A Novel Fast Multi-objective Evolutionary Algorithm for QoS Multicast Routing in MANET[*]

Yanbing LIU[†]

School of Computer Science, Chongqing University of Posts and Telecommunications
Chongqing, 400065, P.R. China
E-mail: liuyb@cqupt.edu.cn

Jun HUANG[‡]

School of Computer Science, Chongqing University of Posts and Telecommunications
Chongqing, 400065, P.R. China
E-mail: xiaoniuadmin@gmail.com

Abstract

Multicast routing is regarded as a critical component in networks especially the real-time applications become increasingly popular in recent years. Existing multicast routing under certain QoS Constraints tend to use conventional IP QoS architecture based on GA. In this paper, we propose a novel fast multi-objective evolutionary algorithm called QMOEA for solving multicast routing problem (MRP) in MANET. The steps are that, through the analysis of the strengths and limitations of the well-known multicast architecture, we firstly give an improved Core Based Tree model to simplify the MRP. Based on this model, we then propose the QMOEA which integrates the "Greedy" and "family competition" approaches to speed up the convergence and to maintain the diversity of population. After that, we present the theoretical validations for the proposed method to show its efficiency, and finally, the performance of MANET scaled from 20 to 200 nodes with different types of service is evaluated by OPNET, our experimental results show that our proposed method is capable of achieving faster convergence and more preferable for multicast routing in MANET compared with other genetic algorithms (GAs) well-known in the literature.

Keywords: Multi-objective; Evolutionary algorithm; MANET; QoS routing

1. Introduction

Multicast routing has drawn a lot of attention in recent years, since it enables a source to send messages to multiple destinations concurrently. The wireless communication technologies and mobile devices have realized the important and useful applications of mobile ad hoc network (MANET) with greatly advancement. Multicast routing plays a critical role in the transmission of information, such as video and other streaming data. Nevertheless, the main difficulty in designing a routing protocol for mobile ad hoc networks is the dynamical topology which results from the random movement of mobile nodes within the source's transmission range. MANET, which is fundamentally different from conventional infrastructure-based networks, is self-configuring and formed directly by a set of mobile nodes. In MANET, the heterogeneity of networks and destinations makes it difficult to improve bandwidth utilization and service flexibility. Therefore, mobility of hosts (nodes) makes

the design of multimedia distribution jobs greatly challenging.

Up to now, various works involved focus on designing multicast routing algorithm [1-6]. An early summary of problems and general technical solution related to multicast communication was given by Diot et al. [7]. Tafazolli et al. [8] and Chen et al. [9] present a survey of multicast routing under certain QoS constraints solutions for MANET. As an NP-Complete problem, to develop different types of heuristic algorithm for calculating near-optimum paths with multiple QoS constraints is a research focus. For example, Wang et al. [10] investigate three representative intelligent computational methods (genetic algorithm, simulated annealing and Tabu search) to construct the QoS multicast trees to support multimedia group communication separately; the proposed algorithms consider both the end-to-end delay constraint and network resource requirement; the simulation evaluates the performance of three heuristics on a small-scale real-world multimedia communication network and a randomly generated large-scale network, and then concludes that genetic algorithm shows the best performance in terms of the solution quality. In 2008, Qu et al [11] propose a set of node-based rate constraints to model the interference relationship among nodes in a wireless ad hoc network and to provide rate constraints for its QoS flows, they demonstrat that, the algorithm can always admit the feasible flows as well as make full use of the bandwidth resource. Zahrani et al. [12, 13] import logarithmic simulated annealing (LSA) as pre-processing of GA; the algorithm utilizes the partially crossover operation (PMX) under the elitist model and the landscape analysis were presented to estimate the depth of the deepest local minimal in the landscape generated by the routing tasks and the objective function; experimental results show that the algorithm is effective on the randomly generated networks. Yang et al. [14] and Ikeda et al. [15] focus on creating a robust path to find solution for specified networks; the genetic algorithm is proposed and, respectively, the individuals of the population are represented by trees, algorithm uses the single point crossover and a mutation operation where the "tree junctions" are chosen randomly, the algorithm employs the elitist model where the individual with the highest fitness value in a population is left unchanged in the next generation, the simulation results show that the algorithm is reasonably fast on small and medium size networks. Differing from the above network architec-

ture, Rango et al. [16] and Mala et al. [17] refer a scheme called Core Based Tree (CBT) with genetic algorithm which provides a new way for realizing multicast routing protocol in wireless networks, however, it needs much running time.

The remainder of the paper is organized as follows. In Section 2 we state some basic conceptions of multi-objective optimization and give the mathematical description for problem. A QoS-Aware Multicast Routing Architectures is given in Section 3. We outline the design of proposed algorithm in Section 4. Section 5 analyses the properties of our method. The simulation results and performance evaluation are shown in Section 6 and the last section presents our conclusion.

2. Notations and Problem Formulation

To begin with we will introduce some basic conceptions of multi-objective optimization before we describe the problem that would help us know the model thoroughly.

2.1. *Basic Conceptions*

Definition 1 (Multi-objective Optimization Problem, MOP) the MOP consists of n decision parameters, k objective functions and m constraints, without loss of generality:

Maxmize $\quad y = f(x) = (f_1(x), f_2(x), \cdots, f_k(x))$

Subject to $\quad e(x) = (e_1(x), e_2(x), \cdots, e_m(x)) \leq 0$

Where $x = (x_1, x_2, \cdots x_m) \in \Omega$, $y = (y_1, y_2, \cdots y_n) \in \Phi$. x is decision vector, y is objective vector, Ω denotes the decision space formed by x, Φ denotes the objective space formed by y.

