The Novel Co-evolutionary Quantum Evolution Algorithm and Its Applications

Liang Zhou, Yu Sun and Ming Shao

Abstract**—**To fully explore the potential features of traditional Quantum-inspired Evolution Algorithms (QEA) in optimizing design problems, this paper proposed a novel quantum-inspired evolution algorithm with co-evolutionary mechanism (NCQEA). In the proposed new structure, the quantum state population is firstly divided into multiple sub-populations which can complete the evolution process independently. In the course of evolution, every sub-population will produce an elitist individual; then these elitist individuals from every sub-population are selected to construct an elite library and the individual in this elite library can be used to help the poor sub-population to find the global optimal solution or near-optimal solution. In addition, this algorithm is also to define a new diversity indicator for every sub-population which can be used to measure its corresponding population diversity on the basis of characteristic information of every sub-population. As for the sub-population with poor diversity, the mutation strategies are implemented in order to give the algorithm the power to explore its search space. Finally, simulation experiments are performed on global numerical optimization functions and Knapsack problems, and the results indicate that the new co-evolutionary algorithm develops better performance than the traditional QEA.

Keywords**—**Co-evolutionary; Elitist individual; Global numerical optimization; Knapsack problem.

I. Introduction

NOWADAYS the intelligent optimization algorithms are usually used as the first candidates to solve some complex optimization problems. These traditional intelligent optimization algorithms (TIOAs), represented by Genetic algorithms (GAs), often applied the Darwin's theory of evolution, such as the survival of the fittest, natural selection and competition. These algorithms have many advantages in the actual optimization application, such as quick convergence, robustness and high efficiency capability of searching the optimal solution within the whole defined space, which make them be widely used in many fields [1]. However, these traditional intelligent optimization algorithms have some demerits when they are in the process of evolution; for example, they are often easy to fall into local optimum, and the corresponding populations have poor diversity at the end of iteration which will lead to premature. These demerits have drawn much attention to the research in improving the performance of intelligent optimization algorithm by using co-evolutionary mechanism. As a new evolutionary algorithm framework, co-evolutionary mechanism has been a hot research topic of computational intelligence in recent years. Co-evolution is inspired by the practical ecosystem, and had firstly proposed by Ehrlich and Raven [2]. The co-evolution emphasizes the mutual influence or co-ordination among the different populations and the environments. The co-evolutionary algorithms (CEAs) focus on the interaction and mutual influence among different populations to achieve the co-evolution and improve the optimization performance.

There are a lot of differences between TIOAs and CEAs in terms of operation mechanisms. The TIOAs, such as genetic algorithm and particle swarm optimization algorithm, start their evolution process from generating a random variable population, and then take the evolutionary strategy of "survival of the fittest" to solve global optimization problems; while the co-evolution process is broadly defined based on the population density, the population itself and the evolution of genetic composition between interacting populations. Comparing with TIOAs, CEAs have a better search capability, progressive learning ability to overcome premature convergence and robustness properties [3]. As for the co-evolutionary individuals, they are often affected by the following three factors when they are in the evolutionary process: the individual fitness, located environment and the competition with each other. These three factors coordinating with each other can effectively solve the singleness problem of the traditional evolutionary model, thus can maintain the diversity of the population better, and avoid premature and slow convergence issues [4].

Meanwhile, K. H. Han et al. firstly proposed Genetic Quantum Algorithm (GQA) in 2000, which is also considered to be one of the earliest models of quantum evolution algorithm (QEA); and soon afterward, they expanded the GQA and presented quantum-inspired evolutionary algorithm. QEA utilizes the concepts of quantum bit (Q-bit), superposition of

This work performed in the framework of the research project funded in part by the Jiangsu Planned Projects for Post-Doctoral Research Funds, the special fund for Shanghai colleges' outstanding young teachers' scientific research projects (Project No. ZZGJD12033 and ZZLX13004) and Ministry of Education in China Project of Humanities and Social Sciences (Project No. 13YJCZH147).

