Improved Quantum Ant Colony Algorithm based on Bloch Coordinates

Xiaofeng Chen
Software College, Northeastern University, Shenyang, China
Email: neucxf@163.com

Xingyou Xia and Ruiyun Yu
Software College, Northeastern University, Shenyang, China
Email: {xiaxy, yury}@mail.neu.edu.cn

Abstract—**The Ant Colony Algorithm is an effective method for solving combinatorial optimization problems. However, in practical applications, there also exist issues such as slow convergence speed and easy to fall into local extremum. This paper proposes an improved Quantum Ant Colony Algorithm based on Bloch coordinates by combining Quantum Evolutionary Algorithm with Ant Colony Algorithm. In this algorithm, the current position information of ants is represented by the Bloch spherical coordinates of qubits; position update, position variation and random behavior of ants are all achieved with quantum rotation gate. Simulations of function extremum problem, TSP problem and QoS multicast routing problem were conducted, the results indicated that the algorithm could overcome prematurity, with a faster convergence speed and higher solution accuracy.**

Index Terms—**quantum computing, Ant Colony Algorithm, Quantum Ant Colony Algorithm**

I. INTRODUCTION

Ant Colony Algorithm (ACA) [1] is a heuristic algorithm for solving combinatorial optimization or function optimization problems. It has advantages such as positive feedback, strong robustness, excellent distributed computing mechanism, easy to combine with other algorithms, etc., which has been widely used in the NP-complete problem. In recent years, ACA has been applied to the fields such as knapsack problem [2], Assignment Problem [3], Job-shop Assignment [4], Sequential Ordering [5], Network Routing [6], Vehicle Routing [7], Power System [8] and Controls Parameter Optimization [9], etc. and obtained good effect. Meanwhile, like other swarm intelligence optimization algorithms, ACA also has some shortcomings in the application process, such as: easy to fall into local optimization, slow convergence speed, etc.

A quantum ant colony algorithm (QACO), based on the concept and principles of quantum computing can overcome this defect. In [17], a QACO-based edge detection algorithm was proposed. Quantum bit (qubit) and quantum rotation gate are introduced into QACO to represent and update the pheromone respectively. Experiments and comparisons show that QACO is an efficient and effective approach in image edge detection. In order to select the optimal parameter, quantum-inspired ant colony optimization is employed to select the parameter of relevance vector machine in [18]. Quantum-inspired ant colony optimization is well suited to multi-objective optimization problems with excellent results. By measuring experimentally the vibration signals of the gear system at different rotating speeds for different faults, the testing signals are obtained. In [19], a novel parallel ant colony optimization algorithm based on quantum dynamic mechanism for traveling salesman problem (PQACO) was proposed. The use of the improved 3-opt operator provides this methodology with superior local search ability. A global optimization method was proposed to analyze ground state energy of quantum mechanical systems in [20], which It simulates the way that real ants find a shortest path from nest to food source and back. To eliminate system disturbances and noise from the high levels of data, a novel quantum ant colony optimization (QACO) algorithm was proposed to select the fault features [21].

This paper proposed an improved quantum ant colony algorithm based on the Bloch Spherical Coordinate [11] (BIQACA), and various solution space transformational models and fitness functions are planned for different optimization problems. Algorithm in this paper is verified by function extreme value problem, Traveling Salesman Problem and QoS multicast routing problem respectively. The result of simulation shows that the algorithm not only expresses high efficiency of quantum computing, but also maintains the preferable optimizing and robustness of colony algorithm.

II. QUANTUM ANT COLONY ALGORITHM (QACA)

Any point on the Bloch sphere can be identified via θ and φ as: $|\varphi\rangle = [\cos\varphi\sin\theta, \sin\varphi\sin\theta, \cos\theta]^T$. Suppose there are a total of n ants in the ant colony, where each ant carries a group (m units) of qubits, current position of ant is represented by Bloch spherical coordinate, corresponding to approximate solution of optimization problem.

A. Initialize Ant Colony

doi:10.4304/jcp.8.6.1536-1543

P_i is set as the location of the ith ant, considering that the randomness of coding for ant colony and constraint conditions for probability amplitude of the quantum state, the initialization of BIQACA is expressed as:

$$\begin{bmatrix} P_{ix}^j \\ P_{iy}^j \\ P_{iz}^j \end{bmatrix} = \begin{bmatrix} \begin{vmatrix} \cos\phi_{i1}\sin\theta_{i1} \\ \sin\phi_{i1}\sin\theta_{i1} \\ \cos\theta_{i1} \end{vmatrix} \begin{vmatrix} \cos\phi_{i2}\sin\theta_{i2} \\ \sin\phi_{i2}\sin\theta_{i2} \\ \cos\theta_{i2} \end{vmatrix} \begin{vmatrix} \cdots \\ \cdots \\ \cdots \end{vmatrix} \begin{vmatrix} \cos\phi_{im}\sin\theta_{im} \\ \sin\phi_{im}\sin\theta_{im} \\ \cos\theta_{im} \end{vmatrix} \end{bmatrix} \quad (1)$$

Where $\varphi_{ij} = 2\pi rand$, $\theta_{ij} = \pi rand$, $rand$ are random numbers between (0, 1) ; $i \in \{1,2,\cdots,n\}$, $j \in \{1,2,\cdots,m\}$, n for number of ant; m for number of qubit. 3 coordinates of qubit are regarded as 3 paratactic genes, and each ant contains 3 gene chains, which are called X-chain, Y-chain and Z-chain respectively, each gene chain stands for an optimal solution P_{ix}^j , P_{iy}^j , P_{iz}^j .

