Available online www.jocpr.com

Journal of Chemical and Pharmaceutical Research, 2014, 6(7):685-694

Research Article

ISSN : 0975-7384
CODEN(USA) : JCPRC5

A chaotic quantum bee colony optimization for thinned array

Hongyuan Gao* and Yanan Du

College of Information and Communication Engineering, Harbin Engineering University, Harbin, China

---

ABSTRACT

Design a new novel intelligence algorithm which is called as chaotic quantum bee colony optimization (CQBCO) for discrete optimization problem. The proposed CQBCO applies the chaotic theory to quantum bee colony optimization (QBCO), which is an effective discrete optimization algorithm. Then the proposed chaotic quantum bee colony algorithm is used to solve benchmark functions and optimization problem of thinned array. By hybridizing the quantum bee colony optimization and quantum computing theory, the quantum state and binary state of the bees can be well evolved by simulated quantum rotation gate and chaotic mechanical. The new thinned array method based on CQBCO can search the global optimal solution. Simulation results for thinned array are provided to show that the proposed thinned array method is superior to the thinned array methods based on other three intelligence algorithms.

Key words: Quantum bee colony optimization, thinned array, chaotic, particle swarm optimization

---

INTRODUCTION

As we all know that natural system is one of the affluent sources of inspiration for designing new intelligent algorithms. For intelligence algorithms are important scientific domains that are closely related to biological phenomenon existing in nature, some algorithms such as particle swarm optimization (PSO) [1-2] and bee colony optimization [3] are widely studied for all kinds of applications . At present, particle swarm optimization [4] and bee colony optimization [5] were widely used to solve engineering optimization problem, but the global convergent performance should be improved.

In order to improve classic intelligent algorithm, the quantum computing theory is introduced into conventional intelligence algorithm and a better performance is obtained [6]. Quantum-inspired genetic algorithm (QGA) which is a promising genetic algorithm developing in recent years, and it is the product of merging quantum computing theory with genetic algorithm. In QGA, quantum bit encoding represents the chromosome, and evolutionary process of chromosomes is implemented by using quantum rotation gate. Now, we pay more attention to quantum optimization algorithm because it has a strong ability for global search with small population size, rapid convergence and short computing time [7]. Based on quantum computing and bee colony optimization, quantum bee colony optimization (QBCO) is proposed as an effective swarm intelligence algorithm [8]. The results of simulation comparisons show that the performance of the QBCO algorithm is competitive to other intelligence computing algorithms with an advantage of using fewer control parameters.

At present, a hot research domain is applying intelligent algorithms to thinned array .Thinning an array means deleting some elements in a uniformly spaced or periodic array to create a desired amplitude density across the aperture. If an element is existed which means “on”, and the element has no value which means “off”. Thinning an array to produce low side lobes is much easier than the more general method of non-uniform spacing the elements. We can get an infinite number of possibilities for placement of the elements when we use non-uniform spacing. However, thinning an array has 2^D possible combinations, where D is the number of array elements. Thinning may be also regarded as an amplitude taper of quantization, and the amplitude at each element is expressed with one bit.

A low side lobe amplitude taper is generated by strategically positioning equally weighted elements in aperiodic arrays domain. Deriving the element positions to obtain a perfect side lobe level by using simple analytical methods are not available [9]. Instead, merge the element density with a region of the array to the amplitude density of the low side lobe amplitude taper for the same size aperture is used for most aperiodic array synthesis methods [10]. The element density at the center of the array is greatest and gradually decreases to the edges. As a rule, side lobes close to the main beam decrease but those far from the main beam increase [11] (which is usually quite acceptable). Aperiodic array synthesis methods target a maximum relative side lobe level by using a given probability [12].

Thinning array for low side lobes includes inspecting a rather large number of possibilities in order to find the best thinned aperture. It is only practical for small arrays if checking of all possible element combinations [13]. Most optimization methods are not well suited for thinning arrays such as down-hill simplex, Powell's method, and conjugate gradient. Those methods can only optimize a few continuous variables and drop into local minima easily [14]. Those methods developed for continuous parameters, but the array thinning problem involves discrete parameters. Although dynamic programming can optimize a large parameter set, it is easily effect by local solutions [15].

As we all know genetic algorithms [16] and simulated annealing algorithm [17] are suited for thinning arrays because they do not limit to the number of variables of optimization. Although these algorithms can handle very large arrays, they are quite slow to find the optimal structure of thinned array. In order to achieve more robust and efficient performances for thinned array, CQBCO is proposed to design optimal structure of thinned array.