L. Zhou is with the School of Mathematics and Information Science, Shanghai Lixin University of Commerce, 2800 Wenxiang Rd, Songjiang, Shanghai, CHINA (e-mail: hzhouatlx@163.com)

Y. Sun is with the Department of Leadership Studies, Our Lady of the Lake University, 411 S. W. 24th St., San Antonio, TX 78207 USA (e-mail: ysun@ollusa.edu)

M. Shao is with the School of Management, Shanghai University of Engineering Science, Shanghai, CHINA (e-mail: mshao1978@126.com)

states and collapse of states based on GQA [5]. Although many scholars had tried to improve the quantum evolution algorithm and proved its performance was better than traditional evolutionary algorithms, quantum evolution algorithm is still easy to fall into local optimum, especially for complex global numerical optimization problems. Therefore, how to ensure QEA overcomes above disadvantage has been a hotspot but also a difficulty [6].

Some researchers began to combine co-evolutionary mechanism with QEA and then proposed some improved QEAs in recent years. Gu et al. proposed a novel competitive co-evolutionary quantum genetic algorithm (CCQGA) which included three new strategies named as competitive hunter, cooperative surviving and the big fish eating small fish. This algorithm could not only increase the diversity of genes and avoid premature convergence, but also accelerate the convergence. The experiment results show that CCQGA has better feasibility and effectiveness than quantum inspired genetic algorithm (QGA) and standard genetic algorithm (GA) [7]. Xiong, Gui et al. proposed a double population co-evolution algorithm based on the quantum evolution algorithm and difference evolution algorithm. In their new algorithm, these two populations complete different evolutionary process. One population is used for global searching; the other one is for partial searching [8]. Zhang proposed an elite collaborative quantum evolution algorithm. His new algorithm divided the entire population into several sub-populations. 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 [9].

In general, the existing improved QEAs based on co-evolutionary mechanism often include the following two main design ideas: (1) Focusing on the co-evolution between the two sub-populations; (2) Using the elite individual to guide the entire population toward the global optima. These two improvement strategies based on the multiple sub-populations are attempting to find the current global optima individual at the end of each iteration, and then apply the global optima to help these sub-populations. However, they ignore the evolutionary characteristics of individual sub-population, which may include some useful information for the optimization process. In this paper, the co-evolutionary mechanisms are introduced into QEA in order to propose a novel co-evolutionary quantum-inspired evolutionary algorithm (NCQEA). This NCQEA will construct its own elite library which includes multiple elite individuals from the different sub-populations. In the course of evolution, the elite individual will guide the selected individual sub-population instead of the entire population. In addition, the NCQEA will extract the characteristic information of every sub-population to get a better performance of evolution for the corresponding sub-population.

The rest of this paper is organized as following. Section 2 describes the fundamental theory of co-evolutionary algorithm and quantum evolution algorithm; 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.

II. Preliminary Knowledge

A. Co-Evolutionary Algorithm

Definition 2.1 [10]: An evolutionary algorithm is called a co-evolutionary algorithm, If and only if it satisfies the following conditions:

[1] This algorithm can maintain multiple sub-populations simultaneously;

[2] As for the individuals in this algorithm, their fitness values depend on the individuals in the other sub-populations;

[3] As for the individuals in this algorithm, the evolutionary operations of individuals (including insert, delete, and update) will lead to the fitness landscape of other sub-populations fitness change.

The genetic algorithm here uses co-evolutionary strategy to improve its optimizing performance, and its essence of co-evolution is to change the evaluation method of individual viability. The fitness of every individual will not only depend on itself, but also on the other individuals which is in the other sub-populations. When the population is divided into two sub-populations, its co-evolutionary process can be described as following [11].

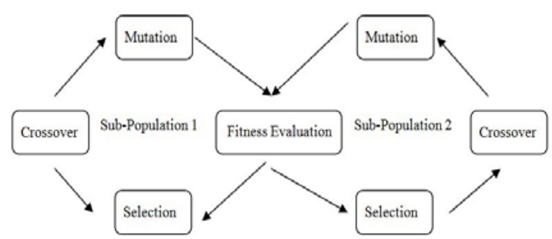

Fig. 1 a diagram of the co-evolutionary process between two sub-populations

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) [12][13]. 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". 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.

B. 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 instead of select, crossover and mutation operations in GA, which can guide the search direction of individual to the optimal area, and increase the algorithm's convergence speed. Han K. pointed out that QEA could show better performance when dealing with the optimization problems compared with traditional evolutionary algorithms [14].

