



Flooding-limited and multi-constrained QoS multicast routing based on the genetic algorithm for MANETs

Yun-Sheng Yen^a, Han-Chieh Chao^{b,c,*}, Ruay-Shiung Chang^d, Athanasios Vasilakos^e

^a Department of Applied Informatics, Fo Guang University, I-Lan, Taiwan

^b Institute of Computer Science & Information Engineering and Department of Electronic Engineering, National Ilan University, I-Lan, Taiwan

^c Department of Electrical Engineering, National Dong Hwa University, Hualien, Taiwan

^d Department of Computer Science & Information Engineering, National Dong Hwa University, Hualien, Taiwan

^e Department of Computer and Telecommunications Engineering, University of Western Macedonia, Greece

ARTICLE INFO

Article history:

Received 28 December 2009

Received in revised form 29 August 2010

Accepted 10 October 2010

Keywords:

Mobile ad hoc network (MANET)

Quality-of-service (QoS)

Genetic algorithm (GA)

ABSTRACT

A wireless MANET is a collection of wireless mobile hosts that dynamically create a temporary network without a fixed infrastructure. The topology of the network may change unpredictably and frequently. Therefore, multicast routing in ad hoc networks is a very challenging problem. This paper proposes a multi-constrained QoS multicast routing method using the genetic algorithm. The proposal will be flooding-limited using the available resources and minimum computation time in a dynamic environment. By selecting the appropriate values for parameters such as crossover, mutation, and population size, the genetic algorithm improves and tries to optimize the routes. Simulation results indicate its better performances compared to other methods.

© 2011 Published by Elsevier Ltd

1. Introduction

A MANET is a multi-hop wireless network composed of mobile hosts communicating with each other without the support of a fixed network infrastructure. Wireless links are formed or destroyed whenever one mobile host moves in or out of transmission ranges of other mobile hosts. Intermediate mobile hosts between two communication nodes act as routers. Thus, the mobile hosts operate both as hosts and routers in MANET. When the hosts are mobile, they are free to move randomly and organize themselves arbitrarily. The creation of routing paths is affected by the addition and deletion of mobile hosts. Thus, the topology of the wireless network changes unpredictably and frequently. As a result, unicast or multicast routing in ad hoc networks becomes a challenge due to this dynamic nature.

The basic function of Quality-of-Service (QoS) routing is to find a better path based on QoS metrics, such as end-to-end delay, jitter (also called delay variation), bandwidth guarantee, and packet loss probability. QoS routing is much more complicated than regular routing because of routing under multiple constraints. In addition, most QoS routing algorithms consider the optimization of resource utilization. The problem of QoS routing is difficult because multiple constraints often make the routing problem intractable, and the dynamic of network state makes it difficult to gather up-to-date information in a large network. Wang and Crowcroft [1] have shown that the problem of finding a path subject to constraints in QoS routing are NP-complete [1–3].

Routing problems can be divided into unicast and multicast routing. Unicast routing refers to finding a feasible path between a single source and a single receiver. On the other hand, multicast routing refers to finding a feasible tree covering

* Corresponding author at: Department of Electrical Engineering, National Dong Hwa University, Hualien, Taiwan. Tel.: +866 3 9357400x251; fax: +866 3 9354238.

E-mail addresses: ysyen@mail.fgu.edu.tw (Y.-S. Yen), hcc@niu.edu.tw (H.-C. Chao), rschang@mail.ndhu.edu.tw (R.-S. Chang).

a single source and a set of receivers. However, in MANETs where the network topology changes arbitrarily, it is a challenging problem even to provide QoS guarantee. QoS-based multicast routing is one of the critical issues which seek to find feasible routes for QoS traffic. Providing the QoS guarantee for data traffic and maximizing the utilization of the whole network on QoS-based multicast routing offer significant challenges. This gives rise to the need for an efficient multicast routing protocol which will be able to determine multicast routes satisfying the different QoS constraints.

Difference from non-linear programming methods, Genetic Algorithm (genetic algorithm) [4–9], Ant Algorithm [10,11], Fuzzy Logic (FL) [12,13] and Neural Networks (NNs) [14] are heuristic methods which use explicit rules to find feasible routes. In this paper, we propose a multi-constrained multicast routing protocol based on genetic algorithm that determines near-optimal multicast routes on demand. Based on the multi-constrained genetic algorithmic approach, our proposal will try to optimize multiple QoS parameters (e.g., end-to-end delay, bandwidth guarantee and residual bandwidth utilization) simultaneously. The genetic algorithm uses the genetic operators for optimization. By selecting the appropriate values for parameter such as crossover probability, mutation probability, and population size, the routing performance can be improved. To obtain robustness routing path and optimality is the main concern of our proposal. Simulation results demonstrate that our algorithm is capable of discovering on demand a set of QoS-based, near-optimal multicast routes within a few iterations, even with imprecise network information. From these set of routes one can choose the best possible multicast route depending on the specified QoS requirements.

The remainder of this paper is organized as follows. In Section 2 we first introduce the operation of genetic algorithm and describe the problem formulation. Section 3 discusses in detail the design concept of network mode and the operation of sequence and topology encoding. Section 4 proposes the practical operation of genetic algorithm. Section 5 shows the simulation results, analysis and discussions. Finally, the last section gives the conclusions and future work.

2. Problem formulation and genetic algorithm

2.1. Problem formulation

QoS metrics include available bandwidth, end-to-end delay, residual power, delay variance (jitter), and so on. Different QoS parameters have different kinds of property. It can be classified into three categories, such as additive, multiplicative, and minimal property. Although multiple metrics can certainly model both networks and applications more accurately, the problem is that finding a path subject to multiple metrics is inherently difficult and in many cases is proved to be NP-complete [1]. It means that it cannot be solved optimally in a reasonable time scale which is very crucial for the majority of applications in general and for delay-sensitive applications in particular. The value of a metric over the entire path can be one of the following properties: (1) additive: It can be represented mathematically as $total_metric = \sum_i metric_i$, where $metric_i$ is the metric of node i . Path delay and hop number are examples of this type of property; (2) Concave: It can be represented mathematically as $min_metric = \min_i \{metric_i\}$. Bandwidth and batter energy are two examples. They can be defined as the minimum of the available bandwidth or residual batter energy among all links or nodes on the path; (3) multiplicative: It can be represented mathematically as $1 mul_metric = \prod_i mul_metric_i$. Packet delivery rate is an example.

An undirected graph $G = (V, E)$ is defined as a set of vertices, V , and a collection of edges, E . An edge $e \in E$ connecting nodes u and v will be denoted by (u, v) , $u, v \in V$. A tree T rooted at s , where $s \in V$, has to meet multi-constraints, D, J, B , and P to be called a feasible tree. Define four constraints in a multicast routing: Path delay bound, D : upper bound on the acceptable end_to_end delay from source to destination, $\sum_{e \in p(s,v)} D(e) < D, \forall e \in E$. Delay variation bound, J : a synchronization time window between members in a multicast group, $\sum_{e \in p(s,v)} D(e) - \sum_{e \in p(s,u)} D(e) \leq J, \forall e \in E$. Available bandwidth bound, B : the minimum bandwidth of multicast tree on each link, $\min\{B(e)\} > B, \forall e \in E$. For any node n , residual power bound, $P : \min\{P(n)\} > P, \forall n \in V$.

