International Journal of Neural Systems, Vol. 20, No. 1 (2010) 39–50

DOI: 10.1142/S012906571000222X

A HYBRID ARTIFICIAL BEE COLONY OPTIMIZATION AND QUANTUM EVOLUTIONARY ALGORITHM FOR CONTINUOUS OPTIMIZATION PROBLEMS

HAI-BIN DUAN[*,†], CHUN-FANG XU[*] and ZHI-HUI XING[*,†]
[*]*National Key Laboratory of Science and Technology on Holistic Control*
School of Automation Science and Electrical Engineering
Beihang University, No. 37, Xueyuan Road, Haidian District
Beijing, 100191, P. R. China
[†]*State Key Laboratory of Virtual Reality Technology and Systems,*
Beihang University, No. 37, Xueyuan Road, Haidian District,
Beijing, 100191, P. R. China
[‡]*hbduan@buaa.edu.cn*
http://hbduan.buaa.edu.cn

In this paper, a novel hybrid Artificial Bee Colony (ABC) and Quantum Evolutionary Algorithm (QEA) is proposed for solving continuous optimization problems. ABC is adopted to increase the local search capacity as well as the randomness of the populations. In this way, the improved QEA can jump out of the premature convergence and find the optimal value. To show the performance of our proposed hybrid QEA with ABC, a number of experiments are carried out on a set of well-known Benchmark continuous optimization problems and the related results are compared with two other QEAs: the QEA with classical crossover operation, and the QEA with 2-crossover strategy. The experimental comparison results demonstrate that the proposed hybrid ABC and QEA approach is feasible and effective in solving complex continuous optimization problems.

Keywords: Artificial Bee Colony (ABC); Quantum Evolutionary Algorithm (QEA); Genetic Algorithm (GA); Q-bit chromosome; premature.

1. Introduction

Since the concept of quantum computing was proposed, there has been a developing revolution in computing. As the name "quantum" suggests, this advance comes from the smallest of all places: the subatomic particles that form the basis of all matter.

Quantum computing has promised prodigious computing power in recent years. Its basic currency, the Q-bit, exists in an ON or OFF verge, which is unkown until it is read out. Therefore, if you could operate on K Q-bits, a potentially vast space of 2^k values opens up for computation. This means that we can solve many computing problems simultaneously, which saves a lot of time. The fundamental operation on Q-bits is a rotation. We have logic gates to combine the rotations. The algorithm is based on these logic gates. In principle, these algorithms can perform calculations far beyond classical computation's conceivable reach.

The genetic algorithm (GA) was firstly put forward by Holland in the 1970s to study the self adaptation behavior of natural systems.[7] It is a classical meta-heuristic algorithm for the approximate solution of combinatorial optimization problems that has been inspired by the principles of evolution of living organisms in nature. The application of GA needs no initiating knowledge of the system, and it isn't limited by the form and property of the problem. Guided by fitness function and principle of probability, it can search globally according to self adaptation by using selection, crossover and mutation. Therefore, it is a comprehensive optimization

[‡]Corresponding author.

method with extensive application in terms of processing complex non-linear problems. GA has been used successfully for complicated [Kim and Adeli,[16] Mersch *et al.*,[20] Mathakari *et al.*,[19] Teklu *et al.*,[27] Cheng and Yan,[6] Kang *et al.*,[10] and large-scale optimization [Adeli and Cheng,[1,2] Adeli and Kumar,[3,4] Sarma and Adeli,[21–24] machine learning [Hung and Adeli[8]], robotics [Smith *et al.*[25]], pattern recognition [Lee *et al.*[18]] and vibration control problems [Jiang and Adeli[9]].

GA has strong robustness and is easy to combine with other methods in optimization [Vlahogianni *et al.*[29]]. Q-bit chromosomes enjoy a rapidly growing population and strong randomization.

To improve GA for certain problems, a quantum evolutionary algorithm (QEA) is proposed on the basis of the concept and principles of quantum computing. In QEA, Q-bit chromosomes, which can represent a linear superposition of solutions, are adopted to maintain solution diversity and overcome premature convergence. At the same time, a quantum rotation gate, which makes full use of the information of the current best individual, is used to update individuals and avoid stagnation [Zhang *et al.*[35]].

The common QEA uses Q-bit gate rotation in mutation and whole interference in crossover [Yang *et al.*[33]]. By using a rotation operation, we can make full use of the information of the currently best individual to perform the next searching process, and the whole interference can avoid prematurity. In this way, the global search capacity can be greatly improved, while the convergence speed is slowed down.

Artificial Bee Colony (ABC) optimization is a bio-inspired optimization algorithm based on the intelligent foraging behavior of a honey bee swarm, proposed by Karaboga in 2005 [Karaboga[12]]. It is as simple as Particle Swarm Optimization (PSO) [Kennedy and Eberhart,[17] Yang *et al.*,[34] Kaveh and Shojaee,[15] Vitins and Axhausen[28]] and Differential Evolution (DE) algorithms, and uses only common control parameters such as colony size and maximum cycle number. ABC, as an optimization tool, can provide a population-based search procedure in which individuals called food positions are modified by the artificial bees with time and the bee's aim is to discover the places of food sources with high nectar amount and finally the one with the highest nectar. The most prominent feature of ABC algorithm is that this system combines local search methods, carried out by employed and onlooker bees, with global search methods, managed by onlookers and scouts, attempting to balance exploration and the exploitation process. Thus, the ABC algorithm can have a very good search capacity.

In order to further improve the whole performance of QEA, we propose a new hybrid strategy combined with the ABC algorithm in this work, and experimental results are given to show the feasibility and effectiveness of the proposed algorithm in solving complex continuous optimization problems.

The remainder of this paper is organized as follows. The next section introduces the main process of the basic QEA. Section 3 proposes a hybrid ABC and QEA model. Then, in Sec. 4, a series of comparison experiments are conducted, the proposed hybrid ABC and QEA approach is compared with two other QEAs: the QEA with classical crossover operation, and the QEA with 2-crossover strategy. Our concluding remarks and future work are contained in the final section.