CHAOTIC QUANTUM BEE COLONY OPTIMIZATION

In order to deal with discrete optimization problems by using chaotic mechanic and quantum bee colony theory, CQBCO is designed. The quantum evolutionary algorithms use quantum coding, called on a quantum bit [18], for the probabilistic representation that is based on the concept of quantum bit, and a quantum position is defined as a string of quantum bits. The quantum position of the *i*th bee is defined as

$$\mathbf{q}_i = \begin{bmatrix} v_{i1} & v_{i2} & \cdots & v_{iD} \\ \beta_{i1} & \beta_{i2} & \cdots & \beta_{iD} \end{bmatrix} \tag{1}$$

where $|v_{id}|^2 + |\beta_{id}|^2 = 1$, $(d = 1, 2, \cdots, D)$. In CQBCO, v_{id} and β_{id} are defined as real numbers and $0 \le v_{id} \le 1, 0 \le \beta_{id} \le 1$ [8].

The evolutionary process of quantum bee colony is mainly completed through the update of quantum position. The update of quantum position is obtained by quantum rotation gate. Quantum rotation gate can be described by equation (2). If the quantum rotation angle is φ_{id}^{t+1}, a quantum bit position $\mathbf{q}_{id}^t = [v_{id}^t, \beta_{id}^t]^{\mathrm{T}}$ is updated by using the rotation gate $\mathbf{U}(\varphi_{id}^{t+1})$. The *d*th quantum bit position $\mathbf{q}_{id}^t$ of the *i*th quantum position is updated as

$$\mathbf{q}_{id}^{t+1} = \mathrm{abs}(\mathbf{U}(\varphi_{id}^{t+1})\mathbf{q}_{id}^t) = \mathrm{abs}(\begin{bmatrix} \cos\varphi_{id}^{t+1} & -\sin\varphi_{id}^{t+1} \\ \sin\varphi_{id}^{t+1} & \cos\varphi_{id}^{t+1} \end{bmatrix}\mathbf{q}_{id}^t) \tag{2}$$

where $\mathrm{abs}()$ is an absolute value function which makes quantum bit in the real domain [0,1].

To reduce computation of CQBCO, we use a series simple quantum bits to represent position of bee in CQBCO. A quantum position of the *i*th bee is simplified as $\mathbf{v}_i^t = (v_{i1}^t, v_{i2}^t, \cdots, v_{iD}^t)$ where quantum bit is limited to $0 \le v_{id}^t \le 1$, $d = 1, 2, \cdots, D$.

Chaotic quantum bee colony optimization is a novel optimization algorithm inspired by social behavior metaphor of bees. Each bee, flies in a *D*-dimensional space according to the historical experiences of its own and its colleagues. In CQBCO, the bee colony contains two groups of bees: employed bees and onlookers. First half of the quantum bee colony consists of the employed bees and the second half includes the onlookers.

There are *h* bees in a bee colony. The position of bee is also a food source's position and its quality is evaluated by the nectar amount of nectar amount function. The food source position of the *i*th bee at the *t*th iteration is expressed as

$\mathbf{x}_i^t = (x_{i1}^t, x_{i2}^t, \ldots, x_{iD}^t)$, $(i = 1, 2, \ldots, h)$, which is a latent solution of optimization problem. The *i*th bee's quantum position is $\mathbf{v}_i^t = (v_{i1}^t, v_{i2}^t, \ldots, v_{iD}^t)$, $(i = 1, 2, \ldots, h)$ which is measured and produced food source's position. Employed bees and onlookers learn different food source information from bee colony. Until now the optimal position of the ith bee is expressed as $\mathbf{p}_i^t = (p_{i1}^t, p_{i2}^t, \ldots, p_{iD}^t)$, $(i = 1, 2, \ldots, h)$ which is called the local optimal position. $\mathbf{p}_g^t = (p_{g1}^t, p_{g2}^t, \ldots, p_{gD}^t)$ represents the global optimal position discovered by the whole bee colony until now, and $\mathbf{p}_g^t$ also is the optimal position of all local optimal positions at the tth iteration.

In order to introduce a chaotic behavior to the optimization process, a simple chaotic system presenting chaotic behavior is called logistic map [19], and chaotic equation is written as

$$v_{id}^{t+1} = \eta v_{id}^t (1 - v_{id}^t) \tag{3}$$

where η is a control parameter in the range $0 \le \eta \le 4$. The behavior of the chaotic system defined by (3) is very sensitive to changes of η. The value of η determines whether v_{id}^t stabilizes at a constant size or represents chaotically in an uncertain mode. Very tiny differences in the initial value of v_{id}^t can lead to enormous differences in its long-time behavior. Equation (3) always displaying chaotic dynamics when $\eta = 4$ and $v_{id}^t \notin \{0, 0.25, 0.5, 0.75, 1\}$ [20]. It is easy to observe that the chaotic sequences have different behaviors, depending on the value of the parameter η.