In the general 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:

$$\left|\Psi\right\rangle = \alpha\left|0\right\rangle + \beta\left|1\right\rangle \quad (1)$$

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

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

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. 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. Meanwhile, QEA uses Q-gate as a variation operator that drives the probability of Q-bits converge either to "1" (or "0") and the quantum individuals toward better solutions [15]. Further details about QEA can be seen in paper [5].

III. THE NOVEL CO-EVOLUTIONARY QUANTUM EVOLUTION ALGORITHM

A. Evolutionary Strategy

Evolutionary strategy 1: 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 [16]. In this case, 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 to construct an elite library. Accompanied by the evolution of the whole population, this elite library is also constantly updated. Subsequently, this new elite library can be used to judge the evolution performance of each sub-population between superiority and inferiority to a certain extent. The NCQEA could rely on the individuals in elite library as one of the standards to evaluate its corresponding sub-populations. Here the best individual with highest fitness in the elite library will be used to guide the worst sub-population instead of the whole. In this respect, we use Evolutionary strategy 1 to complete the following two goals.

Firstly, every elite individual corresponds to a sub-population, and it is equivalent to using a simple and efficient method to determine which sub-population has evolved a better result. Secondly, as for these multiple sub-populations, the best individual in the elite library will be used to guide the worst sub-population, which is actually equivalent to build up an information exchange mechanism between the best sub-population and the worst sub-population.

Evolutionary strategy 2: 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 [17]. Normally, when the optimization algorithms use the co-evolutionary strategy, its sub-populations are relatively independent. If the diversity of sub-populations is not identified at the end of each iteration, it would be gradually reduced in the later iterative process of the algorithm. This could result in a gene decrease for effective individuals, 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.

Here a modified indicator of degree of population diversity is introduced to measure the diversity of sub-populations based on the reference [18]. 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: $Q = \{q_1, q_2, ..., q_n\}$, where $q_j = \{q_{1j}, q_{2j}, ..., q_{Lj}\}$, and $j = 1,2,...,n$. The degree of population diversity is defined as the following formula (3):

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

where $sizepop$ is the scale of individuals in population, and $sizepop = L \times n$, $\sum_{j=1}^{n} q_{lj}$ represent the numbers of binary code "1" of all the individuals in their l th 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 l th bit in the population. The result of calculating $\left[\sum_{l=1}^{L}\sum_{j=1}^{n} q_{lj} - \sum_{l=1}^{L}\sum_{j=1}^{n}(1-q_{lj})\right]$ 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 $div(Q)$ becomes smaller, it shows the diversity of the population is getting worse. As mentioned in above formula (3), when the NCQEA divides the population into many sub-populations, every sub-population will has its own div as an indicator to measure the sub-population diversity during the entire evolutionary process.

Evolutionary strategy 3: In addition, 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 [19]; and the value of mutation probability is fixed [20]. The NCQEA in this section will implement mutation operation for the worst sub-population in order to improve its diversity. It should be noted that the change of div is related to the probability of mutation. Therefore, when the div becomes smaller, it means the diversity of population is getting worse, while the probability of mutation will increase. With this operation, the diversity of population and global searching abilities would be improved by adaptively changing of the mutation probability.

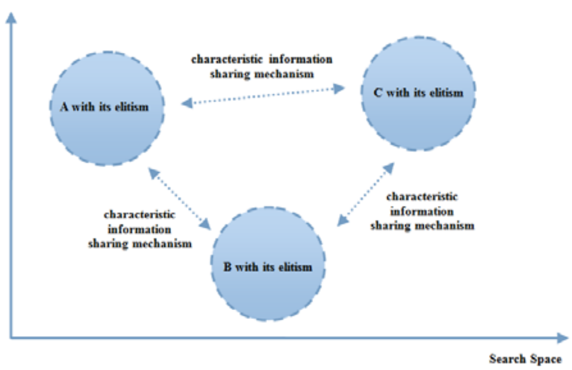

Fig. 2 the proposed evolutionary strategy of the co-evolutionary process between three sub-populations

Here the evolutionary structure has been changed by using above proposed evolutionary strategies. As shown in Figure 2, three sub-populations A, B, C are implementing the evolution in the same solution space. Every sub-population evolves independently and continuously updates its corresponding characteristic information; meanwhile, a characteristic information sharing mechanism based on the indicator and mutation operation between selected two sub-populations in order to ensure the poor sub-population has a better optimization performance.

