2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Quantum Bacterial Foraging Optimization Algorithm

Fei Li Yuting Zhang Jiulong Wu
Key Lab of Broadband Wireless Communication and Sensor Network Technology of Ministry of Education
Nanjing University of Posts and Telecommunications
Nanjing, China

Haibo Li
Department of Applied Physics and Electronics
Umea University
Umea, Sweden

***Abstract*—This paper proposes a novel swarm intelligence optimization method which integrates bacterial foraging optimization (BFO) with quantum computing, called quantum bacterial foraging optimization (QBFO) algorithm. In QBFO, a multi-qubit which can represent a linear superposition of states in search space probabilistically is used to represent a bacterium, so that the quantum bacteria representation has a better characteristic of population diversity. A quantum rotation gate is designed to simulate the chemotactic step to drive the bacteria toward better solutions. Several tests are conducted based on benchmark functions including multi-peak function to evaluate optimization performance of the proposed algorithm. The numeric results show that the proposed QBFO has more powerful properties in convergence rate, stability and the ability of searching for the global optimal solution than the original BFO and quantum genetic algorithm. In addition, we applied our proposed QBFO to solve the traveling salesman problem, which is a well-known NP-hard problem in combinatorial optimization. The results indicate that the proposed QBFO shows better convergence behavior without premature convergence, and has more powerful properties in convergence rate, stability and the ability of searching for the global optimal solution, as compared to ant colony optimization algorithm and quantum genetic algorithm.**

***Keywords—quantum computing; bacterial foraging optimization; quantum bacterial foraging optimization; traveling salesman problem*[1]**

I. INTRODUCTION

In recent years, the swarm intelligence optimization methods inspired by biological evolution and animal swarm behaviors, such as ant colony optimization (ACO) [1] and particle swarm optimization (PSO) [2], have found their way into the realm of optimization algorithms and proved their effectiveness. The swarm intelligence optimization methods have found a strongly increasing number of applications in diverse fields, including in signal processing [3].

Bacteria Foraging Optimization (BFO), proposed by Passino [4], is a new comer to the family of nature swarm inspired optimization algorithms. BFO is inspired by the social foraging behavior of Escherichia coli bacteria. Similar to ACO and PSO, BFO are designed for function optimization by moving a swarm of individuals called bacteria in the search space. One major step in BFO is the chemotaxis which mimics bacteria searching for nutrients. After every fixed number of chemotaxis steps, the swarm of bacteria performs a reproduction and elimination step.

Since its inception, BFO which mimics how bacteria forage over a landscape of nutrients to perform parallel nongradient optimization has drawn the attention of researchers from diverse fields of knowledge [5-8] due to its effectiveness in the optimization domain. It has already been applied to many real world problems and proved its effectiveness over many variants of GA and PSO [9]. However, according to mathematical analysis in [10], the chemotaxis employed by the classical BFO usually results in sustained oscillation, especially on flat fitness landscapes, when a bacterium cell is close to the optima. In dealing with complex problems, BFO has a low convergence behavior and performance decreases rapidly with an increase in the search space. To accelerate the convergence speed of the group of bacteria near the global optima and avoid its premature convergence, a novel quantum bacterial foraging optimization (QBFO) algorithm is proposed by merging BFO and quantum computing.

The subject of quantum computing brings together ideas from classical information theory, computer science, and quantum physics [11]. Research on combining evolutionary computing and quantum computing has been started since late 1990s. It can be classified into two areas. One concentrates on generating new quantum algorithms using automatic programming techniques such as genetic programming [12]. The other concentrates on quantum-inspired evolutionary computing for a classical computer [13]. Encouraged by that quantum-inspired evolutionary algorithms show better performance on solving combinatorial optimization problems than their classical counterparts [14-16], this paper proposes a novel bacterial foraging optimization algorithm, called a quantum bacterial foraging optimization (QBFO) algorithm, which is based on the concept and principles of quantum computing such as a qubit, multiqubit, superposition of states and quantum gates.

In QBFO, a multiqubit is used to represent a bacterium, and quantum rotation gate is used to mimic chemotaxis. A multiqubit system (for example n-qubit system) has available 2^n mutually ***orthogonal*** quantum states, so the quantum

This work was supported in part by the Six Top Talents Foundation of Jiangsu Province, China (No. 2011WLW011) and Guest Professor Foundation of Umeå University, Sweden (No. 541010300).

bacterium with multiqubit has the advantage that it can represent a linear superposition of states (binary solutions) in search space probabilistically. A quantum rotation gate is defined as a chemotactic operator of QBFO to drive the individual bacterium toward better solutions and eventually toward a single state.

The proposed quantum BFO (QBFO) schemes have been applied to solve four traveling salesman problems. The experiment results have been compared with their classical counterpart, a very popular swarm-intelligence algorithm known as ant colony optimization (ACO) [1] and quantum genetic algorithm (QGA) [14], with respect to the following performance measures: solution quality and convergence speed.

The work presented here has focused on the formulation of the QBFO algorithm, which takes advantage of BFO and quantum-inspired evolutionary computing such as QGA. The work in [17] uses a different swarming pattern, and the work in [18-19] takes a different quantum representation of a bacterium. While our proposed QBFO takes a new representation of a bacterium, and a new quantum chemotaxis operator and a new quantum elimination-dispersal, which was not considered in these earlier studies.