2.2. Genetic algorithm

The genetic algorithm proposed by Holland in 1975 is derived from the ideas of natural selection and natural genetic. Genetic algorithms are different from other heuristic methods. The most important difference is that a genetic algorithm works on a population of possible solutions, while other heuristic methods use a single solution in their iterations. Another difference is that genetic algorithm is stochastic, not deterministic. Each individual in the genetic algorithm population represents a possible solution. Some individuals are selected based on the fitness value. And then, genetic algorithm imitates the nature genetic process, crossover, to exchange some of these individual genetic data randomly to generate the offspring. By repeating these processes until the best genes, which have the fittest capability, are obtained. Each individual may represent one or more chromosomes with an associated fitness value and population.

Genetic algorithm is also a searching algorithm that employs the ideas of natural selection and the genetic operators of crossover and mutation [15]. In each generation, a new population of solutions is created by exchanging and combining the information obtained from the solutions of the previous generation. In genetic algorithm, the variables of the problem are like the genes in a chromosome. A context in each bit of string is called chromosome. One gene of a chromosome represents one possible solution. In general, the main operations of genetic algorithm are encoding, initial population, evaluating fitness value, reproduction, crossover and mutation. A flowchart of a typical genetic algorithm is shown in Fig. 1.

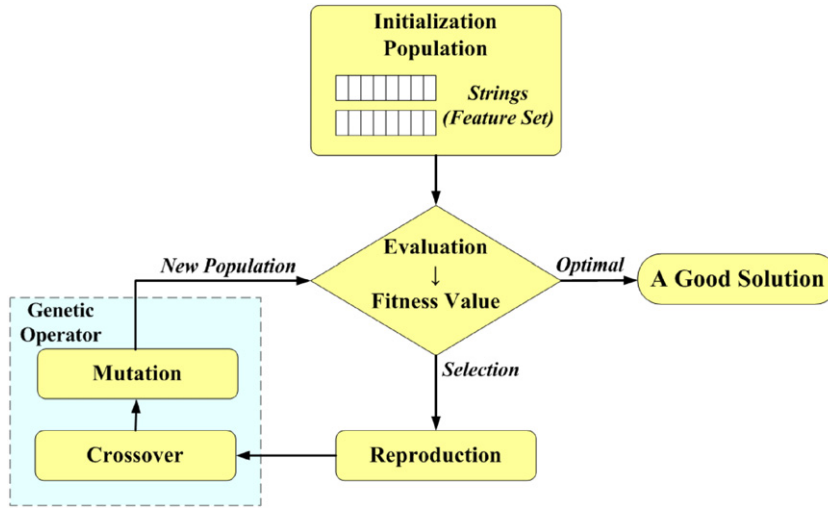


Fig. 1. The flowchart of a simple genetic algorithm.

The selection is an evolutionary operator in the genetic algorithm. It is also the policy for selecting the fittest individuals from the population. It will need a method to calculate this fitness. The fitness method simply calculates the amount of free space each individual/solution offers. The best is selected for further iteration.

There are two main genetic operators in genetic algorithm. The first is the crossover and the other is the mutation. These two genetic operators allow the chromosomes to search for the global optimum through an evolutionary manner. The crossover is the method for combining those selected individuals into new individuals. The crossover splits up the “parent” individuals and recombines them. It is also one of the genetic operators in which genes of two chromosomes are exchanged and the genotypes of two selected parents are merged to yield two new offspring. Two chromosomes with greater fitness values are picked from the chromosomes pool. The starting point and length of the portion to be exchanged are randomly selected. The two new offspring are created and put back into the chromosomes pool.

The mutation simply adds “genes” to the individuals (usually called “children”). Mutation introduces new genetic structures in the population by randomly modifying some of the genes, such that the search algorithm can escape from the local optimum and avoid the genetic algorithm from converging too fast. In other words, mutation operation gives genetic algorithm an opportunity to search for new and more feasible chromosomes in new areas of the solution spaces. After the mutation operation, the multicast tree will be modified because of mutation operator can destroy the tree structure and outgoing degree constraints.

A genetic algorithm has a number of advantages. It can quickly scan a huge solution set. Bad proposals do not affect the end solution negatively as they are simply discarded. The inductive nature of the genetic algorithm means that it does not have to know any rules of the problem. It works by its own internal rules. This is very useful for complex or loosely defined problems.

3. Network model and encoding

3.1. The flooding-limited mechanism for multicast routing

Assume that a network is represented as a weighted graph $G = (V, E)$, where V denotes the set of vertices as mobile hosts, and E denotes the set of edges as communication links connecting the mobile hosts. $|V|$ and $|E|$ denote the number of mobile hosts and links on the MANETs, respectively. In $G(V, E)$, considering a QoS constrained multicast routing problem from a sender to multi-receiver hosts. We define multi-constrained spanning tree to be a multicast tree $T = (V_T, E_T)$ rooted at s , where $V_T \subseteq V$, $E_T \subseteq E$, $s \in V_T$. Associated with each link are parameters that describe the current status of the link. Let SH represent the sender host, denoted by s . RH represents a receiver host. A multicast delivery tree is a tree rooted at a sender host s and spans M , where $M = M^* \cup \{s\}$, $M^* = \{r_1, r_2, \dots, r_m\}$ is the set of receiver hosts, m is the number of receiver hosts in multicast delivery tree T . In a delivery tree $T(T \in G)$ rooted at SH and spanning all of the mobile hosts in M , a path from the sender host SH to the receiver host RH is denoted by $p(s, r)$.

Let R^+ be the set of real positive numbers. For a multicast delivery tree $T = (V_T, E_T)$, any link $e \in E_T$, we can define some QoS metrics: delay variation function, $jitter(e) : E_T \rightarrow R^+$; available bandwidth function $bandwidth(e) : E_T \rightarrow R^+$. Similarly, for any mobile host $n \in V_T$, we can also define a metric: residual battery energy function $power(n) : V_T \rightarrow R^+$. Then for a path $p(s, r)$, we define the following three QoS parameters:

$$jitter_{\max}(p(s, r)) = \max\{jitter(e)\}, \quad \forall e \in E_T \quad (1)$$

$$\text{bandwidth}_{\min}(p(s, r)) = \min\{\text{bandwidth}(e)\}, \quad \forall e \in E_T \quad (2)$$

$$\text{power}_{\min}(p(s, r)) = \min\{\text{power}(n)\}, \quad \forall n \in V_T. \quad (3)$$

Let J_T be the maximum delay variation constraint, B_T be the minimum available bandwidth constraint, and P_T be the minimum power constraint. The path from the sender to receiver has a measured value of jitter_{\max} , bandwidth_{\min} , and power_{\min} that should be limited by the QoS constraint J_T , B_T , and P_T . The QoS-based routing problem should satisfy some QoS constraints.

In the above QoS constraints, the jitter is a convex metric, the bandwidth and power are concave metrics. It can be described by a triple (J_T, B_T, P_T) . Before initiating, the values of (J_T, B_T, P_T) are defined by the sender host.

