos(\theta_{i2}), \cdots \cos(\theta_{in}))\,,\ P_{is} = (\sin(\theta_{i1}), \sin(\theta_{i2}), \cdots \sin(\theta_{in}))$$

In QPSO, the particle position movement is realized by quantum rotation gates. Therefore, particle velocity update in standard PSO is converted to qubit rotation angles update in quantum rotation gates, while particle movement update is converted to qubit probability amplitudes update.

Qubit rotation angle is updated as: $\Delta\theta_{ij}(t+1) = w \cdot \Delta\theta_{ij}(t) + c_1 r_1(\Delta\theta_l) + c_2 r_2(\Delta\theta_g)$

where $\Delta\theta_l$ is determined by individual optimal position and $\Delta\theta_g$ is determined by global optimal position. Reference[8]-[9] gives a query table to find out the right $\Delta\theta_l$ and $\Delta\theta_g$.

Directed by the current individual and global optimal position, the movements of particles are performed by quantum rotation gates $U(\theta)$, then $[\cos(\theta_{ij}(t)),\quad \sin(\theta_{ij}(t))]^T$ is updated as:

$$U(\theta) \bullet \begin{bmatrix} \cos(\theta_{ij}(t)) \\ \sin(\theta_{ij}(t)) \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta_{ij}(t+1)) & -\sin(\Delta\theta_{ij}(t+1)) \\ \sin(\Delta\theta_{ij}(t+1)) & \cos(\Delta\theta_{ij}(t+1)) \end{bmatrix} \bullet \begin{bmatrix} \cos(\theta_{ij}(t)) \\ \sin(\theta_{ij}(t)) \end{bmatrix} = \begin{bmatrix} \cos(\theta_{ij}(t) + \Delta\theta_{ij}(t+1)) \\ \sin(\theta_{ij}(t) + \Delta\theta_{ij}(t+1)) \end{bmatrix} \quad (1)$$

where $i = 1,2,\cdots,m;\ j = 1,2,3,\cdots,n;$ Then updated two new locations of particle P_i are:

$$\overline{P}_{ic} = (\cos(\theta_{i1}(t) + \Delta\theta_{i1}(t+1)), \cdots, \cos(\theta_{in}(t) + \Delta\theta_{in}(t+1)))$$

$$\overline{P}_{is} = (\sin(\theta_{i1}(t) + \Delta\theta_{i1}(t+1)), \cdots, \sin(\theta_{in}(t) + \Delta\theta_{in}(t+1)))$$

As each qubit contains two probability amplitudes, each particle occupies two positions in space, therefore it accelerates the searching process.

Mutation operator was proposed in the QPSO to help increase the particles diversity and global search capability. Quantum non-gate V is used here as a mutation operator. To randomly select a number of qubits based on pre-determined mutation probability and impose the quantum non-gate to interchange two probability amplitudes on the same bit. Such a mutation is in fact a kind of qubit rotation update: suppose a qubit has an angle t, after the mutation, the angle turns to be $\pi/2 - t$, which means it rotates forward at an angle of $\pi/2 - 2t$.

The specific mutation operation is as follows:

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

This kind of uniform forward rotation can increase the diversity of particles and reduce the premature convergence probability.

3 The Proposed Bloch QPSO with Bloch Sphere

As the evolvement method in QPSO is a probability operation, individuals will inevitably produce degradation phenomenon during population evolution. And the determination of rotation angle orientation by far is almost based on the look-up table[8]-[9], which involves multi-conditional judgments, thus reduces the algorithm efficiency. Therefore, we propose a novel Bloch Sphere encoding mechanism and take a simple angle orientation determination method to solve the above problems.

3.1 The Bloch Sphere Encoding Mechanism

In quantum computing, the smallest information units are quantum bits, or qubits. In three-dimensional Bloch Sphere, a qubit can be written in the form as: $|\varphi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle$,where $\left|\cos\frac{\theta}{2}\right|^2$ and $\left|e^{i\varphi}\sin\frac{\theta}{2}\right|^2$ respectively represent the probability of becoming $|0\rangle$ or $|1\rangle$, also they satisfy the normalization condition: $\left|\cos\frac{\theta}{2}\right|^2 + \left|e^{i\varphi}\sin\frac{\theta}{2}\right|^2 = 1$ .Therefore qubits can be expressed by using the quantum probability amplitudes as: $\left[\cos\frac{\theta}{2},\ e^{i\varphi}\sin\frac{\theta}{2}\right]^T$. In the Bloch sphere, a point P can be determined by two angles θ and φ, as it is shown in Figure 1 below.

