

# Adaptive Collaborative Quantum-Inspired Evolutionary Algorithm for Global Numerical Functions

Zhou Liang<sup>1,\*</sup>, Shao Ming<sup>2</sup>, and Ma Chengqian<sup>3</sup>

<sup>1</sup>School of Information Management, Shanghai Lixin University of Accounting and Finance, Shanghai, China

<sup>2</sup>School of Management, Shanghai University of Engineering Science, Shanghai, China

<sup>3</sup>The college of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai, China

**Abstract.** – A novel adaptive collaborative quantum-inspired evolutionary algorithm (ACQEA) is proposed by combining the collaborative evolution and adaptive mutation mechanism together in this paper. In ACQEA, the whole population will be divided into multi sub-populations which can complete the evolution independently, and then the collaborative evolution mechanism is used to make these multi sub-populations full exchange their evolution information in operation process. In addition, the adaptive mutation and update strategies are implemented in order to give ACQEA the power to explore its search space on the basis of characteristic information of the elite individual and corresponding population diversity. Finally, the proposed ACQEA is compared with existing quantum evolution algorithm (QEA) in solving global numerical functions and the experiments results verify that the advantages of ACQEA on convergence rate and searching accuracy.

## 1 Introduction

Quantum-inspired evolutionary algorithm (QEA) is proposed by combining quantum computing theory with evolutionary theory. It utilizes the concepts of quantum bit, superposition of states and collapse of states and it applies quantum rotation gate to guiding the searching direction into the optimal area and accelerating the algorithm's convergence speed [1]. Thus QEA have a number of computational advantages compared with the traditional optimization algorithms. QEA is still easy to fall into local optimum, especially for complex global optimization problems [2]. Recently, numerous literatures about improved QEA have been reported which focus on making its performance better, and these works can be summarized as following three aspects.

(1) The new encoding methods are proposed on QEA to improve the global search ability. Li Panchi et al. used Bloch coordinates comprise gene chain to represent quantum individuals, which can accelerate the convergence process [3]; Zhao Shuanfeng et al. proposed a real-coded chaotic quantum genetic algorithm based on the chaotic and coherent characters of Q-bits [4]; Zhang Lei et al. adopt phase angle and its corresponding real mapping to construct quantum individual with probabilistic representation, which can represent all the feasible solution probabilistically [5]; Jianyong Liu et al. had thought conventional quantum genetic algorithm (QGA) could be used to solve combinatorial optimization problems well than real-valued problems, so they proposed Real-coded QGA [6].

(2) New evolutionary strategies are adopted into QEA to improve its performance. Jianyong Liu et. al. combine quantum coding, immune clone and niche mechanism

together to solve multi-modal function optimization more effectively [7]; Tzzy-Chyang Lu proposed a region-based QEA, the algorithm adopted quantum bit representation for the selection of regions; Its search process uses a region-by-region exploration in the beginning and, as the candidate regions are identified, a randomized search in good regions is employed for exploitation [8]; Qian Jie et al. proposed a statistical learning QEA. Experiments had shown that this new design could avoid premature convergence [9].

(3)The hybrid approaches are proposed which combines the traditional version with other optimization algorithms. Yu Yang et al. combined QEA with PSO and proposed two hybrid QEAs, one is to embed the evolutionary equation of PSO into QEA, while the other one is applied the quantum chromosomes of QEA to binary PSO [10]; Li Zhiyong et al. proposed a novel quantum bat algorithm, which is based on quantum evolution algorithm and bat algorithm [11]; Wu Shun-Chi et al. proposed a two-stage scheme that combines quantum evolutionary algorithm and tabu search, simultaneously possessed the capabilities of exploration and exploitation and could be effectively employed for LP design with less computational load [12].

In addition, co-evolutionary algorithms (CEAs), as a new evolutionary algorithm framework, focus on the idea of mutual influence or co-ordination among different species to ensure algorithms have better search accuracy and search efficiency [13]. In recent years, some researchers began to combine co-evolutionary mechanism with QEA and then proposed some improved QEAs. Gu et al. proposed a novel competitive co-evolutionary quantum genetic algorithm which included

\* Corresponding author: [hzhou0168@yahoo.com](mailto:hzhou0168@yahoo.com)

three new strategies named as competitive hunter, cooperative surviving and the big fish eating small fish [14]. Xiong, Gui et al. proposed a double population co-evolution algorithm based on the quantum evolution algorithm and difference evolution algorithm. In their algorithm, one population is used for global searching; the other one is for partial searching [15]. Zhang proposed an elite collaborative quantum evolution algorithm. Two sub-populations were selected randomly. Subsequently, the corresponding elite individuals from the selected two sub-populations would construct a mutually beneficial relationship in order to implement co-evolutionary operation, and guide the individual populations evolve toward the optimal solution [16].