The sender host SH sends data to a set of receiver hosts. In the beginning, SH should activate the join discovery. We use a flooding-limited mechanism [16–19] to broadcast QRREQ packets to find a suitable routing path. When SH broadcasts the QRREQ packet, all its neighbors will receive the QRREQ packet and check the delay variation, available bandwidth and residual battery energy for conforming to the QoS constraints (J_T, B_T, P_T) . If a neighbor accepts the QRREQ packet, it will add the route entry into its routing table and set the route state as explored. However, this entry will be deleted if a QRREP packet for this route is not received within a time period. This time period is defined as:

For any mobile host $e \in E_T$, let $\text{delay}(e): V_T \rightarrow R^+$ denote the time from the sender host SH to receiver host RH . Then define

$$\text{delay}_{\max}(p(s, r)) = \max_{p(s, r)} \sum_{e \in P(s, r)} \text{delay}(e), \quad \forall e \in p(s, r). \quad (4)$$

Let D_T be the end-to-end delay constraint and its value is defined by the sender host before initiating. The delay function $\text{delay}(e)$ represents the real value of end-to-end delay on each edge. The path from the sender to receiver has a measured value delay_{\max} that should be limited by the QoS constraint D_T .

The route establishment is done using limited flooding as follows. When we establish a route and the destination is not in the source's neighborhood list, the source broadcasts a route request packet, which includes the requested jitter jitter_{\max} , available bandwidth bandwidth_{\min} , residual battery energy power_{\min} and end-to-end delay delay_{\max} . Upon receiving the join request packet, a bandwidth, delay and residual battery energy admission decision is made at each host. If the request is accepted, host i will add a route entry in its route cache with status explored and rebroadcast the request to the next hop. Host i will only remain in explored status for a period of $2D_T - \sum_{e \in P(s, i)} \text{delay}(e)$. $\sum_{e \in P(s, i)} \text{delay}(e)$ denote the total delay from the SH to host i ; If no reply arrives at the explored host in time, the route entry will be deleted and all late reply packets will be ignored.

As shown in Fig. 2, assume host v_1 is the source and others are the receivers. In the beginning, host v_1 should activate the route discovery using a flooding-limited mechanism to broadcast QRREQ packet to find a suitable routing path. When host v_1 broadcasts the QRREQ packet, all its neighbors will receive the QRREQ packet and check whether the delay variation, the bandwidth, and the power energy constraints are satisfied. There may have multiple trees to satisfy the constraints. For example, $T_1\{v_1 \rightarrow v_2 \rightarrow v_4 \rightarrow \infty\}$, $\{v_1 \rightarrow v_3 \rightarrow v_7 \rightarrow \infty\}$, $\{v_1 \rightarrow v_9 \rightarrow v_8 \rightarrow v_6 \rightarrow \infty\}$, and $T_2\{v_1 \rightarrow v_2 \rightarrow \infty\}$, $\{v_1 \rightarrow v_3 \rightarrow v_5 \rightarrow v_4 \rightarrow \infty\}$, $\{v_1 \rightarrow v_9 \rightarrow v_8 \rightarrow \infty\}$, and so on.

The flooding-limited mechanism for multicast routing uses available detailed information about the connectivity and conditions found in the network. Hence it increases the chances that a mobile host will be able to generate a route that meets a specified set of constraints. In the next section, we will show how to solve the QoS multicast routing problem using the genetic algorithm.

3.2. Multi-constrained model for GA

In a multicast tree T , the total number of the mobile hosts is $|V|$; the total number of links for connecting mobile host to host is $|E|$; the root spans M , $M \subseteq V$; the set of receiver is denoted by $M' \subseteq \{V - \{s\}\}$, which is a subset of V . The total delay time means the summation of the node-to-node delay from the sender needs for a packet to the receiver. The total power means the summation of the residual battery energy that the sender needs for a packet to the receiver. The total delay (TD) and residual battery energy (RBE) functions associated with a multicast tree are described as:

$$TD = \sum_{e \in M} \text{delay}(e) \quad (5)$$

$$RBE = \sum_{n \in M'} \text{power}(n). \quad (6)$$

Once the multicast tree is computed, the multicast source transmits a packet to the multicast group in a manner similar to any source routing protocol. Since an ad hoc network has no fixed infrastructure or central administration, it is a dynamically reconfigurable wireless network. In the following, we use the sequence and topology encoding method [20,21] to code a multicast tree so that each node in the corresponding multicast tree does an intelligent action for delivering a packet.

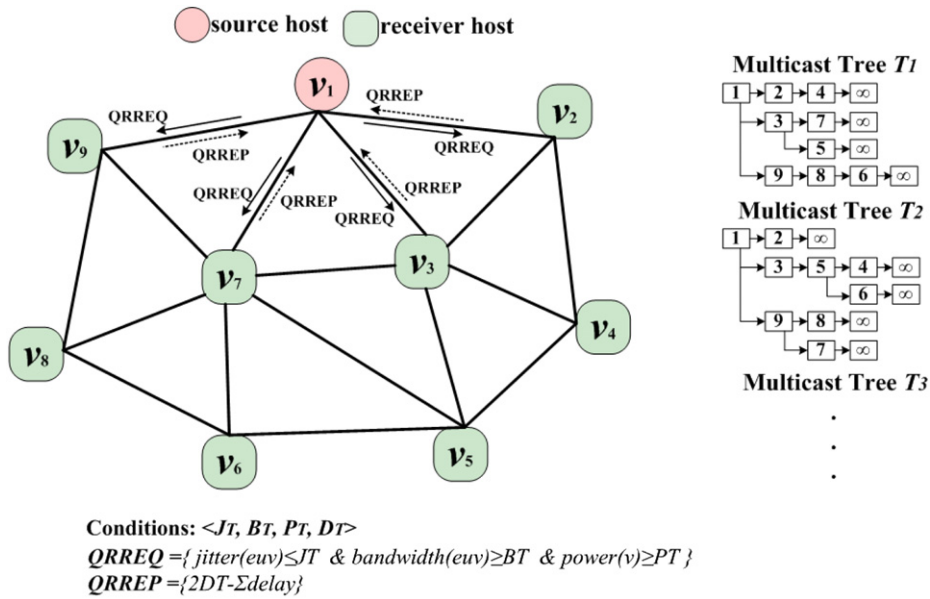


Fig. 2. The network topology of an example.

3.3. Extended tree encoding in GA

The most important step in genetic algorithm is to define an encoding representation. Better encoding approach will be easy for crossover and mutation operation. In this paper, we extend the sequence and topology encoding method [20–22] which maps candidate solutions as a symbol (i.e., integer strings). These strings are known as chromosomes and each chromosome represents a feasible solution respectively. We then have to define a fitness function. The greater value the fitness value is, the higher chance it will be selected to survive and replicate. The genetic algorithm creates a population of genomes (i.e., chromosomes) first. It then applies crossover and mutation to the individuals in the population to generate new individuals. It uses various selection criteria so that it picks the best individuals for mating.