Figure 1 tells us that every qubit corresponds to a point in the Bloch Sphere, thus we can directly use the Bloch Sphere coordinates to encode the particles positions. Suppose that P_i is the i th particle in the population. Then Bloch Sphere encoding process is described as follows:

$$P_i = \begin{vmatrix} \cos\varphi_{i1}\sin\theta_{i1} \\ \sin\varphi_{i1}\sin\theta_{i1} \\ \cos\theta_{i1} \end{vmatrix} \begin{matrix} \cdots \\ \cdots \\ \cdots \end{matrix} \begin{vmatrix} \cos\varphi_{in}\sin\theta_{in} \\ \sin\varphi_{in}\sin\theta_{in} \\ \cos\theta_{in} \end{vmatrix}$$

where $\varphi_{ij} = 2\pi \times rand$, $\theta_{ij} = \pi \times rand$, *rand* is the random number in $[0, 1]$, $i = 1,2,\cdots,m$; $j = 1,2,3,\cdots,n$; m is the population size and n is the space dimension. Then each qubit is encoded as follows: $P_{ix} = (\cos(\varphi_{i1})\sin(\theta_{i1}),\cdots,\cos(\varphi_{in})\sin(\theta_{in}))$, $P_{iy} = (\sin(\varphi_{i1})\sin(\theta_{i1}),\cdots,\sin(\varphi_{in})\sin(\theta_{in}))$, $P_{iz} = (\cos(\theta_{i1}),\cdots,\cos(\theta_{in}))$.

To facilitate the presentation, we defined P_{ix}, P_{iy}, P_{iz} as position X , Y, Z .

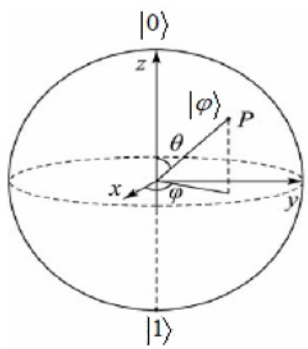

Fig. 1. Qubit Bloch Sphere

In the encoding mechanism above, each Bloch Sphere coordinate is treated as a particle position, thus each qubit contains three probability amplitudes and each particle occupies three positions in space. As a result, the Bloch QPSO has expanded solution numbers and enhanced the probability of obtaining global optimal solution.

3.2 Bloch Qubit Update and Mutation

Similar to QPSO, the particle position movement in BQPSO is realized by quantum rotation gates. Suppose that particle P_i gets current individual optimal position X , namely $P_{il} = (\cos(\varphi_{i11})\sin(\theta_{i11}),\cdots,\cos(\varphi_{i\ln})\sin(\theta_{i\ln}))$. While current global optimal position is $P_g = (\cos(\varphi_{g1})\sin(\theta_{g1}),\cdots,\cos(\varphi_{gn})\sin(\theta_{gn}))$.

Based on the above assumption, the Bloch Qubit probability amplitude is updated as follow. Derived from the matrix equation (3)[10], we get the rotation gate U:

$$U \bullet \begin{bmatrix} \cos\varphi_{ij}(t)\sin\theta_{ij}(t) \\ \sin\varphi_{ij}(t)\sin\theta_{ij}(t) \\ \cos\theta_{ij}(t) \end{bmatrix} = \begin{bmatrix} \cos(\varphi_{ij}(t) + \Delta\varphi_{ij}(t+1))\sin(\theta_{ij}(t) + \Delta\theta_{ij}(t+1)) \\ \sin(\varphi_{ij}(t) + \Delta\varphi_{ij}(t+1))\sin(\theta_{ij}(t) + \Delta\theta_{ij}(t+1)) \\ \cos(\theta_{ij}(t) + \Delta\theta_{ij}(t+1)) \end{bmatrix} \tag{3}$$

$$U = \begin{bmatrix} \cos\Delta\varphi_{ij}(t+1)\cos\Delta\theta_{ij}(t+1) & -\sin\Delta\varphi_{ij}(t+1)\cos\Delta\theta_{ij}(t+1) & \sin\Delta\theta_{ij}(t+1)\cos(\varphi_{ij}(t+1)+\Delta\varphi_{ij}(t+1)) \\ \sin\Delta\varphi_{ij}(t+1)\cos\Delta\theta_{ij}(t+1) & \cos\Delta\varphi_{ij}(t+1)\cos\Delta\theta_{ij}(t+1) & \sin\Delta\theta_{ij}(t+1)\sin(\varphi_{ij}(t+1)+\Delta\varphi_{ij}(t+1)) \\ -\sin\Delta\theta_{ij}(t+1) & -\tan(\varphi_{ij}(t+1)/2)\sin\Delta\theta_{ij}(t+1) & \cos\Delta\theta_{ij}(t+1) \end{bmatrix}$$