In this paper, the co-evolutionary mechanisms are introduced into QEA in order to propose Adaptive Collaborative Quantum-inspired Evolutionary Algorithm (ACQEA). This new ACQEA will fully extract the characteristic information of every sub-population in order to lead the algorithm has a better performance. The rest of this paper is organized as following. The preliminary knowledge of co-evolutionary algorithm and quantum evolution algorithm is explained in Section 2; section 3 gives the details of the proposed algorithm; section 4 shows the experimental simulation using the novel algorithm; finally, conclusions are drawn in section 5.

## 2 Preliminary knowledge

### 2.1 Co-evolutionary algorithm

Now there are two structural forms for CEAs to implement evolution. One is Competitive Co-evolutionary Algorithm (Com-CEA) and the other one is Cooperative Co-evolutionary Algorithm (Coo-CEA) [17][18]. In Coo-CEA, the fitness of individual is not evaluated separately. It is firstly in accordance with the prior knowledge of solving problem; subsequently, it combines the "representatives" individuals from the other sub-populations to construct a solution vector of solving problem and then to evaluate the fitness of individuals. This process is called "cooperation"; the cooperative co-evolution mechanism also provides a generalized evolution framework to solve the high-dimensional or complex optimization problems [19].

As for Com-CEA, the individual fitness in a sub-population is determined by the competition results which include a series of competition with the individuals in other sub-population. At this time, these two sub-populations always take the role of "Host" population and "Parasite". The evolutionary operations will use these populations to produce new sub-populations. With the development of the evolution, these different sub-populations will show different genetic characteristics.

### 2.2 Quantum evolution algorithm

Quantum evolution Algorithm (QEA) has advantages of quantum computing and evolutionary computation. It uses quantum bit (Q-bit) as a probabilistic representation to improve the diversity of population. It also uses quantum rotation gate (Q-gate) as the update operation, which can guide the search direction of individual to the optimal area, and increase the algorithm's convergence speed. In the traditional QEA, a Q-bit may be in the "1" state, "0" state, or any superposition of these two states. It can be represented as following:

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (2-1)$$

Where  $\alpha$  and  $\beta$  are complex numbers that specify the corresponding probability amplitudes of the states "0" and "1". Normalization of the two states to unity always guarantees:

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

So a Q-bit in QEA can be defined with a pair of numbers:  $[\alpha \ \beta]^T$ .

In general, a quantum individual is defined as a string of Q-bits,  $q_i = [\alpha_{i1} \ \alpha_{i2} \ \dots \ \alpha_{im}]$ ; the population  $Q(t) = \{q_1, q_2, \dots, q_n\}$ , where  $i$  is the size of population, and  $m$  is the number of Q-bits. Since the Q-bits representation is able to express as a linear superposition of states probabilistically, evolutionary computing with the Q-bit representation is often has a better performance than traditional approaches.

The main evolutionary way of QEA is realized by Q-gate, which transfer the probability of Q-bits converge either to "1" (or "0"). Q-gate can be designed according to the actual problem. In this paper, we use the common one as following.

$$U(\theta_i) = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \quad (2-3)$$

The update process of Q-gate and its related adjustment strategy is:

$$\begin{bmatrix} \alpha'_i \\ \beta'_i \end{bmatrix} = U(\theta_i) \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} \quad (2-4)$$

$$\theta_i = s(\alpha_i, \beta_i)\delta \quad (2-5)$$

Where  $\theta_i$  is the rotation angle whose size and sign is defined by adjustment strategy in literature [20]. By using Q-gates, the quantum individuals can be guided toward the better solutions throughout the iteration process.

## 3 Adaptive collaborative quantum evolutionary algorithm

This section proposes adaptive collaborative quantum evolutionary algorithm. Firstly, the initial population will be divided into multiple sub-populations; then the following new evolutionary strategies are introduced into these multiple sub-populations in order to improve its global optimization ability.