In [20], a tree must be position indexed before it can be encoded with sequence and topology encoding. An indexed tree can be encoded with n -element strings \mathbf{s} and \mathbf{t} where n is the number of nodes of the indexed tree. The string \mathbf{s} is a sequence encoding the nodes in the tree. The string \mathbf{t} is used to encode the topology of the tree. The procedure of encoding used to convert the multicast tree into the sequence and topology encoding is described below. Given two arrays $\mathbf{s}[1 : n]$, $\mathbf{t}[1 : n]$, define integer i as the position index. We use the depth first search order for nodes. Thus, put the root of multicast tree T into the first element of string \mathbf{s} , $\mathbf{s}[1]$. $\mathbf{t}[i]$ represents the position index of the father of host $\mathbf{s}[i]$. $\mathbf{t}[1]$ is always set to 0 because the root host has no predecessor. For all other nodes, put the node number at index i of the tree into string \mathbf{s} , that is, $\mathbf{s}[i] \leftarrow \text{host at position index } i$. And put the position index of the father of node i into string \mathbf{t} , $\mathbf{t}[i] \leftarrow \text{position index of the predecessor of host } \mathbf{s}[i]$. The procedure of decoding to convert sequence and topology code into its corresponding tree is in the following. First, set $\mathbf{s}[1] \leftarrow 1$ be the root of multicast tree T . For $i = 2$ to n , set a link $\mathbf{s}[i]$ to $\mathbf{s}[\mathbf{t}[i]]$. Fig. 3(a) and Fig. 4(a) have the sequence and topology codes shown in Fig. 3(b) and Fig. 4(b). Thus, we have string $\mathbf{s}_1 = \{1, 9, 8, 6, 3, 7, 5, 2, 4\}$ and $\mathbf{t}_1 = \{0, 1, 2, 3, 1, 5, 5, 1, 8\}$, and string $\mathbf{s}_2 = \{1, 9, 8, 7, 3, 5, 6, 4, 2\}$ and $\mathbf{t}_2 = \{0, 1, 2, 2, 1, 5, 6, 6, 1\}$. \mathbf{t}_1 and \mathbf{t}_2 are used to encode the topology of the tree. The interior nodes of the multicast tree are selected from strings \mathbf{s}_1 and \mathbf{s}_2 by the order of the index numbers, showing in strings \mathbf{t}_1 and \mathbf{t}_2 .

4. The genetic algorithm

The genetic algorithm is often applied to the optimization problem. At first, by encoding all parameters, n initial strings called chromosomes are generated. A process is performed by evaluating these individuals using the fitness function to obtain the fitness values. An individual with a higher fitness value has a higher probability to be selected into the mating pool. This process is called selection. After the selection process, these selected individuals further perform the crossover process and the mutation process randomly. These processes are repeated until an optimal solution is obtained or the stop condition is satisfied.

4.1. Elitist selection operation

The fitness value indicates the quality of an individual. In other words, by measuring the fitness value we can obtain the quality or rank of a chromosome. Starting with a set of random chromosomes, a process of evolution is simulated. After a

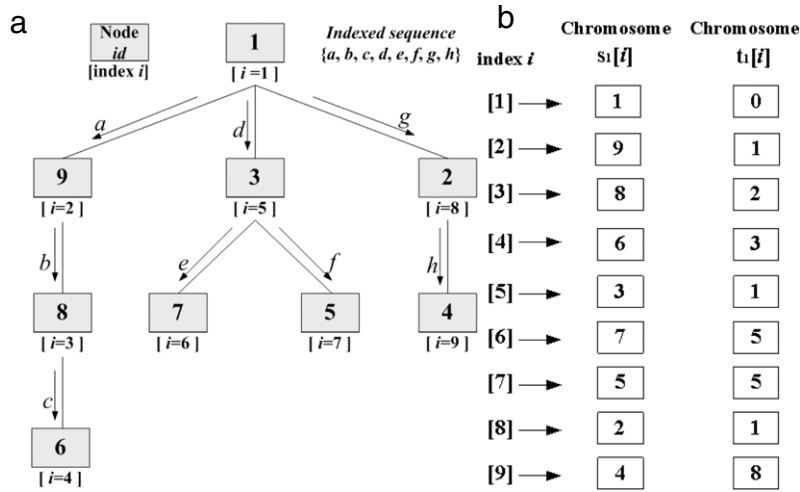


Fig. 3. (a) Indexed multicast tree T_1 . (b) Extended sequence and topology encoding to code tree T_1 .

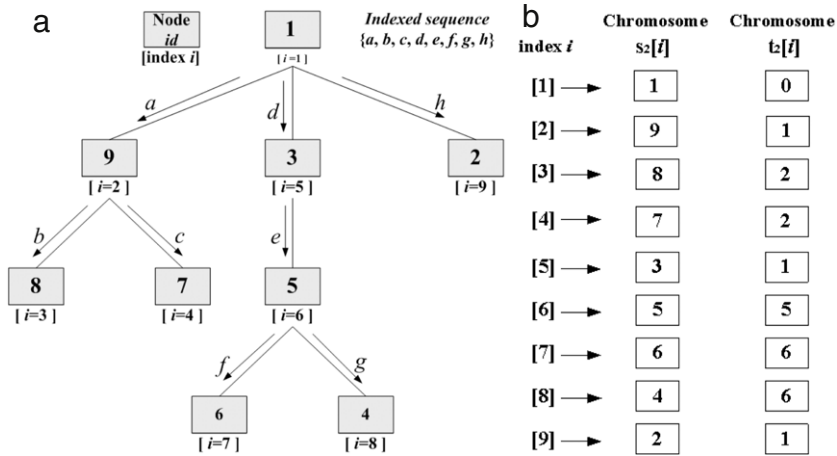


Fig. 4. (a) Indexed multicast tree T_2 . (b) Extended sequence and topology encoding to code multicast tree T_2 .

number of generations, chromosomes with higher fitness values emerge and these correspond to better solutions. There are two ways for generating and maintaining the feasible solutions. One is the repair function, and the other is the penalty value. In this paper, we calculate the total delay and residual battery energy of all the links and nodes in a multicast tree. We adopt the repair function that a node with lower residual battery energy will be replaced by another with a higher value. The penalty function is to prevent the best solution from falling into an infeasible region. In other words, it penalizes the infeasible solution.

The fitness function f_T uses the multiplicative form

$$f_T = f_p * f_d * f_r \quad (7)$$

where T denotes a candidate multicast tree, and f_p is a penalty function.