B. Transformation of Solution Space

In the optimization of specific problems in BIQACA, transformation between the unit quantum space and solution space of optimization problem is needed, making each probability amplitude of qubit on ant correspond to an optimization variable of solution space. In this paper, the function extremum problem, TSP problem and QoS multicast routing problem are taken as examples to explain the process.

Solution space transformation approach for function extreme-value problem: propose the domain of definition of variable X^j is its solution space $[a_j, b_j]$, record the jth qubit in P_i as $\left[\cos\varphi_{ij}\sin\theta_{ij}, \sin\varphi_{ij}\sin\theta_{ij}, \cos\theta_{ij}\right]^T$ by using linear transformation, then the corresponding solution space variable is:

$$\begin{bmatrix} X_{ix}^j \\ X_{iy}^j \\ X_{iz}^j \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1+\cos\varphi_{ij}\sin\theta_{ij} & 1-\cos\varphi_{ij}\sin\theta_{ij} \\ 1+\sin\varphi_{ij}\sin\theta_{ij} & 1-\sin\varphi_{ij}\sin\theta_{ij} \\ 1+\cos\theta_{ij} & 1-\cos\theta_{ij} \end{bmatrix}\begin{bmatrix} b_j \\ a_j \end{bmatrix} \quad (2)$$

Solution space transformation approach for TSP problem and QoS multicast routing problem: this paper has designed two-layer transformational model in the aspect of solution space aiming at the specific characteristic of TSP problem and QoS multicast routing problem, the model contains two transformations--linear transformation and lead transformation.

Linear transformation: qubit is transformed from unit space to lead space. Propose the definitional domain of lead message variable, r^j , is $[0,1]$, formula (2) is used to calculate corresponding lead solution space variable $[\tau_{ix}^j, \tau_{iy}^j, \tau_{iz}^j]^T$.

Lead transformation: impact strength of lead message and inspire message to solution could be regulated by adjusting lead factor and inspire factor. Strategy is selected according to lead probability and roulette to carry out optimal decode. Suppose the current node as i, select node j as the next visiting node:

$$p_{ij}^k = \begin{cases} \dfrac{r_{ij}^{\omega}(t)\bullet\lambda_{ij}^{\upsilon}(t)}{\sum\limits_{s\in allowed_k} r_{is}^{\omega}(t)\bullet\lambda_{is}^{\upsilon}(t)} & \text{j} \in \text{allowed} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where $r_{ij}^{\omega}(t)\bullet\lambda_{ij}^{\upsilon}(t)$ is for message of path, $r_{ij}(t)$ stands for lead message, ω is lead factor; $\lambda_{ij}(t)$ represents inspire message λ_{ij}(t)= $1/d_{ij}$, d_{ij} means the distance from node i to node j, υ is inspire factor; $allowed_k = \{1, 2, \cdots m\} - tabu_k$ means the set of available node may selected by ant k at the time t; $tabu_k$ is used to keep the routing table which obtained by transforming ant k.

C. Definition of Fitness Function

A variety of fitness function needs to be designed for different optimal problems, the more fitness it is, the better solution for individual.

Fitness function of extreme-value problem: suppose $f(X_i)$ as the ith solution, $fit(X_i)$ is the adaptive value for the ith solution. min and max denote the minimum value and maximum value of function, respectively.

$$\min\ fit(X_i) = \begin{cases} \dfrac{1}{1+f(X_i)} & f(X_i) \ge 0 \\ 1+abs(f(X_i)) & f(X_i) < 0 \end{cases} \quad (4)$$

$$\max\ fit(X_i) = \begin{cases} 1+f(X_i) & f(X_i) \ge 0 \\ 1+\dfrac{1}{abs(f(X_i))} & f(X_i) < 0 \end{cases} \quad (5)$$

TSP fitness function: fitness of individual $X_i = \{x_1, x_2, \cdots, x_m\}$ of TSP is defined as the reciprocal of path length represented by individual.

$$fit(X_i) = \frac{1}{D(X_i)} \quad (6)$$

Fitness function for multicast routing problem:

$$fit(T_i) = \frac{W_c(W_d \cdot \Phi(TD - D_{\max}) + W_{dj} \cdot \Phi(TDJ - DJ_{\max}) + W_{pl} \cdot \Phi(TPL - PL_{\max}))}{TC} \quad (7)$$

where TD, TDJ, TPL and TC represent the delay, delay jitter, packet loss rate and cost of multicast tree respectively. Wc=0.5, Wd=0.2, Wdj=0.1 and Wpl=0.2, represent the proportion of the cost, delay, delay jitter and packet loss rate in the fitness function respectively; $\cdot\Phi(X)$ is a penalty function, when $\cdot X \le 0$, $\cdot\Phi(X) = 1$, or else, $\cdot\Phi(X) = 0.5$. It can be seen from the above equation that, the fitness value is the bigger the better.

