Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Quantum-Inspired Evolutionary Multicast Algorithm

Yangyang Li
Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Institute of Intelligent Information Processing, Xidian University, Xi'an 710071, China
yyli@xidian.edu.cn

Jingjing Zhao
Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Institute of Intelligent Information Processing, Xidian University, Xi'an 710071, China
lasttemple@126.com

Licheng Jiao
Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Institute of Intelligent Information Processing, Xidian University, Xi'an 710071, China
lchjiao@mail.xidian.edu.cn

Qiuyi Wu
Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Institute of Intelligent Information Processing, Xidian University, Xi'an 710071, China
diwudiwu@126.com

***Abstract*—As a global optimizing algorithm, genetic algorithm (GA) is applied to solve the problem of multicast more and more. GA has more powerful searching ability than traditional algorithm, however its property of "prematurity" makes it difficult to get a good multicast tree. A quantum-inspired evolutionary algorithm (QEA) to deal with multicast routing problem is presented in this paper, which saliently solves the "prematurity" problem in Genetic based multicast algorithm. Furthermore, in QEA, the individuals in a population are represented by multistate gene quantum bits and this representation has a better characteristic of generating diversity in population than any other representations. In the individual's updating, the quantum rotation gate strategy is applied to accelerate convergence. The algorithm has the property of simple realization and flexible control. The simulation results show that QEA has a better performance than CS and conventional GA.**

***Keywords*—genetic algorithm, multicast, quantum-inspired evolutionary algorithm**

I. INTRODUCTION

With the development of network and communication technologies, multicast routing service is becoming a key requirement of network multimedia applications which require certain quality of service (QoS). Multicasting involves the transport of the same information from a sender to multiple receivers along a multicast tree. In the past, much work has been focused on algorithms for computing low-cost multicast trees so that network resources can be efficiently managed. Since the problem of computing minimum-cost multicast trees (call Steiner trees) in a network [1] is NP-complete, most of the proposed algorithms are heuristic ones [2]-[4]. Besides some conventional heuristic algorithms, evolutionary algorithms [5] [6] are promising for solving this problem. Despite the successes of genetic algorithms, they have the same defects of pre-maturity and stagnation. Artificial immune algorithms are new computational intelligence methods inspired by theoretical immunology. They have also been applied to QoS multicast routing problem and achieved good performances [7] [8]. In particular, superior to conventional genetic algorithms (CGAs) and bounded shortest multicast algorithm (BSMA) [2] which is the best heuristic constrained multicast algorithm, A Clonal Strategies (CS) based multicast algorithm [8] maintains good diversity and less likely to be trapped in local optima. Unfortunately, CS obtains a good local searching ability at a cost of adding the scale of the population by the clone operator; the characteristics of population diversity and selective pressure are not easy to be implemented in immune clonal algorithm [9].

The quantum-inspired evolutionary algorithm (QEA) recently proposed in [10] can treat the balance between exploration and exploitation more easily when compared with CGAs. In this paper, QEA can explore the search space with a smaller number of individuals and exploit the search space for a global solution within a short span of time. QEA is based on the concept and principles of quantum computing, such as the quantum bit and the superposition of states. With the scale of the network increased, QEA can find better multicast trees than GA and CS.

II. THE MULTICAST ROUTING PROBLEM DEFINITION

The communication network is modeled as a graph $G(V,E)$, where V is the set of nodes and E is the set of edges. On the graph G, we define the functions $c(x,y)$ and $d(x,y)$, where $c(x,y)$ is the cost of using edge $(x,y)\in E$ and $d(x,y)$ is the delay along edge $(x,y)\in E$. Let $P(a,b)$ denote the

This work was supported by the National High Technology Research and Development Program (863 Program) of China (No. 2009AA12Z210), the National Natural Science Foundation of China (Nos. 60703108, 60703107 and 60803098), the Provincial Natural Science Foundation of Shaanxi of China (No. 2007F32), the China Postdoctoral Science Foundation funded project (No. 20080431228), the China Postdoctoral Science Foundation Special funded project (No. 200801426) and the Program for Cheung Kong Scholars and Innovative Research Team in University (No. IRT0645).