### 3.1 Elite individuals

Elite individual generally corresponds to the individual with a better fitness and contains corresponding

evolutionary characteristic information. Maintaining best Q-bit individuals as elite individual in every generation can avoid the possibility of losing high quality individuals. At the same point, it is even more powerful if the elitism is further strengthened and the solutions are spread out by quantum mechanism [21]. In ACQEA, every sub-population will produce an elite individual which actually contains the characteristics information of current sub-population in the implementation process of algorithm. Thus, these elite individuals can be collected as guiding individual. Accompanied by the evolution of the whole population, these guiding elites will be applied to help the poor performance sub-population and constantly updated.

### 3.2 Definition of population diversity

Current experience has shown that the population diversity is very important in giving the algorithm the power to explore the search space and not get trapped in local optima [22]. Normally, if the diversity of sub-populations is not identified at the end of iteration, it would be gradually reduced in the later iterative process of the algorithm. This could result in a gene decrease for effective individuals, and affect the optimization process of the algorithm and finally cause a premature convergence. Therefore, it is necessary to design a mechanism to maintain the diversity of the population.

In this section, a modified indicator of degree of population diversity is introduced to measure the diversity of sub-populations based on the reference [23]. Here the modified indicator includes the characteristic information of every individual in each sub-population and also is a parameter involved in the evolution of the population.

**Definition 3.1** When the Q-bits are applied to optimization problems, they will converge to the corresponding binary encoding space  $\{0,1\}^L$ , where L is the length of binary code, the population size is n. A group of individuals in this population can be described:  $Q = \{q_1, q_2, \dots, q_n\}$ , where  $q_j = \{q_{1j}, q_{2j}, \dots, q_{Lj}\}$ , and  $j = 1, 2, \dots, n$ . The degree of population diversity is defined as the following formula:

$$\text{div}(Q) = \exp\left(-\frac{|\sum_{l=1}^L \sum_{j=1}^n q_{lj} - \sum_{l=1}^L \sum_{j=1}^n (1-q_{lj})|}{\text{scale}}\right) \quad (3-1)$$

where scale is the size of individuals in population, and  $\text{scale} = L \times n$ ,  $\sum_{j=1}^n q_{lj}$  represent the numbers of binary code "1" of all the individuals in their lth bit in the population; while  $\sum_{j=1}^n (1 - q_{lj})$  represent the numbers of binary code "0" of all the individuals in their lth bit in the population. The result of calculating  $|\sum_{l=1}^L \sum_{j=1}^n q_{lj} - \sum_{l=1}^L \sum_{j=1}^n (1 - q_{lj})|$  indicates that the distribution of binary code "0" and "1" of all the individuals in the population. In each iteration, the greater the result, the more unbalanced distribution between the binary code "1" and "0", which also means the population diversity is in reducing. Finally, when the result of  $\text{div}(Q)$  becomes smaller, it shows the diversity of the population is getting worse. As mentioned in above formula (3-1), when the

ACQEA divides the population into many sub-populations, every sub-population will has its own  $\text{div}(\text{Sub\_}Q)$  as an indicator to measure the sub-population diversity during the entire evolutionary process.

### 3.3 Adaptive mutation and update mechanism

Mutation operator can produce a random disturbance for the candidate solution in order to get new generations. In general, the efficiency of evolutionary algorithm relies strongly on the performance of mutation operation [24]. The ACQEA in this section will implement mutation operation for the worst sub-population with smallest  $\text{div}(Q)$  in order to improve its diversity. Here should be noted that the change of  $\text{div}(Q)$  is related to the probability of mutation. Therefore, when the  $\text{div}(Q)$  becomes smaller, it means the diversity of population is getting worse, while the mutation probability of the sub-population will increase. Here the diversity of population and global searching abilities would be improved by adaptively changing of the mutation probability.

With this process, if the original value of the Q-bit is '0', then it will turn into '1' after this mutation operation; the other way around is '1', then it will turn into '0' after mutation operation. When the QEA used Q-gate to update these sub-populations, the best individual with in the elite library will be used to guide and update the following two sub-populations instead of the whole. One is the sub-population which is corresponding to the worst individual in elite library, and the other is the worst diversity of the population in order to ensure the poor sub-populations have a better optimization performance. The remaining sub-populations will be updated by their own best individuals.

As a whole, the mutation and update strategy is actually equivalent to build up an information exchange mechanism between the best sub-population and the worst sub-population, at the same time, this mechanism also ensure some of the sub-populations can maintain their independent evolutions.