2. The Main Process of Basic QEA

2.1. *Qubit chromosome*

In QEA, a qubit chromosome as a string of n qubits can be defined as follows [Zhang *et al.*[36]]:

$$\mathrm{q} = \left[\begin{array}{c|c|c|c} \alpha 1 & \alpha 2 & \cdots & \alpha m \\ \beta 1 & \beta 2 & \cdots & \beta m \end{array}\right] \tag{1}$$

where $|\alpha_i|^2 + |\beta_i|^2 = 1$, $i = 1, \ldots, m$, m is the number of qubits as well as the string length of the qubit individual. $|\alpha_i|^2$ provides the probability that the qubit will be found in the state of '0', while $|\beta_i|^2$ gives the probability that the qubit will be found in the '1' state. A qubit chromosome is able to represent a linear superposition of all possible solutions. It has a better characteristic of diversity than a chromosome [Tayarayi *et al.*,[26] Al-Rabadi[5]]. The process to get a classical chromosome is: bring a random number between 0 and 1, if it is bigger than $|\alpha_i|^2$, this bit in the classical chromosome is '1', else '0' is chosen.

2.2. *Quantum mutation*

The standard mutation operation is totally random without direction, so the speed of convergence is slowed down. While in QEA, the qubit representation can be used as a mutation operator. Directed

Table 1. Rotation angle.

x_i	$best_i$	$f(x) > f$ (best)	θ_i			
			$\alpha_i\beta_i > 0$	$\alpha_i\beta_i < 0$	$\alpha_i = 0$	$\beta_i = 0$
0	0	False	0	0	0	0
0	0	True	0	0	0	0
0	1	False	0	0	0	0
0	1	True	-0.05π	0.05π	$\pm 0.05\pi$	0
1	0	False	-0.05π	0.05π	$\pm 0.05\pi$	0
1	0	True	0.05π	-0.05π	0	$\pm 0.05\pi$
1	1	False	0.05π	-0.05π	0	$\pm 0.05\pi$
1	1	True	0.05π	-0.05π	0	$\pm 0.05\pi$

by the current best individual, quantum mutation is completed through the quantum rotation gate $U(\theta)$, then the $[\alpha_i, \beta_i]^T$ is updated as:

$$\begin{bmatrix} \alpha' i \\ \beta' i \end{bmatrix} = \begin{bmatrix} \cos\theta i & -\sin\theta i \\ \sin\theta i & \cos\theta i \end{bmatrix} \begin{bmatrix} \alpha i \\ \beta i \end{bmatrix} \tag{2}$$

Table 1 is used to find out the right θ_i. It is determined by both the quantum and classical chromosome.

In Table 1, x_i is the ith bit of the current classical chromosome, $best_i$ is the ith bit of the current best classical chromosome, $f(x)$ is the adaptation function [Xiao *et al.*[31]], which is like the adaptation function in GA.

Figure 1 describes the polar plot of the rotation operation on a qubit. It shows why the rotation gate

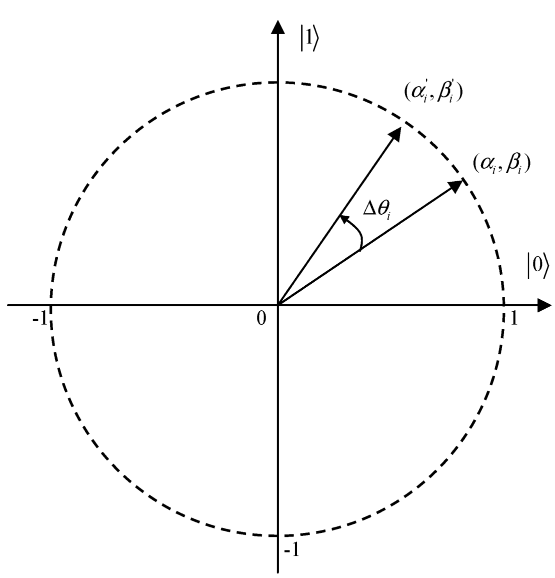

Fig. 1. Polar plot of the rotation gate for qubit chromosome.

can increase the speed of convergence obviously [Wei *et al.*[30]].

2.3. *Quantum whole interference crossover*

This kind of crossover operation is constructed by the interference characteristic of a qubit. All the quantum chromosomes are involved in the computation. For example, when the population number is 5 and the length of chromosome is 7, and let the capital numbers denote the new chromosomes, such as A(1), A(2), ..., A(7), B(1), ..., B(7), C(1), ..., C(7), D(1), ..., D(7), E(1), ..., E(7), then Table 2 shows the typical operation in this case.

The whole interference crossover operation can make full use of the information in the chromosome, improve the unilateralism of classical crossover and avoid premature convergence and stagnation problems.

2.4. *Main steps of basic QEA algorithm*

The process of basic QEA for solving complex optimization problems can be described as follows:

Step 1: Initialization of parameters: bring a random angle ω between 0 and $2 * \pi$, $\alpha = \cos(\omega), \beta = \sin(\omega)$, then a qubit is produced. Set other parameters: classical crossover probability-P_{cc}, mutation probability-Pm, whole interference crossover probability-P_{ic}, the max circulation generation-$ger_{\max}$, the number of population-n, and the length of chromosome-L.

Step 2: Produce a classical population by using these quantum chromosomes. This is the original classical population. Evaluate the fitness of each chromosome.

Step 3: Use roulette operation to select parent quantum chromosomes, operate crossover in the classical crossover probability. Update the quantum population. Then produce a classical population, and find

Table 2. The whole interference crossover operation.

1	A(1)	E(2)	D(3)	C(4)	B(5)	A(6)	E(7)
2	B(1)	A(2)	E(3)	D(4)	C(5)	B(6)	A(7)
3	C(1)	B(2)	A(3)	E(4)	D(5)	C(6)	B(7)
4	D(1)	C(2)	B(3)	A(4)	E(5)	D(6)	C(7)
5	E(1)	D(2)	C(3)	B(4)	A(5)	E(6)	D(7)

out the best solution which will be used in the mutation operation.

Step 4: Operate mutation in the mutation probability and update the quantum population. Then produce a new classical population, evaluate the fitness of each chromosome, compare with the old ones and update the classical population.