The rest of the paper is organized as follows. Section II proposes a quantum BFO. Section III provides detailed comparison between the classical BFO and its quantum variants over a test suite of 4 well-known numerical benchmarks. Section IV presents an application example with QBFO, ACO and QGA for traveling salesman problems, and summarizes the experimental results. Finally, conclusions are drawn in section V.

II. QUANTUM BACTERIAL FORAGING ALGORITHM

A. Representation

Inspired by the concept of quantum computing and quantum-inspired evolutionary algorithm [15], QBFO is designed with a novel quantum representation, called a Quantum bacterium (***Q-bacterium***), which is defined below

$$q_i^t = \begin{bmatrix} \alpha_{i1}^t & \alpha_{i2}^t & \cdots & \alpha_{iM}^t \\ \beta_{i1}^t & \beta_{i2}^t & \cdots & \beta_{iM}^t \end{bmatrix}, \quad i = 1, 2, \cdots, S \tag{1}$$

where M is the number of multiqubit, which is defined with a pair of numbers (α, β) as $[\alpha \ \beta]^{\mathrm{T}}$. (α, β) corresponds a qubit expressed in $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle = (\alpha, \beta)^T$. A Q-bit is defined as the smallest unit of information in QBFO.

QBFO with Q-bit representation has a better characteristic of population diversity than other representations, since it can represent linear superposition of states probabilistically. Only one Q-bit individual such as (3) is enough to represent 2^M states, but in binary representation at least 2^M strings.

Research results shows that *E.coli* bacteria have an interesting group behavior. A group of E.coli cells arrange themselves in a traveling ring by moving up the nutrient gradient. The cells keep certain distance and exchange food information through various ways. It increases their understanding of the environment and so increases their survival chances. The bacteria swarm in QBFO is composed of a group of Q-bacteria. The tth population is

$$Q(t) = (q_1^t, q_2^t, ..., q_S^t) \tag{2}$$

where S is the size of population.

B. Quantum Chemotaxis

Chemotaxis simulates the movement of an E.coli cell through straight swimming and tumbling via flagella. If the bacterium senses that it is moving in the correct direction (toward attractant/away from repellent), it will keep swimming in a straight line for a longer time before tumbling. If it is moving in the wrong direction, it will tumble sooner and try a new direction at random. In other words, E. coli bacteria use temporal sensing to decide whether their situation is improving or not. In this way, it finds the location with the highest concentration of nutrition (usually the source) quite well. Even under very high concentrations, it can still distinguish very small differences in concentration. In the presence of a chemical gradient bacterium will chemotaxis, or direct their overall motion based on the gradient.

In QBFO, chemotaxis operation cannot be performed as same as classical BFO because Q-bacteria can be in quantum superposition state. A Q-gate is defined as a chemotaxis operator of QBFO, by which operation the updated Q-bit should satisfy the normalization condition, $|\alpha'|^2 + |\beta'|^2 = 1$, where α' and β' are the values of the updated Q-bit. The following rotation gate is used as a Q-gate in QBFO, such as

$$\mathrm{U}(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \tag{3}$$

where θ is a rotation angle of each Q-bit toward either 0 or 1 state depending on its sign. θ should be designed in compliance with the application problem. The adjustment operation is as follows.

$$\begin{pmatrix} \alpha_m' \\ \beta_m' \end{pmatrix} = U(\theta_m) \cdot \begin{pmatrix} \alpha_m \\ \beta_m \end{pmatrix} = \begin{pmatrix} \cos\theta_m & -\sin\theta_m \\ \sin\theta_m & \cos\theta_m \end{pmatrix} \cdot \begin{pmatrix} \alpha_m \\ \beta_m \end{pmatrix} \tag{4}$$

where θ_m is rotating angle and $\theta_m = s(\alpha_m, \beta_m) \cdot \Delta\theta_m \cdot \ s(\alpha_m, \beta_m)$ is used to control the rotation direction and $\Delta\theta_m$ used to control the size of the rotation angle which should be designed in compliance with the application problem.

Quantum chemotaxis operator acts on the linear superposition of states of all Q-bits in Q-bacteria and change the phase information of Q-bit, as well as the amplitude information. As the result, the position of the Q-bacterium is updated.

C. Quantum Reproduction

Quantum reproduction is an evolutionary process based on survival of the fittest. Let

$$Jhealth(i) = \sum_{j=1}^{N_c} fitness(i, j) \tag{5}$$

be the health of the ith bacterium (a measure of how many nutrients it got over its lifetime and how successful it was at avoiding noxious substances), where N_c is the number of

chemotactic steps and $fitness(i, j)$ is the fitness function value of ith bacterium at jth chemotactic step.

The less healthy bacteria eventually die while each of the healthier bacteria asexually split into two bacteria, which are then placed in the same location. This keeps the swarm size constant.

D. Quantum Elimination-dispersal

Gradual or sudden changes in the survival environment, such as a significant local rise of temperature, may kill or disperse a group of bacteria that are currently in a region with a high concentration of nutrient gradients.