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

path from a to b, then delay function $Delay(a,b)$ and cost function $Cost(a,b)$ can be expressed as follow:

$$Delay(a,b) = \sum_{(x,y)\in P(a,b)} d(x,y)\,, \tag{1}$$

$$Cost(a,b) = \sum_{(x,y)\in P(a,b)} c(x,y)\,. \tag{2}$$

Supposes s is the source node, $D \in V - \{s\}$ denotes a set of destination nodes. The multicast routing problem searches for a tree $T = (V_T, E_T)$ ($T \in G$), which subjects to the following constraints:

$$Cost(T) = \min\Big(\sum_{(x,y)\in P_T(x,y)} c(x,y)\Big)\,, \tag{3}$$

$$\sum_{(x,y)\in P_T(x,y)} d(x,y) \le \Delta, \forall y \in D, \tag{4}$$

where Δ is a delay tolerance and $P_T(x,y)$ denotes the unique path from x to $y \in D$ in the multicast tree.

Throughout this paper, we let $n = |V|$, $m = |D|$.

III. QEA

A. Representation

In this study, an individual represents a search point in decision space. In [9], we give a new representation, a qubit for the probabilistic representation that it can represent a linear superposition of states and has a better characteristic of population diversity than other representations [10], which are defined below.

Definition 1: The probability amplitude of one qubit is defined with a pair of numbers (α, β) as:

$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix}, \tag{5}$$

satisfying

$$|\alpha|^2 + |\beta|^2 = 1\,, \tag{6}$$

where $|\alpha|^2$ denotes the probability that the qubit will be found in the '0' state and $|\beta|^2$ denotes the probability that the qubit will be found in the '1' state.

Definition 2: A qubit individual as a string of N qubits is defined as:

$$\begin{bmatrix} \alpha_1 & \alpha_2 & \ldots & \alpha_N \\ \beta_1 & \beta_2 & \ldots & \beta_N \end{bmatrix}, \tag{7}$$

where $|\alpha_l|^2 + |\beta_l|^2 = 1, (l = 1,2,...,N)$.

B. Description of the Algorithm

QEA is a probabilistic algorithm that is similar to EA. It maintains a quantum population $Q(t) = \{q_1^t, q_2^t, ..., q_M^t\}$ at the t-th generation where M is the size of population, and N is the length of the qubit individual q_i^t which is defined as: $q_i^t = \begin{bmatrix} \alpha_1^t & \alpha_2^t & \ldots & \alpha_N^t \\ \beta_1^t & \beta_2^t & \ldots & \beta_N^t \end{bmatrix}, i = 1,2,...,M.$

The main loop of QEA is as follows.

Algorithm 1: quantum-inspired evolutionary algorithm

Q(*t*) (population of qubit individuals at the *t*-th generation)

P(*t*) (population of classical bit individuals at the *t*-th generation)

b (best classical bit individual at the current generation)

Step1: Initialize $Q(t)$ and *b*, *t*=0.

Step2: Produce $P(t)$ by observing the updated $Q(t)$.

Step3: Evaluate the fitness of $P(t)$ and store the best solutions among $P(t)$ to *b*.

Step4: Update $Q(t)$ by quantum mutation.

Step5: Judge the termination condition, if it is satisfied, output the best solution and end the process, otherwise *t*=*t*+1.

Step6: Go to *step2*.

IV. Quantum-Inspired Evolutionary Multicast Algorithm

A. Construction of candidate path set

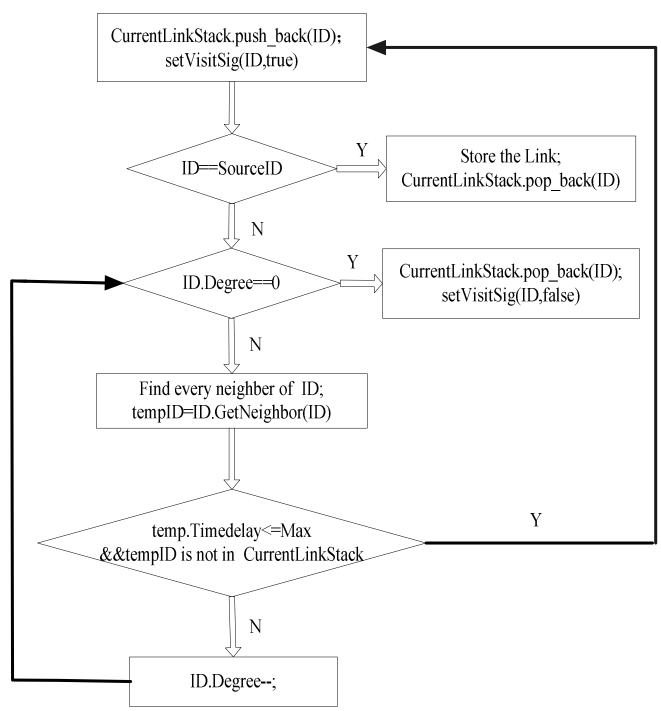

Figure 1. Flow chart of searching the candidate path set.