B. Pseudo -code of the NCQEA

Begin

Set the number of iterations is t and let $t=0$; the number of sub-populations is i;

Step 1 Generate the initial sub-population of $Q_i(t)$, and construct an elite library $E(t)$, for all $i \in \{1,2,...,n\}$;

Step 2 Make $P_i(t)$ by observing the states of $Q_i(t)$;

Step 3 Evaluate $P_i(t)$;

Step 4 Store the best solutions among $P_i(t)$ into $E(t)$;

Step 5 While (not termination-condition) do

Begin

Set $t = t+1$

Step 5.1 Make $P_i(t)$ by observing the states of $Q_i(t-1)$;

Step 5.2 Evaluate $P_i(t)$;

Step 5.3 Update $Q_i(t)$ using Q-gates;

Step 5.4 Store the best solutions among $P_i(t)$ and $E(t-1)$ into $E(t)$;

Step 5.5 The best individual in the elite library $E(t)$ will be used to guide the worst sub-population, which is correspondence to the worst individual in the elite library $E(t)$;

Step 5.6 Find the sub-population with worst diversity according to its div and complete the mutation operation.

IV. Numerical Examples

A. Numerical optimization functions

In order to illustrate the effectiveness of the proposed NCQEA, we compare it with the classic QEA: GQA [21]. The maximum number of the iteration of these two algorithms is 100, and the initial population size is 200. The population of NCQEA is divided into four sub-populations with size of 50 each. Global numerical optimization problems arise in many fields such as engineering, business management. Solving these numerical optimization function problems does not require specialized knowledge in a particular field of study, and can reflect the actual performance of the algorithms. This section using some existing numerical optimization benchmark functions selected from literature [22] to evaluate the performance of the above two algorithms. These test functions are listed as following:

Function 4: Multi-peaks function 1

$$f_4(x,y) = 1 + x\sin(4\pi x) - y\sin(4\pi y + \pi) + \frac{\sin(6\sqrt{x^2+y^2})}{6\sqrt{x^2+y^2+10^{-15}}}$$

Function 4 has four global maximum points with corresponding value 2.11876. These maximum points are distributed on (0.64, 0.64), (-0.64, -0.64), (0.64, -0.64), (-0.64, 0.64) symmetrically. Here the ranges of the independent variables are restricted to (-1, 1). Function 4 has a large number of local maximum points. When the optimized result is greater than 2.1180, it can be considered that the algorithm converges globally.

Function 6: Shaffer's F1 function

$$f_6(x,y)=10\cos(2\pi x)+10\cos(2\pi y)$$
$$-x^2-y^2-20$$

Function 6 has a lot of local maximum points, among which only one 0 is the global maximum value. Here the ranges of the independent variables are restricted in (-5.12, 5.12). When the optimized result is greater than -0.005, it can be considered that the algorithm converges globally.

Function 7: Shaffer's F5 function

$$f_7(x_i)=\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}$$

where

$$(a_{ij}^k)=\begin{bmatrix}-32 & -16 & 0 & 16 & 32\\ -32+16k & -32+16k & -32+16k & -32+16k & -32+16k\end{bmatrix}$$

$a_{ij}=(a_{ij}^0\ a_{ij}^1\ a_{ij}^2\ a_{ij}^3\ a_{ij}^4)$ and $i=1,2$; $j=1,2,\ldots,25$; $k=0,1,\ldots,4$.

Function 7 has a lot of local maximum points. Only one 1.002 is the global maximum value. Here the ranges of the independent variables are restricted to (-65.536, 65.536). When the optimized result is greater than 1.000, it can be considered that the algorithm converges globally.

Function 8: Bohachevsky function

$$f_8(x,y)=0.3\cos 3\pi x-0.3\cos 4\pi y-x^2-y^2-0.3$$

Function 8 has two global maximum points with value 0.24003441039434. These maximum points are distributed on (0, -0.23), (0, 0.23) symmetrically. Here the ranges of the independent variables are restricted to (-1, 1). Function 8 has a large number of local maximum points. When the optimized result is greater than 0.24, it can be considered that the algorithm converges globally.