Step 5: If the stopping criterion is satisfied, the proposed QEA algorithm stops, and output the best solution, else return to Step 3.

The above-mentioned procedures are the main process of the basic QEA algorithm. To obtain greater randomness and get better results, we have also developed an improved QEA algorithm with 2-crossovers, which means that the crossover procedure of the algorithm contains both classical single point crossover and whole interference crossover.

3. The Proposed Hybrid ABC and QEA

3.1. *ABC optimization*

ABC is one of the most recently defined algorithms by D. Karaboga, motivated by the intelligent behavior of honey bees [Karaboga,[11] Karaboga and Basturk[12]]. In the ABC system, artificial bees fly around in the search space, and some (employed and onlooker bees) choose food sources depending on the experience of themselves and their nest mates, and adjust their positions. Some (scouts) fly and choose the food sources randomly without using experience. If the nectar amount of a new source is higher than that of the previous one in their memory, they memorize the new position and forget the previous one [Karaboga and Basturk[13]]. Thus, the ABC system combines local search methods, carried out by employed and onlooker bees, with global search methods, managed by onlookers and scouts, attempting to balance exploration and exploitation process.

In order to introduce the model of forage selection that leads to the emergence of collective intelligence of honey bee swarms, first, we need to define three essential components: food sources, unemployed foragers and employed foragers.

(1) Food Sources (**A** and **B** in Fig. 2): For the sake of simplicity, the "profitability" of a food source can be represented by a single quantity. In our function optimization problem, the position of a food

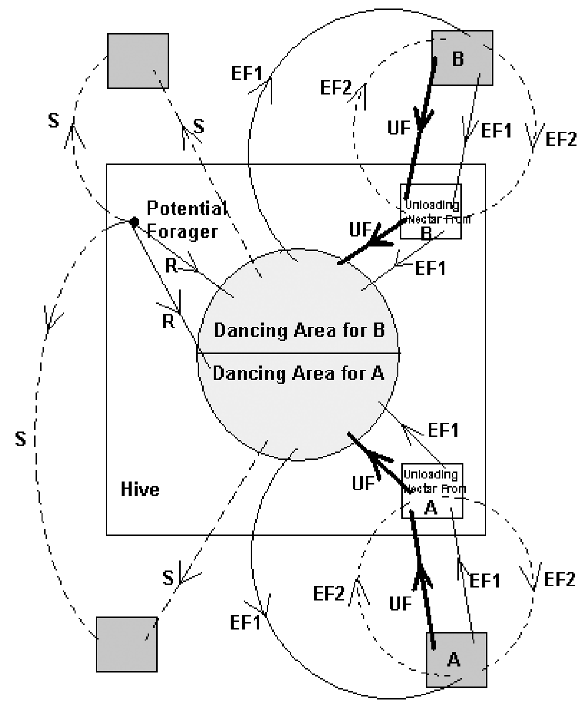

Fig. 2. The behavior of honey bee foraging for nectar.

source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution.

(2) Unemployed foragers: If it is assumed that a bee has no knowledge about the food sources in the search field, the bee initializes its search as an unemployed forager. Unemployed foragers are continually on the look out for a food source to exploit. There are two types of unemployed foragers: scouts and onlookers.

(3) Scouts (**S** in Fig. 2): If the bee starts searching spontaneously for new food sources without any knowledge, it will be a scout bee. The percentage of scout bees varies from 5% to 30% according to the information into the nest.

(4) Onlookers (**R** in Fig. 2): The onlookers wait in the nest and search the food source through sharing information of the employed foragers, and there is a greater probability of onlookers choosing more profitable sources.

(5) Employed foragers: These are associated with a particular food source which they are currently exploiting. They carry with them information about this particular source, the profitability of the source and share this information with a certain probability. After the employed foraging bee loads a portion of nectar from the food source, it returns to the hive and unloads the nectar to the food area in the hive. There are three possible options related to residual amount of nectar for the foraging bee.

If the nectar amount decreased to a low level or became exhausted, then the foraging bee abandons the food source and becomes an unemployed bee. (**UF** in Fig. 2). If there are still sufficient amounts of nectar in the food source, it can continue to forage without sharing the food source information with the nest mates (**EF2** in Fig. 2).Or it can go to the dance area to perform a waggle dance to inform the nest mates about the food source (**EF1** in Fig. 2), as is shown in Fig. 3. In this way, the bees can construct a relatively good solution of the multimodal optimization problems [Karaboga[14]].

The main steps of the ABC algorithm are described as follows:

Step 1: Initial food sources are produced for all employed bees

Step 2: REPEAT

- **Step 2.1:** Each employed bee goes to a food source in her memory and determines a neighbour source, then evaluates its nectar amount and dances in the hive.
- **Step 2.2:** Each onlooker watches the dance of employed bees and chooses one of their sources depending on the dances, and then goes to that source. After choosing a neighbor around that, she evaluates its nectar amount.
- **Step 2.3:** Abandoned food sources are determined and are replaced with the new food sources discovered by scouts.
- **Step 2.4:** The best food source found so far is registered.

Step 3: UNTIL (requirements are met)

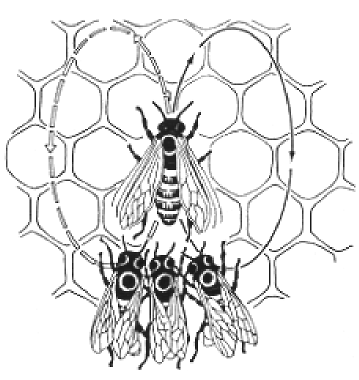

Fig. 3. Waggle dance of honey bees.

3.2. *Detailed process of the hybrid ABC and QEA*

The QEA provide new ideas to improve the traditional GA. Firstly, the information in a quantum chromosome is more than that in a classical chromosome, the number of the population is decreased and the diversity is improved. Secondly, the mutation operation is no longer totally random but directed by some rules to make the next generation better and increase the speed of convergence. Thirdly, the whole interference crossover operation can avoid premature convergence and stagnation problems.