It is obvious that the qubit phase rotate $\Delta\varphi_{ij}(t+1)$ and $\Delta\theta_{ij}(t+1)$ respectively under the rotation gate action. $\Delta\varphi_{ij}(t+1)$ and $\Delta\theta_{ij}(t+1)$ are updated as follows:

$$\Delta\varphi_{ij}(t+1) = w\cdot\Delta\varphi_{ij}(t) + c_1 r_1(\Delta\varphi_l) + c_2 r_2(\Delta\varphi_g)$$
$$\Delta\theta_{ij}(t+1) = w\cdot\Delta\theta_{ij}(t) + c_1 r_1(\Delta\theta_l) + c_2 r_2(\Delta\theta_g)$$

Both $\Delta\varphi_{ij}(t+1)$ and $\Delta\theta_{ij}(t+1)$ are crucial to the convergence quality, in that the angle sign decides the convergence orientation while the angle size decides the convergence speed. From the update rule we know that $\Delta\varphi_{ij}(t+1)$ and $\Delta\theta_{ij}(t+1)$ are determined on $\Delta\varphi_l$, $\Delta\varphi_g$, $\Delta\theta_l$ and $\Delta\theta_g$. By far $\Delta\varphi_l$, $\Delta\varphi_g$, $\Delta\theta_l$ and $\Delta\theta_g$ are determined by a look-up table[8]-[9] containing all various possible conditions. However, as it involves multi-conditional judgments, this method reduces the efficiency of this algorithm to a large extent. To solve this problem, we use the following simple method to determine the related increment angle[10]:

$$\Delta\theta_l = \begin{cases} 2\pi + \theta_{ilj} - \theta_{ij} & (\theta_{ilj} - \theta_{ij} < -\pi) \\ \theta_{ilj} - \theta_{ij} & (-\pi \le \theta_{ilj} - \theta_{ij} < \pi) \\ \theta_{ilj} - \theta_{ij} - 2\pi & (\theta_{ilj} - \theta_{ij} > \pi) \end{cases} \qquad \Delta\theta_g = \begin{cases} 2\pi + \theta_{gj} - \theta_{ij} & (\theta_{gj} - \theta_{ij} < -\pi) \\ \theta_{gj} - \theta_{ij} & (-\pi \le \theta_{gj} - \theta_{ij} < \pi) \\ \theta_{gj} - \theta_{ij} - 2\pi & (\theta_{gj} - \theta_{ij} > \pi) \end{cases}$$

The calculation method for $\Delta\varphi_l$, $\Delta\varphi_g$ is the same as that for $\Delta\theta_l$, $\Delta\theta_g$.

To generalize the effect of quantum non-gate from the Hilbert space's unit circle to three-dimensional Bloch Sphere, we give a three-dimensional mutation operator, which satisfy the following matrix equation[10]:

$$V \cdot \begin{bmatrix} \cos\varphi_{ij}(t)\cos\theta_{ij}(t) \\ \sin\varphi_{ij}(t)\sin\theta_{ij}(t) \\ \cos\theta_{ij}(t) \end{bmatrix} = \begin{bmatrix} \cos(\pi/2 - \varphi_{ij}(t))\sin(\pi/2 - \theta_{ij}(t)) \\ \sin(\pi/2 - \varphi_{ij}(t))\sin(\pi/2 - \theta_{ij}(t)) \\ \cos(\pi/2 - \theta_{ij}(t)) \end{bmatrix} \tag{4}$$

Derived from (4), we get three-dimensional mutation operator V as follows:

$$V = \begin{bmatrix} 0 & \cot\theta_{ij}(t) & 0 \\ \cot\theta_{ij}(t) & 0 & 0 \\ 0 & 0 & \tan\theta_{ij}(t) \end{bmatrix}$$

Specific mutation process of BQPSO is similar to that of QPSO.

The process of our proposed BQPSO for solving complex optimization problems can be described as follows:

Step 1: Initialization of particle populations. Bring a random angle φ in $[0, 2\pi]$, a random angle θ in $[0, \pi]$, then qubits are produced by Bloch Sphere encoding. Initialize the size of angle increment $|\Delta\varphi| = \varphi_0$ and $|\Delta\theta| = \theta_0$. Set other parameters: max circulation generation- $ger\max$,mutation probability- Pm, population size- m, space dimension- n, and optimization problem solution scope: (a_j, b_j) $(j = 1, 2, \cdots, n)$.