## 4 Simulation tests

In order to illustrate the effectiveness and performance of the proposed ACQEA, we compare it with the traditional QEA [25] by using the following typical global numerical functions with diverse complexities from virtual library of Simulation Experiments in Simon Fraser University. These functions listed below are some of the common functions used for testing optimization algorithms, and all of them have many local minima points. The maximum number of the iteration of these two algorithms is 100, and the initial population size is 200. The population of ACQEA is divided into four sub-populations with size of 50 each.

### 4.1 Description of the test functions

These test functions are listed as following:

**Function 1: Ackley Function**

$$f(x) = -a \exp \left( -b \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2} \right) - \exp \left( \frac{1}{d} \sum_{i=1}^d \cos(cx_i) \right) + a + \exp(1)$$

Where  $d$  is the dimensions; The Ackley function is widely used for testing optimization algorithms. In its two-dimensional form in the plot above, it is characterized by a nearly flat outer region, and a large hole at the centre. The function poses a risk for optimization algorithms, particularly hillclimbing algorithms, to be trapped in one of its many local minima. Recommended variable values are:  $a = 20$ ,  $b = 0.2$  and  $c = 2\pi$ . Its corresponding input domain is on  $x_i \in [-32.768, 32.768]$ , for all  $i = 1, 2, \dots, d$ . Its global minimum  $f(x^*) = 0$  at  $x^* = (0, 0, \dots, 0)$ .

**Function 2: Bukin Function N. 6**

$$f(x) = 100 \sqrt{|x_2 - 0.01x_1^2|} + 0.01|x_1 + 10|$$

The sixth Bukin function has many local minima, all of which lie in a ridge; and it is usually evaluated on the rectangle  $x_1 \in [-15, -5]$ ,  $x_2 \in [-3, 3]$ . Its global minimum  $f(x^*) = 0$  at  $x^* = (-10, 1)$ .

**Function 3: Cross-in-Tray Function**

$$f(x) = -0.0001 \left( \left| \sin(x_1) \sin(x_2) \exp \left( \left| 100 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi} \right| \right) \right| + 1 \right)^{0.1}$$

The Cross-in-Tray function has multiple global minima. It is shown here with a smaller domain in the second plot, so that its characteristic "cross" will be visible. The function is usually evaluated on the square  $x_i \in [-10, 10]$ , for all  $i = 1, 2$ . Its global minimum  $f(x^*) = -2.06261$  at:

$$x^* = (1.3491, -1.3491), (1.3491, 1.3491), (-1.3491, 1.3491), (-1.3491, -1.3491).$$

**Function 4: Drop-Wave Function**

$$f(x) = -\frac{1 + \cos(12\sqrt{x_1^2 + x_2^2})}{0.5(x_1^2 + x_2^2) + 2}$$

The Drop-Wave function is multimodal and highly complex. The second plot above shows the function on a smaller input domain, to illustrate its characteristic features. The function is usually evaluated on the square  $x_i \in [-5.12, 5.12]$  for all  $i = 1, 2$ . Its global minimum  $f(x^*) = -1$  at:  $x^* = (0, 0)$ .

**Function 5: Eggholder Function**

$$f(x) = -(x_2 + 47) \sin \left( \sqrt{\left| x_2 + \frac{x_1}{2} + 47 \right|} \right) - x_1 \sin \left( \sqrt{|x_1 - (x_2 + 47)|} \right)$$

The Eggholder function is a difficult function to optimize, because of the large number of local minima. The function is usually evaluated on the square  $x_i \in [-512, 512]$ , for all  $i = 1, 2$ . Its global minimum  $f(x^*) = -959.6407$  at:  $x^* = (512, 404.2319)$ .

All the above numerical optimization benchmark functions have many local minima around the global minimum, therefore they can be used to test the global search ability of the algorithm and its ability to escape from local optima. Notices that the above numerical optimization benchmark functions are used to solve the minimum, and we will reverse these optimization functions into solving the maximum in order to handling uniformity. Each algorithm runs 10 times independently. The statistical results including the best optimal result, the average result, the worst results, and the standard deviations are described as following table.