Although the QEA has many advantages, it still has scope for improvement. In our test, we can easily find out that the QEA could not always reach the best solution of the problem, which means that the algorithm still has a considerable probability of premature convergence. To solve this problem, we propose a hybrid QEA based on the ABC algorithm. The result is extraordinary both in theory and experiments.

In our newly proposed hybrid QEA, we introduced the ABC idea into the traditional QEA to enhance the ability of global search. The parameters above are the same as the basic QEA.

In the first step, a randomly distributed initial population (food source positions) is generated.

After initialization, the population is subjected to repeated cycles of the search processes of the employed, onlooker, and scout bees, respectively.

An employed bee produces a modification on the source position in her memory and discovers a new food source position. Provided that the nectar amount of the new one is higher than that of the previous source, the bee memorizes the new source position and forgets the old one. Otherwise she keeps the position of the one in her memory.

In this process, the modification strategy is achieved by the classical single point crossover as well as the quantum rotation. In the single point crossover process, a Roulette selection operation is used to choose two quantum chromosomes from the parent generations, and then the child generation is produced by crossover. After this, two better individuals can be chosen into the next generation by evaluating their fitness. This operation is mainly to improve the convergence speed and preserve the instructive information. Usually, we choose 0.6 to 0.9 as the crossover probabilities. After the crossover process, we choose the best one as the mutation director, and implement the quantum mutation operation using the rules shown in Table 1. This operation is also to improve the convergence speed as well as to increase the diversity of the population. We choose 0.01 to 0.2 as the mutation probabilities. After all the operations, finally the employed bee selects the better population as the new source position to remember.

After all employed bees complete the searching process, they share the position information of the sources with the onlookers on the dance area. Each onlooker evaluates the nectar information taken from all employed bees and then chooses a food source depending on the nectar amounts of sources. As in the case of the employed bee, she produces a modification on the source position in her memory and checks its nectar amount. Providing that its nectar is higher than that of the previous one, the bee memorizes the new position and forgets the old one.

The sources abandoned are determined and new sources are randomly produced to replaced the abandoned ones by artificial scouts.

The process of our proposed hybrid QEA combined with ABC algorithm for solving complex optimization problems can be described as follows:

Step 1: Initialization of parameters: take a random angle ω between 0 and $2 * \pi$, $\alpha = \cos(\omega), \beta = \sin(\omega)$, then a Q-bit is produced. Set other parameters: classical crossover probability-Pcc, mutation probability-Pm, whole interference crossover probability-Pic, the max circulation generation-germax, the number of quantum population-n, the number of classical population produced by each quantum chromosome-N, the length of chromosome-L the number of employed bees-N_employed, the number of unemployed bees-N_unemployed and the searching limit-Limit.

Step 2: Produce a classical population by using these quantum chromosomes as the initial positions of the employed bees and evaluate the fitness of each individual.

Step 3: Use roulette operation to select parent quantum chromosomes, operate crossover in the classical crossover probability. Select the better solutions into the next generation and update the quantum population.

Step 4: Operate mutation in the mutation probability according to the best position found ever and update the quantum population. Then produce a new classical population, evaluate the fitness of each chromosome, compare with the old ones and choose the better ones to update the classical population as the new positions of the employed bees.

Step 5: Each onlooker chooses a food source depending on the fitness of the employed bees, and then produces a modification on the source position in her memory and calculate its fitness. Providing that its fitness is higher than that of the previous one, the bee memorizes the new position and forgets the old one.

Step 6: If an employed bee and the unemployed bees followed it still could not find a better solution after searching certain times (Limit), abandon the source and randomly produce a new source to replace the abandoned one by artificial scouts.

Step 7: Find out and remember the best solution ever found. Calculate the average value of the employed bees.

Step 8: If the stopping criterion is satisfied, the proposed QEA algorithm stops, and output the best solution, else return to Step 3.

The above-mentioned procedures of the proposed hybrid QEA process can also be described with Fig. 4.

4. Experimental Results

In order to investigate the feasibility and effectiveness of the proposed hybrid ABC and QEA approach, series of experiments are conducted on one