To simulate this phenomenon in QBFO, quantum elimination-dispersal is performed after several steps of quantum chemotaxis and quantum reproduction. In this process, some bacteria are dispersed at random with a very small probability P_{ed} while the new replacements are randomly initialized over the search space as

$$\left|\varphi_{q_i^t}\right\rangle = \sum_{n=1}^{2^M} \frac{1}{\sqrt{2^M}} \left| X_n \right\rangle \tag{6}$$

where X_n is the nth state represented by the binary string $(x_1, x_2, ..., x_M)$, where x_m, $m = 1, ..., M$ is either 0 or 1 according to the probability of either $\left|\alpha_m^t\right|^2$ or $\left|\beta_m^t\right|^2$, respectively.

E. The Procedure of QBFO

The detailed pseudo-code of the complete algorithm is described as TABLE I.

TABLE I
THE PSEUDO-CODE OF QBFO

Step	Do
(1)	Initialize parameters p, S, N_c, N_s , N_{re}, N_{ed} , P_{ed} . where p: dimension of the search space S: total number of bacteria in the population N_c: number of quantum chemotactic steps N_{re}: the number of quantum reproduction steps N_{ed} : the number of quantum elimination–dispersal events P_{ed} : quantum elimination-dispersal probability
(2)	Let $t,j,k,l = 0$
(3)	Initialization quantum bacteria population Q (t), α_m^0 and β_m^0 of all q_i^0 are initialized with $1/\sqrt{2}$. It means that one Q-bacterium q_i^0 represents the linear superposition of all possible states with the same probability. The state of q_i^0 is as (6).
(4)	Make P(0) by observing the states of Q(0). Quantum state Q(0) collapses to P(0), which is the set of binary solutions
(5)	Evaluate P(0). Each binary solution is evaluated to give a level of its fitness.
(6)	Store the best solutions among P(0) into B(0), the initial best solutions are then selected among the binary solutions.

(7)	**while** (**not** termination-condition) **do**
	begin
	$t \leftarrow t+1$
	while (l≤Ned) **do Quantum Elimination–dispersal**
	begin
	$l \leftarrow l+1$
	while (k≤Nr) **do Quantum Reproduction**
	begin
	$k \leftarrow k+1$
	while (j≤Nc) **do Quantum Chemotaxis**
	begin
	$j \leftarrow j+1$

(8)	Make P(t) by observing the states of Q(t-1). Quantum state Q(t-1) collapses to P(t), which is the set of binary solutions
(9)	Evaluate P(t). Compute fitness function, obtain the best fitness of the bacterium as the target of next evolution values
(10)	Update Q(t) using Q-gates. Q-bacteria in Q(t) are updated by applying Quantum rotation gates.
(11)	Store the best solutions among B(t-1) and P(t) into B(t)
(12)	Store the best solution **b** among B(t) **end**
(13)	Compute $Jhealth(i)$. Sort Q-bacteria in order of ascending cost.
(14)	Reproduction. The half of the bacteria with the better values split (this process is performed by placing the copies that are made at the same location as their parent and the other half is eliminated. **end**
(15)	Elimination–dispersal. Generate a random number *rand*, a bacterium is eliminated if *rand*<P_{ed} . Disperse another one to a random state as eq. (6). **End**
(16)	***if*** (migration-condition) ***then*** migrate **b** or b_j^t to B(t) globally or locally, respectively **end**

III. EXPERIMENTS AND RESULTS OVER BENCHMARK FUNCTIONS

This section presents some comparisons among the performances of the proposed QBFO, the original BFO, and QGA which is a typical algorithm of Quantum Evolutionary Computation. All methods have been applied to several benchmark test functions as depicted in TABLE II in order to check the effect of the proposed QBFO in the efficiency and the convergence speed.

A. Test Functions

Our test suite includes 4 well-known benchmark functions of varying complexity. The formulas of these functions are presented in TABLE II.

The Sphere function (f_1) is continuous, convex and unimodal with only one global minimum. The others are multimodal with a considerable number of local extremes in the region of interest. The Needle-in-haystack function (f_2) has one global maximum with four local maxima, and the function behaves like a needle in the haystack (the function values for points in the space outside the narrow peaks give very little information on the location of the global optimum). The Schaffer's F6 function (f_3) has one global maximum with numerous local maxima, the difficulty in this function is that the size of the potential maxima that need to be overcome to get to a minimum increases the closer one gets to the global minimum. The Multi-peak function (f_4) has one global maximum with huge number of local maxima, the difficulty in this function is asymmetric and having the global maximum at the edge of the search space. TABLE II summarizes the optima and search ranges used for all the functions. The contours of all the test functions are illustrated in TABLE II.

