A Novel Deterministic Quantum Swarm Evolutionary Algorithm

Xikun Wang[1], Lin Qian[1, 2], Ling Wang[1,*], Muhammad Ilyas Menhas[3], Haoqi Ni[1], and Xin Du[1]

[1] Shanghai Key Laboratory of Power Station Automation Technology, Shanghai University, Shanghai 200072,China
[2] Shanghai Power Construction Testing Institute, Shanghai 200031, China
[3] Department of Electrical Engineering, Mirpur University of Science and Technology, Mirpur A.K., Pakistan
wangling@shu.edu.cn

Abstract. This paper presents a novel deterministic quantum swarm evolutionary (DQSE) algorithm based on the discovery of the drawback of the standard quantum swarm evolutionary (QSE) algorithm, in which a deterministic search strategy, inspired by the nature of qubit-based evolutionary algorithms and the characteristics of qubits, is proposed to avoid the misleading of search and strengthen the global search ability. The experimental results show that the developed DQSE outperforms the quantum-inspired evolutionary algorithm, the quantum-inspired evolutionary algorithm with NOT gate and QSE in terms of the search accuracy and the convergence speed, which demonstrates that DQSE is an effective and efficient optimization algorithm.

Keywords: quantum evolutionary algorithm, particle swarm optimization, quantum swarm evolutionary algorithm, Q-bit, qubit.

1 Introduction

Inspired by the concept and principles of quantum computing, Han et al. proposed the quantum evolutionary algorithms [1, 2] which provided links between quantum computing and evolutionary algorithms. After that, quantum-inspired evolutionary algorithms (QEAs) have been studied and applied to a variety of optimization problems, such as multidimensional knapsack problems [3], fault diagnosis [4], traveling salesman problems [5], clustering [6] and network design problems [7]. Recently hybrid QEAs have been a main research direction of QEAs for improving the performance, and various hybrid QEAs are developed, such as the quantum ant colony optimization algorithms [8, 9], genetic quantum evolutionary algorithm [10], immune quantum evolutionary algorithms [11, 12] and quantum swarm evolutionary algorithms [13-15]. Among these hybrid QEAs, quantum swarm evolutionary (QSE) algorithms employ the search mechanism of particle swarm optimization (PSO) to

* Corresponding author.

M. Fei et al. (Eds.): LSMS/ICSEE 2014, Part II, CCIS 462, pp. 111–121, 2014.

update the quantum angles automatically, and therefore the robustness and the search ability of the algorithm are enhanced and it is easy to be implemented. The previous works show that QSE outperforms the standard QEA on 0-1 knapsack problems, traveling salesman problems and multiuser detection problems [13-15]. However, as quantum observing is also used to generate binary solutions in QSE, which would mislead the search direction, the performance of QSE on complicated problems may be very poor. To make up for it, this paper presents a novel deterministic quantum swarm evolutionary (DQSE) algorithm in which a deterministic search strategy is proposed to improve the global search ability of QSE.

The rest part of the paper is organized as follows. The standard QEA is first introduced in Section 2 for understanding DQSE better. Section 3 describes the presented deterministic quantum swarm evolutionary algorithm in detailed. In Section 4, the simulation results and the comparisons with QEAs and QSE are given and analyzed. Finally, Section 5 concludes the paper.

2 Quantum-inspired Evolutionary Algorithms

2.1 Initialization

Unlike other evolutionary algorithms using the classical representation approaches such as binary, numeric or symbolic coding, QEAs use Q-bits. One Q-bit is defined with a pair of complex number $[\alpha, \beta]^T$, which is characterized as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where $|0\rangle$ and $|1\rangle$ are the quantum states. α and β are complex numbers that specify the probability amplitudes of the corresponding states; α^2 denotes the probability that the Q-bit will be found in the "0" state and β^2 gives the probability that the Q-bit will be found in the "1" state. For an *M*-dimensional individual q of QEAs, it can be defined as Eq. (1)

$$q = \begin{bmatrix} \alpha_1 \dots \alpha_j \dots \alpha_M \\ \beta_1 \dots \beta_j \dots \beta_M \end{bmatrix} \; 1 \le j \le M \quad . \tag{1}$$

where M is the length of Q-bit chromosome, α_j and β_j are the corresponding probability amplitudes of the *j-th* Q-bit and satisfy the normalization condition $|\alpha_j|^2 + |\beta_j|^2 = 1$. All the Q-bits are set to $\frac{1}{\sqrt{2}}$ which means that each Q-bit chromosome is initialized with the linear superposition of all possible states with the same probability.