Definition 2 (Pareto Dominance): a vector $a = (a_1, a_2, \cdots, a_n)$ is said to dominate $b = (b_1, b_2, \cdots, b_n)$ if and only if a is partially less than b, i.e.

$\forall i \in \{1, 2, \cdots, n\}, a_i \leq b_i \wedge \exists j \in \{1, 2, \cdots, n\}, a_j < b_j$.

Definition 3 (Pareto Optimal): a decision vector x_b is said to be Pareto Optimal if and only if there is no $x_a \in \Omega$ where $F(x_a) = a = (a_1, a_2, \cdots, a_n)$ dominates (use Definition 2's scheme) $F(x_b) = b = (b_1, b_2, \cdots b_n)$.

Definition 4 (Pareto Front) the set of all Pareto-optimal decision vectors is called the Pareto-optimal set of the problem and the corresponding set of objective vectors is called Pareto Front.

As we know, most of problems in the world are known as non-orthogonal problems. In a linear problem, each component is independent, so that any improvement to any one part will lead to an improvement of the

entire system. But few real-world problems like this, while most of real world problems are nonlinear, one component changing may have ripple effects on the entire system, and thus we should treat the problem as a multi-objective optimization model.

2.2. *Problem Formulation*

A network can be modeled as an undirected graph $G = (V, E)$, where V is the set of nodes that represent routers and E is the set of arcs (arcs represent path between nodes). Each link between two nodes is bi-directional, it means that if there is a link $e = (u, v)$, the link $e' = (v, u)$ also exists. We employ the metrics of $bandwidth(e)$, $delay(e)$, packet $loss(e)$ ratio and delay $jitter(e)$, which could describe the QoS request of most services from our previous study, to evaluate each link e [18]. Let $p(s, d)$ be a path from the source node s to the destination d, the total bandwidth of the path $p(s, d)$ is the minimum of bandwidth of all links along $p(s, d)$ and it is denoted as $Bandwidth(p(s, d))$:

$$Bandwidth(p(s, d)) = \min_{e \in p(s, d)} [bandwidth(e)]. \quad (1)$$

$$Delay(p(s, d)) = \sum_{e \in p(s, d)} delay(e). \quad (2)$$

$$Loss(p(s, d)) = 1 - \prod_{e \in p(s, d)} (1 - loss(e)). \quad (3)$$

$$Jitter(p(s, d)) = \max[Delay(p(s, d))] - \min[Delay(p(s, d))]. \quad (4)$$

QoS multicast routing problem can be defined as follows:

$$\min F = \min\{-F_1, F_2, F_3, F_4\}. \quad (5)$$

Where:

$$\begin{cases} F_1 = \min_{e \in p(s, d)} [Bandwidth(e)] \\ F_2 = \sum_{e \in p(s, d)} Delay(e) \\ F_3 = 1 - \prod_{e \in p(s, d)} (1 - (Loss(e)) \\ F_4 = \max[Delay(p(s, d))] - \min[Delay(p(s, d))] \end{cases} \quad (6)$$

In contrast, this model imports a scalarization scheme to depict the problem rather than to aggregate the multi-metric into a single value. Ikeda et al. [15] describe the relationship between Pareto solution and the solution space (see Fig.1). Fig.1 indicates that solutions obtained by GA are rare in the Pareto solution space. It can be predicted that we will get no solution in the Pareto solution space if the coefficients are not appropriate. Due to contradiction among metrics, GA will make only one of them prone to optimum.

Accordingly, it is improper to aggregate the multi-metric into a single value among multi-objective problem, and thus our definition for solving multicast problem is more preferable.
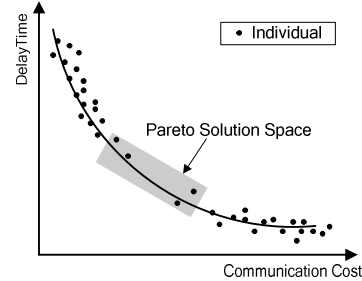

Fig.1. Relationship between Pareto solution and solution space

3. QoS Multicast Routing Architecture

Fig.2 illustrates the change of topology of MANET. It is more complex to construct a Steiner tree for the group with dynamic change in group members. Thus the CBT technique is chosen, and one node will be regarded as the Rendezvous Point (RP). Rango et al. [16] and Mala et al. [17] show details that how the CBT protocol works when nodes are allowed to join and leave the multicast group dynamically. We focus on the RP (core) selection herein rather than address the details again. Nevertheless, as a key point of CBT, it adopts link costs for the core selection, namely, it aims at only one component of the problem instead of all.
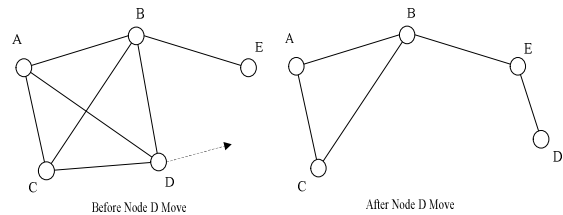

Fig.2. Dynamic topology of MANET

In order to overcome the difficulty of selecting core, we redefine the $cost(p(s, d))$:

$$\begin{cases} cost(p(s, d)) = \omega_1 \cdot Bandwidth(p(s, d)) + \\ \omega_2 \cdot Delay(p(s, d)) + \omega_3 \cdot Loss(p(s, d)) + \\ \omega_4 \cdot Jitter(p(s, d)) \\ \prod_{i=1}^{4} \omega_i = 1 \end{cases} \quad (7)$$