Assume Graphic is a complete map that has n nodes, that is, each node in the network has a degree of n-1. The number of the feasible paths from a node to a designated node in the network is $PathCount = 1 + P_{n-2}^{1} + P_{n-2}^{2} + P_{n-2}^{3} + \cdots + P_{n-2}^{n-2} > n!$. If we construct a multicast tree in such a network, the number of possible combinations is described as:

$$\underbrace{PathCount \times PathCount \times \cdots \times PathCount}_{m}, \quad (8)$$

where m is the number of the destination nodes.

For an Unconstrained network-wide connectivity, its candidate path set is very large. But after adding the delay constraint, many candidate paths will be removed from the original path set. Then the scale of the problem is reduced. The more tightly the time delay bound, the smaller the candidate path set is.

For each $d_i \in D$ $(i = 1,2,...,m)$, we first compute all the paths $P_T(s,d_i)$ satisfying a delay constraint that is equal to or less than Δ. And the candidate path set is arranged in ascending cost order. Figure 1 shows the process of searching the candidate path set.

B. Encoder based on Elite strategy

In the view of statistic, the Steiner tree consists of paths with smaller cost in candidate path set, and the fact can be found in both Prim's and Kruskal's algorithms which are based on the strategy of greedy [1]. That is, successive spanning tree will have to join the shortest side of the node tree. Then in a Steiner tree, the cost of path $P_i(s,\ d),(i = 1,2,...,m)$ is smaller, not necessary the smallest one in all the paths. So it is essential to reduce the number of candidate paths to speed up the convergence rate and to narrow the search space. Then we can assume that a Steiner tree must consist of some shortest paths to their own destination. Therefore, we encode the candidate path with elite strategy. That is, we collect the first 2^N paths for every destination and make all the $m \times 2^N$ paths to be an "elite group". Here N denotes the number of qubits used to represent a candidate path to a destination node). So the length of the qubit individual is $|D| \times N$, which defined as

$$q^t = \begin{bmatrix} \alpha_{11}^t & \alpha_{12}^t & \ldots & \alpha_{1N}^t & \ldots & \alpha_{m1}^t & \alpha_{m2}^t & \ldots & \alpha_{mN}^t \\ \beta_{11}^t & \beta_{11}^t & \ldots & \beta_{1N}^t & \ldots & \beta_{m1}^t & \beta_{m2}^t & \ldots & \beta_{mN}^t \end{bmatrix},$$ where

$|\alpha_{ij}|^2 + |\beta_{ij}|^2 = 1, (i = 1,2,...,m,\ j = 1,2,...,N).$

C. Observing operator

In order to evaluate the population with fitness value, observing operator is a process of deciding the population by observing the amplitude of each gene in the qubit individual population. That is, in the act of observing a quantum state, it collapses to a single state (namely classical representation). For example, we observe the qubit individual q^t at the t-th generation and produce binary strings solution $P(t) = \{x_{11}^t, x_{12}^t, ..., x_{1N}^t, ..., x_{m1}^t, x_{m2}^t, ..., x_{mN}^t\}$, where x_{ij}^t $(i = 1,2,...,m,\ j = 1,2,...,N)$ is numeric strings of length $m \times N$ derived from the amplitude α_{ij}^t or β_{ij}^t $(i = 1,2,...,m,\ j = 1,2,...,N)$. The process is described as follows: (a) generate a random number $r \in [0,1]$; (b) if it is smaller than $|\beta_{ij}^t|^2$, the corresponding bit in $P(t)$ takes '0', otherwise takes '1'. And the specific process is shown in Figure 2.

```
Procedure:
begin
    j:=0
    while ( j < m×N )do
        begin
          j:=j+1
          if random[0,1]<|β_ij^t|^2
          then  p_i = 0
          else  p_i = 1
          end
        end
    end
end
```

Figure 2. Observing operator.

Then we decode, for example, "110" denote the 7th path of the destination node. When the number of the paths to a destination node is 7, both 110 and 111 denote the 7th path, and then we evaluate multicast tree cost.