Function 9: Multi-peaks function 3

$$f_9(x,y)=-(x^2+y^2)^{0.25}(\sin^2(50(x^2+y^2)^{0.1})+1.0)$$

Function 9 has unlimited global maximum points, among which only one 0 is the global maximum value on (0, 0). The ranges of the independent variables are restricted in (-5.12, 5.12). When the optimized result is greater than -0.05, it can be considered that the algorithm converges globally.

Function 10: Needle-in-haystack function

$$f_{10}(x,y)=(\frac{3}{0.05+x^2+y^2})^2+(x^2+y^2)^2$$

Function 10 has four local maximum points, which are distributed on (5.12, 5.12), (-5.12, -5.12), (5.12, -5.12), and (-5.12, 5.12) symmetrically. As for Function 10, only 3600 is the global maximum value on (0, 0). Here the ranges of the independent variables are restricted to (-5.12, 5.12). When the optimized result is greater than 3599, it can be considered that the algorithm converges globally.

Notices that some of 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 tables.

Tab. 1. Statistical Results of QEA and NCQEA for Optimization Functions (with 10 experiments)

Optimization Functions	QEA				
	Best	Worst	Average	Convergence times	Standard Deviation
f_4	2.1188	2.1188	2.1188	10	0
f_6	-1.23e-07	-1.49e-06	-7.864e-07	10	4.4311e-07
f_7	1.002	0.71087	0.93881	7	0.10143
f_8	0.24003	0.24003	0.24003	10	0
f_9	-0.0053775	-0.020144	-0.012503	10	0.0047693
f_{10}	3600	3599.9993	3599.9997	10	0.00022825
Optimization Functions	NCQEA				
	Best	Worst	Average	Convergence times	Standard Deviation
f_4	2.1188	2.1188	2.1188	10	0
f_6	-9.46e-09	-2.74e-07	-7.372e-08	10	1.0147e-07
f_7	1.002	0.83164	0.95089	7	0.078069
f_8	0.24003	0.24003	0.24003	10	0
f_9	-0.005243	-0.01289	-0.0060346	10	0.0022857
f_{10}	3600	3599.9999	3600	10	3e-05

The results shown in above Table 1 indicate that the NCQEA is superior to QEA from the view of quality of solutions when these two optimization algorithms perform the same functions. The statistical results show that the NCQEA is more efficient in finding the global optimal solution and robustness than QEA. The standard deviation indicates that NCQEA has a good stability and widespread adaptability. Here take the evolutionary results for function 4 and function 7 based on NCQEA 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 4 and function 7, their corresponding sub-populations can converge to their optimal solutions throughout the iteration cycle at the same time. As for function 4 in Figure 3, the optimal solutions in sub-population 1, 2, 3, 4 are all (-0.64062, 0.64096) and their corresponding optimal values are 2.1188. For function 7 in Figure 4, the optimal solutions in sub-population 1, 2, 3, 4 are all (-31.9792, -32.0007) with corresponding optimal values 1.002. These results also present the following two conclusions. Firstly, the new evolutionary strategies are effective for the Co-evolutionary Quantum-Inspired Evolutionary Algorithm to a certain degree. 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 some extent.