Let $r(p(s,d))$ be the sum of cost, the mean cost associated to core node c can be expressed as follows:

$$cost(c) = \frac{1}{|V_{MG}|} \sum_{x \in V_{MG}} r(p(c,x)). \qquad (8)$$

Where $|V_{MG}|$ is the amount of members in multicast group. The multicast routing architecture can be described as Fig. 3. We divide the multicast routing problem into two segments: one is formed by the multicast group and the core via improved CBT protocol that uses new policy in selecting core showed in Eq. (7) and (8); the other is the combination of the source and the core, using the proposed method to find the optimum path from the source to the core. It means that the MRP is divided into "Unicast" segment and "Multicast" segment.
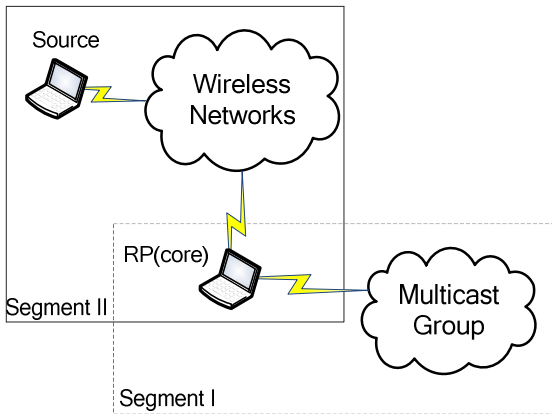

Fig.3. II-segment architecture of multicast routing

Proposition: the II-segments division of MRP does not change its property of NP-Complete.

Proof Wang et al. [19] proved that two or more additive and multiplicative metrics in any possible combination is NP-Complete. In our architecture, the delay is an additive metric and the loss is multiplicative, therefore, it is still an NPC problem after division of MRP in MANET.

4. Design of QMOEA

4.1. *Multi-Objective Evolutionary Algorithm*

Evolutionary algorithm (EA) is a type of heuristic global search algorithm, extensively applied to various kinds of optimization problems. In general, they includes three operators: selection, crossover and mutation. Fig.4 presents the framework of EA.
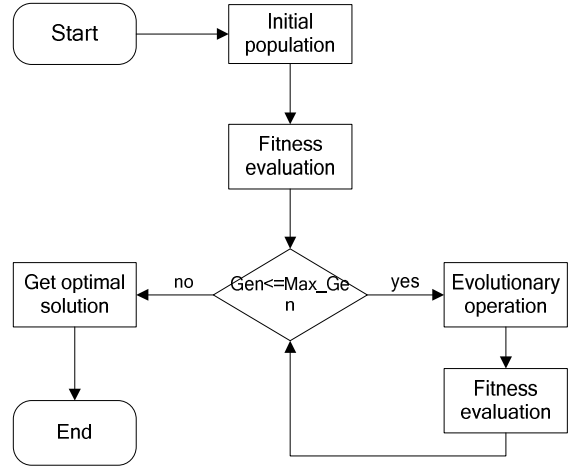

Fig. 4 Framework of EA

EAs have been recognized to be possibly well-suited to multi-objective optimization since early in their development. Multiple individuals can search for multiple solutions in parallel, eventually taking advantage of any available similarities in the family of possible solutions to the problem. The ability to handle complex problems, involving features such as discontinuities, multimodality, disjoint feasible spaces and noisy function evaluations, reinforces the potential effectiveness of EA in multi-objective search and optimization, which perhaps is a problem area where Evolutionary Computation distinguishes itself from other multi-objective algorithms.

More stuff about multi-objective evolutionary algorithm (e.g. NSGA-II, SPEA2) can be found in Deb's book [20].

4.2. *QMOEA for MANET*

Throughout this paper, $\overrightarrow{X}(n)$ denotes the nth generation population, $\overrightarrow{X}$ denotes the current population. X_i is the individual in $\overrightarrow{X}$. Probabilities for crossover and mutation are denoted by p_c and p_m. $T_s(\cdot)$, $T_c(\cdot)$ and $T_m(\cdot)$ stand for the selection, crossover and mutation for population respectively.

4.2.1. *Coding*

Chromosome coding, the chief matter and key issue when applying the evolutionary algorithm, affects not only the methods of decoding and fitness evaluation, but also the realization of selection, crossover and mutation procedures. There are many works focusing on coding. Zhou et al. [21] summarize three normal coding approaches

and conclude that Prüfer coding is more feasible because of lower complexity. However, for the specificity of MRP in MANET, coding methods can be divided into two categories: one is that the individual is represented by a tree [14], however, whether this method could traverse the whole state space or not needs to be proven despite it can eliminate cycles and invalid paths after genetic operations; the other is path coding, which utilizes the visiting sequence of nodes as the coding principle that conforms to Dejong's block assumption.
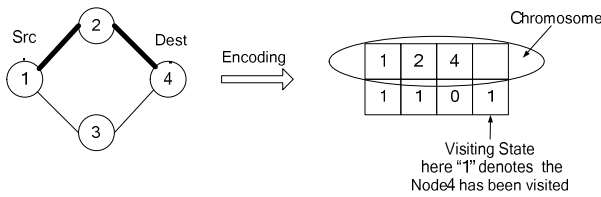

Fig. 5 An example of encoding

In this paper, we adopt path coding and attach a visiting vector to each chromosome (As Fig.5 shown). It is simple and doable; moreover, it does not generate invalid paths after genetic operation.