2.2 Quantum Observing

To solve optimization problems, the corresponding solutions of Q-bit chromosomes are needed. In QEAs, a conventional binary solution is constructed by observing Q-bits. For a bit p_{ij} of the binary individual p_i, a random number r is generated. If $|\alpha_{ij}|^2 > r$, then set p_{ij} with "0" ; otherwise set p_{ij} with "1", i.e.

$$p_{ij} = \begin{cases} 1 & if\ r > |\alpha_i|^2 \\ 0 & otherwise \end{cases} . \tag{2}$$

Then the population P of binary solutions can be generated by observing the states of the current Q-bit individuals.

2.3 Quantum Rotation Gate

By using the corresponding binary solutions, the fitness of quantum individuals can be calculated and adopted to evaluate the performance. Then the quantum rotation gate is used to update quantum individuals in the QEA and leads the algorithm close to the best solution gradually, which is defined as follows:

$$\begin{bmatrix} \alpha'_{ij} \\ \beta'_{ij} \end{bmatrix} = \begin{bmatrix} \cos(\theta_{ij}) & -\sin(\theta_{ij}) \\ \sin(\theta_{ij}) & \cos(\theta_{ij}) \end{bmatrix} \begin{bmatrix} \alpha_{ij} \\ \beta_{ij} \end{bmatrix} = U(\theta_{ij}) \begin{bmatrix} \alpha_{ij} \\ \beta_{ij} \end{bmatrix} . \tag{3}$$

$$\theta_{ij} = \Delta\theta_{ij} \cdot S(\alpha_{ij}, \beta_{ij}) . \tag{4}$$

where θ_{ij} is the quantum rotation angle, $S(\alpha_{ij}, \beta_{ij})$ is the sign of θ_{ij} that determines the direction, and $\Delta\theta_{ij}$ is the magnitude of the rotation angle which is determined according to the lookup table [1]. With the updating of the quantum angle, $|\alpha_i|^2$ or $|\beta_i|^2$ approaches to 1 or 0, that is, the Q-bit chromosome converges to a single state, and finally the optimal solution can be found. More details of QEAs can be found in [1, 2].

3 Deterministic Quantum Swarm Evolutionary

QEAs only use the information of the optimal individual to guide the search, and therefore the algorithm is likely to be trapped in the local optimum when solving complex problems. In addition, the rotation angle, as the main updating strategy, determines the optimization performance in QEAs, which is given based on the empirical values. Note that it is not balanced for QEAs to update the "1" state and "0" state which may spoil the search ability of algorithms for some problems, and it is very difficult to set new quantum angle rotation rules for QEAs in a new application. To make up for it, QSE presents a simply but efficient approach for quantum angle rotation, and as the local best information, as well as the global best information, is used and therefore the search ability of QSE is enhanced. However, the search direction of QSE may be misled due to the characteristic of quantum observing which will be discussed in the following section, thus a new deterministic updating approach is developed and used in the proposed DQSE.

3.1 Initialization

For simplification, the quantum individual of DQSE uses the quantum angle as the coding scheme, and the population Q of DQSE can be represented as Eq.(5)

$$Q = \begin{bmatrix} \theta_{11}, \theta_{12}, \theta_{13}, ..., \theta_{1M} \\ \theta_{21}, \theta_{22}, \theta_{23}, ..., \theta_{2M} \\ ... \\ \theta_{i1}, \theta_{i2}, \theta_{i3}, ..., \theta_{iM} \\ ... \\ \theta_{N1}, \theta_{N2}, \theta_{N3}, ..., \theta_{NM} \end{bmatrix} . \quad (5)$$

where $\theta_{ij} \in [0, \frac{1}{2}\pi]$ is the rotation angle, $\cos\theta_{ij} = |\alpha_{ij}|$, $\sin\theta_{ij} = |\beta_{ij}|$, and $\sin^2\theta_{ij} + \cos^2\theta_{ij} = 1$. Obviously, θ_{ij} can be used to replace $|\alpha_{ij}|$ and $|\beta_{ij}|$.