Penalty function

$$f_p = \sum_{e \in p(s,r)} \Delta(\text{delay}(e) - \text{delay}_{\max}) \quad (8)$$

delay_{\max} is the maximum delay of a link. Let $\delta = \text{delay}(e) - \text{delay}_{\max}$, then $\Delta(\delta) = \begin{cases} 1, & \delta < 0 \\ \lambda, & \delta > 0 \end{cases}$, where $\Delta(\delta)$ is penalty function, $p(s, r)$ is a route from source node s to the receiver r , and λ is a penalty degree. In general, $0.01 < \lambda < 0.5$. When chromosome is falling into a feasible region, $f_p = 1$; and if the chromosome is falling into an infeasible region, $0 \leq f_p < 1$. In this paper, we used the multiplicative type. The function of end-to-end delay is:

$$f_d = \frac{1}{TD}. \quad (9)$$

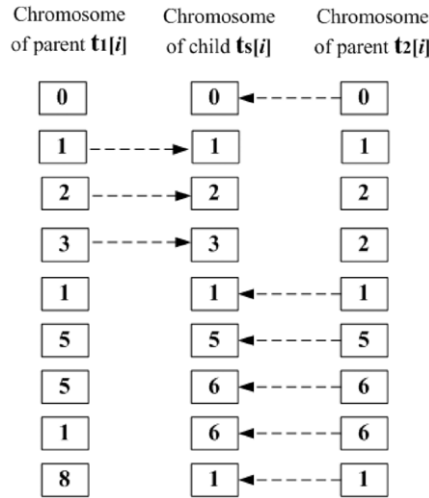


Fig. 5. The crossover of T chromosome.

f_r is the repair function wherein a node with a lower residual battery energy would be replaced by another with a higher value. In this way, outgoing degrees of nodes will be restricted during the mutation operation. The mutation for the repair function is $f_r = RBE_{M'}$. This paper attempts to extend the lifetime to the maximum by calculating the total residual battery energy of a multicast tree with all the nodes. To suit our condition, we made a little modification in the formula of the cost function, so that the total residual battery energy of a multicast tree is defined as follows:

$$RBE = \sum_{n \in M'} power(n), \quad \text{where } power(n) = \omega * \left(\frac{P_c(n)}{P_{full}(n)} \right)^\kappa \quad (10)$$

where $P_c(n)$ is residual battery capacity of node n , $P_{full}(n)$ is the full battery capacity of node n . ω is the transmission power of node n . κ is positive weighting factor, which is inversely proportional to node's degree. Obviously, the maximum lifetime of the multicast tree is controlled by a node n with a lower residual battery energy. When the node n has the smallest residual battery energy, the multicast routing will first have an interruption at this node.

Besides satisfying the conditions in Eqs. (7)–(10), the route selection is based on fitness value. The fitness function f_r is to select the best route. By maximizing the fitness value (Eq. (7)), we try to minimize the delay time value and maximize the residual power. This means that a packet from source to receivers is transmitted with a small delay and a long lifetime.

4.2. Genetic crossover and mutation operation

We revise [23] for the crossover of the T chromosome. The parents are crossed over using the two-point crossover to yield two new chromosomes. As shown in Fig. 5, the two chromosomes are t_1 and t_2 . To create an offspring t_s using the two-point crossover, insert the offspring into the population according to the following rule. (1) The genome would be selected from the two crossover points and inherited by the offspring. (2) Two crossover points are selected, sub-string from beginning of chromosome to the first crossover point is copied from second parent (e.g., chromosome t_2). The part from the first to the second crossover point is copied from the first parent (e.g., chromosome t_1), and the rest is also copied from the second parent (e.g., chromosome t_1).

The gene of the S chromosome has a unique and unrepeatable characteristic, so we slightly modify the crossover strategy [24]. As shown in Fig. 6(a), we assume chromosome s_1 has a higher priority in heredity. The priority genome is inherited by offspring from the selected two crossover points (e.g., 1 and 3). Consequently, the remaining genome in the offspring will be inherited from chromosome s_2 . Thus, a new tree after topology crossover is shown in Fig. 6(b).

As mentioned previously, balancing the energy consumption in a multicast tree can extend the maximum lifetime for the multicast service. In our mutation operation, we assume that $RMB_{M'}$ is a mutation factor for prolonging the energy consumption in a multicast tree. An intermediate host (IH) with a lower residual battery energy in the t chromosome will be replaced by a leaf host with a higher residual battery energy.

Let $\deg(v)$ denote the degree of a node v . The larger the $\deg(v)$, the more is the energy consumption. We use the genetic algorithm approach to restrain $\deg(v)$. Meanwhile, when considering the problem of equally balancing the consumption of the battery energy, we utilize the Mutation-Replace Function wherein the term residual battery energy acts as a mutation factor to control the mutation process. The mutation process adopted of [17] is described below.

The T chromosome of the offspring is shown in Fig. 5. There are three entries with $t[i] = 1$, one with $t[i] = 2$, one with $t[i] = 3$, one with $t[i] = 5$, two with $t[i] = 6$. Assume the number of genes in the T chromosome is represented by the

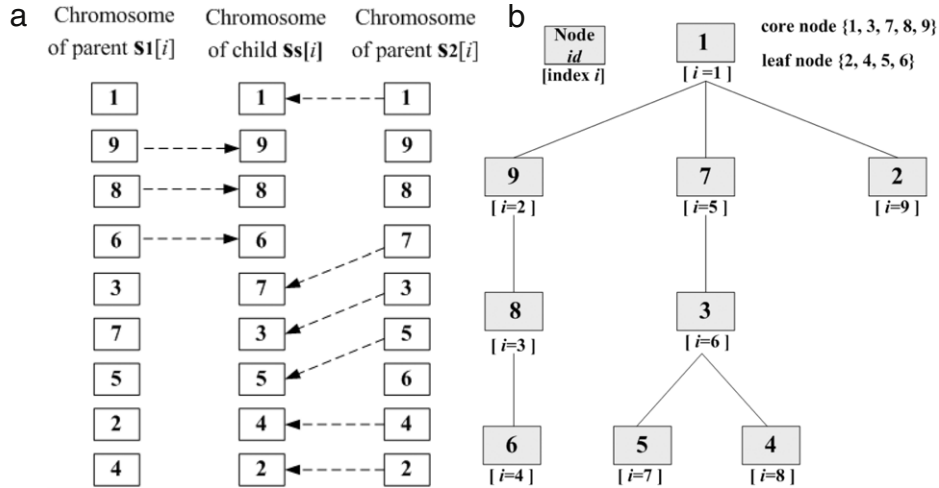


Fig. 6. (a) The crossover of S chromosome. (b) New tree after topology crossover.

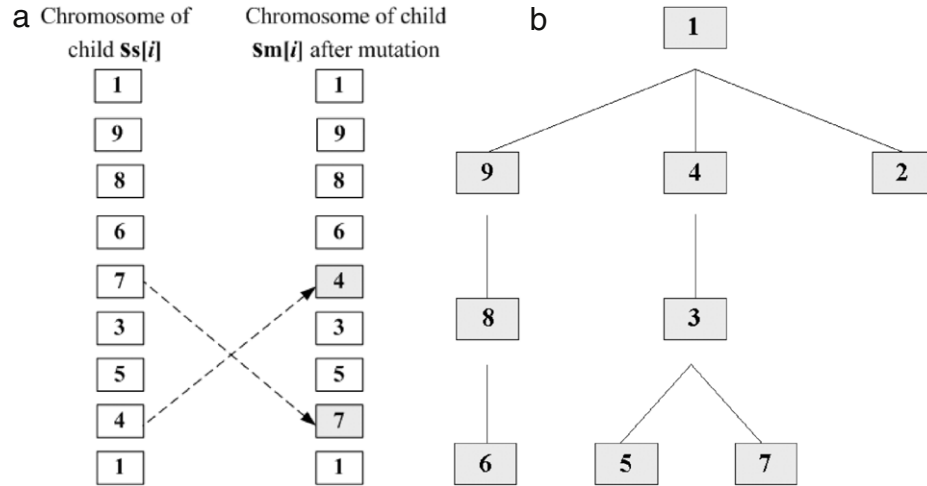


Fig. 7. (a) The mutation process of chromosome s. (b) The new tree of mutation.