4.2.2. *Fitness Function*

The algorithm sorts all the chromosomes according to Pareto Dominance relationship between two individuals when evaluating each individual's fitness; i.e. it defines the first batch of Pareto Dominant individuals' fitness, which is called "Pareto Rank", equal to 1, and then removes these individuals from the population. In the residual population, the second batch of Pareto Dominant individual's fitness is defined as 2, and the rest may be similarly deduced till all individuals are defined.

4.2.3. *Selection*

By means of individual's fitness evaluation, we can conclude that individuals in the same Pareto Dominant have the same Pareto rank. So, the selection can be described as:

$$P\{T_s(\overrightarrow{X}) = X_i\} = \frac{f(X_i)}{\sum\limits_{k=1}^{n} f(X_k)}. \tag{9}$$

4.2.4. *Crossover*

We first define the "adaptive back-off selection probability" as:

$$p = \frac{|N| - |M_{weight\_k}|}{|N|}. \tag{10}$$

Where $|N|$ is the number of individuals, and $|M_{weight\_k}|$ stands for the number of individuals whose Pareto rank equals to k. Because adaptive back-off selection is an ideal way to implement "family competition", it can avoid two potentially negative effects - the loss of population diversity and trapping at a local optimal.

Hence, when one individual is chosen randomly, the other one that participates in crossover could be selected by:

$$P\{T_s(\overrightarrow{X}) = X_j\} = \begin{cases} \dfrac{f(X_j)}{\sum\limits_{k=1}^{n} f(X_k)} \cdot p & X_j \in M_{weight\_k} \\ \dfrac{f(X_j)}{\sum\limits_{k=1}^{n} f(X_k)} \cdot (1-p) & X_j \notin M_{weight\_k} \end{cases}. \tag{11}$$

Where $\overrightarrow{X}(n)$ is the nth generation population, $\overrightarrow{X}$ stands for the current population and X_i is the individual in $\overrightarrow{X}$.

To speed up convergence of QMOEA, greedy algorithm is imported. Greedy, a useful and powerful means in many optimization problems [6, 22, 23, 24], converges very quickly but it is liable to trap at a local optimal. That is the reason why we employed "family competition". So, the crossover operation can be described as follows:

Step 1: Select N individuals independently from the group $\overrightarrow{X}(n)$ so as to get the population of $\overrightarrow{X}(n) = (X_1, X_2, \cdots, X_N)$.

Step2: Select two individuals according to "family competition":

$$X_i(n) = (v_1, v_2, v_3, \cdots, v_n) \tag{12}$$
$$X_j(n) = (v_1, v_2', v_3', \cdots, v_n) \tag{13}$$

Step3: Let v_1 be the first gene of $X_i(n+1)$, find the next gene of v_1 in (12) and (13), evaluating their Pareto Dominant Relationship, then choose one (such as v_2) that dominates the other as the second gene of $X_i(n+1)$ and set the corresponding unit of its visiting vector to 1.

Step 4: Find the position v_2 in (12) and (13), compare their tail gene to confirm which is better, then choose it as the next gene of $X_i(n+1)$ and set its corresponding unit of visiting vector to 1 analogously.

Step 5: Repeat the above steps, till $X_i(n+1)$ is formatted.

$X_j(n+1)$ can be generated similarly, just with the difference of getting the next gene from the reversed direction. Then crossover can be described as follows:

$$P\{T_c(X_i, X_j)\} = \begin{cases} (l_1 + l_2) p_c & X_i \neq X_j \\ 1 - p_c & X_i = X_j \end{cases} \quad (14)$$

Where l_1 and l_2 is the length of chromosome.

4.2.4. *Mutation*

The single point mutation is adopted in QMOEA.

$$P\{T_m(\vec{X}) = Y\} = p_m^{d(X,Y)} (1 - p_m)^{l - d(X,Y)} \quad (15)$$

Where $d(X,Y)$ denotes the number of gene pairs that the corresponding gene in X and Y is different with each other.

5. Analysis of QMOEA

We assumed that $|V|$ is the number of nodes in networks; $c \cdot |V|$ is the size of population, c is a constant; $|Gen.|$ denotes iterative times of QMOEA, the pseudo-code of QMOEA is shown as Fig. 6.

Algorithm QMOEA

1. **Void** Main
2. {
3. *set generation=0;*
4. **While** (*generation <= $|Gen.|$*)
5. {
6. *Initialization;*
7. *Fitness evaluation;*
8. *Selection;*
9. *Select two individuals by "family competition";*
10. *Crossover;*
11. *Mutation;*
12. }
13. }

Fig. 6 Pseudo-code for MOEAQ

5.1. *Time Complexity*

Firstly, for population initialization that takes $O(c \cdot |V| \cdot (|V| - 1))$; secondly, the algorithm needs to sort all of the individuals when evaluating fitness, which takes $O(c^2 \cdot |V|^2)$; moreover, operations selection, crossover (it contains "family competition" selection for crossover) and mutation take $O(c \cdot |V|)$, $O(c \cdot |V| + c \cdot |V|^3)$ and $O(c \cdot |V|)$ respectively. Therefore, the one time of evolu-

tionary operation takes $O(c \cdot |V| \cdot (|V| - 1) + c \cdot |V| + c \cdot |V| + c \cdot |V|^3 + c \cdot |V|)$. So, the time complexity of MOEAQ is

$$\begin{aligned} T_{MOEAQ} &= O(|Gen.| \cdot (c \cdot |V| \cdot (|V| - 1) + c \cdot |V| + c \cdot |V| \\ &\quad + c \cdot |V|^3 + c \cdot |V|)) \\ &= O(|Gen.| \cdot |V|^3) \end{aligned} \quad (16)$$

QSGA, which proposed by Ref. 17, takes $O(c \cdot |V| \cdot (|V| - 1))$ for population initialization; $O(c \cdot |V|^2)$ for fitness evaluation; $O(c \cdot |V|)$, $O(c \cdot |V|^2)$ and $O(c \cdot |V|)$ for selection, crossover and mutation respectively, besides, it eliminates the circle and invalid path that needs $O(c \cdot |V|^3)$, therefore, the time complexity of GAQ is

$$\begin{aligned} T_{GAQ} &= O(|Gen.|' \cdot (c \cdot |V| \cdot (|V| - 1) + c \cdot |V| + c \cdot |V|^2 \\ &\quad + c \cdot |V| + c \cdot |V|^3)) \\ &= O(|Gen.|' \cdot |V|^3) \end{aligned} \quad (17)$$

Where $|Gen.|'$ denotes the iterative times of QSGA.