Quantum rotation angle of employed bee is updated by the local optimal position $\mathbf{p}_i^t$ and the global optimal position $\mathbf{p}_g^t$. At each iteration, the quantum bit position of the *i*th employed bee is updated by the following:

$$\varphi_{id}^{t+1} = e_1 (p_{id}^t - x_{id}^t) + e_2 (p_{gd}^t - x_{id}^t) \tag{4}$$

$$v_{id}^{t+1} = \begin{cases} 4v_{id}^t (1 - v_{id}^t), & \text{if} \quad \varphi_{id}^{t+1} = 0 \text{ and } \mu_{id}^{t+1} < c_1; \\ \text{abs}[v_{id}^t \cos(\varphi_{id}^{t+1}) - \sqrt{1 - (v_{id}^t)^2} \sin(\varphi_{id}^{t+1})], & \text{otherwise} \end{cases} \tag{5}$$

where $i = 1, 2, \ldots, h/2$, $d = 1, 2, \ldots, D$, e_1 and e_2 are constants, c_1 is mutation probability which is a constant among $[0, 1/D]$, μ_{id}^{t+1} represents uniform random number between 0 and 1, superscript $t+1$ and t represent the number of iterations.

After watching the dances of employed bees, the onlooker $i (i = h/2+1, h/2+2, \ldots, h)$ goes to the food source located at $\mathbf{p}_j^t = (p_{j1}^t, p_{j2}^t, \ldots, p_{jD}^t)$ $(j = 1, 2, \ldots, h/2)$ by certain probability and determines a neighbor food source to take its nectar. The location of a food source selected by the onlooker depends on the fitness function value $fit(\mathbf{p}_j^t)$ of local optimal position. Therefore, the selection probability of employed bee j is decided by roulette wheel selection , and can be expressed as

$$\lambda_j^{t+1} = \frac{fit(\mathbf{p}_j^t)}{\sum_{l=1}^{h/2} fit(\mathbf{p}_l^t)} \tag{6}$$

At each iteration, the quantum bit position of the *i*th onlooker is updated by the following:

$$\varphi_{id}^{t+1} = e_3 (p_{id}^t - x_{id}^t) + e_4 (p_{jd}^t - x_{id}^t) \tag{7}$$

$$v_{id}^{t+1} = \begin{cases} 4v_{id}^{t}(1-v_{id}^{t}), & \text{if} \quad \varphi_{id}^{t+1} = 0 \text{ and } \mu_{id}^{t+1} < c_2; \\ \text{abs}[v_{id}^{t}\cos(\varphi_{id}^{t+1}) - \sqrt{1-(v_{id}^{t})^2}\sin(\varphi_{id}^{t+1})], & \text{otherwise} \end{cases} \tag{8}$$

where $i = h/2+1, h/2+2, \ldots, h$, $d = 1,2,\ldots,D$, e_3 and e_4 are constants, c_2 is mutation probability which is a constant among $[0,1/D]$. The value of e_3 and e_4 express the relative important degree of $\mathbf{p}_i^t$ and $\mathbf{p}_j^t$ in the moving process. Food source's position of the i th bee is updated by (9).

$$x_{id}^{t+1} = \begin{cases} 1, & \text{if} \quad \varepsilon_{id}^{t+1} > (v_{id}^{t+1})^2; \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where $i = 1,2,\ldots,h$, $d = 1,2,\ldots,D$, $\varepsilon_{id}^{t+1} \in [0,1]$ is uniform random number, $(v_{id}^{t+1})^2$ represents the probability that the quantum bit will be found in the '0' in the $(t+1)$th iteration

For CQBCO, the local optimal position $\mathbf{p}_i^t$ of bee i and the global optimal position $\mathbf{p}_g^t$ are updated by the following manners. For bee i, if the nectar amount of $\mathbf{x}_i^{t+1}$ is superior to that of $\mathbf{x}_i^{t+1}$, then $\mathbf{p}_i^{t+1} = \mathbf{x}_i^{t+1}$; else, $\mathbf{p}_i^{t+1} = \mathbf{p}_i^t$. If the nectar amount of $\mathbf{p}_i^{t+1}$ is superior to that of $\mathbf{p}_g^t$, then $\mathbf{p}_g^{t+1} = \mathbf{p}_i^{t+1}$;else, $\mathbf{p}_g^{t+1} = \mathbf{p}_g^t$.