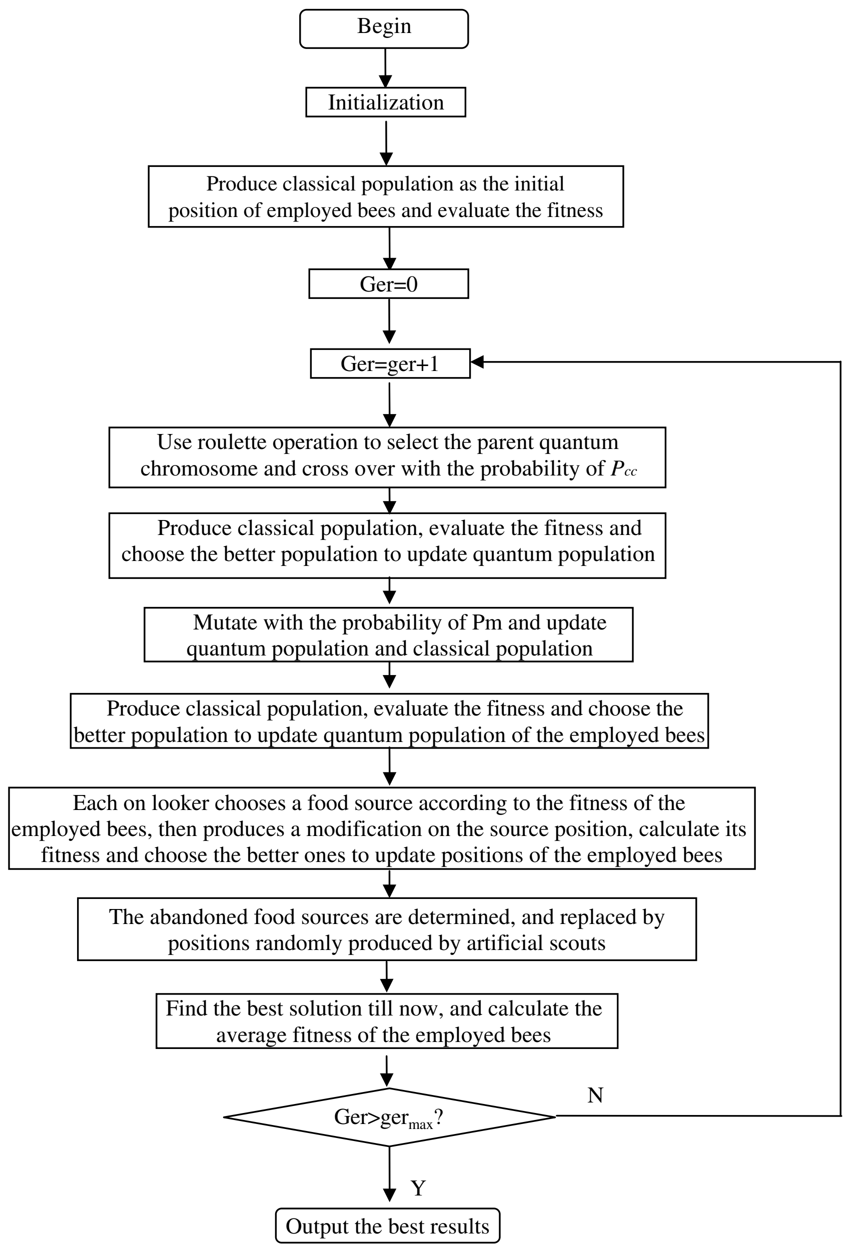

Fig. 4. Detailed procedure of the proposed hybrid ABC and QEA.

typical Benchmark continuous optimization problem (see Eq. (3)): to find the maximum value of function.

$$f(x) = 10 + \frac{\sin(1/x)}{(x-0.16)^2 + 0.1}, \quad x \in (0,1) \tag{3}$$

In the three experiments we conducted, the first experiment adopts QEA with classical crossover operation, the second experiment adopts a 2-crossover strategy into the traditional QEA [Xing and Duan[32]], while the third one adopts the proposed hybrid ABC and QEA. Each experiment has been conducted 20 times and the results are shown in the following figures and tables.

The three algorithms have all been encoded in Matlab language and implemented on a PC with 512 Mb of RAM using Windows XP. The parameters were set to the following values: $n = 20$, N_employed $= 10$, N_unemployed $= 10$, $L = 22$, $P_{cc} = 0.9$, $P_m = 0.2$, $ger_{\max} = 100$, Limit $= 30$.

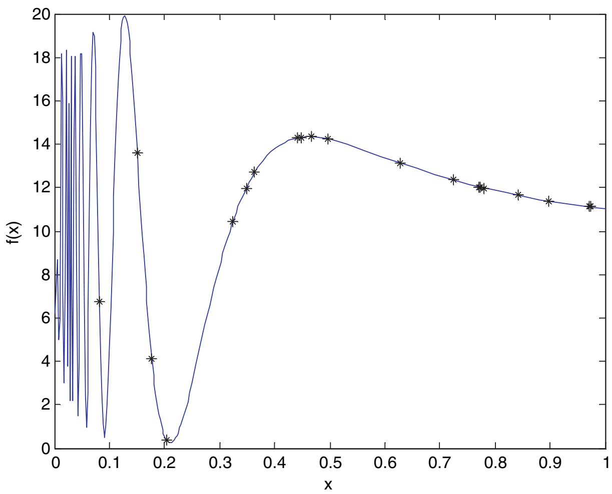

Fig. 5. The initial positions of the chromosomes.

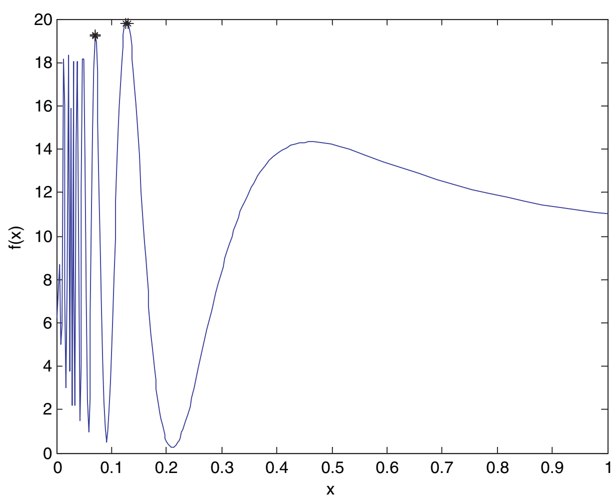

Fig. 6. Final positions of the chromosomes in experiment 1.

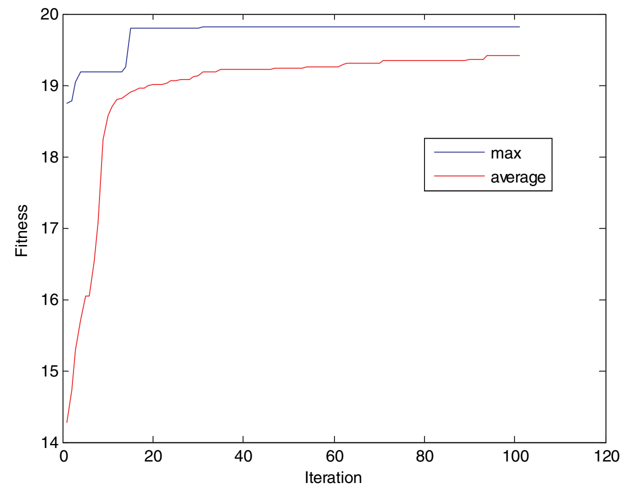

Fig. 7. The evolution curve of experiment 1.

The original chromosomes are produced with strong randomization, as is shown in Fig. 5. Figure 6 shows the final position of the chromosomes in the first experiment. It is obvious that it is easy to get into local premature convergence. The evolution curves presented in Fig. 7 also reflect the slow convergence speed. Table 3 shows the final specific results that contains both the maximum value and the average value of the chromosomes in the experiment.

Figures 8–11, and Tables 4, 5 show the results in experiment 2 and 3.