TABLE II
DESCRIPTION OF THE BENCHMARK FUNCTION USED

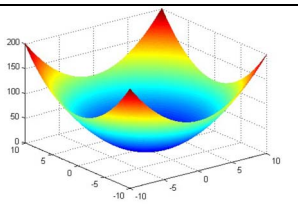
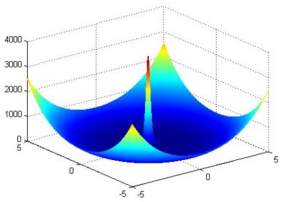
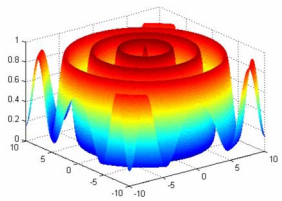
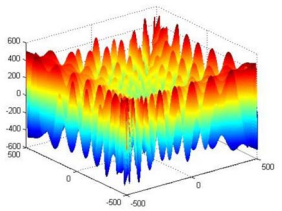
Function name	Formula	Optima	Search domain	Contour
Sphere function (f_1)	$f_1(x,y)=x^2+y^2$	$f_1(0,0)=0$	$-100\le x,y\le 100$	
Needle-in-haystack function (f_2)	$f_2(x,y)=(\frac{3}{0.05+x^2+y^2})^2+(x^2+y^2)^2$	$f_2(0,0)=3600$	$-5.12\le x,y\le 5.12$	
Shaffer's F6 function (f_3)	$f_3(x,y)=0.5+\frac{sin^2\sqrt{x^2+y^2}-0.5}{(1+0.001\cdot(x^2+y^2))^2}$	$f_3(0,0)=1$	$-100\le x,y\le 100$	
Multi-peak function (f_4)	$f_4(x,y)=-xsin\sqrt{\lvert y+1-x\rvert}cos\sqrt{\lvert y+1+x\rvert}$ $-(y+1)cos\sqrt{\lvert y+1-x\rvert}sin\sqrt{\lvert y+1+x\rvert}$	$f_4(-512,-512)$ $=511.7319$	$-512\le x,y\le 512$	

TABLE III
EXPERIMENTAL RESULTS FOR 20 INDEPENDENT RUNS ON FOUR BENCHMARK FUNCTIONS

Function name	Algorithm	Best value	Worst value	Mean best value	Standard deviation	Average iterations	Average run time (s)	Convergence runs
Sphere function (f_1)	BFO	2.6980e-07	8.6672e-04	6.8164e-05	1.8469e-04	156.67	0.932624	**20**
	QGA	1.3296e-04	6.1000e-02	5.4486e-03	1.2980e-02	421.52	5.669402	9
	QBFO	**1.1369e-09**	1.6220e-04	**1.5472e-05**	**4.1785e-05**	284.81	3.743355	**20**
Needle-in-haystack function (f_2)	BFO	**3600**	2748.8	3004.2	400.2022	481.45	2.551758	6
	QGA	**3600**	3594	3598.7	1.6706	361.8	4.893244	12
	QBFO	**3600**	3599	**3599.9**	**0.2471**	295.3	3.808226	**20**
Shaffer's F6 function (f_3)	BFO	0.9903	0.7268	0.9457	0.0656	500	2.873371	0
	QGA	0.9982	0.9900	0.9918	0.0031	448.40	6.312885	4
	QBFO	**1.0000**	0.9903	**0.9989**	**0.0030**	230.40	3.001256	**15**
Multi-peak function (f_4)	BFO	511.7078	497.2463	503.7723	4.5548	479.85	2.370211	3
	QGA	511.5752	501.3417	508.4660	2.9080	436.10	5.800089	4
	QBFO	**511.7319**	501.8813	**510.7167**	**2.2493**	182.00	2.490855	**15**

B. *Parameter Settings*

For the BFO, we chose the population size of the bacteria S=40, the number of chemotaxis N_c=50, the number of reproduction steps N_{re}=5, the number of elimination and dispersal events N_{ed}=2, the probability of elimination and dispersal P_{ed}=0.25, the depth of the attractant released by the cell $d_{attract}$=0.1, the width of the attractant signal $w_{attract}$=0.2, the height of the repellent effect $h_{repellant}$=$d_{attract}$=0.1, and the width of the repellent $w_{repellant}$=2. The size of the step taken in the random direction specified by the tumble C was set as 0.1 for benchmarks f_1 and f_3. For benchmarks f_2 and f_4 we chose C=0.001. For the QBFO, the parameter values of S, N_c, N_{re}, N_{ed}, and P_{ed} were kept exactly same as BFO. We fixed the length of the Quantum bacterium M=44, and the size of the rotation angle $\Delta\theta_m=0.08\pi$. For the QGA, we chose population size S=40, the length of the quantum chromosome len=44, the probability of cross p_c=0.7, the probability of variation p_m=0.15, the size of rotation angle $\Delta\theta_m=0.08\pi$, and the maximal generation number $maxgen$=500.

C. *Results and Discussions*

Twenty independent runs of the three competitor algorithms were carried out on each problem, and the experimental results are presented in TABLE III. In the table, the best results among the algorithms are shown in bold. The graphs presented in Fig.1-4 illustrate the evolution of best fitness found by three algorithms averaged for 20 runs for each function.

TABLE III illustrates the comparisons of the three algorithms on the benchmark functions. From TABLE III, it is observed that for all test problems, the proposed QBFO is superior to other two algorithms on the optimization problems although it converges slower sometimes (i.e. as shown in Fig. 1). The best value and the mean best value of the proposed method are closest to or even the same as the optimal value. QBFO is the most stable as the standard deviation of QBFO is smallest.

For convenience to show better search ability, Fig.1-4 illustrate the comparisons on functions f_1-f_4. In general, the graphs in Fig.1-4 show that the QBFO could converge to the global optimum keeping a good diversity and high speed when it conducts the optimization of Sphere, Needle-in-haystack, Shaffer's F6 and Multi-peak problems.