3.2 Quantum Angle Updating

In the standard QSE, the search mechanism of PSO is introduced to search for the optimal quantum angle to solve optimization problems, which can be represented as Eq. (6-7):

$$v_{ij} = \omega \times v_{ij} + c_1 \times r_1 \times (\theta p_{ij} - \theta_{ij}) + c_2 \times r_2 \times (\theta g_j - \theta_{ij}) . \quad (6)$$

$$\theta_{ij} = \theta_{ij} + v_{ij} . \quad (7)$$

where ω is inertia factor; c_1 and c_2 are constants; r_1 and r_2 are random numbers between 0 and 1; v_{ij}, θ_{ij}, and θp_{ij}, are the velocity, the current quantum angle, and the corresponding individual best quantum angle, respectively; θg_j is the global best quantum angle value.

Compared with QEAs, QSE updates the quantum angle based on the evolution strategy of PSO easily and efficiently. However, due to the characteristic of quantum observing, the search direction of the quantum angle may be misled in QSE. For instance, consider $\cos^2\theta_{ij} = 0.01$, which indicates the corresponding binary value is "0" with probability of 0.01 and "1" with probability of 0.99, respectively. If the finally observed bit is "0" and the corresponding binary individual is the optimal solution, the quantum angle of this dimension of the population will be attracted to move to the current angle value, which means that more and more "1" will be generated by QSE in the following quantum observing operation process while we actually need the algorithm to move to "0" state. This misleading may seriously spoil the performance of QSE.

Note that in qubit-based evolutionary algorithms including QEAs and QSE, the updating operation is used to rotate the quantum angle and thus each Q-bit converges to a single state gradually, i.e. approaching either 1 or 0. In DQSE or QSE, the quantum angle is used, and therefore the goal of updating operation is to lead quantum angles to 0 or $\pi/2$, i.e. approaching 0 or 1. Inspired by this nature of qubit-based evolutionary algorithms, a novel deterministic velocity updating strategy is developed and used in DQSE to remedy the misleading and improve search efficiency as Eq. (8-10)

$$v_{ij} = \omega \times v_{ij} + c_1 \times r_1 \times \left(D\theta p_{ij} - \theta_{ij} \right) + c_2 \times r_2 \times \left(D\theta g_j - \theta_{ij} \right) \quad . \tag{8}$$

$$D\theta p_{ij} = \begin{cases} 0 \ \ if \ \ Pbest_{ij} = 0 \\ \dfrac{\pi}{2} \ \ if \ \ Pbest_{ij} = 1 \end{cases} \quad . \tag{9}$$

$$D\theta g_j = \begin{cases} 0 \ \ if \ \ Gbest_j = 0 \\ \dfrac{\pi}{2} \ \ if \ \ Gbest_j = 1 \end{cases} \quad . \tag{10}$$

where $Pbest_{ij}$ and $Gbest_j$ are the corresponding binary values of the local best solution and global best solution, respectively.

However, considering the characteristics of Q-bits as well as quantum observing operation and avoiding the premature of the algorithm, the converging states of quantum angles are improved as Fig.1 where γ is a constant angle, and consequently the deterministic velocity updating operation used in DQSE is modified as follows:

$$v_{ij} = \omega \times v_{ij} + c_1 \times r_1 \times \left(D\theta p_{ij} - \theta_{ij} \right) + c_2 \times r_2 \times \left(D\theta g_j - \theta_{ij} \right) \quad . \tag{11}$$

$$D\theta p_{ij} = \begin{cases} \gamma \quad \ \ if \ \ Pbest_{ij} = 0 \\ \dfrac{\pi}{2} \text{-} \gamma \ if \ \ Pbest_{ij} = 1 \end{cases} \quad . \tag{12}$$

$$D\theta g_j = \begin{cases} \gamma \quad \ \ if \ \ Gbest_j = 0 \\ \dfrac{\pi}{2} \text{-} \gamma \ if \ \ Gbest_j = 1 \end{cases} \quad . \tag{13}$$

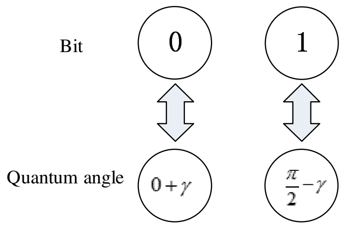

Fig. 1. The modified relationship between bits and quantum angles in DQSE

3.3 Implementation of NQSE

In summary, the implementation of DQSE is described as follows:

Step 1: Initialize the population Q, and each quantum angle is random generated between 0 and $\pi/2$.