Based on our previous research in traveling salesman problem (TSP) [22, 23], we can conclude that $|Gen.|' \gg |Gen.|$. So, theoretically, QMOEA can converge much faster than QSGA.

5.2. *Convergence analysis of QMOEA*

To validate the convergence of QMOEA, two definitions are required as follows:

Definition 5 (Satisfactory Population Value) $F(\vec{X}) = \max\{f(X_i); i \leq N\}$ is the satisfactory population value of $\vec{X} = \{X_1, X_2, \cdots, X_N\}$.

Definition 6 (Satisfactory Population Set) $M^* = \{\vec{X}; F(\vec{X}) = \max\{f(X); X \in S\}\}$ is satisfactory population set.

Lemma Utilizing Eq. (7) to depict fitness is the sufficient condition of Pareto rank in QMOEA.

Proof It is obvious that, the individual who with the lowest Pareto rank in the population is definitely to be with the highest fitness depicted by Eq. (7), i.e. the former is the sufficient condition of the latter.

According to the above two definitions and the *Lemma* we assume that QEA uses formula (7) to evaluate fitness and the same genetic operators mentioned in section 4, then we have:

Theorem 1 Assume that $\{\vec{X}(n); n \geq 0\}$ is the initial population generated by QEA then $\{\vec{X}(n); n \geq 0\}$ is a homogeneous Markov chain.

Proof Firstly, QEA is a Markov chain for the reason that there is no following effect after genetic operations; secondly, we know that $P\{\vec{X}(n+1) = \vec{Y} / \vec{X}(n) = \vec{X}\}$ is independent of n, so it is homogeneous. In summary, $\{\vec{X}(n); n \geq 0\}$ is a homogeneous *Markov Chain*.

Theorem 2 $\{\vec{X}(n); n \ge 0\}$ converges to $M_0^* = \{\vec{Y} = (Y_1, Y_2, \cdots, Y_N)\}$ with probability one.

Proof In terms of Eq. (9), (11), (14) and (15), if $\vec{X}, \vec{Y} \in M_0^*$; then crossover operator can guarantee that $P_n(\vec{X}, \vec{Y}) > 0$ and $P_n(\vec{Y}, \vec{X}) > 0$, i.e. $\vec{X} \leftrightarrow \vec{Y}$; if $\vec{X} \in M_0^*$, $\vec{Y} \notin M_0^*$; we have $P_n(\vec{X}, \vec{Y}) = 0$, i.e. $\vec{X}$ could not arrive $\vec{Y}$ and there must exist a stationary distribution π which makes that:

$$\lim_{n \to \infty} P\{\vec{X}(n) = \vec{Y} / \vec{X}(0) = X_0\} = \begin{cases} \pi(\vec{Y}), & Y \in M_0^* \\ 0, & Y \notin M_0^*. \end{cases} \quad (17)$$

Obviously, $P(\infty)$ has a unique, irreducible, non-periodic, and positive recurrence class M_0^*, and S / M_0^* is a non-recurrence class, so $\{\vec{X}(n); n \ge 0\}$ is strongly ergodic, to an arbitrary initial state $\vec{X}(0) = X_0$, we have

$$\begin{cases} \lim_{n \to \infty} P\{\vec{X}(n) = \vec{Y} / \vec{X}(0) = X_0\} = \pi(\vec{Y}) \\ \sum_{\vec{Y} \in M} \pi_\infty(\vec{Y}) = 1. \end{cases} \quad (18)$$

Therefore, we get:

$$\lim_{n \to \infty} P\{\vec{X}(n) = \vec{Y} / \vec{X}(0) = X_0\} = \sum_{\vec{Y} \in M} \pi_\infty(\vec{Y}) = 1. \quad (19)$$

From the validation of *Theorem 1* and *Theorem 2*, we know that QEA can achieve convergence. According to the lemma, utilizing formula (7) to depict fitness is the sufficient condition of Pareto rank in QMOEA, and we have that QMOEA can achieve convergence too.

6. Simulations and Performance Evaluation

6.1. *Simulations*

The proposed approach was tested on different scale of MANET ranged from 20 to 200 nodes. The MANET randomly generates a scenario [25], in which the nodes are distributed in an area of 1000×1000 m^2, four QoS parameters (bandwidth, delay, packet loss rate and jitter) are generated at the same time.