After the every m iterations, $0.5h$ new positions will be generated by mutation operator applying to local optimal positions, and the position of the qth $(q = 0.5h+1, 0.5h+2, \ldots, h)$ bee is generated by the qth onlooker. We select $z(z \in \{1,2,\ldots,Z\}, Z \le D)$ directions in a random manner. For each dimension $d \in$ {pre-selected z dimensions}, the qth onlooker produces temporary position $\mathbf{u}_q = (u_{q1}, u_{q2}, \ldots, u_{qD})$ is generated by

$$u_{qd}^{t+1} = \begin{cases} 1 - p_{qd}^{t} & \text{if} \quad d \in \{\text{pre-selected } z \text{ dimensions}\}; \\ p_{qd}^{t} & \text{otherwise} \end{cases} \tag{10}$$

where $q = 0.5h+1, 0.5h+2, \ldots, h$, $d = 1,2,\ldots,D$, $\lambda_{qd}^{t+1} \in [0,1]$ is uniform random number in the range of [0,1].

Then , for bee q, the local optimal position is updated as

$$\mathbf{p}_q^{t+1} = \begin{cases} \mathbf{u}_q^{t+1} & \text{if} \quad fit(\mathbf{u}_q^{t+1}) > fit(\mathbf{p}_q^{t+1}); \\ \mathbf{p}_q^{t+1} & \text{otherwise} \end{cases} \tag{11}$$

THE PERFORMANCE OF THE CHAOTIC QUANTUM BEE COLONY OPTIMIZATION

We use minimum values of four benchmark functions to evaluate the performance of the CQBCO. For comparison, the initial population and the maximum number of iterations must be identical for the four evolutionary algorithms. For GA, PSO, QBCO and CQBCO, the population size is set to 30 and the maximum number of iterations is set to 1000. As for GA[21], the possibility of cross is 0.8, and the possibility of mutation is 0.01. In PSO, the two acceleration coefficients are equal to 2, and $V_{\max} = 4$ [23]. For QBCO, we use parameters of preference [8]. For CQBCO, the parameters can be set as the following: m=2 , $e_1 = 0.06$, $e_2 = 0.03$, $e_3 = 0.06$, $e_4 = 0.03$, $c_1 = c_2 = 0.1/D$.

The four benchmark functions are as follows:

$$F_1(\mathbf{y}) = 0.5 + \frac{1}{(1+0.001\sum_{i=1}^{n} y_i^2)^2}\left(\sin^2(\sqrt{\sum_{i=1}^{n} y_i^2}) - 0.5\right), (-100 \le y_i \le 100, i = 1, 2, \ldots, n) \quad (12)$$

$$F_2(\mathbf{y}) = \frac{1}{4000}\left(\sum_{i=1}^{n}(y_i - 100)^2\right) - \left(\prod_{i=1}^{n}\cos\left(\frac{y_i - 100}{\sqrt{i}}\right)\right) + 1, (-600 \le y_i \le 600, i = 1, 2, \ldots, n) \quad (13)$$

$$F_3(\mathbf{y}) = \sum_{i=1}^{n-1} 100[(y_{i+1} - y_i^2)^2 + (y_i - 1)^2], (-50 \le y_i \le 50, i = 1, 2, \ldots, n) \quad (14)$$

$$F_4(\mathbf{y}) = 2 \times 418.9829 - \sum_{i=1}^{n} y_i \sin\left(\sqrt{|y_i|}\right), (-500 \le y_i \le 500, i = 1, 2, \ldots, n) \quad (15)$$

The nectar amount function is identical with the fitness function. The fitness function is the reciprocal of sum of benchmark function and 10^{-7}. For the minimum value optimization problems, the objective of nectar amount is the minimum of benchmark function (objective function), and the position of maximal nectar amount is the optimal position.In the following simulations, we use binary-encoding, and the length of each variable is 50 bits. We set $n = 2$ for all benchmark functions, i.e. $i = 1, 2$ .All the results are the average of 200 times.

The first function we use is Schaffer function. From Fig.1, we can see that CQBCO has a slow convergence rate but has a more accurate value compared with the other three algorithms. It is obvious that the value which CQBCO reach at the 100th iteration is equal to the value of QBCO obtained at the 1000th iteration, GA and PSO have not reached it at the 1000th iteration. So CQBCO overcomes the disadvantage of local convergence of QBCO and obtains a more accurate convergence value.