Step 2: Solution Space Transformation. Map three approximate positions for each particle form unit space $I^n = [-1, 1]^n$ to optimization problem solution space Ω, then we get approximate solution $X(t)$.

Step 3: Evaluate fitness of each particle. If the particle's current position is superior to its own-memory optimal position, then replace the latter with current position. If the current global optimal position is superior to the optimal global position ever searched, then replace the latter with the current global optimal position.

Step 4: Use rotation gates to update the population according to formula (3).

Step 5:Use the three-dimensional mutation operator V under the mutation probability to mutate the quantum population according to formula (4) .Then a new particle population was produced.

Step 6: Evaluate all the new fitness of each particle after the update operation and mutation operation. Update individual optimal position and global optimal position by the same methods that are used in Step 3.

Step 7: If the stopping criterion is satisfied, the proposed BQPSO algorithm stops, then output the best solution, else return to Step 3.

The above-mentioned procedures of the proposed BQPSO process can also be described in the Figure 2.

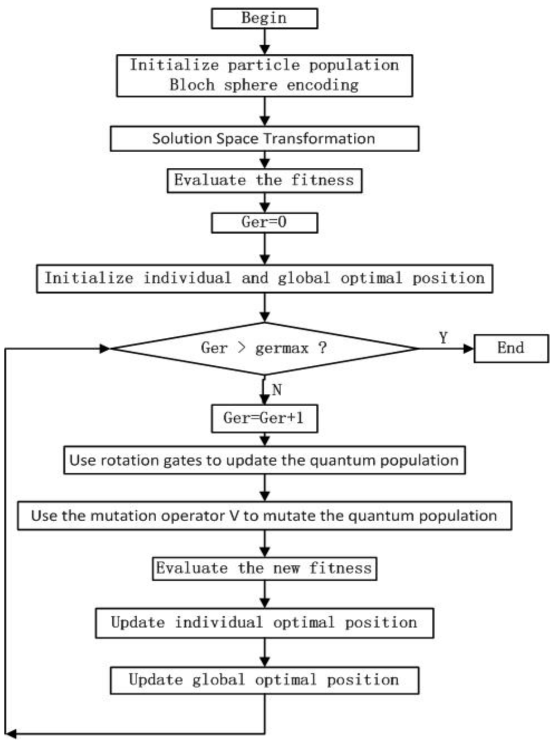

Fig. 2. The flowchart of the proposed BQPSO

4 Experimental Result

In order to investigate the feasibility and effectiveness of the proposed BQPSO, a series of experiments are conducted on a PID controller parameters optimization problem: to find the optimal parameters configuration (proportional coefficient, integral coefficient and derivative coefficient) for a second-order control system, of which the closed-loop transfer function is $G(s) = \frac{400}{s^2 + 50s}$.

In order to obtain satisfactory dynamic characteristics, we use the time integration of absolute error as the minimum objective function J, which has taken overshoot, adjustment time and static error altogether into account. The minimum objective function J is defined as follows: $J = \int_0^{\infty} (w_1|e(t)| + w_2 u^2(t) + w_4|\Delta y|)dt + w_3 \cdot t_u$ where $w_4 >> w_1$, $\Delta y(t) = y(t) - y(t-1)$, $e(t)$ is static error, $u(t)$ is PID controller output, t_u is adjustment time, $y(t)$ is the output of the controlled system, w_1, w_2, w_3, w_4 are weights. We take the fitness function as: $f = 1/J$. In the three conducted experiments, the first experiment uses standard PSO, the second experiment uses QPSO, while the third experiment uses the proposed BQPSO.

The three algorithms have been encoded and run the simulation in Matlab. Parameters were set to the following values: Sampling time is $1ms$, simulation time is $0.15s$, $ger\max$ =100, Pm =0.05, m=30, n=3, $w_1 = 0.999$, $w_2 = 0.001$, $w_3 = 2.0$, $w_4 = 100$, weighting factor $w = 0.5$, self-factor $c_1 = 2.0$, global-factor $c_2 = 2.0$. Draw the minimum objective function convergence curve and optimized system step response curve respectively on PSO, QPSO, and BQPSO. The results are shown in Figure 3 and Figure 4.