D. Quantum mutation

In the quantum theory, the transform of states is fulfilled by the quantum transformation matrix. For example, the updating operator can be denoted by the quantum rotation gates. Also, the best antibody in the current generation is important, i.e., the communication among antibodies is needed to speed up the convergence. Quantum mutation focuses on a wise guidance by the current best antibody in subpopulation. A quantum rotation gate U is

$$U(\Delta\theta) = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix}, \quad (9)$$

where $\Delta\theta$ is the rotate angle which controls the convergence of QEA and $\Delta\theta$ is defined as

$$\Delta\theta_i = \delta \times s(\alpha_i, \beta_i), \quad (10)$$

where δ is a coefficient determining the speed of convergence. If the value of δ is too small, the speed of convergence will slow down, while if the value of δ is too large, divergence or premature convergence may happen. So a dynamic regulation strategy of rotating angle is used, which regulates the value of δ between 0.1π and 0.005π according to the inheritance generation. The following lookup table can be used as a strategy for convergence. It should be indicated that it is a general way of mutation for different problems.

TABLE I. THE LOOK UP TABLE OF $\Delta\theta$

x_i	b_i	$f(x_i) > f(b_i)$	$\Delta\theta_i$	$s(\alpha_i,\beta_i)$			
				$\alpha_i\beta_i > 0$	$\alpha_i\beta_i < 0$	$\alpha_i = 0$	$\beta_i = 0$
1	1	false	0	0	0	0	0
1	1	true	0	0	0	0	0
1	0	false	0	0	0	0	0
1	0	true	δ	-1	+1	±1	0
0	1	false	δ	-1	+1	±1	0
0	1	true	δ	+1	-1	0	±1
0	0	false	δ	+1	-1	0	±1
0	0	true	δ	+1	-1	0	±1

In Table I, b_i and x_i $(i = 1,2\ldots,m\times N)$ are the *i*-th bit of the best solution *b* and binary solution *x* respectively, *f*(*x*) is the fitness of ***x*,** $s(\alpha_i,\beta_i)$ is the sign of the angle that controls the direction of rotation. Why it can converge to a better solution? Looking into Figure 3, you may get some ideas.

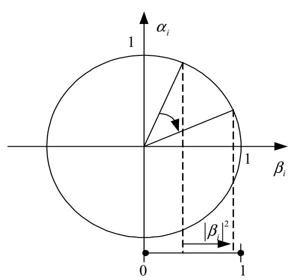

Figure 3. Quantum Rotation gate.

For example, if $x_i = 0$, $b_i = 1$, *f*(*x*)>*f*(*b*). The probability of current solution x_i should be larger to get a better individual, i.e. $|\beta_i|^2$ should be larger, so if (α_i,β_i) in the first or third quadrant, θ should rotate clockwise or else rotate counter-clockwise.

The update procedure can be described as

$$\begin{bmatrix}\alpha_i^{t+1}\\ \beta_i^{t+1}\end{bmatrix} = U(\Delta\theta_i^t) * \begin{bmatrix}\alpha_i^{t}\\ \beta_i^{t}\end{bmatrix}, (i = 1,2,\ldots,N\times m) \tag{11}$$

E. Evaluation of Fitness Function

Sometimes we should also take into account the number of edges that may affect the cost of the multicast tree. If we use as many public paths as we can, the cost could be decreased, as showed in Figure 4.There are two path ("4+5,6+6") from point a to point b and also there are two path ("5+8,6+8") from point a to point c and ("6+7,8+4") from point a to point d respectively. It is obviously that the cost of the path consist of real line is less than the dotted line. But the cost of multicast tree that consist of real line is more than that of dotted line. The two multicast tree costs are 5+4+8+8+4=29 and 6+6+8+7=27. The dotted line a-e denotes the public path. Figure 4 shows the public path is used to decrease the cost of the multicast tree.

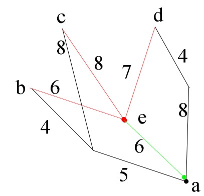

Figure 4. An example for finding the public path.

Therefore, the fitness function is defined as:

$$Fitness = alpha\times 20.0 / MultiTreeCost + (1.0 - alpha)\times 1.0 / TreeedgesCount, \tag{12}$$

where $alpha \in [0.5,1.0]$.