D. Ant Position Update

In the solution space of optimal problem, suppose $\tau(X_i)$ is the strength of pheromone of kth ant at X_i , initial moment all set as some constant: $\eta(X_i)$ stands for the visibility at X_i . The basic framework of BIQACA described as follows:

1) Selecting the target position of ant movement

By applying the principle of randomness, a number of qubits in the current position were randomly selected to constitute a position update vector S. The transition rule and transition probability of ant k from position X_i to position X_s are:

$$X_s = \begin{cases} \arg\max\limits_{X_s \in P} \{ \tau^{\alpha}(X_s) \bullet \eta^{\beta}(X_s) \} & \text{q} \le q_0 \\ \widetilde{X_s} & \text{q} > q_0 \end{cases} \tag{8}$$

$$p(X_s) = \frac{\tau^{\alpha}(X_s) \bullet \eta^{\beta}(X_s)}{\sum\limits_{X_s, X_u \in P} \tau^{\alpha}(X_u) \bullet \eta^{\beta}(X_u)}, \tag{9}$$

where q$\in$[0, 1] is even-distributed random number, q_0 $\in$[0, 1] is probability parameter, P is the set of occupied points for ant in unit space, $\widetilde{X_s}$ is the selected target location as per formula (8) ; α is the update parameter of pheromone, β is the update parameter of visibility.

2) Realizing the movement of ant towards target position via quantum rotation gate

After the ant has selected the target position, its movement process can be realized by changing the phase of qubit it brought for quantum rotation gate. In unit space, suppose the current position for ant at time t is P_i , selected target position is P_k , update vector *of* P_i is S, then the update of phase angle increment at P_i is

$$\Delta\phi_{ij}^{t} = \begin{cases} (\phi_{kj} - \phi_{ij}) \times rand_j & \phi_{kj} \ne \phi_{ij} \\ \Delta\phi_{ij} & \phi_{kj} = \phi_{ij} \end{cases} \tag{10}$$

$$\Delta\varphi_{ij}^{t+1} = \begin{cases} \Delta\varphi_{ij}^{t} + 2\pi & \Delta\varphi_{ij}^{t} < -\pi \\ \Delta\varphi_{ij}^{t} & -\pi \le \Delta\varphi_{ij}^{t} \le \pi \\ \Delta\varphi_{ij}^{t} - 2\pi & \Delta\varphi_{ij}^{t} > \pi \end{cases} \tag{11}$$

$$\Delta\theta_{ij}^{t} = \begin{cases} (\theta_{kj} - \theta_{ij}) \times rand_j & \theta_{kj} \ne \theta_{ij} \\ \Delta\theta_{ij} & \theta_{kj} = \theta_{ij} \end{cases} \tag{12}$$

$$\Delta\theta_{ij}^{t+1} = \begin{cases} \Delta\theta_{ij}^{t} + \pi & \Delta\theta_{ij}^{t} < -\pi/2 \\ \Delta\theta_{ij}^{t} & -\pi/2 \le \Delta\theta_{ij}^{t} \le \pi/2 \\ \Delta\theta_{ij}^{t} - \pi & \Delta\theta_{ij}^{t} > \pi/2 \end{cases} \tag{13}$$

where $j \in \{S(1), S(2), \cdots, S(sm)\}$, $rand_j$ is random number between [0, 1]; $\Delta\varphi_{ij}$, $\Delta\theta_{ij}$ can be obtained using (17), (18).

Update of probability amplitude of qubit based on quantum rotation gate

$$U = \begin{bmatrix} \cos\Delta\varphi_{ij}^{t+1}\cos\Delta\theta_{ij}^{t+1} & -\sin\Delta\varphi_{ij}^{t+1}\cos\Delta\theta_{ij}^{t+1} & \sin\Delta\theta_{ij}^{t+1}\cos(\varphi_{ij}^{t} + \Delta\varphi_{ij}^{t+1}) \\ \sin\Delta\varphi_{ij}^{t+1}\cos\Delta\theta_{ij}^{t+1} & \cos\Delta\varphi_{ij}^{t+1}\cos\Delta\theta_{ij}^{t+1} & \sin\Delta\theta\sin(\varphi_{ij}^{t} + \Delta\varphi_{ij}^{t+1}) \\ -\sin\Delta\theta_{ij}^{t+1} & -\tan(\varphi_{ij}^{t}/2)\sin\Delta\theta_{ij}^{t+1} & \cos\Delta\theta_{ij}^{t+1} \end{bmatrix} \tag{14}$$

$$\begin{bmatrix} \cos\varphi_{ij}^{t+1}\sin\theta_{ij}^{t+1} \\ \sin\varphi_{ij}^{t+1}\sin\theta_{ij}^{t+1} \\ \cos\theta_{ij}^{t+1} \end{bmatrix} = U \begin{bmatrix} \cos\varphi_{ij}^{t}\sin\theta_{ij}^{t} \\ \sin\varphi_{ij}^{t}\sin\theta_{ij}^{t} \\ \cos\theta_{ij}^{t} \end{bmatrix} = \begin{bmatrix} \cos(\varphi_{ij}^{t} + \Delta\varphi_{ij}^{t+1})\sin(\theta_{ij}^{t} + \Delta\theta_{ij}^{t+1}) \\ \sin(\varphi_{ij}^{t} + \Delta\varphi_{ij}^{t+1})\sin(\theta_{ij}^{t} + \Delta\theta_{ij}^{t+1}) \\ \cos(\theta_{ij}^{t} + \Delta\theta_{ij}^{t+1}) \end{bmatrix} \tag{15}$$

Apparently, U-gate can rotate the phase of qubit by $\Delta\varphi_{ij}^{t+1}$ and $\Delta\theta_{ij}^{t+1}$.