Fig. 7(a) shows the choice of crossover probability (Pc) in QEA when solving multicast routing problem among 200 nodes (we set population size to be 1000). As Pc increases from 0.20 to 0.45, the fitness of population increases due to the crossover being the main operator in the evolutionary algorithm, whereas the fitness of population decreases slightly when Pc set to 0.50, i.e. the best choice of crossover probability to QMOEA is 0.45. In the same way, Fig. 7(b) indicates the choice of mutation probability (Pm). When the mutation probability increases, the fitness of population also increases for

the higher probability of mutation can escape from the local optima effectively, however, we could only set the upper bound of Pm to be 0.1 for the reason that the higher Pm will make the algorithm degenerated into a stochastic searching approach.
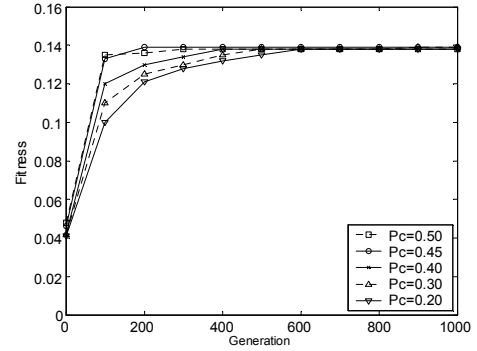
(a)
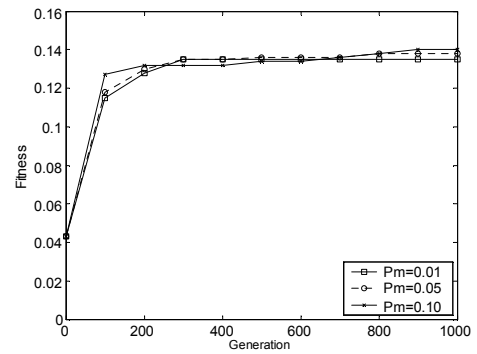
(b)

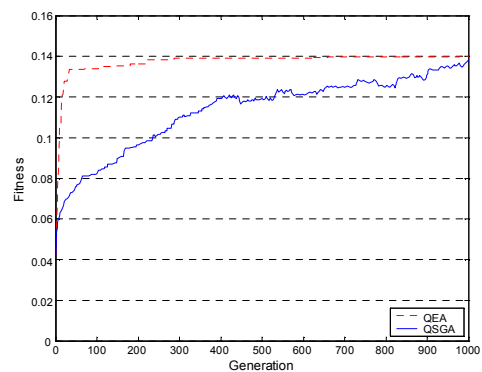Fig. 7 Parameter choices for crossover and mutation probability

Fig. 8 Comparison of fitness of QEA and QSGA

Fig. 8 illustrates the convergence process of QEA and QSGA as we run the algorithms for 30 times. From the curves, the population fitness of QSGA tends to increase before 400 generations, but it can escape when falling into the local optimal that exists in the section

from 400 to 500. The reason is: before 400 generations, the diversity of population is rich, whereas after 400, it is poor, and then the mutation operation works. When QSGA runs under generations from 500 to 1000, the perspective of population fitness is increasing. Compared with QSGA, the transformation of QEA is simpler due to the "Greedy" crossover operation. QEA converges at about 300 generations, and its fitness is higher than QSGA.

Table 1 compares the results obtained by QMOEA with that of the QSGA. As we have discussed above, The Pc and Pm for QMOEA are set to 0.45 and 0.1 respectively, while for QSGA they are set to 0.6 and 0.05.With respect to the running time (RT) and evolutionary generation (Gen), QSGA is more than three times as much as QMOEA. This result follows the analysis in Section5 that QMOEA can converge much faster than QSGA. As the nodes increase, QMOEA acts more effectively. QMOEA, integrating greedy and "family competition" approach, can not only stabilize the search behaviors, but also yield solutions of higher quality and cost less running time. In summary, QMOEA is a promising method for MANET multicast routing within reasonable time.

Table 1 convergence speed of QMOEA and QSGA

| Nodes	QMOEA		QSGA	
	Gen.	RT(s)	Gen.	RT(s)
20	10	<0.0001	48	0.015
40	25	0.015	90	0.046
80	38	0.093	196	0.250
100	62	0.234	321	0.421
160	205	1.203	989	3.640
200	315	2.834	1506	8.976

6.2. *Performance Evaluation*

The main concern of this section is to test the paths obtained by QMOEA and QSGA in providing multicast users with QoS and satisfying the service requirements of multimedia applications. We focus on quantitative aspects of efficiency such as throughput, delivery delay, media access delay and packet loss ratio. The simulations are conducted using OPNET Modeler 14.0 Educational Version and Wireless Module [26]. The results are aggregated for a multicasting scenario with typical two
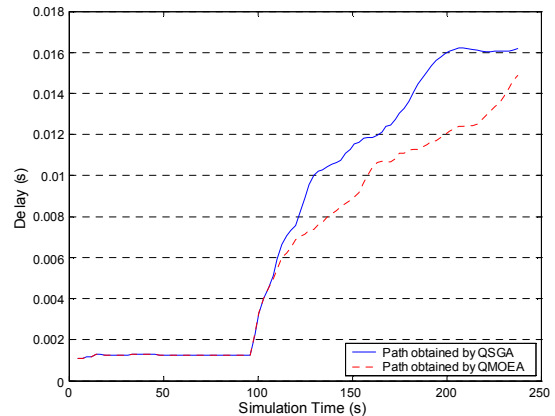
QoS classes. The simulation parameters are defined in Table 2.