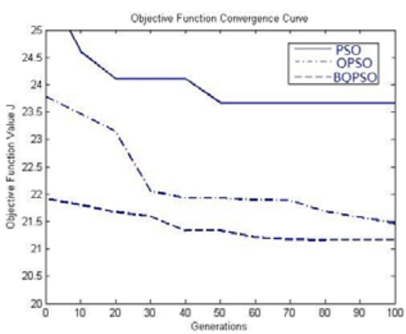

Fig. 3. Objective Function Convergence Curves

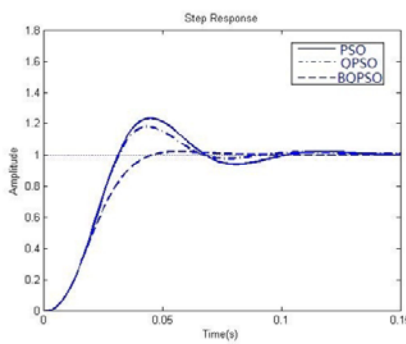

Fig. 4. Step Response

The Objective Function Convergence Curve in Figure 3 shows that, in the first experiment, it's easy to get into premature convergence with standard PSO. QPSO has more superior global search capability, while BQPSO performs best in the global search process. Also BQPSO has a faster convergence speed, compared with QPSO.

The Step Response in Figure 4 reflects that the control system optimized by BQPSO has smallest overshoot and transition time, while its stability, accuracy and rapidity of the system have been greatly improved, which indicates that BQPSO has strong robustness when applied to solve complex optimization problems.

It is obvious that our proposed BQPSO can find better solutions than standard PSO and QPSO in solving continuous optimization problems, for the reason that it has a more excellent performance with strong ability to find optimal solution and quick convergence speed.

5 Conclusions

This paper has presented an improved QPSO with Bloch Sphere for solving the continuous optimization problems. The serial experimental results verify that our proposed BQPSO is a practical and effective algorithm in solving some complex optimization problems, and also a feasible method for other complex real-world optimization problems.

Our future work will focus on applying the newly proposed BQPSO approach in this paper to other combinatorial optimization problems and combine it with other optimization methods. Furthermore, we will make a greater effort to give a complete theoretical analysis on the proposed BQPSO model.

Acknowledgments. This work was partially supported by the National Natural Science Foundation of China (Grant No. 60975072), the Program for New Century Excellent Talents in University of China (Grant No. NCET-10-0021) and the Beijing NOVA Program Foundation of China (Grant No. 2007A0017), the Aeronautical Science Foundation of China (Grant No. 2008ZC01006), and Open Fund of the Provincial Key Laboratory for Information Processing Technology, Suzhou University.

References

1. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: Proc. IEEE Conf. On Neural Network, pp. 1942–1948 (1995)
2. Angeline, P.J.: Evolutionary Optimization Versus Particle Swarm Optimization. In: Philosophy and Performance Differences. Evolutionary Programming VIII. LNCS, vol. 1477, pp. 601–610. Springer, Heidelberg (1998)
3. Eberhart, R.C., Shi, Y.H.: Comparison between Genetic Algorithm and Particle Swarm Optimization. In: Porto, V.W., Waagen, D. (eds.) EP 1998. LNCS, vol. 1447, pp. 611–616. Springer, Heidelberg (1998)
4. Krink, T., Vesterstorm, J., Riget, J.: Particle Swarm Optimization with Spatial Particle Extension. In: Proceedings of the IEEE Congress on Evolutionary Computation, vol. 2, pp. 1474–1479 (2002)
5. Van den Bergh, F.: An Analysis of Particle Swarm Optimizers. PhD Thesis. University of Pretoria, South Africa (2001)
6. Li, P.C., Li, S.Y.: Quantum particle swarms algorithm for continuous space optimization. Journal of Quantum Electronics 24(4), 463–468 (2007)

7. Al-Rabadi, A.N.: New dimensions in non-classical neural computing, part II: quantum, nano, and optical. International Journal of Intelligent Computing and Cybernetics 2(3), 513–573
8. Xing, Z., Duan, H.: An Improved Quantum Evolutionary Algorithm with 2 crossovers. In: Yu, W., He, H., Zhang, N. (eds.) ISNN 2009. LNCS, vol. 5551, pp. 735–744. Springer, Heidelberg (2009)
9. Eberhart, R.C., Shi, Y.H.: Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings of the International Congress on Evolutionary Computation, Piscataway, pp. 84–88. IEEE Press, Los Alamitos (2000)
10. Li, P.C., Li, S.Y.: Quantum Computation and Quantum Optimization Algorithm, pp. 113–117. Harbin Institute of Technology Press (2009)