3) Adjustment strategy of search space

In BIQACA, the search space for each qubit is designed as $[lowBd_{ij}, upBd_{ij}]$, the search space at initialization is $[0.25\pi, 0.75\pi]$, during optimizing process of ants, these search spaces are related with the contraction level of each qubit, and decrease exponentially, which can significantly improve the solution accuracy of the algorithm.

$$\begin{bmatrix} lowBd_{ij}^{t+1} \\ upBd_{ij}^{t+1} \end{bmatrix} = \begin{bmatrix} \dfrac{1}{nf^{\,nL_{ij}^{t}}} \end{bmatrix} \begin{bmatrix} lowBd_{ij}^{t} \\ upBd_{ij}^{t} \end{bmatrix} \tag{16}$$

where $j \in \{S(1), S(2), \cdots, S(sm)\}$, S is the update vector of ants, $nf = 2$ is the constriction factor, nL_{ij}^{t} represents the contraction level of t-th iteration.

4) Processing of ant position variation

Suppose the current position is P_i , update vector of P_i is S, the search space of P_i is $[lowBd_{ij}, upBd_{ij}]$. Then the update of phase angle increment at P_i is:

$$\begin{cases} \Delta\varphi_{ij} = (2rand - 1)(upBd_{ij} - lowBd_{ij}) \\ \Delta\varphi_{ij} = sign(\Delta\varphi_{ij})(abs(\Delta\varphi_{ij}) + lowBd_{ij}) \end{cases} \tag{17}$$

$$\begin{cases} \Delta\theta_{ij} = (2rand - 1)(upBd_{ij} - lowBd_{ij}) \\ \Delta\theta_{ij} = sign(\Delta\theta_{ij})(abs(\Delta\theta_{ij}) + lowBd_{ij}) \end{cases} \tag{18}$$

5) Random behavior of ants

If P_i is not improved after continuous limited-time cycles, the position should be abandoned, the ants will generate a new $P_i^{'}$ through random behavior to substtute P_i .

$$\varphi_{ij}^{'} = \text{mean}(\varphi_{\text{i}}) + \Delta\varphi_{ij} \tag{19}$$

$$\theta_{ij}^{'} = \text{mean}(\theta_{\text{i}}) + \Delta\theta_{ij} \tag{20}$$

where $i \in \{1, 2, \cdots, n\}$, $j \in \{S(1), S(2), \cdots, S(sm)\}$, S is the update vector of current position, $\Delta\varphi_{ij}$ and $\Delta\theta_{ij}$ are updated using (17), (18), $mean(\theta_i)$ is the mean value of vector of phase angle θ_i at P_i .

6) Update rules for pheromone intensity and visibility

When the ant completes a traverse, the current position is mapped into the solution space of optimal problem from unit space, fitness function is calculated, and the intensity and visibility of pheromone at current position should be updated.

$$\begin{cases} \tau(X_i) = (1-\rho)\tau(X_i) + \rho\vartriangle\tau(X_i) \\ \vartriangle\tau(X_i) = Qfit(X_i) \end{cases} \tag{21}$$

$$\eta(X_i) = fit(X_i) \tag{22}$$

where $(1-\rho) \in [0,1]$ is the evaporation coefficient of pheromone, Q is the enhancement coefficient of pheromone.

E. Description of BIQACA

Taking the function extremum problem as an example, BIQACA implementation steps are described as follows:

Step 1: Setup relevant parameters such as the number of ants, maximum number of iterations, number of limits limit, contraction level nL_{ij}, constriction factor nf, Maximum contraction level MaxL, reset contraction level resetL, search space $[lowBd_{ij}, upBd_{ij}]$, etc.

Step 2: Randomly generate initial position of ants according to (1), transform the solution space according to (2), and calculate the fitness of each ant according to (5) or (6). Update the pheromone intensity and visibility according to (21) and (22).Record the current optimum solution, i.e. global optimum solution GBest. Initialize the conceptual vector $trial(i) = 0$, record the number of non-updates at the position of ant.

Step 3: Update the search space $[lowBd_{ij}, upBd_{ij}]$ according to (16).

Step 4: Select a moving target for each ant in the ant colony according to (8) and (9), then realize the movement of ants using quantum rotation gate in light of (10), (12) and (14).

Step 5: For each ant, according to mutation probability, realize the variation of ant's position using quantum rotation gate in light of (17) and (18).

Step 6: Transform the solution space according to (2), calculate the fitness of each ant according to (5) or (6). Update the current position if the new position is better than the current one; otherwise $trial(i) = trial(i) + 1$, update contraction level $nL(i, j) = nL(i, j) + 1$, if $nL(i, j)$ >MaxL, $nL(i, j)$ =resetL.

Step 7: Determine whether $trial(i)$ is greater than the limit, if $trial(i)$ >limit, abandon the current position of the i-th ant, and generate a new position according to (19), (20) and (15), *perform space transformation* in light of (2), calculate the fitness of each ant according to (5) or (6), $trial(i) = 0$.

Step 8: Update the pheromone intensity and visibility according to (21) and (22). Record the current local optimum position, Best, and local worst position, Worst.

Step 9: Determine whether the local optimum position, Best is greater than the global optimum position, GBest, if Best>GBest, update the global optimal position, GBest with local optimum position, Best, otherwise, update the local worst position, Worst with global optimum position, GBest.

Step 10: Update the number of iterations t=t+1. If the current number of iterations t>maxgen or accuracy of convergence is met, stop the search, output the global optimum position, or else, turn to Step 3.