We also take another typical Benchmark continuous optimization problem as example, which is Rosenbrock function:

$$f(\bar{x}) = \sum_{i=1}^{10} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2, \quad x_i \in [-3, 3] \qquad (4)$$

Table 3. The test results of experiment 1.

Times	1	2	3	4	5	6	7	8	9	10
Max	19.8949	19.8871	19.8857	19.8575	19.8949	19.8942	19.8237	19.5677	19.8949	19.7726
Average	19.8933	19.8774	19.6257	19.8560	19.8858	19.8873	19.3410	18.9658	19.8943	19.7636
Times	11	12	13	14	15	16	17	18	19	20
Max	19.8949	19.5046	19.8949	19.8949	19.8948	19.8947	19.8425	19.8936	19.8866	19.8495
Average	19.8851	19.1500	19.8948	19.8949	19.8947	19.8863	19.7341	19.8843	19.8435	19.4081
					Average(max)	19.8412				
					Average(average)	19.7233				

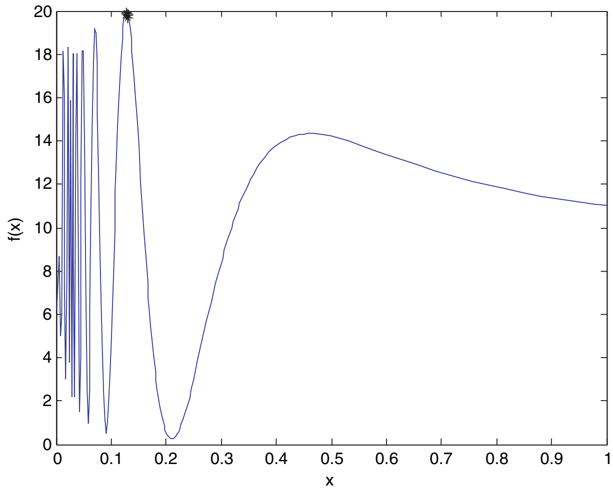

Fig. 8. Final positions of the chromosomes in experiment 2.

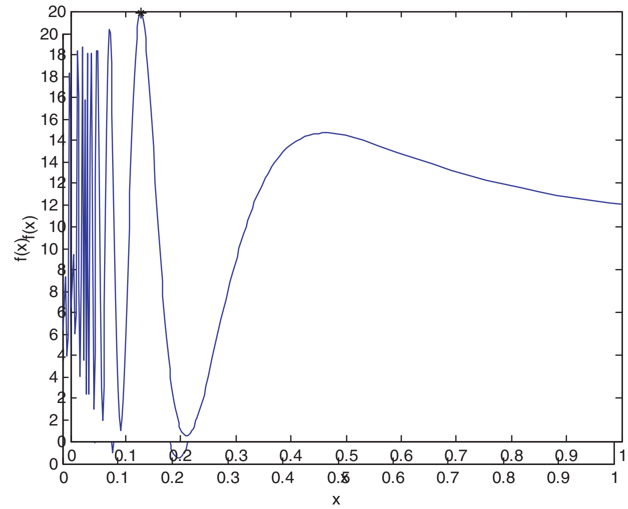

Fig. 10. Final positions of the chromosomes in experiment 3.

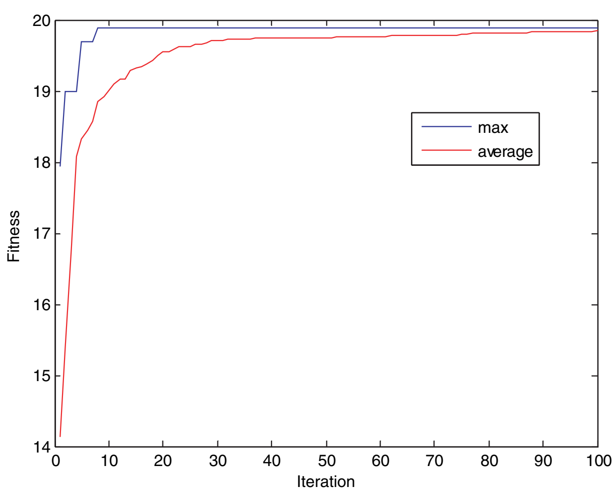

Fig. 9. The evolution curve of experiment 2.

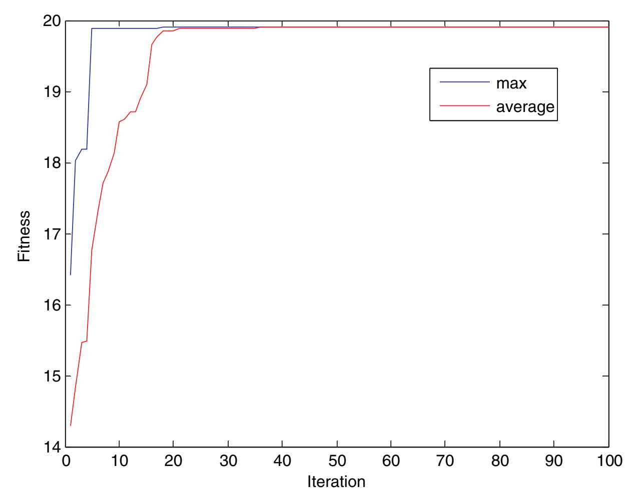

Fig. 11. The evolution curve of experiment 3.

Table 4. The test results of experiment 2.

Times	1	2	3	4	5	6	7	8	9	10
Max	19.8948	19.8913	19.8948	19.8948	19.8949	19.8949	19.8949	19.7680	19.8404	19.8949
Average	19.7840	19.8424	19.8899	19.8948	19.8877	19.8949	19.8949	19.5771	19.7516	19.8949
Times	11	12	13	14	15	16	17	18	19	20
Max	19.7714	19.8894	19.8949	19.8949	19.8949	19.7737	19.8949	19.8362	19.8904	19.8933
Average	19.7532	19.8853	19.8949	19.8949	19.8948	19.7737	19.8947	19.7553	19.8802	19.8926
					Average(max)	19.8699				
					Average(average)	19.8416				

Table 5. The test results of experiment 3.