When the fitness of offspring is greater than that of parents, the offspring $Q(t)$ will be kept to lead the following mutation. If the best individual fitness has not been improved after successive 20 generations, we will randomly select a non-zero bit x_i^t $(i = m\times N)$ in binary strings solution $P(t)$ and set it to be 0. Then if it has improved the fitness function, we store the new x_i^t, otherwise we keep the non-zero x_i^t. At the same time, we update q^t by rotating gate.

F. The termination condition

The termination condition is the given number of generation.

G. Computational Complexity

In this section, we only consider population size in computational complexity. Assuming that the size of network nodes is *n*, the size of destination nodes is *m*, the size of candidate path set is *P*, the size of population is *M*, the maximum size of generations is *g* and then the time complexity for the algorithm can be calculated as follows:

The time complexity for searching the candidate path set is $O(n)$; the time complexity for sorting the candidate path set is $O(\log(P))$; the time complexity for finding the optimal multicast tree is $O(g\times M\times m)$. So the worst total time complexity for the proposed algorithm is

$$O(g\times M\times m) + O(\log(P)) + O(n). \tag{13}$$

If we correlate linearly *g* and *M* with *n*, and according to the operational rules of the symbol O, the worst time complexity for the proposed algorithm can be simplified as

$$O(n^2\times m) \tag{14}$$

But the worst time complexity for BSMA is $O(n^3\times\log n)$ [2], so the time complexity for QEA used to solve multicast routing problem is superior to those of BSMA.

V. CONVERGENCE OF THE ALGORITHM

Definition 3 let $X(t)=(x_1(t),x_2(t),\cdots,x_n(t))$ in S^n be the population at time t and for $X(t)$, defined:

$$M=\{\bar{X}\,|\,f(\bar{X})=\max\{f(X_i(t)),i\le n\}\}, \tag{15}$$

$$M^*=\{\bar{X}\,|\,f(\bar{X})=\max\{f(X),X\in S^n\}\}. \tag{16}$$

M is called the satisfied set of population X_t and M^* is defined as the global satisfied set of state S^n.

Definition 4 supposes arbitrary initial distribution, the following equation satisfies:

$$\lim_{t\to\infty} P\{M\subseteq M^*\}=1. \tag{17}$$

Then we call the algorithm is convergent.

Theorem 1 the population series of QEA $\{Q_t, t\ge 0\}$ is finite homogeneous Markov chain.

Proof: Like the evolutionary algorithms, the state transfer of QEA are processed on the finite space, therefore, population is finite, since

$$Q(t+1)=T(Q(t))=T_s\circ T_u\circ Q(t). \tag{18}$$

T_s and T_u indicate the selection operator and the update operator respectively. Note that T_s and T_u have no relation with t, so $Q(t+1)$ only relates with $Q(t)$. Namely, $\{Q_t, t\ge 0\}$ is finite homogeneous Markov chain.

Theorem 2 the M of Markov chain of QEA is monotone increasing, namely, $\forall t\ge 0, f(Q_{t+1})\ge f(Q_t)$.

Proof: Apparently, the individual of QEA does not degenerate for our adopting holding best strategy in the algorithm.

Theorem 3 The quantum-inspired evolutionary algorithm (QEA) is convergent.

Proof: For Theorem 1 and Theorem 2, the QEA is convergent with the probability 1.

VI. PERFORMANCE COMPARISON

In this section, we compare conventional GA and CS [8] with QEA in solving the multicast routing problems that were frequently used as benchmark problems.

The networks with different scales we used in our experiments are produced by the Waxman model [1]. For each random network, the coordinate area is 4000*km*×4000*km,* the mean degree of a nodes is 4. The delay of link (x,y) is defined as $d(x, y)$ = 1000×dis(x, y)/c, in which dis(x, y) is the distance between x and y, c is the light speed. Delay constraint Δ=100.

We apply the following metric to evaluate the three algorithms' performance, where T is the number of independent runs, *Cost* is the solved optimal multicast tree cost and *Time* is the running time for each algorithm. Obviously, the smaller R is, the better the performance of QEA is, compared with GA and CS.

$$R_{\cos t}=\frac{1}{T}\sum_{i=1}^{T}\frac{Cost(T_{\text{QEA}})}{Cost(T_{\text{(GA)or(CS)}})},\quad R_{time}=\frac{1}{T}\sum_{i=1}^{T}\frac{Time(T_{\text{QEA}})}{Time(T_{\text{(GA)or(CS)}})}. \tag{19}$$

In the following experiments, we performed 30 independent runs on each test problem. All experiments are executed on a 2.33GHz Pentium IV PC with 2G RAM by programming with C++. For GA, we use a population of size 10, $p_c=0.6$ and $p_m=0.01$. For CS, the population size is 10, the clone size is 20 and $p_m=0.6$. For QEA, the size of population M= 10, where N denotes the number of qubits used to represent a destination node in the candidate path.