III. Simulation Experiment

To verify the effectiveness and feasibility of BIQACA algorithm, function extremum problem, traveling salesman problem and multicast routing problem were selected for testing. The simulation programs were programed and implemented in MATLAB 2009a, test results were obtained in a PC with an Intel Core (TM) i5 CPU running at 3.2GHz, and a 2.8GB RAM.

A. Function Extremum Problem

Three internationally commonly used functions f1~f3 were selected to test BIQACA performance when the number of independent variables was 2 and 30 respectively

$$f_1(x, y) = 0.3\cos 3\pi x - 0.3\cos 4\pi y - x^2 - y^2 - 0.3, -1 \le x, y \le 1 \tag{23}$$

$$f_2(x_i) = \sum_{i=1}^{n}(-x_i \sin(\sqrt{|x_i|})), -500 \le x_i \le 500 \tag{24}$$

$$f_3(x_i) = \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i) + 10), -5.12 \le x_i \le 5.12 \tag{25}$$

1) When the number of independent variables is 2

Test function is f1, the optimization objective is to obtain the maximum value.

Algorithm parameters: maximum number of iterations maxgen=500, number of ants n=20, probability parameter q_0 =0.5, evaporation coefficient $1-\rho$ =0.05, pheromone update parameter α =1, visibility update parameter β =5, pheromone enhancement coefficient Q =10, mutation probability P_m =0.05; the program was terminated when the BIQACA algorithm found the optimal solution or had run for gen=500 iterations. Simulations were conducted using the CQACO algorithm and ACO algorithm in Ref. [12] respectively, each algorithm was run for 50 times independently under the same conditions, and their optimal value (Best), optimal mean value (M-best), number of success and average number of iterations were recorded, optimization results were compared in Table 1.

TABLE I.
COMPARISON OF EXPERIMENTAL RESULTS OF 3 ALGORITHMS WITH TEST FUNCTION F 1

Func/opt	Status	Algorithm		
		ACO	CQACO	BIQACA
f_1/ 0.24	Best	0.2400	0.2400	0.2400
	M-best	0.1720	0.2388	0.2400
	Con-times	42	48	50
	Ave-Steps	198.16	84.04	17.06

It can be seen from Table 1 that, BIQACA algorithm's optimization efficiency is the highest, its optimization results is also the greatest, with the success rate of 100%; followed is CQACO, with a success rate of 96%; the last is ACO, with a success rate of 84%.

2) When the number of independent variables is 30

Test functions were f2, f3, optimization goal is to obtain the minimum value.

TABLE II.
COMPARISON OF EXPERIMENTAL RESULTS OF 3 ALGORITHMS WITH TEST FUNCTION F2 AND F3

Func/ Opt	Status	Algorithm		
		OGA/Q	LEA	BQACA
f_2/ -12569.5	M-nfun	302116	287365	236613
	M-best	-12569.454	-12569.454	-12569.487
	St.dev	6.447×10^{-4}	6.831×10^{-4}	6.5856×10^{-6}
f_3/ 0	M-nfun	224710	223803	223230
	M-best	0	2.103×10^{-18}	0
	St.dev	0	3.0359×10^{-18}	0

Algorithm parameters: maximum number of iterations maxgen=1500, number of ants n=100, other parameters were the same as test 3.1.1. Simulations were conducted using the BIQACA algorithm, as well as the OGA/Q algorithm in Ref. [13] and LEA algorithm in Ref. [14] respectively, each algorithm was run independently for 50 times under the same conditions, and their average number of function evaluations (M-nfun), optimal mean value (M-best) and standard deviation (St. dev) were recorded, optimization results were compared in Table 2.

It can be seen from Table 2 that, the BIQACA algorithm is obviously superior to the OGA/Q algorithm and LEA algorithm with respect to optimal mean value, average number of function evaluations and standard deviation of function f2, f3; for f3, the BIQACA algorithm and the OGA/Q algorithm could both find the optimal solutions.

For function f2, the three algorithms all failed to find the optimal solutions, but the solution finding quality of BIQACA algorithm is significantly better than that of the LEA algorithm and OGA/Q algorithm, the standard deviation obtained by the BIQACA algorithm is also less than that of the LEA algorithm and OGA/Q algorithm.

B. Traveling Salesman Problem

Take symmetric-distance TSP as example, five problems with different data scale was selected from TSPLIB database as cases to verify the performance of the algorithm. Compare the result with Common Genetic Algorithm (CGA), Common Particle Swarm Optimization (CPSO) and Common Ant Colony Algorithm (CACA) respectively.

Algorithm parameters: each algorithm prescribes a limit to algebra of 100 and population of 50. in CGA algorithm, integer encoding is adopted; in CPSO algorithm, integer encoding is also adopted, inertia factor W =0.5, self-factor C_1 =0.3, global factor C_2 =0.7; in CABC algorithm, integer encoding adopted as well, other parameters with the function extreme value problem; In BIQACA algorithm, transfer factor parameter ω = 1, stimulating factor parameter υ = 5, other parameters with the function extreme value problem.

Then in every case, 20 experimental data would be taken, and table 3 shows contrast of optimization results. Figure 1 is the Oliver30 Optimization Results and Figure 2 is the EIL51 Optimization Results. Figure 3 shows the best solution of Oliver30 quantum ant colony algorithm, the total distance for 424. Figure 4 shows the best solution of EIL51 quantum ant colony algorithm, the total distance is 458.