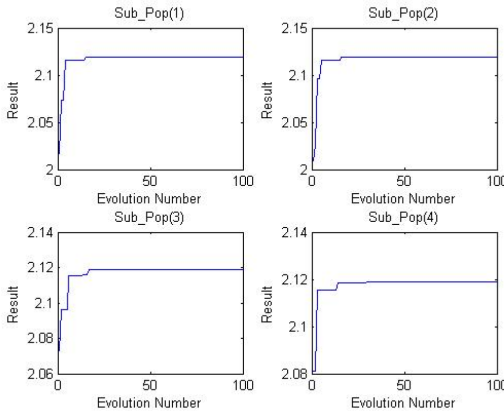

Fig. 3 Evolutionary results for Function 4 on the basis of NCQEA

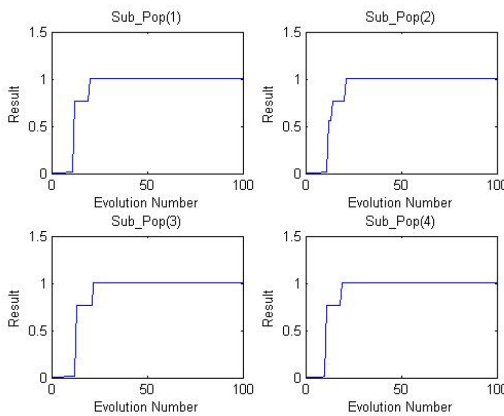

Fig. 4 Evolutionary results for Function 7 on the basis of NCQEA

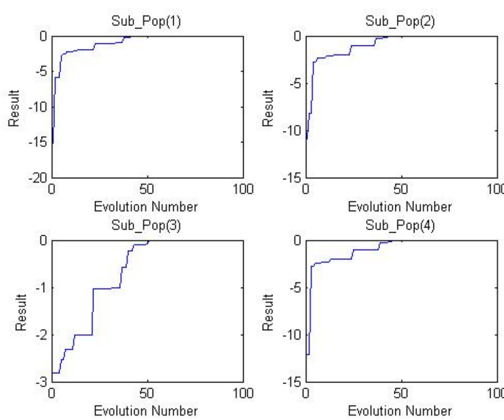

Fig. 5 Evolutionary results for Function 6 on the basis of NCQEA

Our NCQEA 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. The standard deviation results have also shown that NCQEA has better computational stability than traditional QEA.

However, our experiment results show that in some cases these four sub-populations do not become synchronized toward convergence. For function 6 in Figure 5, the optimal solutions in sub-population 1, 2, and 4 are (-4.8828e-006, 4.8828e-006) with corresponding optimal values -9.4601e-009; while the optimal solutions in sub-population 3 is (-2.4414e-005, 4.8828e-006) with optimal value -1.2298e-007.

B. Knapsack problems

As a well-known combinatorial optimization problem, the Knapsack problem is one of the classic NP-Hard problems. It can be described as selecting from various items, and the selected items are the most profitable items, given that the Knapsack has a limited capacity. Here the Knapsack problem can be defined as following.

Select a set of items x_i, $i = 1,2,...,m$ with profits p_i, and weight w_i. Here given a set of m items and a Knapsack with capacity C, select subset of the items to maximize the profit function $f(x)$.

$$f(x) = \sum_{i=1}^{m} p_i x_i$$

Such that $\sum_{i=1}^{m} w_i x_i \le C$; and $x_i \in \{0,1\}$, $i = 1,2,...,m$

Here we have selected the following simulation examples from literature [23], and then use our new algorithm to test those simulation examples. The dimensions of the tested examples are from the simple 10 dimensional space to more complex 100 dimensional spaces. Therefore we can fully test the performance of these two algorithms. Notice for every evolutionary run, the population size and iteration number are set as $4m$, where m is the dimensional number of the test examples. The corresponding optimal solution of each simulation example is illustrated in Table 2. Notice $f(x)$ is the total profit for the Knapsack as selecting from various items; $sum(weight)$ is the total weight for the Knapsack. The simulation results can be summarized in the following Table 3.

The data are averaged over 30 different runs and the corresponding statistical results of these runs are shown in Table 3. It can be seen from Table 3 that NCQEA have better simulation results than QEA for the different Knapsack problems and NCQEA has relatively stable computing ability. Although numerous calculations occasionally fall into local optimal solution such as for P3, P4, P5, and P6, NCQEA is still much better than QEA according to the standard deviation results. As for more complex Knapsack problems P5 and P6, if we set the evolutionary parameters to be $6m$, then we get the corresponding results in the following Table 4.

Tab. 2. The Six simulation examples for Knapsack problems

Problem Number	Number of Dimensions	with known optimal solution
P1	10	$f(x)=295$ $Sum(weight)=269$
P2	15	$f(x)=481.07$ $Sum(weight)=354.96$
P3	50	$f(x)=3103$ $Sum(weight)=1000$
P4	60	$f(x)=8362$ $Sum(weight)=2393$
P5	80	$f(x)=5183$ $Sum(weight)=1170$
P6	100	$f(x)=15170$ $Sum(weight)=3818$