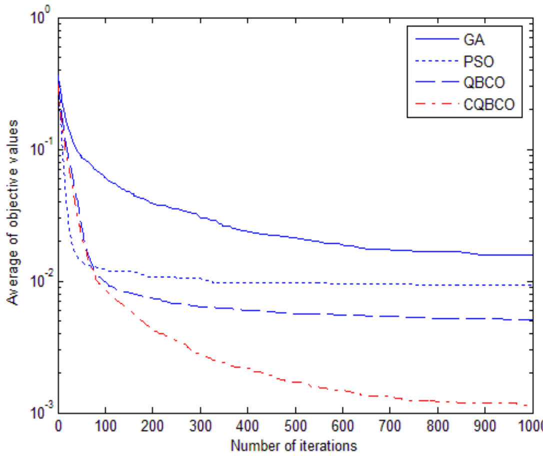

Fig. 1: The performance of four algorithms using Schaffer function

The second function is Griewank function. From Fig.2 we can see that CQBCO outperforms GA, PSO and QBCO. The Fig.2 presents that CQBCO have a slow convergence rate but the classical algorithms are easy trap into local convergence. As we can see that CQBCO has a smaller convergence value compared to QBCO. The convergence precision of CQBCO at the 200th iteration outperforms the performance that others reached at the 1000th iteration.

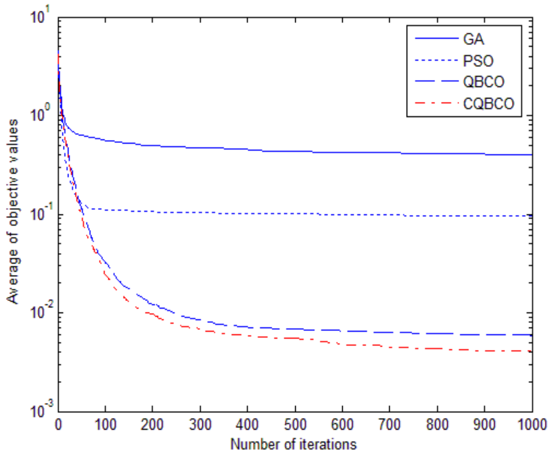

Fig. 2: The performance of four algorithms using Griewank function

The third function is Rosenbrok function, which is a well-known classic optimization function. It is difficult to converge to the global optimum of this function because the global optimum lies in a narrow, long, parabolic-shaped flat valley. The variables are strongly dependent, and the gradients generally do not point towards the optimum, this problem is repeatedly used to test the performance of the optimization algorithm. From Fig.3 we can see that the classical algorithm has a fast convergence rate, but they all trap into local convergence. The Fig.3 proves that CQBCO has a more accurate convergence value. So CQBCO overcomes the disadvantage of local convergence of QBCO and obtains a more accurate convergence value.

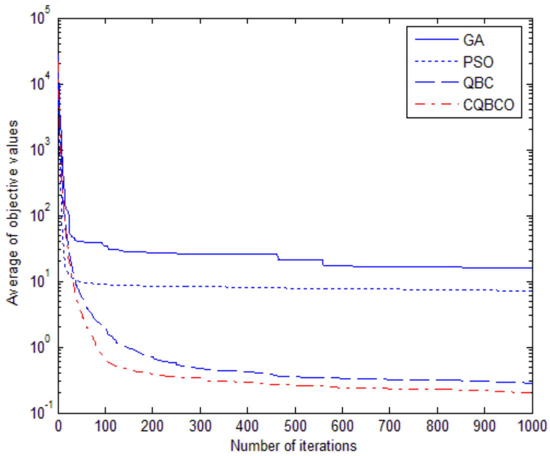

Fig. 3: The performance of four algorithms using Rosenbrock function

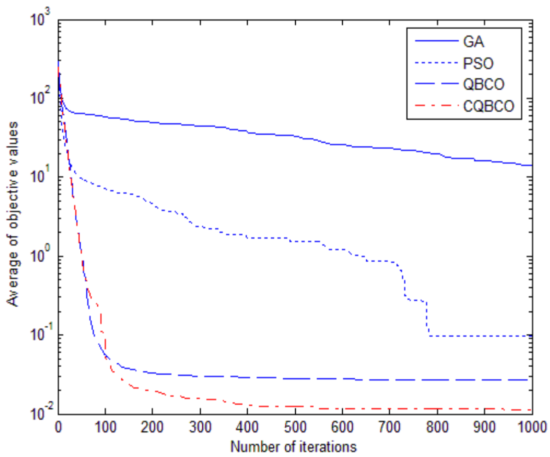

Fig. 4: The performance of four algorithms using Schewefel function

The fourth function is Schewefel function. In Fig. 4, we can see that both QBCO and CQBCO have a fast convergence rate. And CQBCO outperforms GA, PSO and QBCO in both convergence rate and convergence precision.