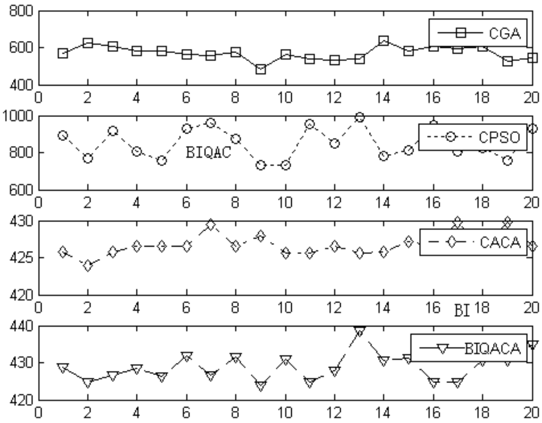

Figure 1. The Oliver30 Optimization Results

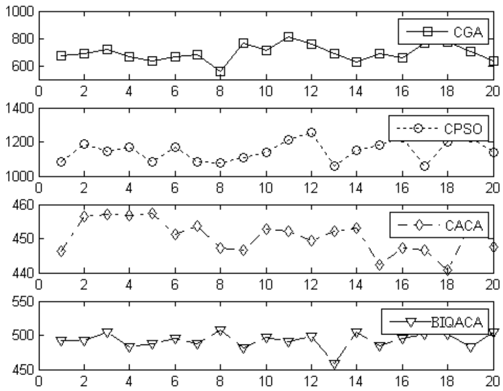

Figure 2. The EIL51 Optimization Results

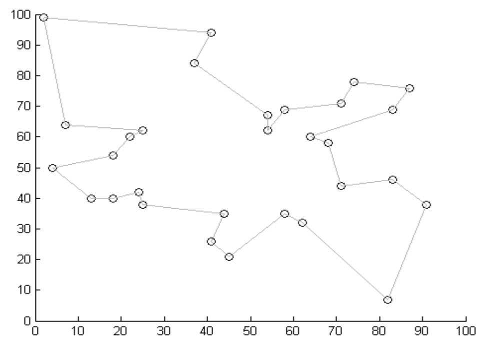

Figure 3. The Oliver30's Best Solution by using BIQACA

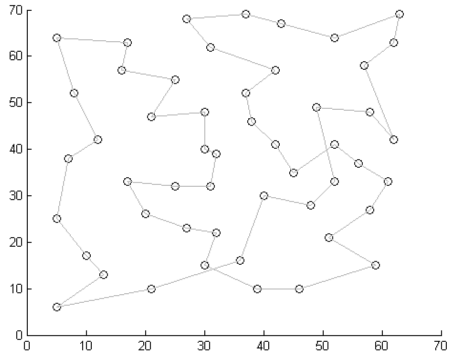

Figure 4. the EIL51's Best Solution by using BIQACA

Statistics analysis of data based on table 3: from time perspective, the average time of CPSO algorithm is shortest, secondly CGA algorithm, BIQACA algorithm followed, and CACA algorithm is the last one; from steps perspective, BIQACA algorithm and CACA algorithm are at the same level and the convergence rate of them is more preferable than the other two algorithms; from the perspective of calculation result, the optimal solution of BIQACA algorithm and CACA algorithm can reach an ideal resolution recommended by TSPLIB database when urban scale is small.

When the urban scale is large, the BIQACA's optimal solution can approach to that of CACA algorithm, and its average solution and standard deviation is better than that of CPSO algorithm and CGA algorithm. To sum up, BIQACA algorithm in this paper is feasible and effective.

TABLE III.
TSP CALCULATION RESULTS

Test library	algorithm	Optimal solution	Worst solution	Mean value	Standard deviation	Mean-square deviation	Mean time	Average step
Uleysses22	CGA	76.9162	93.5551	85.0164	4.3255	18.7098	1.4645	100.0000
	CPSO	78.9062	125.7221	105.8074	9.7533	95.1269	0.2646	88.8000
	CACA	75.9832	77.3018	76.3748	0.3878	0.1504	1.9651	60.0000
	BIQACA	75.9832	76.6314	76.2579	0.2226	0.0496	1.6571	62.0000
Oliver30	CGA	482.6800	639.5484	570.9215	37.2503	1387.5816	2.0194	100.0000
	CPSO	729.5129	990.8997	850.9355	84.6154	7159.7655	0.3277	84.7500
	CACA	423.7406	429.7853	426.7401	1.4871	2.2115	3.1717	54.1000
	BIQACA	423.7406	438.6092	428.8924	3.8389	14.7375	2.2512	50.0500
EIL51	CGA	558.7636	812.5886	694.0846	60.0962	3611.5476	3.4789	100.0000
	CPSO	1055.4600	1252.3856	1146.5343	60.0677	3608.1277	0.4929	93.4500
	CACA	440.7957	457.3709	450.5731	4.8484	23.5068	10.0465	60.8000
	BIQACA	458.3380	507.1071	492.5977	11.4446	130.9781	5.9311	55.4500
EIL76	CGA	949.3124	1200.5125	1082.3078	77.0343	5934.2825	5.3974	100.0000
	CPSO	1698.2136	2090.3629	1885.3653	106.6634	11377.0827	0.6863	94.6500
	CACA	566.8443	576.1675	572.0972	2.8604	8.1822	18.5971	50.4500
	BIQACA	628.7805	670.0715	653.8038	11.0728	122.6080	11.8519	59.4000
GR96	CGA	1082.0048	1467.8411	1232.7158	98.6122	9724.3659	7.1479	100.0000
	CPSO	2274.0041	2885.1164	2567.0244	160.3023	25696.8139	0.8681	90.8500
	CACA	544.8082	558.9636	552.8529	4.3274	18.7262	37.43502	63.7500
	BIQACA	594.9407	648.2708	626.9822	11.5389	133.1465	22.903469	54.1000

C. QoS Multicast Routing Problem

In order to compare with the GA algorithm in Ref. [15] and QCMR-ACS in Ref. [16], the network architecture model the same with them was adopted in the experiment, as shown in Figure 5.