$f(n)$ of the downstream of the father node. We let $f(n) \geq 1$ node as the core node and where its index of position is i' . Let $f(n) = 0$ node be the leaf node where its index of position is $i - i'$. For example, suppose the T chromosome of offspring after the mating process, and the S chromosome of the offspring are as shown in Figs. 10 and 11, respectively. The gene {1, 2, 3, 5, 6} of the T chromosome is the index value of the position $i' = \{1, 2, 3, 5, 6\}$ for the father node and hence, the responding nodes $s[i'] = \{1, 3, 7, 8, 9\}$ of i' are called core nodes.

As shown in Fig. 7(a, b), the Leaf node $i - i' = \{4, 7, 8, 9\}$ associated with the gene that is not shown in the T chromosome represents the index of position. Thus, the alternative nodes $s[i - i'] = \{2, 4, 5, 6\}$ of $i - i'$ are called the leaf nodes.

The mutation operation of the S chromosome would allow the node with a lower energy to be replaced by a leaf node with a greater energy. For instance, the gene {7} with the lowest energy among the core nodes $s[i'] = \{1, 3, 7, 8, 9\}$ will be replaced by a gene {4} with the highest residual battery energy which is in the set of leaf nodes $s[i - i'] = \{2, 4, 5, 6\}$.

Using a tree encoding in the genetic algorithm, the loops can be avoided. After the mutation operation, the degree of the node will be automatically reduced. In this encoding method, the data structure of the multicast tree is adopted as the structure of the chromosome. Each pair of s and t chromosomes represents a multicast tree. Obviously, the advantages of the extended sequence and topology approach are not just to save on the memory space of the encoding, but also to cut down on the decoding operation. The extended sequence and topology encoding will be presented to show how well it works for those problems in genetic algorithms. As a whole, the proposed extended sequence and topology encoding is easy for genetic operations.

Table 1
The parameters in relation to number of nodes.

Functions	Values
Simulation time	600 (s)
Simulation rectangular region	$1000 \times 1000(\text{m}^2)$
Number of mobile hosts	100 (nodes)
Membership groups	10–20 (members)
Pause time	60, 120, 180, 240, 300, 360, 420, 480, 540, 600 (s)
Maximum node speed	2–20 (m/s)
Transmission radius	200–250 (m)
Mobility	random waypoint with various pause time
Random waypoint pause time	10–250 (s)
Power parameter	Ref. [22], Table II

5. E² Multicasting and simulation results

5.1. E² Multicasting

On-Demand Multicast Routing Protocol (ODMRP) is a mesh-based multicast protocol in which a mesh of nodes for forwarding packets is created between the senders and receivers. The mesh is created using the forwarding group concept. The main disadvantage with ODMRP is the excessive overhead incurred in keeping the forwarding group current and the global flooding of the Joint-Request packets. ODMRP uses the soft state approach for maintaining the mesh, and it exhibits robustness. However, this robustness is achieved at the expense of high control overhead. Also, the same data packet propagates through more than one path to a destination, thereby reducing the multicast efficiency. Another disadvantage is that this multicast routing does not support an energy-conserving mechanism for increasing the lifetime of the network.

Since the mobile hosts in MANETs are battery operated, it is important to find energy-conserving schemes and protocols that optimize the use of battery power in order to increase the lifetime of the network. Some energy-efficient multicasting protocols (E2MRP) for MANETs [25–27] have been proposed. In [19], E²MRP is a modification of ODMRP that takes the power consumption into consideration. It also uses the concept of a forwarding group to forward the data packets. The protocol operates in two phases. The first phase uses a heuristic called minimum energy consumed per packet (MECP). The main idea is to minimize the total end-to-end power consumed in sending a packet from the source to its final destination. The second phase uses a heuristic called minimum maximum node cost (MMNC). This heuristic gives more importance to the power available at nodes along the route.

5.2. Simulation results

In this thesis, we simulate the QoS multicast routing protocol in ad hoc wireless networks by using the network simulator ns2 to be the platform and the flow controlled algorithm uses a GA simulated by MATLAB 7.0. First, the GA and control parameters are set. The generation is set $\text{gen_no} = 100$. The other parameter values are set as follows: population size is $\text{popuSize} = 500$; mutation probability is $\bar{P}_m = 0.05$; and crossover probability of is $\bar{P}_c = 0.8$. Thus, we utilize the GAs to adjust the QoS multicast routing protocol function's domain of discourse range. In the simulation ns2 environment, we transmit the constant bit rate (CBR) traffic source with 512-byte data packets at one second intervals. The source host to destination pairs is spread out randomly over the network. The number of source hosts is varied in the simulation. In the mobility model, we apply the random waypoint model. The node transmission range is 200–250 m. Each mobile host randomly selects its position and the mobile hosts move around a rectangular region of 1000 m by 1000 m. Other parameters are shown in Table 1.

During each generation, we use some measure to evaluate the fitness of chromosomes. Fitness is estimated based on the multi-objective function in most cases of optimization problems. Therefore, a better chromosome has larger fitness value. The fitness value of each chromosome is defined as Eqs. (5)–(7). To compare our extended tree encoding, ST encoding and Prüfer number encoding, Figs. 8 and 9 illustrate the measured results of the fitness values in various generations. The results obtained after running 100 generations show that the extended tree encoding clearly outperforms Prüfer number encoding, because our encoding mechanism data structure is simpler than Prüfer number encoding. Obviously, our extended tree encoding has a higher fitness value when compared to ST encoding and Prüfer number encoding.

There is not a definite crossover rate and mutation rate used for finding the best solution of the GA. Generally, crossover rate is set between 0.5 and 1 and the mutation rate is set between 0.01 and 0.08. Thus we consider several combinations of a variety of mutation and crossover rates, and the trends of average solutions in generations solved by the GA. Fig. 8 shows the convergences in the crossover range of 0.1, 0.5 and 0.8 when the mutation rate is 0.05. We obtained the convergence under different crossover probabilities. Obviously, the convergence speed is related to the crossover probability. The higher the crossover probability, the higher the choices of genes are in the chromosome. In other words, when the crossover probability is 0.8 and 81 generations have been run, there is a tendency to converge to 1. In the same way, Fig. 9 shows the convergence at the mutation probability ranges of 0.01, 0.05, and 0.09, when the crossover probability is 0.5. When the mutation probability

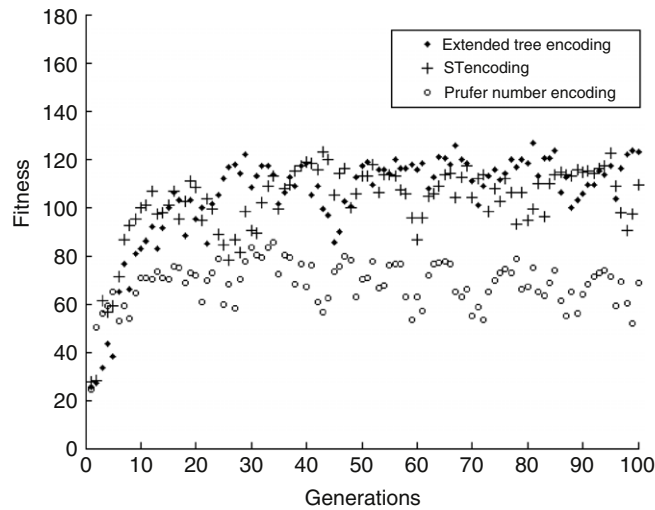


Fig. 8. The fitness values in various generations (x scale: linear).