Step 2: Generate the corresponding binary population P by performing quantum observing as Eq. (2), calculate the fitness value of each individual with the binary solutions, and set the initial local best solutions and the initial global best solution.

Step 3: Compute the velocity of quantum angles according to Eq. (11-13), and then update quantum angles as Eq. (7).

Step 4: Generate new binary population P by executing the quantum observing operator as Eq. (2) and calculate the fitness of new individuals.

Step 5: Update the local best solutions and the global best solution according to the fitness values.

Step 6: If the termination condition is satisfied, output the best binary solution; otherwise goto Step 3.

4 Parameter Study

Obviously, the parameter γ affects the search ability of DQSE. To briefly observe the performance of DQSE and set a fair γ, a parameter study of γ is performed. Two functions among the 15 benchmark functions listed in Table 1, i.e. f_4 and f_{14}, are used for testing DQSE with different γ , i.e. 0, 0.01 , 0.05 , 0.08 , 0.1 , 0.11 , 0.12 , 0.13 , 0.15 , 0.20 and 0.25. The population size, Q-bit length of each decision variable, and number of generations are set to be 30, 32, and 3000, respectively. The recommended parameters of PSO are used, that is, ω =0.7289, c_1=1.42 and c_2=1.47. DQSE was run 50 times independently for each function. The results of the parameter study are given in Table 2 including the success rate (SR), the best solution (Best), the average solution (Ave), the average generation of finding the optimal solution (AveG), and the minimum generation of finding the optimal solution (MinG).

Table 2 shows that DQSE works well with γ in a fair range. However, it is not surprised that a too small or too big value of γ significantly spoils the performance of DQSE as the algorithm loses the capability of escaping from the local optima and the advantage of Q-bits when γ is equal or very close to "0" while DQSE degrades to a random search when γ is equal or very close to "$\pi/2$". Based on the results of two benchmark functions, $\gamma = 0.11\pi$ is recommended and adopted in this paper.

5 Experimental Results and Analysis

To evaluate the performance of DQSE, we ran DQSE on the 15 benchmark functions, as well as the standard QSE [13], the standard QEA [2], and an improved QEA, i.e. the QEA with NOT gate (NQEA) [10]. Each algorithm with the recommended parameters was run 50 times independently on all the functions. The results are presented in Table 3.

Table 1. Benchmark functions

	Functions	Type
f_1	Sphere Model	unimodal
f_2	Schwefel's Poblem 2.22	unimodal
f_3	Schwefel's Poblem 1.2	unimodal
f_4	Rosenbrock's Valley	multimodal
f_5	Schaffer F6	multimodal
f_6	Shubert Function	multimodal
f_7	F1 Problem	unimodal
f_8	Rastrigin's Function	multimodal
f_9	Ackley's Function	multimodal
f_{10}	F2 Problem	multimodal
f_{11}	Six Hump Camel Back Function	multimodal
f_{12}	Branin's Function	multimodal
f_{13}	Levy F5	multimodal
f_{14}	Glankwahmdee Function	multimodal
f_{15}	Freudenstein-roth Function	multimodal

Table 3 shows that DQSE outperforms QSE, QEA and NQEA on 15, 14, and 12 functions and is inferior to QSE, QEA and NQEA on 0, 1, 3 functions, respectively. However, although the SR results of DQSE are poorer than those of NQEA on f_{14} and f_{15}, the average values of DQSE are superior to those of NQEA, which indicates that DQSE can efficiently avoid being trapped in the local optima. The performance of QSE is even worse than that of QEA due to the misleading. Compared with QSE, DQSE has much better results which demonstrates that the proposed deterministic search strategy can fix the misleading problem.