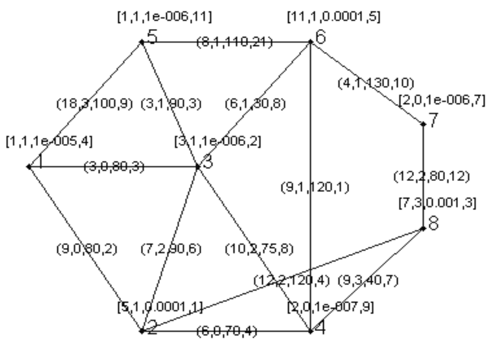

Figure 5. 8-node network model

In this typical 8-node network model, network can be represented using picture G (V, E), where V (D, DJ, PL, C) represents network node set, E (D, DJ, B, C) represents link set, and D, DJ, PL, C and B represent delay (ms), delay jitter (ms), packet loss rate, cost and bandwidth (Mb/s) respectively.

The core idea of multicast routing algorithm is: in each iteration, firstly, qubits pass linear transformation and state transition transformation, and complete the conversion of quantum information in the routing path; secondly, use the routing paths to generate the multicast tree of the iteration; thirdly, compute the delay, delay jitter, packet loss rate, bandwidth, and cost of the multicast tree; finally, calculate the fitness of the multicast tree.

Multicast tree generation: the original multicast tree is generated using the vector information of multiple routing paths, and through the conversion from vector to matrix, the original multicast tree was pruned and processed to obtain the multicast tree.

Algorithm parameters: source node s=1, destination node M=[2, 4, 5, 7], maximum number of iterations maxgen=16, number of ants n=8, mutation probability $P_m = 0.1$.

Table 4 shows the optimization results of BIQACA algorithm when $D_{\max} = 46$, $DJ_{\max} = 8$, $B_{\min} = 70$ and $PL_{\max} = 0.001$ were constrained; Figure 6 shows the cost convergence curves of multicast tree for three algorithms (GA, QCMR-ACS, BIQACA)

TABLE IV.
BIQACA ALGORITHM OPTIMIZATION RESULTS

Route request	Optimal multicast tree	Delay	Delay jitter	Packet loss rate	Cost
s=1	(1, 2), (1, 3), (3, 4), (3, 5)	45	7	0.0001	66
M=[2, 4, 5, 7]	(4, 6), (6, 7)				

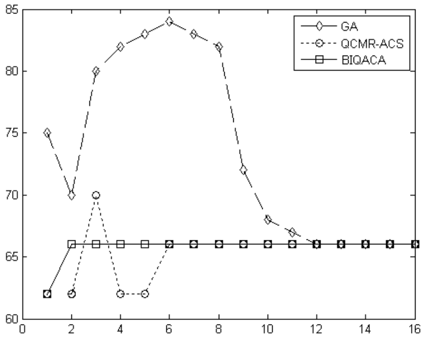

Figure 6. Cost convergence curves of multicast tree for three algorithms

Table 5 shows the optimization results of BIQACA algorithm when $D_{\max} = 50$, $DJ_{\max} = 6$, $B_{\min} = 70$ and $PL_{\max} = 0.001$ were constrained; Figure 7 shows the cost convergence curves of multicast tree for three algorithms (GA, QCMR-ACS, BIQACA)

TABLE V.
BIQACA ALGORITHM OPTIMIZATION RESULTS

Route request	Optimal multicast tree	Delay	Delay jitter	Packet loss rate	Cost
s=1	(1, 2), (1, 3), (2, 4), (3, 5)	49	5	0.0002	62
M=[2, 4, 5, 7]	(4, 6), (6, 7)				

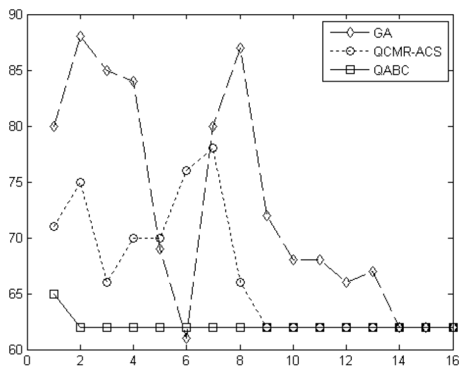

Figure 7. Cost convergence curves of multicast tree for three algorithms

It can be seen from Figure 6 and Figure 7 that, under the conditions of the two multicast routing constraints, the three algorithms can all converge to the global optimal solution, for GA algorithm in Ref. [15], evolution generations during convergence were 12 and 14, respectively, QCMR-ACS algorithm in Ref. [16] requires 6 and 9 generations respectively, while the BIQACA algorithm herein requires only 2 generations, its convergence speed is much faster than that of the GA and QCMR-ACS based QoS multicast routing algorithms, thus the feasibility and effectiveness of the BIQACA algorithm are verified.