As evident from TABLE III and Fig.1, it obviously shows that the Sphere function is easy to solve. It is shown that the QBFO converges slower than BFO and QGA, but the average run time of QBFO is less than QGA and the convergence runs of QBFO is more than QGA. QBFO hits the success 20 times by 20 runs.

For Needle-in-haystack function, it is evident form TABLE III and Fig.2 that QBFO is the fastest algorithm in reaching the target global value. The frequency of hitting the optima of QBFO is 20 times in 20 runs, and that of BFO is only 6 times.

For Shaffer's function, it can be easily observed form TABLE III and Fig.3 that QBFO arrives at the global optimum value fastest. The frequency of hitting the optima of QBFO is 15 times in 20 runs, and BFO cannot reach to the global optimum value.

According to TABLE III and Fig.4, QBFO remained the best performance in the convergence rate, the best value, the mean best value and the frequency of hitting the optima.

Obviously, QBFO takes advantage of quantum computation, which provides the bacteria with more intelligence to search the global optimum, and contribute to the global optimization ability.

According to the comparison analysis above, it is obvious to know that the proposed QBFO can keep a better diversity to develop the virgin space and have the best ability to reach the optimum. The relative results showed that QBFO is a good method to improve the global ability of BFO. QBFO shows good convergence performance not only for simple smooth function such as Sphere function but also for complex function such as multi-peak function and nonlinear optimization problem.

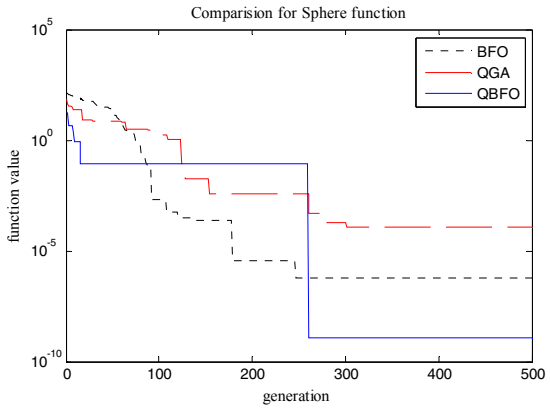


Fig. 1. Convergence Curve of three algorithms for f_1

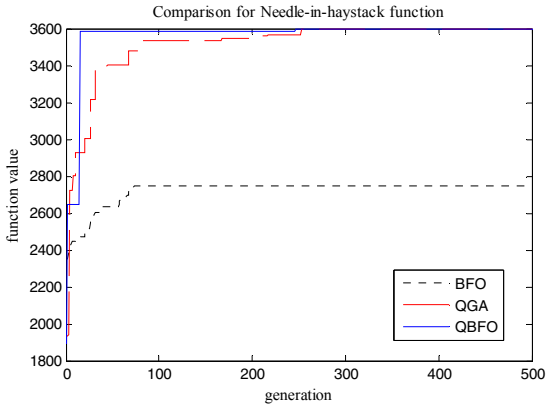


Fig. 2. Convergence Curve of three algorithms for f_2

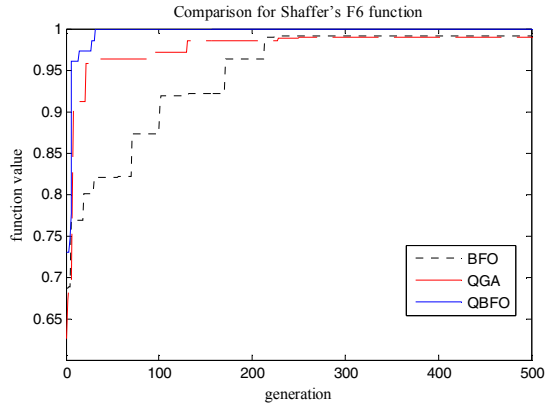


Fig. 3. Convergence Curve of three algorithms for f_3

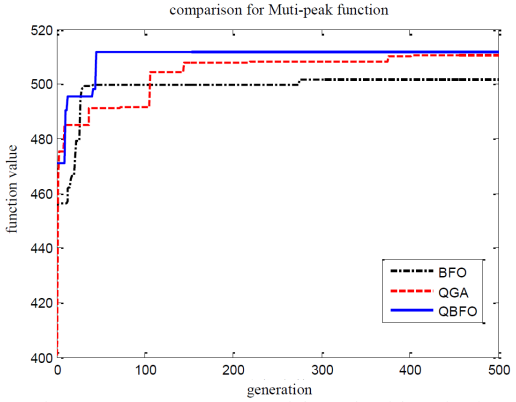


Fig. 4. Convergence Curve of three algorithms for f_4

IV. Application to the Traveling Salesman Problem

A. QBFO for the Traveling Salesman Problem

The traveling salesman problem (TSP), which is an NP-hard problem in combinatorial optimization, is considered to demonstrate the applicability of QBFO to the combinatorial optimization problem. TSP can be modeled as an undirected weighted graph, such that cities are the graph's vertices, paths are the graph's edges, and a path's distance is the edge's length. It is a minimization problem starting and finishing at a specified vertex after having visited each other vertex exactly once. In this paper we will restrict attention to TSPs in which cities are on a plane and a path (edge) exists between each pair of cities (i.e., the TSP graph is completely connected).