THINNED ARRAY BASED ON CHAOTIC QUANTUM BEE COLONY OPTIMIZATION

As for a D-element uniform spaced array and all the pattern of element is isotropic. The array pattern function is shown below:

$$F(\theta)=\sum_{d=1}^{D} I_d \mathrm{e}^{\mathrm{j}((d-1)kw\cos\theta+\phi_d)} \tag{16}$$

where $I_d \in \{0,1\}$ is defined as the amplitude weight of element , we set $I_d = 1$ if the element is existed, otherwise, $I_d = 0$. w is defined as the spacing between elements, k is defined as wave number, $k = 2\pi/\sigma$ (σ expresses the working wavelength of array antenna), ϕ_d is defined as the phase of the d th element of incentive.

Since the thinned array is characterized by pattern, we describe the pattern in the form of a function as

$$FF(\theta) = 20\lg\left|\frac{F(\theta)}{M_{\max}}\right| \tag{17}$$

where $M_{\max} = \max_{\theta\in S} |F(\theta)|$ and S is defined as the region of pattern side lobe, $2\theta_0$ is defined as the zero power width of main lobe, the visible area of pattern is $[0,\pi]$, $S = \{\theta \mid 0 \le \theta \le 90^{\circ} - \theta_0 \text{ or } 90^{\circ} + \theta_0 \le \theta \le 180^{\circ}\}$. Adding full rate of $\mathbf{I} = [I_1, I_2, \cdots, I_D]$ to objective function, the objective function is written as

$$Fitness(\mathbf{I}) = \begin{cases} -MSLL(\mathbf{I}), & \text{if} \quad cRat \le eRat; \\ -\rho \cdot MSLL(\mathbf{I}), & \text{else} \end{cases} \tag{18}$$

where $MSLL$ is maximum value of relative side lobe level and $\rho << 1$. Full rate $cRat$ is calculated by filled array. Expected full rate $eRat$ is calculated by expected filled array.

The initial positions of CQBCO are randomly chosen from the solution space. All quantum bit positions are initialized as $1/\sqrt{2}$. The nectar amount function is identical with the fitness function. The fitness function is identical with objective function of thinned array. The goal of the objective function is to evaluate the status of each bee. In the thinned array optimization, the evolutionary target of CQBCO is the maximization of nectar amount function (objective function). According to the above introduction, the work processes of CQBCO for thinned array are shown below:

Step1: Set parameters of CQBCO according to requirement of thinned array. To initialize the bee colony, it includes the random food source position, the bee's quantum position and the bee's local optimal position.
Step2: Evaluate every bee's nectar amount. Record the global optimal position.
Step3: Update quantum position and position of each employed bee.
Step4: Update the quantum position and position of each onlooker.
Step5: For each new position of each bee, the nectar amount is evaluated.
Step6: Update employed bee's local optimal position. Record the global optimal position.
Step7: If $\mathrm{mod}(t,m)=0$, for onlooker $q(q = 0.5h+1, 0.5h+2, \cdots, h)$, use mutation operator to generate new position $\mathbf{u}_q^{t+1}$. Compute the fitness value of $\mathbf{u}_q^{t+1}$.Compared $\mathbf{u}_q^{t+1}$ and $\mathbf{p}_q^{t+1}$, if $\mathbf{u}_q^{t+1}$ is superior to $\mathbf{p}_q^{t+1}$,then $\mathbf{p}_q^{t+1} = \mathbf{u}_q^{t+1}$.
Step8: If the algorithm does not attain the stop condition (the stop condition is set as maximum iteration times), then go to step 3, else the algorithm stops and outputs the global optimal position.

EXPERIENMENT AND SIMULATION

In the process of simulation, we set identical initial population for GA, PSO and CQBCO. For GA [21,22], the possibility of cross is 0.8, the possibility of mutation is 0.01. As for PSO [23], the parameters are set according to corresponding references. For CQBCO, we set $m=2$, $e_1=0.06$, $e_2=0.03$, $e_3=0.06$, $e_4=0.03$, $c_1=c_2=0.1/D$, $eRat=0.7$, $\rho=0.001$. For comparison, all intelligence algorithms will be terminated at the same maximal iterations number (1000). The population size of the GA, the PSO and the CQBCO are supposed to be 50. All the results of optimal objective value are the average of 50 times. During the simulation, the first element and the last element are turned "on.", and spacing between two elements is $w=\sigma/2$.

We can see from Fig.5, it provides the performance offered by the CQBCO approach, and we provide simulation results in terms of objective value versus number of iterations. It is obvious that the CQBCO is superior to the PSO and the GA on the average of objective value. The faster convergence rate of CQBCO is obvious.