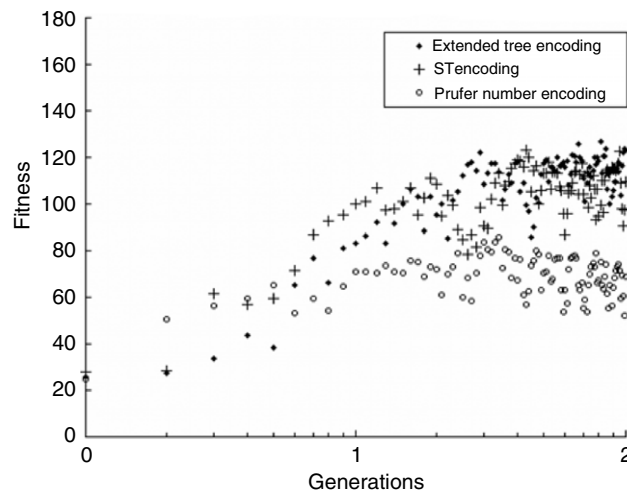


Fig. 9. The fitness values in various generations (x scale: log).

is 0.01, and 77 generations have been run, there is a tendency to converge to 1. When the mutation probability is set at 0.05, the convergence speed responds to 91 generations. To determine how fast genetic algorithm convergence occurs and to calculate how long the multicast service lifetime is extended, we obtain a set of relevant parameters values when the mutation rate is 0.05 and the crossover rate is 0.8.

In Figs. 12 and 13, the relationship between the maximum lifetime in the multicast tree vs. pause time is shown. The lifetime of a multicast tree is the duration from the formation of the tree to the time when the node fails due to battery energy exhaustion. We assume that the energy consumed is proportional in MANET, that each simulation is processed for a long-term period and that an average value is collected every 60 s. According to the results, the ODMRP gave the worst performance, as expected, while our proposed protocol gave the best performance. Since the flooding mechanism can eliminate undesirable nodes and links, the multicast-constrained technique can compute reservation higher residual battery energy, and the mutation operation can allow lower energy to be replaced with large energy. Therefore, our new protocol is able to prolong the lifetime and obtain better efficiency when compared to the ODMRP and E²MRP.

6. Conclusions and future work

This paper proposes a Flooding-Limited and Multi-Constrained QoS multicast routing method using the GA. The proposal is flooding-limited using the available resources and minimum computation time in a dynamic environment. By selecting the appropriate values for parameters such as crossover, mutation, and population size, the GA improves and optimizes the

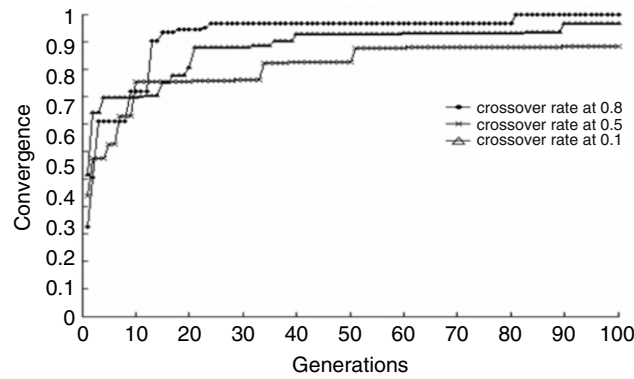


Fig. 10. The convergence in crossover rate.

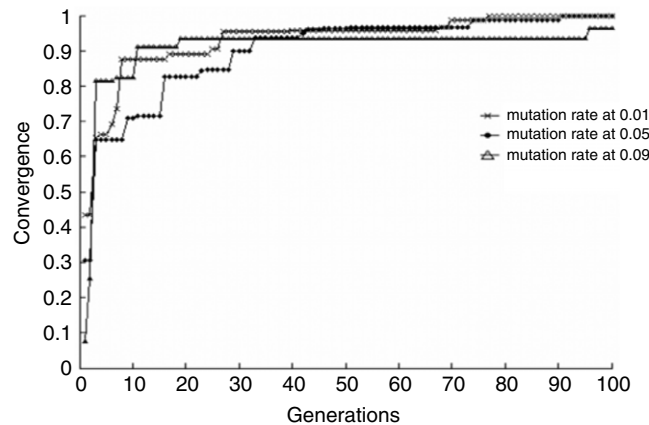


Fig. 11. The convergence in mutation rate.

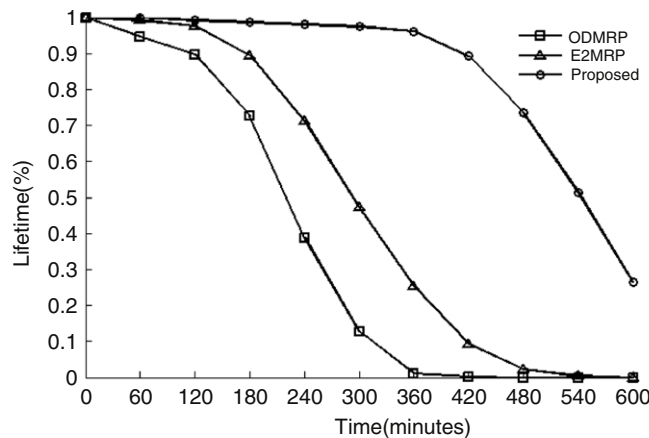


Fig. 12. Lifetime vs. simulation time (10 members).

routes. Simulation results indicate its better performance compared to other methods. Our proposed GA has the following features:

- (i) A limited flooding mechanism that effectively decreases the search space and eliminates the probability of selecting undesirable nodes and links.
- (ii) An extended ST (sequence and topology) encoding method that provides a tree-based structure coding scheme so that the coding and decoding process can be simplified.
- (iii) A modified fitness function that allows the proposed GA to be applied to other multicast problems, such as computing multicast-constrained multicast trees with low total delay time and higher residual battery energy.

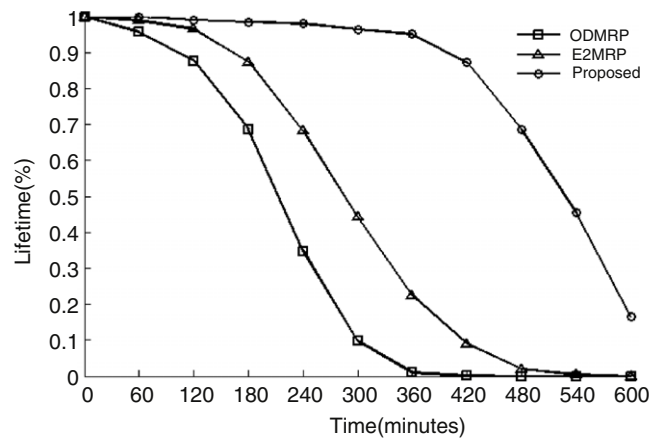


Fig. 13. Lifetime vs. simulation time (20 members).