Tab. 3. Simulation results for Knapsack problems

Optimization Functions	QEA				
	Best	Worst	Average	Convergence times	Standard Deviation
P1	295/289	295/289	295	0	295/289
P2	481.07/354.96	481.07/354.96	481.07	0	481.07/354.96
P3	3103/1000	3077/1000	3091.1667	6.0887	3103/1000
P4	8362/2393	8350/2394	8359.0667	4.0574	8362/2393
P5	5183/1170	5146/1171	5175.9	9.7582	5183/1170
P6	15170/3818	15143/3818	15164.1333	7.7835	15170/3818
Optimization Functions	**NCQEA**				
	Best	Worst	Average	Convergence times	Standard Deviation
P1	295/269	295/269	295	0	295/269
P2	481.07/354.96	481.07/354.96	481.07	0	481.07/354.96
P3	3103/1000	3080/1000	3091.8	5.9464	3103/1000
P4	8362/2393	8354/2399	8359.4	3.2311	8362/2393
P5	5183/1170	5146/1171	5179.5667	9.1056	5183/1170
P6	15170/3818	15145/3818	15164.2667	6.6329	15170/3818

Tab. 4. Analysis results for different Knapsack problems (with 30 experiments)

Problem Number	NCQEA			
	Best	Worst	Average	Standard Deviation
P5	5183/1170	5167/1171	5181.4	3.2721
P6	15170/3818	15151/3812	15167.3333	5.1273

Note that the bigger the evolutionary parameters, the more superior solution we get in table 4. The optimization performance of NCQEA is more stable than the results in Table 3 based on the results of standard deviation. Considering the initial population is randomly selected for the algorithm, they are able to gather from the initial position to the optimal solution after a lot of iterations; and the statistic results also suggest that NCQEA has persistent optimization ability for more complex optimization problems.

V. Conclusions

In order to improve the performance of QEA, this paper proposes a novel co-evolutionary quantum-inspired evolutionary algorithm (NCQEA). In NCQEA, 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 to implement the evolution process. In the comparative study, NCQEA is compared with existing QEA using the established benchmarks functions and Knapsack problems. The convergence performance for NCQEA is superior to QEA, and NCQEA is more accurate and stable in finding the global value in the testing problems.

References

[1] B. Xia, M. Liang, "Development on Genetic Algorithm Theory and Its Applications," Application Research of Computers, vol. 27, no. 7, pp. 2425-2434, 2010.

[2] Ehrlich P.R, Raven P.H. "Butterflies and plants: A study in coevolution," Evolution, vol. 18, no. 4, pp. 586−608, 1964.

[3] W. Ling, S. Jing-nan, W. Sheng-yao, D. Jin, "Advances in co-evolutionary algorithms," Control and Decision, vol. 30, no. 2, pp. 193-202, 2015.

[4] L. Quan, W. Xiao-Yan, F. Qi-Ming, Z. Yong-Gang, Z. Xiao-Fang, "Double Elite Coevolutionary Genetic Algorithm," Journal of Software, vol. 23, no. 4, pp.765−775, 2012.

[5] H. K. Hyun, K. J. Hwan, "quantum inspired evolutionary algorithm for a class of combinatorial optimization", IEEE Transactions on Evolutionary Computation, vol. 6, no. 6, pp.580-593, 2002.

[6] Y. Xiao-ming, L. Sheng, W. Yu-ming, "Hybrid Evolutionary Algorithm Based on Quantum Computing and Performance Analysis," ACTA ELECTRONICA SINICA, vol. 40, no. 4, pp. 856-860, Apr. 2012.

[7] Jinwei G., Manzhan G., Cuiwen C., Xingsheng G., "A novel competitive co-evolutionary quantum genetic algorithm for stochastic job shop scheduling problem," Computers & Operations Research, vol. 37, no. 5, pp. 927–937, 2010.

[8] X. Fu-qiang, G. Wei-hua, Y. Chun-hua, L. Yong-gang, "A double population co-evolution algorithm for process of zinc hydrometallurgy," Control and Decision, vol. 28, no. 4, pp. 590-594, 2013.

[9] Z. Weifeng, "Elite Collaborative Quantum Evolution Algorithm for Combinatorial Optimization Problem," Journal of Hubei University of Automotive Technology, vol. 27, no. 3, pp. 50-56, Sep. 2013.

[10] D. Wu, *Research on Co-evolutionary – Based Hybrid Intelligent Optimization Algorithms and Their Applications*, Liaoning: Dalian Maritime University, 2012.