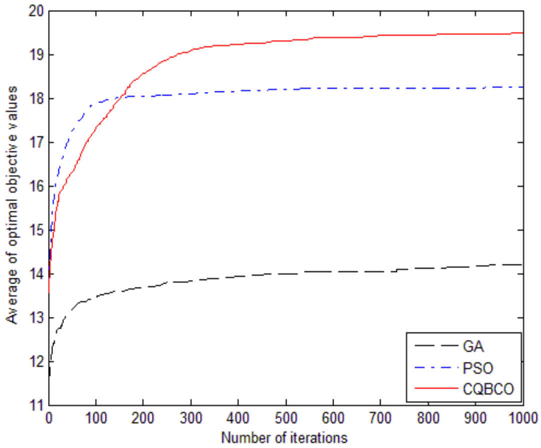

Fig.5: Objective function performance of three algorithms with 50 elements.

We can see from Fig.6, it illustrates the performance offered by CQBCO, PSO and GA approaches, and we provide simulation results in terms of objective value verse number of iterations. It is clearly that GA has poor performance. Although it is not obvious that CQBCO is better than PSO, we can know that the average of optimal objective value of CQBCO still has a tendency to rise. So we can conclude that CQBCO outperforms PSO and GA in convergence precision. And the proposed CQBCO algorithm can overcome the disadvantages of the previous intelligence algorithms.

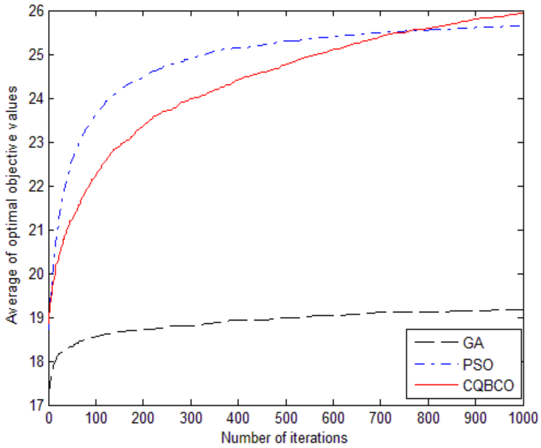

Fig. 6: Objective function performance of three algorithms with 200 elements.

From Fig.7 we can see that CQBCO has good performance in the amplitude of thinned array. We provide simulation results about amplitude of thinned array. Maximum side lobe level obtained by CQBCO is as low as -18.991dB, we obtain side lobe value of -18.441dB when we adopt PSO, but we just obtain side lobe value of -10.006dB when we adopt GA. Side lobe value obtained by CQBCO is 0.55dB lower than that of PSO and 8.985dB lower than that of GA. We can conclude that the performance of CQBCO is better than PSO and GA.

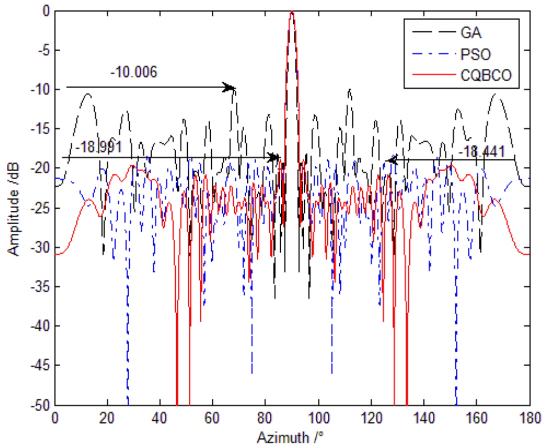

Fig.7: Amplitude performance of three algorithms with 50 elements

Fig.8 shows the amplitude $FF(\theta)$ of three algorithms with 200 elements. We provide simulation results in terms of amplitude versus azimuth. We obtained the maximum side lobe level as slow as -24.513dB when we use CQBCO, but we just obtained -22.906dB and 16.868dB maximum side lobe level when we adopt PSO and GA, respectively. Side lobe value obtained by CQBCO is 1.607dB lower than that of PSO and 7.645dB lower than that of GA. It is obvious that. CQBCO obtains a lower maximum side lobe level. It provides that CQBCO has better global convergence property and excellent maximum side lobe level.

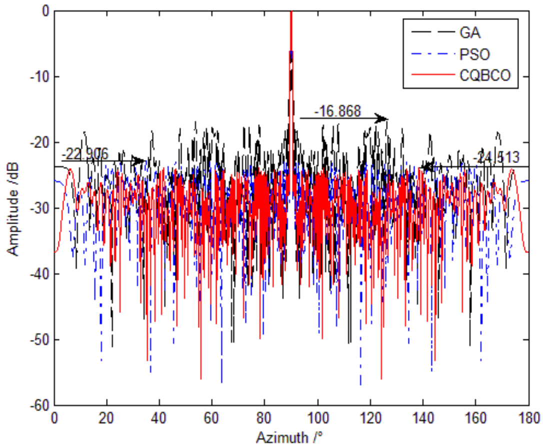

Fig. 8: Amplitude performance of three algorithms with 200 elements.