Give a weighted graph G=(C,D), where C=(C_1,C_2,…,C_M) is the set of cities and $D=\{(C_i,C_j):C_i,C_j\in C\}$ is the set of edges. Let $d(C_i,C_j)$ be a cost measure associated with edge $(C_i,C_j)\in D$, which is the Euclidean distance between C_i and C_j. The object of TSP is to find a shortest closed tour which visits all the cities in C.

In this work a quantum bacterium is an agent which moves from city to city on a TSP graph. We encoded each quantum bacteria population $Q(t)=(q_1^t,q_2^t,....,q_S^t)$ as in eq. (1), where S is the bacterial population size, the length of the quantum bacterium M is the same as the number of cities. At the beginning, we generate randomly a population composed of S "quantum bacteria", each bacterium is randomly initialized over the search space as in eq. (2). This allows a bacterium to encode not only one solution but all the possible solutions by using the superposition principle.

When we make P(t) by observing the states of Q(t) in QBFO, we can get a set of binary solutions $P(t)=(\mathbf{x}_1^t,\mathbf{x}_2^t,....,\mathbf{x}_S^t)$, where $\mathbf{x}_i^t=(x_{i1}^t,x_{i2}^t,....,x_{iM}^t)$ is a binary string. Each binary solution $\mathbf{x}_i^t$ is evaluated to give a level of its fitness as

$$f(x)=\sum_{i=0}^{n}\sum_{j\neq i,j=0}^{n}d_{ij}c_{ij} \tag{7}$$

subject to $\sum_{i=0,i\neq j}^{n}x_{ij}=1,\ j=0,..,M,\ \sum_{j=0,i\neq j}^{n}x_{ij}=1,\ i=0,..,M$.

Where c_{ij} is equal 1 if the path goes from city i to city j, and 0 otherwise, for cities 0, ..., M., and d_{ij} is the distance from city i to city j.

B. Experimental Results and Performance Evaluation

We applied the proposed QBFO to the TSPs listed in TABLE IV [20]. Using the 4 instances listed in TABLE IV, the performance of QBFO was compared with the performance of other two naturally inspired optimization methods: Ant Colony Optimization algorithms (ACO) and Quantum Genetic Algorithms (QGA). Numerical experiments were executed with QBFO, ACO and QGA. The QBFO parameters were set to the following values: S=40, N_c=50, N_{re}=5, N_{ed}=2, P_{ed}=0.25, $\Delta\theta_m=0.08\pi$, and iteration times as 500. The results of comparison are shown in TABLE IV and the obtained optimal routs of 4 instances by QBFO are shown in Fig.5-8.

In TABLE IV, we report the best integer tour length, the best real tour length, and the number of average iterations required to find the best integer tour length. The best result for each problem is in bold. It is therefore clear that the proposed QBFO algorithm gives better results, compared to the ACO and QGA.

TABLE IV
The Results of The Optimal Routes
Comparison of QBFO with ACO and QGA

Problem name	Algorithm	Best search value	Best value known	Average iterations
City Set 1 (10-city problem)	ACO	**2.6907**	2.6907	75.5
	QGA	**2.6907**	2.6907	252.31
	QBFO	**2.6907**	2.6907	187.9
City Set 2 (30-city problem)	ACO	431.3477	423.741	146.25
	QGA	443.5672	423.741	338
	QBFO	**425.6490**	423.741	267.47
City Set 3 (50-city problem)	ACO	462.2606	427.855	198.33
	QGA	465.0977	427.855	450.54
	QBFO	**456.4608**	427.855	342.85
City Set 4 (75-city problem)	ACO	559.1439	549.18	231.62
	QGA	563.7425	549.18	438.78
	QBFO	**553.9830**	549.18	408.52