IV. CONCLUSION

In this paper, by combining quantum computation and ant colony algorithm, an improved quantum ant colony algorithm based on Bloch coordinates is presented, enriching the research field of quantum intelligence algorithm. From the perspective of quantum computation, this algorithm proposes the adjustment strategy of search space in accordance with the exponential decrease method. A number of qubits at current position of ants are selected using the principle of randomness to constitute the update vector, the position update, position variation and random behavior of ants are all subject to the constraint of update vector, thus improving the convergence speed of the algorithm. At the same time, the random behavior of ants introduced can obviously overcome the prematurity of the algorithm. Different solution space transformation models and fitness functions are designed for different optimization problems, where the overall idea of the algorithm remains the same, the algorithm has strong versatility. Research results show that the new algorithm has certain practical value which can improve the efficiency and accuracy. Compared with conventional intelligence algorithms, BIQACA has stronger search capability and higher efficiency, and is appropriate for complex function optimization and combinatorial optimization problems. At the same time, as a novel optimization algorithm, BIQACA is lack of necessary theoretical proof, experimental verification alone is not comprehensive enough; further study is needed in the future.

REFERENCES

[1] Dorigo M, Maniezzo V, Colorni A.The Ant System: Optimization by a Colony of Cooperating A gents.IEEE Trans.on SMC, vol. 26, no. 1, pp. 28-41, 1996.
[2] Zhi Jun Hu, Rong Li, Ant Colony Optimization Algorithm for the 0-1 Knapsack Problem Based on Genetic Operators, Advanced Materials Research, pp. 230-232, 2011
[3] Ch. Piao, X.Han, Y. Wu. Improved ant colony algorithm for solving assignment problem, Proceedings of International Conference on Computer Application and System Modeling, 2010
[4] L. Xing, Y. Chen, A Knowledge-Based Ant Colony Optimization for Flexible Job Shop Scheduling Problems, Applied Soft Computing, vol. 10, no. 3, pp. 888-896, 2010.
[5] L. M. Gambardella1, R. Montemanni, An Enhanced Ant Colony System for the Sequential Ordering Problem, Proceedings of the 41st Annual Conference Italian Operational Research Society, 2010
[6] Hsioa Y. T, Computer network load-balancing and routing by ant colony optimization. Proceedings of the 12th IEEE

International Conference on Networks, vol. 1, pp. 313-318, 2004

[7] Q. H. Gu, S. G. Jing, Study on Vehicle Routing and Scheduling Problems in Underground Mine Based on Adaptively ACA, Applied Mechanics and Materials, vol. 157, pp. 1293-1296, 2012

[8] Gomez J. F, Khodr H. M, De Oliveira P M, et al. Ant colony system algorithm for the planning of primary distribution circuits. IEEE Trnas on Power Systems, vol. 19, no. 2, pp. 996-1004, 2004.

[9] Yu Zhen Yu et al, Regulation of PID Controller Parameters Based on Ant Colony Optimization Algorithm in Bending Control System, Applied Mechanics and Materials, pp. 128-129, 205, 2011.

[10] Ajit Narayanan, Mark Moore, Quantum-inspired genetic algorithms, Proceeding of IEEE International Conference on Evolutionary Computation 1996, 61-66

[11] FENG An-hui, SU Hong-sheng. Improved Quantum Genetic Algorithm and Its Application. Computer Engineering, vol. 37, no. 5, pp. 199-201, 2011.

[12] Panchi Li. Quantum Ant Colony Optimization with Application. Proceeding of Sixth International Conference on Natural Computation (ICNC), vol. 37, no. 6, pp. 2989 – 2993, 2010.

[13] Leung YW, Wang YP. An orthogonal genetic algorithm with quantization for global numerical optimization. IEEE Trans. on Evolutionary Computation, vol. 5, no. 1, pp. 41-53, 2001.

[14] Wang YP, Dang CY. An evolutionary algorithm for global optimization based on level-set evolution and Latin squares. IEEE Trans. on Evolutionary Computation, vol. 11, no. 5, pp. 579-595, 2007.

[15] Wang Z. ying, Shi B. xin. Solving QoS Multicast Routing Problem Based on Heuristic Genetic Algorithm, Chinese J Computers, vol. 24, no. 1, pp. 55-61, 2001.

[16] YANG Yun, XU Jia, GAO Fei, et al. Multiple QoS Constrained Multicast Routing Algorithm based on ACS. MINI- MICRO SYST EMS, vol. 27, no. 11, pp. 2030-2035, 2006

[17] Jian Zhang, Jiliu Zhou, Kun He, Huanzhou Li, "Image edge detection using quantum ant colony optimization", International Journal of Digital Content Technology and its Applications, vol. 6, no. 11, pp. 187-195, June 2012.

[18] Jiyun Bai, Shiyong Li, "Gear fault diagnosis based on relevance vector machine with quantum-inspired ant colony optimization", Journal of Information and Computational Science, vol. 7, no. 14, pp. 3169-3175, December 2010.

[19] Xiao-ming You, Sheng Liu, Yu-Ming Wang, "Quantum dynamic mechanism-based parallel ant colony optimization algorithm", International Journal of Computational Intelligence Systems, vol. 3, no. s1, pp. 101-113, December 2010.

[20] Xia Chen, Chen Tang, "Improved ant colony optimization algorithms for ground state energy of quantum mechanical systems", Chinese Journal of Computational Physics, vol. 27, no. 4, pp. 624-632, July 2010.

[21] Ling Wang, Qun Niu, Minrei Fei, "A novel quantum ant colony optimization algorithm and its application to fault diagnosis", Transactions of the Institute of Measurement and Control, vol. 30, no. 3-4, pp. 313-329, August 2008.