[11] L. Bi, L. Tu-sheng, "Comments on Coevolution in Genetic Algorithms," Computer Science, vol. 36, no. 4, pp. 34-37, 63, 2009.

[12] Li M. Q., Lin D., Kou J. S., "Dynamics of Fitness Sharing Evolutionary Algorithms for Coevolution of Multiple Species," Applied Soft Computing, vol. 10, no. 3, pp. 832-848, 2010.

[13] S. Xiaoyan, C. Dunwei, "varying population size cooperative coevolutioanry genetic algorithm and its application in optimization," Control and Decision, vol. 19, no. 12, pp. 1437-1440, 2004.

[14] H. K. Hyun., K. J. Hwan, "Genetic quantum algorithm and its application to combinatorial optimization problems," In Proceeding Of the 2000 IEEE Congress on Evolutionary Computation, pp.1354-1360.

[15] J. BALICKI, "Quantum-inspired Multi-objective Evolutionary Algorithms for Decision Making: Analyzing the State-Of-The-Art," 2014 ADVANCES in APPLIED and PURE MATHEMATICS, WSEAS Press, pp. 383-389.

[16] J. Balicki, G. Drozda, J. Masiejczyk, "Multi-criterion, Evolutionary and Quantum Decision Making in Complex Systems," 2011 Recent Advances in Applied & Biomedical Informatics and Computational Engineering in Systems Applications, WSEAS PRESS, pp. 222-227.

[17] Mohammad H. Tayarani-N, M. R. Akbarzadeh-T, "Improvement of the performance of the Quantum-inspired Evolutionary Algorithms:

structures, population, operators," Evolutionary Intelligence, vol. 7, no. 4, pp. 219 - 239, Dec. 2014.

[18] L. Quan, W. Xiao-Yan, F. Qi-Ming, Z. Yong-Gang, Z. Xiao-Fang, "Double Elite Coevolutionary Genetic Algorithm," Journal of Software, vol. 23, no. 4, pp. 765−775, 2012.

[19] W. Shihua, Z. Jinhua, L. Miqing, "Comparison and research of mutation operators in multi-objective evolutionary algorithsm," Computerr Engineering and Applications, vol 45, no. 2, pp. 74-78, 2009.

[20] R. TURCINHODZIC, Z. AVDAGIC, S. OMANOVIC, "The Effects of Using Multiple Selection Operators within a Single Genetic Algorithm," 2013 Recent Advances in Telecommunications and Circuits, WSEAS Press, pp. 188-193.

[21] K. H. Han, J. H. Kim, "Genetic Quantum Algorithm and its Application to Combinatorial Optimization Problem," In Proc. 2000 Congress on Evolutionary Computation, Piscataway, NJ: IEEE Press, pp. 1354-1360.

[22] L. Panchi, *Quantum Computation with Applications to Intelligent Optimization and Control*, Haerbin: Harbin Institute of Technology, 2009.

[23] W. Husheng, Z. Fengming, Z. Renjun, W. Song, Z. Chao, "A binary wolf pack algorithm for solving 0-1 knapsack problem," Systems Engineering and Electronics, vol. 36, no. 8, pp. 1660-1667, 2014.

Liang Zhou received his Master degree in Control Theory and Control Engineering from Qingdao University of Science & Technology (China) in 2007 and Ph. D in Management Science in 2012 from Donghua University, Shanghai, China. Currently, he now served as an assistant professor in the School of Mathematics and Information Science, Shanghai Lixin University of Commerce. His research interests include optimization computing, simulation, modeling, and artificial intelligence.

Yu Sun received her B.S. degree in Mathematics from Qingdao University, Qingdao, China in 2004, M.S. in Applied Mathematics in 2007 and Ph. D in Management Science in 2010 from Donghua University, Shanghai, China. She joined the Department of Mathematics, Wayne State University, Detroit, Michigan in 2007 and received her M.A. in Mathematical Statistics and Ph.D. degree in Applied Mathematics in August 2012. Before she joins OLLU, she served as an assistant professor in the Quantitative Business Analysis Department at Siena College, Loudonville, NY. Her research interests include mathematics in decision-making, artificial intelligence, fuzzy systems, stochastic optimization, and quantitative analysis.