Table 2. Parameter study of γ

F	γ	SR(%)	Best	Ave	AveG	MinG
f_4	0	0	4.8281E-4	0.25502874	/	/
	0.01 π	0	6.2015E-4	0.15092999	/	/
	0.05 π	8	0	0.05446532	1115	241
	0.08 π	34	0	0.02457259	874	129
	0.10 π	46	0	0.01028581	639	128
	0.11 π	70	0	7.5550E-5	784	86
	0.12 π	60	0	1.2516E-7	720	170
	0.13 π	62	0	1.2485E-7	755	311
	0.15 π	24	0	3.2377E-6	1770	503
	0.20 π	0	1.7227E-6	1.1919E-4	/	/
	0.25 π	0	5.8238E-5	0.00416987	/	/
f_{14}	0	2	0	0.88311280	6	6
	0.01 π	4	0	0.69021723	52	31
	0.05 π	16	0	0.16990252	38	17
	0.08 π	18	0	0.16977962	33	22
	0.10 π	20	0	0.05555525	57	32
	0.11 π	20	0	1.6659E-4	70	50
	0.12 π	12	0	1.9696E-4	118	75
	0.13 π	12	0	5.2872E-4	231	132
	0.15 π	8	0	4.5201E-4	357	175
	0.20 π	0	1.1472E-5	0.00186329	/	/
	0.25 π	0	1.8890E-4	0.19118453	/	/

Table 3. Results of QEA, NQEA, QSE, DQSE on benchmark functions

		SR (%)	Best	Ave	AveG	MinG
f_1	QEA	88	0	1.24454E-6	528	41
	NQEA	100	0	0	93	53
	QSE	2	0	0.01315277	122	122
	DQSE	100	0	0	59	13
f_2	QEA	98	0	7.8125E-5	422	46
	NQEA	100	0	0	101	44
	QSE	2	0	0.05087223	1331	1331
	DQSE	100	0	0	56	27
f_3	QEA	40	0	0.00241363	75	59
	NQEA	100	0	0	120	57
	QSE	4	0	0.959670410	953	882
	DQSE	100	0	0	81	24
f_4	QEA	0	1.49011E-8	0.040532328	/	/
	NQEA	62	0	0.008534873	395	55
	QPSO	0	2.38424E-7	0.048355591	/	/
	DQSE	70	0	7.555E-5	784	86

Table 3. (*continued*)

f_5	QEA	48	-1	-0.99533627	1046	46
	NQEA	68	-1	-0.99689090	215	58
	QSE	2	-1	-0.99172843	4	4
	DQSE	42	-1	-0.99455897	692	56
f_6	QEA	0	-186.7193	-172.378978	/	/
	NQEA	20	-186.7308	-182.077938	1701	460
	QSE	0	-186.7279	-175.330374	/	/
	DQSE	38	-186.7308	-184.402919	725	70
f_7	QEA	2	-38.8503	-38.6553702	57	57
	DQEA	18	-38.8503	-38.7609278	987	125
	QSE	0	-38.8448	-38.6744453	/	/
	NQSE	34	-38.8503	-38.8263129	898	140
f_8	QEA	14	-80.7066	-80.3395065	281	78
	NQEA	12	-80.7066	-80.6559051	767	182
	QSE	4	-80.7066	-79.9507025	400	324
	DQSE	36	-80.7066	-80.6768844	723	21
f_9	QEA	82	0	0.002101440	404	56
	NQEA	100	0	0	108	58
	QSE	10	0	0.448213776	1105	3
	DQSE	100	0	0	56	18
f_{10}	QEA	0	-1.1283	-1.125200744	/	/
	NQEA	6	-1.1511	-1.130163501	344	74
	QSE	2	-1.1511	-1.128282123	2968	2968
	DQSE	6	-1.1511	-1.131805166	238	120
f_{11}	QEA	36	-1.031628	-1.030175749	851	84
	NQEA	96	-1.031628	-1.031624569	979	85
	QSE	2	-1.031628	-0.995954155	2840	2840
	DQSE	100	-1.031628	-1.03162812	318	16
f_{12}	QEA	48	0.39789	0.4041765031	182	35
	NQEA	64	0.39789	0.3994956086	250	44
	QSE	24	0.39789	0.4169677564	856	77
	DQSE	74	0.39789	0.3987495188	232	10
f_{13}	QEA	4	-176.1375	-162.1123380	1459	323
	NQEA	78	-176.1375	-173.9238963	1293	93
	QSE	0	-176.1322	-153.7435019	/	/
	DQSE	94	-176.1375	-176.1359707	556	71
f_{14}	QEA	10	0	0.0342618884	471	76
	NQEA	22	0	0.0014913356	122	85
	QSE	0	8.1057E-6	0.2711418264	/	/
	DQSE	20	0	1.6659E-4	70	50
f_{15}	QEA	18	0	2.0294479787	753	92
	NQEA	26	0	1.7870134E-4	142	78
	QSE	2	0	12.361067861	18	18
	DQSE	20	0	1.2085406E-4	137	65

6 Conclusions

In this paper the drawback of QSE is pointed out. To make up for it, a novel deterministic quantum swarm evolution algorithm is presented in which a deterministic search strategy is developed to avoid the misleading of quantum angle rotation, inspired by the nature of qubit-based evolutionary algorithms and the characteristics of qubits. The performance of DQSE is evaluated and compared with QEA, NQEA and QSE on benchmark functions. The results show that DQSE outperforms QEA, NQEA and QSE in terms of the search accuracy and the convergence speed, which demonstrates that the presented DQSE is an efficient optimization tool.