1) Comparison between QEA and another two algorithms-GA and CS with network scale increased.

To make a fair comparison, the number of generations is kept 100 when *m*/*n*=0.15 (*m* is the number of destination nodes, and *n* is the scale of the network) for all the three algorithms in Figure 5 to Figure 8. The network scale is increased from 100 to 1000 and sampled it at intervals of 100 in Figure 5 to Figure 8.

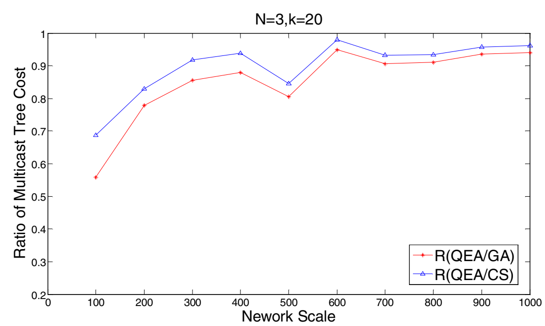

Figure 5. Performances of multicast cost versus network scale with N=3 and k=20,where k represents that in both GA algorithm and CS algorithm that we only take the first k% of the candidate path set for each destination node.

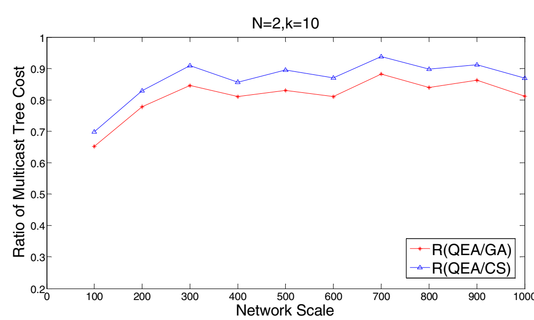

Figure 6. Performances of multicast cost versus network scale with N=2 and k=10,where k represents that in both GA algorithm and CS algorithm that we only take the first k% of the candidate path set for each destination node.

R is less than 1 and is stable under the different parameters setting in Figure 5 and Figure 6. It is shown that QEA is better expansibility and is better performance at complex and massive problems compared with GA and CS.

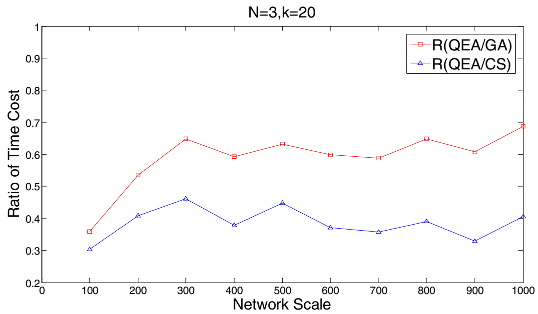

Figure 7. Performances of time cost versus network scale with N=3 and k=20 where k represents that in both GA algorithm and CS algorithm that we only take the first k% of the candidate path set for each destination node.

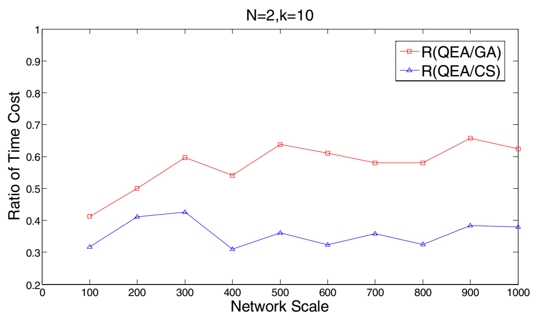

Figure 8. Performances of time cost versus network scale with N=2 and k=10 where k represents that in both GA algorithm and CS algorithm that we only take the first k% of the candidate path set for each destination node.

Figure 7 and Figure 8 show that the percentage of time cost of QEA to that of CS and GA is between 30 and 70. QEA makes full use of the parallelism of quantum computing and sets few parameters, compared with GA and CS. Thereby the utilization of memory is improved.