Times	1	2	3	4	5	6	7	8	9	10
Max	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949
Average	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949
Times	11	12	13	14	15	16	17	18	19	20
Max	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949
Average	19.8949	19.8949	19.8949	19.5014	19.8949	19.8949	19.8949	19.8949	19.8949	19.8949
					Average(max)	19.8949				
					Average(average)	19.8736				

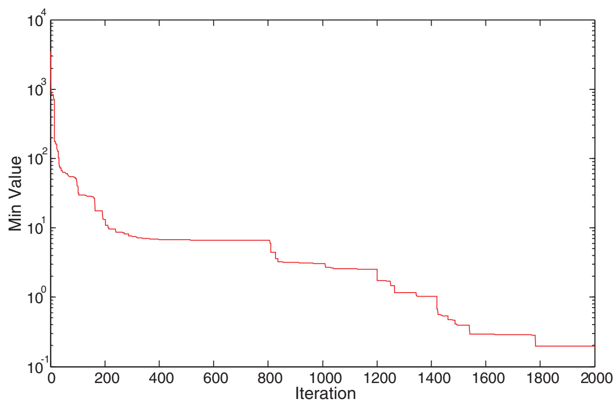

Fig. 12. The evolution curve in experiment 1.

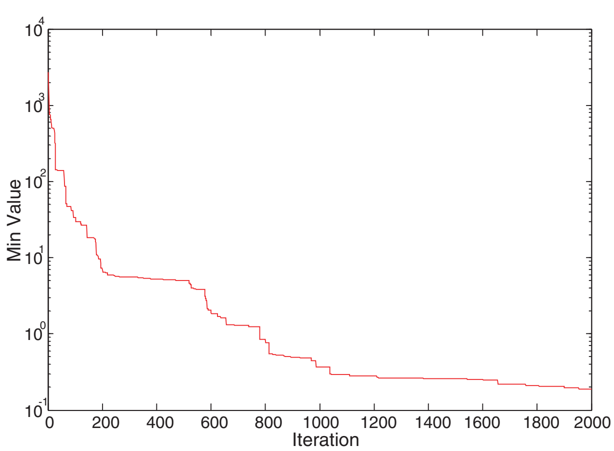

Fig. 13. The evolution curve in experiment 2.

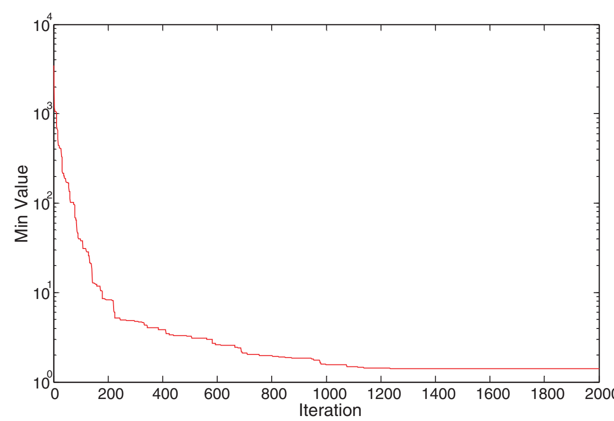

Fig. 14. The evolution curve in experiment 3.

Our aim is to find the minimum value of the above Rosenbrock function. Figures 12–14 show the evolution curves in experiment 1, 2 and 3.

It is obvious that our proposed hybrid ABC and QEA model can find better solutions than the other QEA models in solving the same continuous optimization problem, and the hybrid model can avoid premature which happens in the first and the second experiments. Our proposed hybrid ABC and QEA has a more excellent performance with strong ability to find optimal solution, and the convergence speed is also very quick.

5. Conclusions

This paper presented a hybrid ABC and QEA for solving the continuous optimization problems. A number of experiments are conducted on a set of well-known Benchmark continuous optimization problems and the related results are compared with two other QEAs: the QEA with classical crossover operation, and the QEA with 2-crossover strategy. The series experimental results verify that the proposed hybrid ABC and QEA model is a more practical and effective algorithm in solving complex continuous optimization problems, and also a feasible method for other complex real-world optimization problems.

Our future work will focus on applying the newly proposed ABC and QEA approach in this paper to solve other more complicated optimization problems. Furthermore, we are also interested in the theoretical analysis on the proposed hybrid ABC and QEA model.

Acknowledgments

The authors are grateful to the editor and reviewers for their kind help and constructive comments that helped to improve the presentation of the paper.

This work was partially supported by the National High Technology Research and Development Program of China (863 Program), Natural Science Foundation of China (NSFC) under grant #60975072 and #60604009, Program for New Century Excellent Talents in University (NCET) of China, Aeronautical Science Foundation of China under grant #2006ZC51039, Beijing NOVA Program Foundation of China under grant #2007A017, and "New Scientific Star in Blue Sky" Talent Program of Beihang University of China.