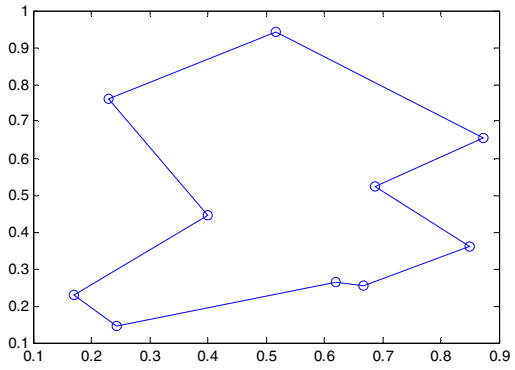


Fig. 5. Best Solution to a TSP with 10 cities

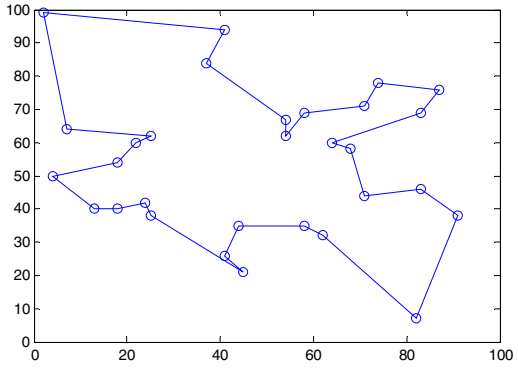


Fig. 6. Best Solution to a TSP with 30 cities

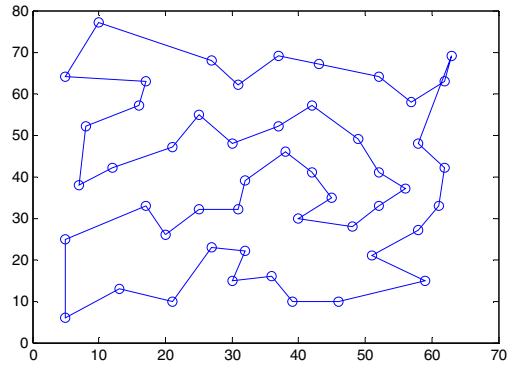


Fig. 7. Best Solution to a TSP with 50 cities

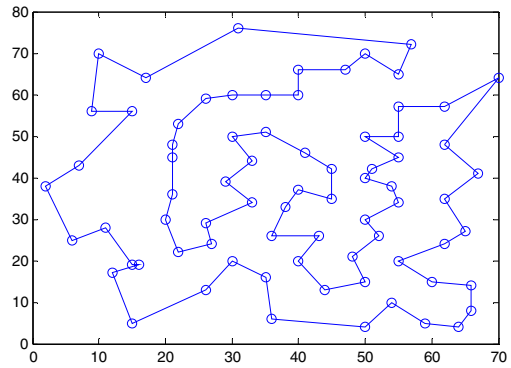


Fig. 8. Best Solution to a TSP with 75 cities

Refer to Fig.9-12, we can see the comparative results of the convergence processes of applying QBFO, ACO and QGA to solve the 4 TSP problems. It is shown that the proposed QBFO can converge toward the optimal solution more quickly than ACO and QGA, and the final convergence values in QBFO also show better findings of the function global minimum in comparison to ACO and QGA.

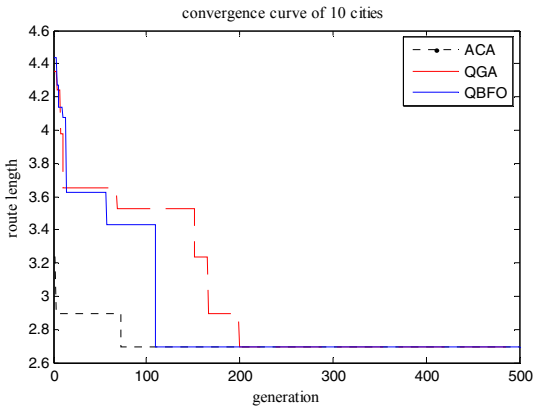


Fig. 9. Convergence curves to a TSP with 10 cities

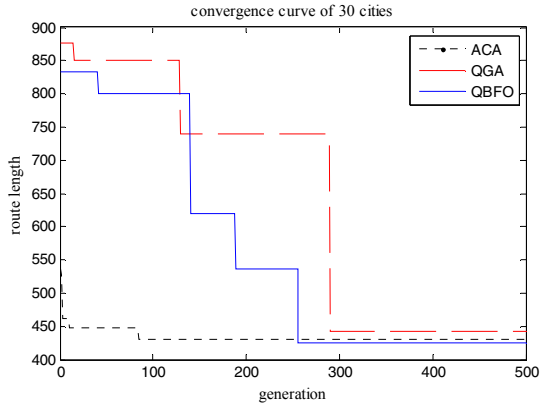


Fig. 10. Convergence curves to a TSP with 30 cities

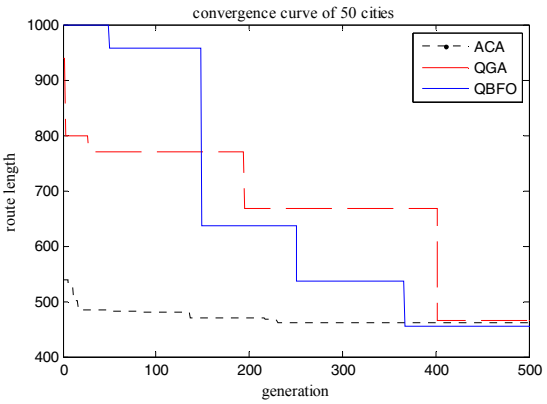


Fig. 11. Convergence curves to a TSP with 50 cities

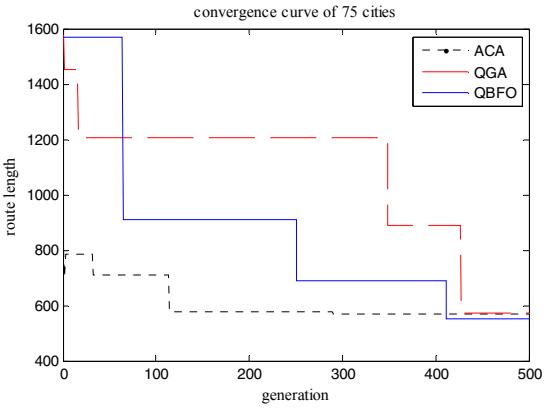


Fig. 12. Convergence curves to a TSP with 75 cities

V. CONCLUSION

In this paper, a novel QBFO is proposed, which is based on the BFO and quantum computing. A novel quantum bit expression mechanism called quantum bacteria is employed and the quantum chemotaxis is adopted to update the Q-bacteria. Quantum reproduction is performed after several steps of quantum chemotaxis, which makes most bacteria get together and accelerates convergence of the algorithm. Then quantum dispersal operation is performed on the bacteria swarm with a certain probability, which can expand the searching space and prevent the algorithm to fall into the local optimal value. The key to the application of QBFO to a new problem is to identify an appropriate representation for the

problem (to be represented as a graph searched by many quantum bacteria). The simulated results in solving traveling salesman problem show that QBFO is superior to ACO and QGA.