- (iv) A two-point crossover, which is the least disruptive recombination operator among the standard operators; it is best suited for mixing genes of two parental chromosomes.
- (v) A chromosome mutation operation that allows the core node with a lower energy to be replaced by a leaf node with a greater energy.

There are several issues for future research. For example, to devise various route discovery and recovery mechanisms are more robust with respect to various network fluctuations and failures. Also, more research is needed to determine how to make the cost function more precise. Furthermore, by making the intermediate nodes calculate multiple metrics for each received and rebroadcast connection-request message, the further research is needed to reduce works on the intermediate nodes and improve the design principle of routing protocols. Finally, additional tests of the proposed algorithm on other possibly more comprehensive comparisons with other non-linear programming approaches, such as ant colony optimization algorithm (ACO), worm algorithm (WA), fuzzy logic (FL) or neural networks (NNs), are required to provide a detailed.

References

- [1] Z. Wang, J. Crowcroft, Quality-of-service routing for supporting multimedia applications, *IEEE Journal on Selected Areas in Communications* 14 (7) (1996) 1228–1234.
- [2] B.L. Sun, S.C. Pi, C. Gui, et al., Multiple constraints QoS multicast routing optimization algorithm in MANET based on GA, *Progress in Natural Science* 18 (3) (2008) 331–336.
- [3] J. Huang, Y. Liu, MOEAQ: A QoS-aware multicast routing algorithm for MANET, *Expert Systems with Applications* 37 (2) (2009) 1391–1399.
- [4] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989.
- [5] F. Xiang, L. Junzhou, W. Jieyi, et al., QoS routing based on genetic algorithm, *Computer Communications* 22 (15–16) (1999) 1392–1399.
- [6] L. Barolli, A. Koyama, N. Shiratori, A QoS routing method for ad-hoc networks based on genetic algorithm, in: *International Workshop on Database and Expert Systems Applications (DEXA'03)*, September 2003, pp. 175–179.
- [7] Wook Ahn Chang, R.S. Ramakrishna, A genetic algorithm for shortest path routing problem and the sizing of populations, *IEEE Transactions on Evolutionary Computation* 6 (6) (2002) 566–579.
- [8] M.S. Zahrani, M.J. Loomes, J.A. Malcolm, et al., Genetic local search for multicast routing with pre-processing by logarithmic simulated annealing, *Computers and Operations Research* 35 (6) (2008) 2049–2070.
- [9] Y. Liu, J. Huang, A novel genetic algorithm and its application in TSP, in: *The 2008 IFIP-International Conference on Network and Parallel Computing*, June 2008, pp. 263–266.
- [10] Y.P. Liu, M.G. Wu, J.X. Qian, The distributed multicast routing scheme with delay constraint using ant colony optimization, in: *The Sixth World Congress on Intelligent Control and Automation (WCICA 2006)*, vol. 1, June 2006, pp. 3062–3066.
- [11] X. Yan, L. Wang, L. Li, Ant agent-based multicast routing with QoS guarantees, in: *1st International Symposium on Pervasive Computing and Applications (SPCA'06)*, August 2006, pp. 279–284.
- [12] C.G. Yue, Y.W. Jing, Uncertain parameters for multicast routing based on a fuzzy number genetic algorithm, in: *Sixth International Conference on Intelligent Systems Design and Applications (ISDA'06)*, vol. 1, October 2006, pp. 903–907.
- [13] X. Wang, N. Gao, G. An, M. Huang, A fuzzy integral and microeconomics based QoS multicast routing scheme in NGI, in: *The Sixth IEEE International Conference on Computer and Information Technology (CIT'06)*, September 2006, pp. 106–106.
- [14] B.P.V. Kumar, P. Venkataram, Reliable multicast routing in mobile n networks: a neural-network approach, *IEE Proceedings-Communications* 150 (5) (2003) 377–384.
- [15] M. Gen, R. Cheng, *Genetic Algorithms and Engineering Design*, John Wiley & Sons, New York, 1997.
- [16] Y.-S. Yen, H.-C. Chang, R.-S. Chang, H.-C. Chao, Routing with adaptive path and limited flooding for Mobile ad hoc networks, in: *International Symposium on Wireless Pervasive Computing (ISWPC'06)*, January 2006.
- [17] G. Zhou, M. Gen, T. Wu, A new approach to the degree-constrained minimum spanning tree problem using genetic algorithm, *Proceedings of IEEE International Conference on System, Man, and Cybernetics* 4 (1996) 2683–2688.
- [18] S.-Y. Tseng, Y.-M. Huang, C.-C. Lin, Genetic algorithm for delay and degree-constrained multimedia broadcasting on overlay networks, *Computer Communications* 29 (17) (2006) 3625–3632.
- [19] Y. Xiao, M. Guizani, Optimal paging load balance with total delay constraint in macrocell-microcell hierarchical cellular networks, *IEEE Transaction on Vehicular Technology (IEEE TVT)* 55 (5) (2006) 2202–1309.
- [20] Y.M. Huang, T.C. Chiang, A sequence and topology encoding for multicast protocol (STMP) in wireless ad hoc networks, in: *Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies*, 2003, pp. 351–355.

- [21] Y.-S. Yen, Y.-K. Chan, Y.-C. Liaw, H.-C. Chao, Energy-Aware QoS multicast routing protocol based on genetic algorithm for the mobile ad hoc networks, *Journal of Internet Technology* 7 (2) (2006) 131–137.
- [22] S.-Y. Tseng, Y.-M. Huang, C.-C. Lin, ST encoding: an efficient scheme for encoding trees in genetic algorithms, *Journal of Internet Technology* 6 (1) (2007).
- [23] M. Gen, K. Ida, J. KIM, A spanning tree-based genetic algorithms for bicriteria topological network design, in: *Proceedings of IEEE International Conference on Evolutionary Computation*, 1998, pp. 15–20.
- [24] A.T. Haghighat, K. Faez, M. Dehghan, GA-Based heuristic algorithms for QoS based multicast routing, in: *Knowledge-based Systems*, vol. 16, 2003, pp. 305–312.
- [25] T.-C. Chiang, C.-H. Liu, Y.-M. Huang, A Near-optimal multicast scheme for mobile ad hoc networks using a hybrid genetic algorithm, *Expert Systems With Applications* 33 (3) (2007) 734–742.
- [26] S. Ci, M. Guizani, H.H. Chen, H. Sharif, Self-regulating network utilization in mobile ad-hoc wireless networks, *IEEE Transaction on Vehicular Technology (IEEE TVT)* 55 (4) (2006) 1302–1310.
- [27] B. Sun, H. Jiang, S. Cheng, Y. He, Multicasting along energy-efficient meshes in mobile ad hoc networks, in: *Proceedings of IEEE Wireless Communications and Networking Conference*, vol. 2, 2002, pp. 807–811.