2) Comparison between QEA and another two algorithms-GA and CS with the number of destination nodes increased.

We use the Waxman model [1] produce a 1000 node network to inspect algorithm's performance at different number of destination nodes. The percentage of destination nodes to network nodes is increased from 5 to 50 and is sampled it at intervals of 5. Figure 9 shows R is less than 1 under different number of the destination nodes. With increased of destination nodes, the ratio of multicast tree cost is decline. That also indicates QEA has stronger search capability than CS and GA.

VII. Conclusions

In this paper, we proposed a new optimization algorithm-quantum-inspired evolutionary algorithm (QEA) to solve the multicast routing problems, inspired by the concept of quantum computing. In particular, a qubit individual was defined as a string of quantum bits for the probabilistic representation, so it can improve diversity of the population and avoiding premature convergence. Due to the novel representation, we put forward the quantum mutation operator which was used in the population to accelerate convergence of the proposed algorithm. As a result, the proposed QEA has the automatic balance ability between exploration and exploitation. Furthermore, we applied QEA to solving the multicast routing problems. When compared with conventional GA and CS, QEA is more effective in terms of convergence and diversity with the scale of network increased.

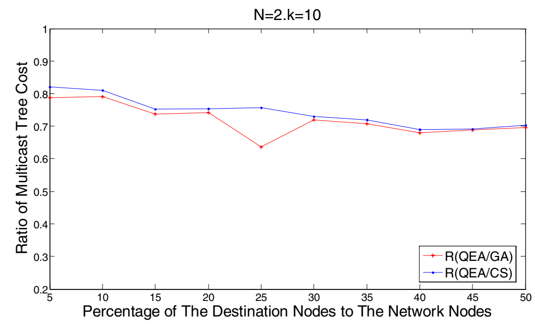

Figure 9. Percentage of QEA to GA and CS versus number of destination nodes with N=2 and k=10 where k represents that in both GA algorithm and CS algorithm that we only take the first k% of the candidate path set for each destination node.

References

[1] B.M. Waxman, "Routing of Multipoint Connections," IEEE Journal of Selected Areas in Communications, 6(9):1617-1622, 1988.

[2] M. Parsa, Q. Zhu, "Garcia-Luna-Aceves JJ. An Iterative Algorithm for Delay-Constrained Minimum-Cost Multicasting," IEEE/ACM Trans on Networking, 6(4): 461-474, 1998.

[3] V.P. Kompella, J.C. Pasquale, G.C. Polyzos, "Multicasting for Multimedia Applications," In: Eleventh Annual Joint Conference of the IEEE Computer and Communications Societies. Piscataway, NJ, USA: IEEE Press, 2078-2085, 1992.

[4] Q. Sun, H. Langendorfer, "Efficient Multicast Routing for Delay-Sensitive Applications," In: Proceedings of the 2th Workshop on Protocols for Multimedia Systems, 242-458, 1995.

[5] Q.F. Zhang; Y.W. Leung, "An Orthogonal Genetic Algorithm for Multimedia Multicast Routing," IEEE Transactions on Evolutionary Computation, 3(1): 53-62, 1999.

[6] A.T. Haghighat, K. Faez, M. Dehghan, A. Mowlaei, Y. Ghahremani, "GA-Based Heuristic Algorithms for Bandwidth-Delay-Constrained Least-Cost Multicast Routing," Computer Communication, 27(1):111-127, 2004.

[7] Y.T Qi, L.C. Jiao, F. Liu, "Multi-Agent Immune Memory Clone Based Multicast Routing," Chinese of Journal Electronics, 17(2):289-292, 2008.

[8] F. Liu, H. C. Yang, "A Multicast Routing Algorithm Based on Clonal Strategies," Journal of electronics and information technology, 26(11): 1825-1829, 2004. (in Chinese)

[9] L.C. Jiao, Y.Y. Li, M.G. Gong and X.R. Zhang, "Quantum-inspired immune clonal algorithm for global optimization", IEEE Transactions on System, Man, and Cybernetics, Part B, 2008, 38(5):1234-1253.

[10] K.-H. Han and J.-H. Kim, "Quantum-Inspired Evolutionary Algorithms With a New Termination Criterion, H_ε Gate, and Two-Phase Scheme," IEEE Trans. Evol. Comput., vol. 8, pp. 156-169, Apr. 2004.