Future research may focus on extending the analysis presented in this paper to a group of quantum bacteria working on a multidimensional fitness landscape and also include effect of the quantum chemotaxis and elimination–dispersal events in the same.

ACKNOWLEDGMENT

The authors thank Dr. Shafiq Ur Rehman, head of Immersive Interaction Lab of Department of Applied Physics and Electronics (TFE) at Umeå University, Sweden, for the suggestions to improve the presentation and quality of this paper.

REFERENCES

[1] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed Optimization by Ant Colonies", *actes de la première conférence européenne sur la vie artificielle*, Elsevier Publishing, Paris, France, pp. 134-142, 1991.

[2] J.Kennedy, R. Eberhart, "Particle Swarm Optimization", *Proc. IEEE International Conference on Neural Networks*, IEEE Press, pp. 1942–1948, 1995.

[3] D. Merkle and M. Middendorf, "Swarm Intelligence and Signal Processing", *IEEE Signal Processing Magazine*, vol.25(6) pp. 152-158, 2008.

[4] K.M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control", *IEEE Control Systems Mag.*, vol.22(3), pp. 52–67, 2002.

[5] S. Mishra, "a Hybrid Least Square-Fuzzy Bacterial Foraging Strategy for Harmonic Estimation", *IEEE Trans on Evolutionary Computation*, vol.9, pp. 61-73, 2005.

[6] S. Mishra and C.N. Bhende, "Bacterial Foraging Technique-Based Optimized Active Power Filter for Load Compensation", *IEEE Trans Power Delivery*, vol.22(1), pp. 457–465, 2007.

[7] S. Dasgupta, S. Das, A. Biswas and A. Abraham, "Automatic circle detection on digital images with an adaptive bacterial foraging algorithm", *Soft Computing*, vol.14(11), pp. 1151-1164, 2010.

[8] H. Nouria and T. S. Hong, "Development of bacteria foraging optimization algorithm for cell formation in cellular manufacturing system considering cell load variations", *Journal of Manufacturing Systems*, vol.32, pp 20-31, 2013.

[9] S. Das, A. Biswas, S. Dasgupta and A. Abraham, "Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications, foundations of computational intelligence", *Global optimization, studies in computational intelligence*, vol. 3. pp. 23–55, 2009.

[10] S. Dasgupta, S. Das, A. Abraham and A. Biswas, "Adaptive Computational Chemotaxis in Bacterial Foraging Optimization: An Analysis", *IEEE Tran. on Evolutionary Computation*, vol.13(4), pp. 919-941, 2009.

[11] A. M. Steane, "Quantum Computing", *Reports on Progress in Physics*, vol.61, pp. 117-173, 1998.

[12] L. Spector, H. Barnum, H. J. Bernstein, and N. Swamy, "Finding a better-than-classical quantum AND/OR algorithm using genetic programming," *Congress on Evolutionary Computation*, Piscataway, vol. 3, pp. 2239–2246, 1999.

[13] A. Narayanan and M. Moore, "Quantum-Inspired Genetic Algorithms", *Proc. IEEE International Conference on Evolutionary Computation*, IEEE Press, Piscataway, pp.1-66, 1996.

[14] K-H Han and J-H Kim, "Genetic quantum algorithm and its application to combinatorial optimization problem", *Proc. of IEEE Congress on Evolutionary Computation*, IEEE Press, USA, pp.1354-1360, 2000.

[15] K.-H. Han and J.-H. Kim, "Quantum-Inspired Evolutionary Algorithm for a Class of Combinatorial Optimization", *IEEE Trans. On Evolutionary Computation*, vol.6(6), pp. 580-593, 2002.

[16] F. Li, D.P. Zhu, F. Tian and H.B. Li. "Improved Quantum Genetic Algorithm for Competitive Spectrum Sharing in Cognitive Radios", *Proc. International Conference in Electrics, Communication and Automatic Contro*, pp. 1203-1209, 2011.

[17] J.L. Cao and H.Y. Gao, "A Quantum-inspired Bacterial Swarming Optimization Algorithm for Discrete Optimization Problems", Advances in Swarm Intelligence, Lecture Notes in Computer Science, Springer, vol.7331, pp 29-36, 2012.

[18] H.Y. Gao, W. Cui and C.W. Li, "A quantum bacterial foraging optimisation algorithm and its application in spectrum sensing", *Int. J. Modelling, Identification and Control*, vol.18(3), pp. 29-36, 2013

[19] G.Y. Zhang, Y.G. Wu and Y.X., "Bacterial Foraging Optimization Algorithm with Quantum Behavior", *Journal of Electronics & Information Technology*, vol.35(3), pp. 614-621, 2013.

[20] M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem", *BioSystems*, vol.43, pp. 73-81, 1997.