**4.2 Simulation results**

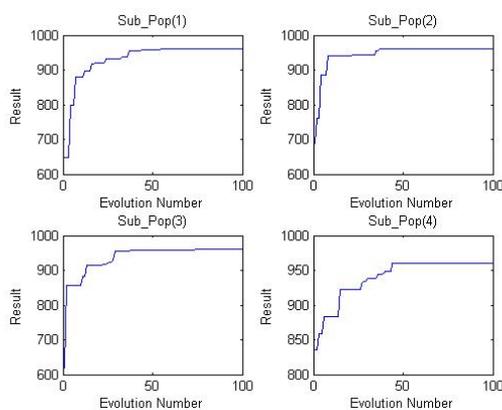
Table 1 shows the overall statistical results of QEA and ACQEA for the above functions. The statistical results indicate that the ACQEA is superior to QEA from the view of quality of solutions when these two optimization algorithms perform the same functions from table 1. The results show that the ACQEA is more efficient in finding the global optimal solution and robustness than QEA. The standard deviation indicates that ACQEA has a good stability and widespread adaptability.

**Table 1.** Statistical Results of QEA and ACQEA for Optimization Functions (with 10 experiments)

Test Functions	QEA			
	Best	Worst	Average	Standard Deviation
$f_1(x)$	-1.2505e-004	-1.1839e-003	-4.8783e-004	3.69073e-004
$f_2(x)$	0.036344	-0.12651	-0.081246	0.026744
$f_3(x)$	2.0626	2.0626	2.0626	0
$f_4(x)$	1	0.93625	0.98725	0.0255
$f_5(x)$	959.6407	955.1846	957.9439	1.92930
Test Functions	ACQEA			
	Best	Worst	Average	Standard Deviation
$f_1(x)$	-1.2505e-004	-9.7945e-004	3.41813e-004	2.95544e-004
$f_2(x)$	0.021733	-0.10834	-0.058894	0.028338
$f_3(x)$	2.0626	2.0626	2.0626	0
$f_4(x)$	1	0.99999	0.999999	2.999999e-006
$f_5(x)$	959.6407	955.3586	958.1830	1.76522

In addition, we take the evolutionary results for function 5 based on ACQEA for example, although each sub-population is evolving independently, these four sub-populations will become synchronized toward convergence in the later iterative process. Here for function 5, their corresponding sub-populations can almost converge to their optimal solutions throughout the iteration cycle at the same time. As for function 5 in Figure 1, the optimal solutions in sub-population 1, 2, 3,

4 are (512, 404.2079), (511.999, 404.2186), (511.999, 404.246), (512, 404.2323) respectively; and their corresponding optimal values are 959.64, 959.6372, 959.6371, 959.6407. The results also present the following three conclusions. Firstly, the new evolutionary strategies are effective for the adaptive collaborative quantum evolutionary algorithm. On the other hand, the elite individual in the constructing elite library just guide the selected sub-population instead of entire population toward the global optimum, which is enough to complete the optimize search and can improve the operating efficiency of the algorithm to a certain degree. Thirdly, Our ACQEA can use smaller populations to complete complex optimization problems with the help of co-evolutionary mechanism and performs better than traditional QEA regardless the complexity of test functions.



**Fig.1.** Evolutionary results for Function 5 on the basis of ACQEA

## 5 Conclusions

An Adaptive Collaborative Quantum Evolutionary Algorithm is proposed in this paper. In ACQEA, the whole population is firstly divided into multiple sub-populations which complete the evolution process independently; and then makes full use of the characteristic information of each sub-population with the new evolutionary strategy to implement the evolution process. ACQEA is compared with existing QEA using the established test functions. The results of simulation experiments for ACQEA show that its convergence performance is superior to QEA; and ACQEA is more accurate and stable in finding the global value in the functions. In the future work, how to apply the algorithm to the more complex multi-objective optimization problems and adjust the parameters of this algorithm in order to ensure it can get the best performance need more research.