Acknowledgments. This work was supported by National Natural Science Foundation of China (Grant No. 61304031, 61374044 & 61304143), and Innovation Program of Shanghai Municipal Education Commission (14YZ007).

References

1. Han, K.H., Kim, J.H.: Genetic Quantum Algorithm and Its Application to Combinatorial Optimization Problem. In: IEEE International Conference on Evolutionary Computation, pp. 1354–1360. IEEE Press, La Jolla (2000)
2. Han, K.H., Kim, J.H.: Quantum-Inspired Evolutionary Algorithm for a Class of Combinatorial Optimization. IEEE Trans. Evolutionary Computation 6(6), 580–593 (2002)
3. Wang, L., Wang, X.T., Fei, M.R.: A Novel Quantum-Inspired Pseudorandom Proportional Evolutionary Algorithm for the Multidimensional Knapsack Problem. In: GEC 2009, pp. 545–552 (2009)
4. Wang, L., Niu, Q., Fei, M.R.: A Novel Quantum Ant Colony Optimization Algorithm and Its Application to Fault Diagnosis. Transactions of the Institute of Measurement and Control 30(3-4), 313–329 (2008)
5. You, X.M., Liu, S.: Quantum Computing-Based Ant Colony Optimization Algorithm for TSP. In: 2009 2nd International Conference on Power Electronics and Intelligent Transportation System, vol. 3, pp. 359–362 (2009)
6. Xiao, J., Yan, Y.P., Zhang, J., et al.: A Quantum-Inspired Genetic Algorithm for K-means Clustering. Expert Systems With Applications 37, 4966–4973 (2010)
7. Lin, D.Y., Waller, S.T.: A Quantum-Inspired Genetic Algorithm for Dynamic Continuous Network Design Problem. Transportation Letters-the International Journal of Transportation Research 1(1), 81–93 (2009)
8. Wang, L., Niu, Q., Fei, M.: A Novel Quantum Ant Colony Optimization Algorithm. In: Li, K., Fei, M., Irwin, G.W., Ma, S. (eds.) LSMS 2007. LNCS, vol. 4688, pp. 277–286. Springer, Heidelberg (2007)
9. Zhang, Y., Liu, S.H., Fu, S., et al.: A Quantum-Inspired Ant Colony Optimization for Robot Coalition Formation. In: Control and Decision Conference, CCDC 2009, pp. 626–631. IEEE Press, Guilin (2009)
10. Wang, L., Tang, F., Wu, H.: Hybrid Genetic Algorithm Based on Quantum Computing for numerical Optimization and Parameter Estimation. Applied Mathematics and Computation 171(2), 1141–1156 (2005)

11. Li, X., Qian, L.H.: A Modified Quantum-Inspired Evolutionary Algorithm Based on Immune Operator and Its Convergence. In: Fourth International Conference on Natural Computation, ICNC 2008, pp. 136–140. IEEE Press, Jinan (2008)
12. Jiao, L.C., Li, Y.Y., Gong, M.G., et al.: Quantum-Inspired Immune Clonal Algorithm for Global Optimization. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 38(5), 1234–1253 (2008)
13. Wang, Y., Feng, X.Y., Huang, Y.X., et al.: A Novel Quantum Swarm Evolutionary Algorithm and Its Applications. Neurocomputing 70(4-6), 633–640 (2007)
14. Yang, Y., Tian, Y.F., Yin, Z.F.: Hybrid Quantum Evolutionary Algorithms Based on Particle Swarm Theory. In: Industrial Electronics and Applications, pp. 1–7. IEEE Press, Singapore (2006)
15. Huang, Y.R., Tang, C.L., Wang, S.: Quantum-Inspired Swarm Evolution Algorithm. In: IEEE International Conference on Computational Intelligence and Security Workshops, pp. 208–211. IEEE Press, Harbin (2007)