References

1. H. Adeli and N.-T. Adeli, Augmented lagrangian genetic algorithm for structural optimization, *Journal of Aerospace Engineering* ASCE **7**(1) (1994a) 104–118.
2. H. Adeli and N.-T. Cheng, Concurrent genetic algorithms for optimization of large structures, *Journal of Aerospace Engineering* ASCE **7**(3) (1994b) 276–296.
3. H. Adeli and S. Kumar, Distributed genetic algorithms for structural optimization, *Journal of Aerospace Engineering* **8**(3) (1995a) 156–163.
4. H. Adeli and S. Kumar, Concurrent structural optimization on a massively parallel supercomputer, *Journal of Structural Engineering* ASCE **121**(11) (1995b) 1588–1597.
5. A. N. Al-Rabadi, New dimensions in non-classical neural computing, part II: Quantum, nano, and optical, *International Journal of Intelligent Computing and Cybernetics* **2**(3) (2009) 513–573.
6. T. M. Cheng and R. Z. Yan, Integrating messy genetic algorithms and simulation to optimize resource utilization, *Computer-Aided Civil and Infrastructure Engineering* **24**(6) (2009) 401–415.
7. J. Holland, *Adaptation in Natural and Artificial Systems* (Ann Arbor, MI: The University of Michigan Press, 1975).
8. S. L. Hung and H. Adeli, A parallel genetic/neural network learning algorithm for MIMD shared memory machines, *IEEE Transactions on Neural Networks* **5**(6) (1994) 900–909.
9. X. Jiang and H. Adeli, Neuro-genetic algorithm for nonlinear active control of highrise buildings, *International Journal for Numerical Methods in Engineering* **75**(8) (2008) 770–786.
10. M. W. Kang, P. Schonfeld and N. Yang, Prescreening and repairing in a genetic algorithm for highway alignment optimization, *Computer-Aided Civil and Infrastructure Engineering* **24**(2) (2009) 109–119.
11. D. Karaboga, An idea based on honey bee swarm for numerical optimization. Technical Report-TR06 (2005).
12. D. Karaboga and B. Basturk, A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm, *Journal of Global Optimization* **39**(3) (2007) 459–171.
13. D. Karaboga and B. Basturk, On the performance of artificial bee colony (ABC) algorithm, *Applied Soft Computing* **8**(1) (2008) 687–697.
14. N. Karaboga, A new design method based on artificial bee colony algorithm for digital IIR filters, *Journal of the Franklin Institute* **346**(4) (2009) 328–348.
15. A. Kaveh and S. Shojaee, Optimal design of scissor-link foldable structures using ant colony optimization algorithm, *Computer-Aided Civil and Infrastructure Engineering* **22**(1) (2007) 56–64.
16. H. Kim and H. Adeli, Discrete cost optimization of composite floors using a floating point genetic algorithm, *Engineering Optimization* **33**(4) (2001) 485–501.
17. J. Kennedy, R. Eberhart, Particle swarm optimization, in *Proceedings of the IEEE International Conference on Neural Networks* (Piscataway, N.J.) Vol. 4, pp. 1942–1948.
18. H. Lee, E. Kim and M. Park, A genetic feature weighting scheme for pattern recognition, *Integrated Computer-Aided Engineering* **14**(2) (2007) 161–171.
19. S. Mathakari, P. Gardoni, P. Agarwal, A. Raich and T. Haukaas, Reliability-based optimal design of electrical transmission towers using multi-objective genetic algorithms, *Computer-Aided Civil and Infrastructure Engineering* **22**(4) (2007) 282–292.
20. B. Mersch, T. Glasmachers, P. Meinicke and C. Igel, Evolutionary optimization of sequence kernels for detection of bacterial gene starts, *International Journal of Neural Systems* **17**(5) (2007) 369–381.
21. K. Sarma and H. Adeli, Fuzzy genetic algorithm for optimization of steel structures, *Journal of Structural Engineering*, ASCE **126**(5) (2000a) 596–604.
22. K. Sarma and H. Adeli, Fuzzy discrete multicriteria cost optimization of steel structures, *Journal of Structural Engineering*, ASCE **126**(11) (2000b) 1339–1347.

23. K. C. Sarma and H. Adeli, Bi-level parallel genetic algorithms for optimization of large steel structures, *Computer-Aided Civil and Infrastructure Engineering* **16**(5) (2001) 295–304.
24. K. C. Sarma and H. Adeli, Life-cycle cost optimization of steel structures, *International Journal for Numerical Methods in Engineering* **55**(12) (2002) 1451–1462.
25. J. F. Smith, H. ThanhVu and T. H. Nguyen, Autonomous and cooperative robotic behavior based on fuzzy logic and genetic programming, *Integrated Computer-Aided Engineering* **14**(2) (2007) 141–159.
26. M. H. N. Tayarayi and M. R. Akbarzadeh, A cellular structure and diversity preserving operator in quantum evolutionary algorithms, in *Proceedings of the IEEE Congress on Evolutionary Computation* (Hong Kong, 2007), pp. 2670–2675.
27. F. Teklu, A. Sumalee and D. Watling, A genetic algorithm for optimizing traffic control signals considering routing, *Computer-Aided Civil and Infrastructure Engineering* **22**(1) (2007) 31–43.
28. B. J. Vitins and K. W. Axhausen, Optimization of large transport networks using the ant colony heuristic, *Computer-Aided Civil and Infrastructure Engineering* **24**(1) (2009) 1–14.
29. E. I. Vlahogianni, M. G. Karlaftis and J. C Golias, Spatio-temporal short-term urban traffic flow forecasting using genetically-optimized modular networks, *Computer-Aided Civil and Infrastructure Engineering* **22**(5) (2007) 317–325.
30. M. Wei, Y. X. Li, D. Z. Jiang, Y. F. He, X. Y. Huang and X. Xu, A new evolutionary algorithm based on quantum statistical mechanics, *Proceedings of the IEEE Congress on Evolutionary Computation* (Hong Kong, 2008), pp. 1722–1726.
31. J. Xiao, Y. P. Yan, Y. Lin, L. Yuan and J. Zhang, A quantum-inspired genetic algorithm for data clustering, *Proceedings of the IEEE Congress on Evolutionary Computation* (Hong Kong, 2008), pp. 1513–1518.
32. Z. H. Xing, H. B. Duan and C. F. Xu, An improved quantum evolutionary algorithm with 2-crossovers, *Lecture Notes in Computer Science (ISNN 2009)*, Part I, Vol. 5551 (Springer-Verlag, Berlin Heidelberg, 2009), pp. 735–744.
33. S. Y. Yang, F. Liu and L. C. Jiao, The quantum evolutionary strategies, *Acta Electronica Sinica* **29**(12) (2001) 1873–1877.
34. Z. Yang, B. Yu and C. Cheng, A parallel ant colony algorithm for bus network optimization," *Computer-Aided Civil and Infrastructure Engineering* **22**(1) (2007) 44–55.
35. R. Zhang and H. Gao, Improved quantum evolutionary algorithm for combinatorial optimization problem. *Proceedings of the 6th International Conference on Machine Learning and Cybernetics* (Hong Kong, 2007), pp. 3501–3505.
36. W. F. Zhang, Z. K. Shi and Z. Y. Luo, Prediction of urban passenger transport based-on wavelet SVM with quantum-inspired evolutionary algorithm, *Proceedings of the International Joint Conference on Neural Networks* (Hong Kong, 2008), pp. 1510–1514.