Table 2 Simulation parameters for MANET

Number of Nodes	40
Type of Node	Mobile Terminal
Area	$1000{\times}1000\ m^2$
Transmission Protocol	TCP, UDP
Type of Service	FTP, Video Conferencing
Simulation Time	240s
Service Start Time	100s
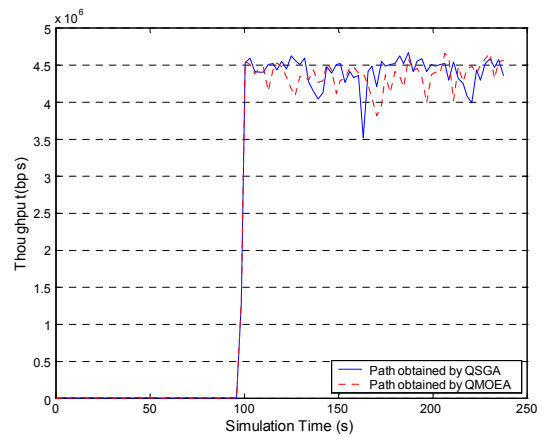
(a)

(b)

Fig. 9 Comparison of performance between two paths when

MANET is running with FTP service

Fig. 9 is the comparison of performance between QMOEA and QSGA with FTP service. The data drop curve is not given herein because FTP uses TCP to transmit data to ensure the number of data drop to be zero. The graph indicates that the throughput of two paths obtained by QSGA and QMOEA are almost the same (see Fig. 9(a)), meanwhile the path gained by

QSGA has higher delay (see Fig. 9(b)), i.e. the paths obtained by QMOEA has relatively better performance than QSGA does.
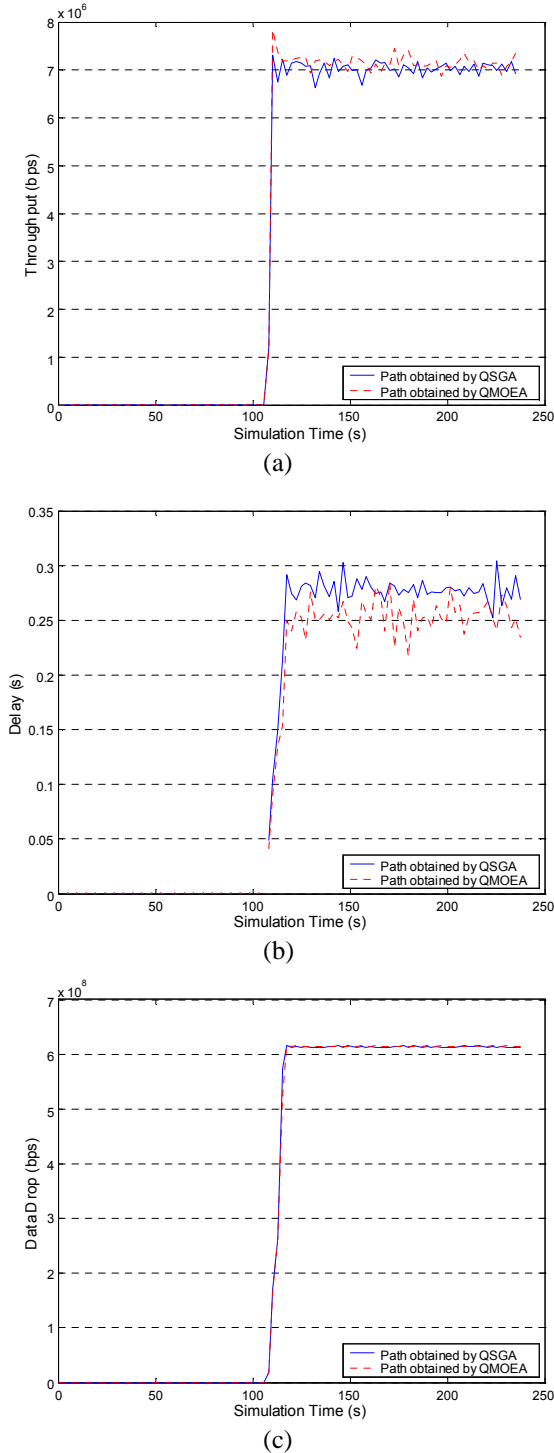

(a)

(b)

(c)

Fig. 10 Comparison of performance between two paths when

MANET is running with Video Conferencing service

Fig. 10 is the comparison of performance between two paths with Video Conferencing service. Fig. 10(a) describes the throughput of paths obtained by QSGA and QMOEA. It is quite obvious that the path gained by QMOEA has slightly higher throughput than that of QSGA. As throughputs are saturated at 120 seconds, the data are beginning to drop; we can see that the data drop from Fig. 10(c) increases dramatically. Fig. 10(b) represents delay of two paths. Apparently, the path obtained by QSGA has higher delay compared with QMOEA. To sum up, Fig.8 depicts that the path obtained by QMOEA has higher throughput, lower delay and as the same data drop, namely, that path is much better.

Consequently, QMOEA not only has better performance than QSGA but also can deal with multi-objective problem effectively. Moreover, it is more preferable for the dynamic topology of MANET since it can get a Pareto set rather than one "optimal" solution acquired from QSGA.

7. Conclusions

In this work, we analyzed strengths and limitations of the well-known multicast model firstly, and then an improved CBT protocol was proposed to simplify the QoS multicast routing problem in MANET; Based on the protocol, we came up with a novel fast multi-objective evolutionary algorithm to overcome the defection of slow convergence and liable to "premature" of conventional GA. The algorithm absorbs the "greedy" and "family competition" approaches which can speed up the convergence of algorithm and maintain the diversity of population; Apart from those traits, the proposed algorithm also can synthesize multi-objective effectively. Through the theoretical analysis, we obtained conclusions that 1). QMOEA needs less running time than typical method QSGA; 2). QMOEA can achieve convergence. The simulation results validate the correctness of these conclusions. Finally, the performance evaluation of path obtained by two methods (QSGA & QMOEA) are given, experimental results show that the path gained by QMOEA has better performance than that of QSGA.