CONCLUSION

This paper has proposed a CQBCO algorithm which is a novel algorithm for discrete optimization problems. Though testing classical benchmark functions, it can be seen that the CQBCO overcomes the disadvantage of local convergence and has much accurate convergence value, and CQBCO algorithm is superior to other classical evolutionary algorithms. It can be seen that CQBCO has a good universality, so it is easy transplanted to solve other engineering optimization problem. There is no doubt that advances in parallel computing would make CQBCO more attractive and practical for thinned array optimization.

Acknowledgments
This work was supported by the National Natural Science Foundation of China (No. 61102106), China Postdoctoral Science Foundation (No. 2013M530148), Heilongjiang Postdoctoral Fund (No. LBH-Z13054) and the Fundamental Research Funds for the Central Universities (No. HEUCF140809).

REFERENCES

[1] Kennedy, J.; Eberhart, R, Discrete binary version of the particle swarm optimization. *IEEE International Conference on Computational Cybernetics and Simulation*. **,1997** ,5,4104–4108,.
[2] Fu, Y.G.; Ding, M.Y.; Zhou, C.P., *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and*

Humans. **2012**, 42(2), 511-526.
[3] Karaboga, D. ; Basturk, B., *Applied Soft Computing.* ,**2008** ,8(1), 687-697.
[4] Liu, L.L.; Yang, S.X.; Wang D.W., *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics.*, **2010**, 40(6), 1634-1648.
[5] Wong, L.P.; Chi, Y. P.; Malcolm, Y. H. L.; Chin, S.C., Bee colony optimization algorithm with big valley landscape exploitation for job shop scheduling problems. *Proceedings - Winter Simulation Conference*., **2008**, 2050-2058.
[6] Han, K. H.; Kim, J. H., Genetic quantum algorithm and its application to combinatorial optimization problems. *Proceedings of the 2000 IEEE Conference on Evolutionary Computation, Piscataway: IEEE Press*., **2000**, 2, 1354–1360.
[7] Yang, J.-A.; Li B.; Zhuang, Z., *J. Electron*. **2003**, 20(1), 62–68.
[8] Gao, H.Y.; Liu, Y.Q.; Diao, M., *International Journal of Innovative Computing and Applications*. ,**2011**, 3(3),160-168.
[9] Lo, Y. T.; Lee, S.W., Aperiodic arrays, in Antenna Handbook, Theory, Applications, and Design. New York Van Nostrand. **1988**.
[10] Willey, R. E., *IEEE Antennas Propagat*., **1962**, 10(4), 369-377.
[11] Galejs, J., Minimization of sidelobes in space tapered linear arrays. *IEEE Trans. Antennas Propagat*., **1964**, 12(4),835-836.
[12] Steinberg, B. D., Principles of Aperture and Array System Design. New York Wiley. **1976**.
[13] Lo, Y. T.; Lee, S. W., *IEEE Trans. Antennas Propagat*., **1966**, 14(1), 22-30.
[14] Press W. H. et al., Numerical recipes. New York: Cambridge University Press. **1992**
[15] Cheston T. C.; Frank J., Phased array radar antennas. in Radar Handbook. M. Skolnik, Ed. New York McGraw-Hil. **1990**.
[16] Davis , L., Generic algorithms and simulated annealing. Los Altos, CA: Morgan Kaufmann. **1987**.
[17] Ruf, C.S., *IEEE Trans. Antennas Propagat*., **1993**, 41(1), 85-90.
[18] Jiao, L.C.; Li, Y.Y.; Gong, M.G.; Zhang, X.Y., *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*. , **2008**, 38(5), 1234–1253.
[19] May, R. M., *Nature.*, **1976**, 261(5560), 459-467.
[20] Coelho, L. D. S., Bora, T. C., Lebensztajn, L. *IEEE Transactions on Magnetics*., **2012**,48(2), 751-754.
[21] Zhao, Z.J., Peng, Z., Zheng, S.L., Shang, J.N., *IEEE Transactions on Wireless Communications*., **2009,**8(9),4421-4425.
[22] Haupt,Randy L. *IEEE antennas and propagation society*., **1994**,42(7), 993-99.
[23] Gao, B., Jin, R.H. , Geng, J.P. *Journal of Electronic & Information Technology*., **2007**, 29(12), 3015-3019.