## References

1. M. Shao, L. Zhou, Novel Quantum-Inspired Co-evolutionary Algorithm, *International Journal of Security and Its Applications*, **10**, 2 (2016)
2. Y.G. Cai, M.J. Zhang, H. Cai, Y. Zhang, Hybrid Chaotic Quantum Evolutionary Algorithm, *Systems Engineering – Theory & Practice*, **32**, 10 (2012)
3. P.C. Li, S.Y. Li. Quantum-inspired evolutionary algorithm for continuous space optimization based on Bloch coordinates of qubits, *Neurocomputing*, **72**, 1/2/3 (2008)
4. S.F. Zhao, G.H. Xu, T.F. Tao, L. Lin, Real-coded chaotic quantum-inspired genetic algorithm for training of fuzzy neural networks, *Computers and Mathematics with Applications*, **57**, 11-12 (2009)
5. L. Zhang, Y.W. Fang, D.H. Mao, P.F. Yang, A new phase angle encoded quantum evolutionary algorithm, *Control and Decision*, **30**, 4 (2015)
6. J.Y. Liu, H.X. Wang, Y.Y. Sun, C.Q. Fu, J. Guo, Real-Coded Quantum-Inspired Genetic Algorithm-Based BP Neural Network Algorithm, *Mathematical Problems in Engineering*, **2015**, (2015)
7. J.Y. Liu, H.X. Wang, Y.Y. Sun, L. Li, Adaptive niche quantum-inspired immune clonal algorithm, *Natural Computing*, **15**, 2 (2016)
8. T.C. Lu, J.C. Juang, A region-based quantum evolutionary algorithm for global numerical optimization, *Journal of Computational and Applied Mathematics*, **239** (2013)
9. J. Qian, J.G. Zheng, A Quantum Evolutionary Algorithm Based on Population Statistical Learning, *Journal of Xi'an Jiaotong University*, **46**, 2 (2012)
10. Y. Yu, Z.F. Yin, Y.F. Tian, Hybrid Quantum Evolutionary Algorithms and its Application, *Computer Engineering and Applications*, **42**, 28 (2006)
11. Z.Y. Li, L. Ma, H.Z. Zhang, Quantum Bat Algorithm for Function Optimization, *Journal of System and Management*, **23**, 5 (2014)
12. S.C. Wu, T.H. Chan, Hsieh, M. Shao, C. Lin, Quantum evolutionary algorithm and tabu search in pressurized water reactor loading pattern design, *Annals Of Nuclear Energy*, **94**, (2016)
13. X.Y. Liu, J. Liu, G.R. Zhao, J. Ma, Y.C. Wang, Study on cooperation evolution model, *Computer Engineering and Applications*, **48**, 8 (2012)
14. J.W. Gu, M.Z. Gu, C.W. Cao, X.S. Gu, A novel competitive co-evolutionary quantum genetic algorithm for stochastic job shop scheduling problem, *Computers & Operations Research*, **37**, 5 (2010)
15. F.Q. Xiong, W.H. Gui, C.H. Yang, Y.G. Li, A double population co-evolution algorithm for

- process of zinc hydrometallurgy, *Control and Decision*, **28**, 4 (2013).
16. W.F. Zhang, Elite Collaborative Quantum Evolution Algorithm for Combinatorial Optimization Problem, *Journal of Hubei University of Automotive Technology*, **27**, 3 (2013)
  17. M.Q. Li., D. Lin., J.S. Kou, Dynamics of Fitness Sharing Evolutionary Algorithms for Coevolution of Multiple Species, *Applied Soft Computing*, **10**, 3 (2010)
  18. X.Y. Sun, D.W. Cong, Varying population size cooperative coevolutionary genetic algorithm and its application in optimization, *Control and Decision*, **19**,12 (2004)
  19. X.X. Hu,Y.L. Yin, Cooperative Co-Evolutionary Cuckoo Search Algorithm for Continuous Function Optimization Problems, *Pattern recognition and Artificial Intelligence*, **26**, 11 (2013)
  20. L.F. Zhang, X.P. Zhang, Network traffic prediction based on BP neural networks optimized by quantum genetic algorithm, *Computer Engineering and Science*, **38**, 1 (2016)
  21. B. Jerzy, D. Grzegorz, M. Jan, Multi-criterion, Evolutionary and Quantum Decision Making in Complex Systems, *Recent Advances in Applied & Biomedical Informatics and Computational Engineering in Systems Applications* (WSEAS PRESS, 2011)
  22. H. Mohammad. N.Tayarani, M. R. Akbarzadeh-T, Improvement of the performance of the Quantum-inspired Evolutionary Algorithms: structures, population, operators, *Evolutionary Intelligence*, **7**, 4 (2014)
  23. Q. Liu, X.Y. Wang, Q.M. Fu, Y.G. Zhang, X.F. Zhang, Double Elite Coevolutionary Genetic Algorithm, *Journal of Software*, **23**, 4 (2012)
  24. S.H. Wen, J.H. Zheng, M.Q. Li, Comparison and research of mutation operators in multi-objective evolutionary algorithm, *Computer Engineering and Applications*, **45**, 2 (2009)
  25. J. Qian, J.G. Zheng, C.Q. Zhang, X. Wang, R.X. Yan, Reviews of current studying progress on quantum evolutionary computation, *Control and Decision* , **26**, 3 (2011)