For our proposed method runs in the exterior environment of the network, one direction of our further research is to put the proposed method into the real-world networks to investigate if the ideas developed in this paper could be executable.

Acknowledgements

References

1. Y. Xiao and K. Thulasiraman, and G. Xue, Constrained Shortest Link-disjoint Paths Selection: A Network Programming Based Approach, *IEEE Transactions on Circuits and Systems I: Regular Papers*. **53**(5) (2006) 1174–1187.

2. Y. Xiao, K. Thulasiraman, and G. Xue, QoS Routing in Communication Networks: Approximation Algorithms Based on the Primal Simplex Method of Linear Programming, *IEEE Transactions on Computers*. **55**(7) (2006) 815–829.

3. G. Xue and S.K. Makki, Multi-constrained QoS Routing: A Norm Approach, *IEEE Transactions on Computers*. **56**(6) (2007)859–863.

4. G. Xue, A. Sen, W. Zhang, J. Tang and K. Thulasiraman, Finding a Path Subject to Many Additive QoS Constraints, *IEEE/ACM Transactions on Networking.* **15**(1) (2007) 201–211.

5. D.H. Lorenz, A. Orda, D. Raz and Y. shavitt, Efficient QoS partition and routing of unicast and multicast, *IEEE/ACM Transactions on Networking*. **14**(6) (2006) 1336–1347.

6. G. Xue, W. Zhang, Multiconstrained QoS Routing: Greedy is Good, in *Proc. GLOBECOM'07*, (2007), pp. 1866–1871.

7. C.Diot, W. Dabbous and J. Crowcroft, Multipoint communications: a survey of protocol, function, and mechanisims, *IEEE Journal on Selected Areas in Communications*. **15**(3) (1997) 277–290.

8. L.H.II and R. Tafazolli, A Survey of QoS Routing Solution for Mobile Ad hoc Networks, *IEEE Communications Surveys & Tutorials*. **9**(2) (2007) 50–70.

9. L. Chen, W.B. Heinzelman, A Survey of Routing Protocols that Support QoS in Mobile Ad Hoc Networks, *IEEE Network*. **21**(6) (2007) 30–38.

10. X. Wang, J. Cao, H. Chen and M. Huang, QoS multicast routing for multimedia group communications using intelligent computational methods, *Computer Communications*. **29**(12) (2006) 2217–2229.

11. J. Qu, Z. Zhao, J. Zhao, Lianfang Zhang and Y. Shu, Node-Based Rate Constraints for QoS Flows in Wireless Ad-Hoc Networks, in PROC. *IEEE GLOBECOM'08* , (2008) pp. 1-5.

12. M.S. Zahrani, M.J. Loomes, J.A. Malcolm and A.A. Albrecht, Landscape analysis for multicast routing, *Computer Communications*. **30**(1) (2006) 101–116.

13. M.S. Zahrani, M.J. Loomes, J.A. Malcolm, A.Z.M. Dayem Ullah, K. Steinhofel and A.A. Albrecht, Genetic local search for multicast routing with pre-processing by logarithmic simulated annealing, *Computers & operations research*. **35**(6) (2008) 2049–2070.

14. Y. Yang, Y. Xu, Q. Li and F. Liu, A multi-objective genetic algorithm of QoS routing, *Journal of China Institute of Communications*. **25**(1) (2004) 43–51.

15. M. Ikeda, L. Barolli, G.D. Marco, A. Durresi, A. Koyama and M. Durresi, Evaluation of a Network Extraction Topology Algorithm for Reducing Search Space of a GA-based Routing Approach, in *Proc. ICDCSW'06*, (2006), p. 54.

16. F.D. Rango, M. Tropea, A.F. Santamaria and S. Marano, An enhanced QoS CBT multicast routing protocol based on Genetic Algorithm in a hybrid HAP-Satellite system, *Computer Communications*. **30**(16) (2007) 3126–3143.

17. C. Mala and S. Swlvakumar, Construction of an optimal multicast tree for group communication in a cellular network using genetic algorithm, *Computer Communications*. **29**(16) (2006) 3306–3312.

18. Y. Liu, H. Tang, M.H. Wang and S.X. Sun, Routing Attribute Data Mining Based on Rough Set Theory, in *Proc. ADMA'05*, (2005), pp. 276–283.

19. Z. Wang and J. Crowcroft, Quality-of-service routing for supporting multimedia applications, *IEEE Journal on Selected Areas in Communications*. **14**(7) (1996) 1228–1234.

20. Deb K. Multi-objective optimization using evolutionary algorithms. New York: Wiley; 2001.

21. G. Zhou, M. Gen and T. Wu, A new approach to the degree-constrained minimum spanning tree problem using genetic algorithm, in *Proc. IEEE SMC'96*, (1996), pp. 2683–2688.

22. Y. Liu and J. Huang, An Effective Parallel Hybrid GA for TSP, in *Proc. ICICA'07*, (2007), pp. 695–698.

23. Y. Liu and J. Huang, A Novel Genetic Algorithm and its Applications, in *Proc. IFIP-ICNPC'08*, (2008), pp. 263-266.

24. Y. Liu and D. Chen, A novel greedy computing algorithm for rectangle packing problems, *International Journal of Computer Science and Network Security*, **6**(4) (2006) 78–81.

25. H.F. Salama, D.S. Reeves and Y. Viniotis, Evaluation of Multicast Routing Algorithm for Real-Time Communication on High-Speed Networks, *IEEE Journal on Selected Area in Communications*, **15**(3) (1997) 33–2345.

26. OPNET Technologies Inc., Bethesda, MD, USA, available at http://www.opnet.com.