Applied Soft Computing 11 (2011) 1953–1964

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Joint QoS multicast routing and channel assignment in multiradio multichannel wireless mesh networks using intelligent computational methods

Hui Cheng[a,*], Shengxiang Yang[b]

[a] Department of Computer Science, University of Leicester, University Road, Leicester LE1 7RH, UK
[b] Department of Information Systems and Computing, Brunel University, Uxbridge, Middlesex UB8 3PH, UK

A R T I C L E   I N F O

Article history:
Received 6 March 2009
Received in revised form 22 April 2010
Accepted 19 June 2010
Available online 30 June 2010

Keywords:
Wireless mesh networks
Multicast
Channel assignment
Genetic algorithm
Simulated annealing
Tabu search

A B S T R A C T

In this paper, the quality of service multicast routing and channel assignment (QoS-MRCA) problem is investigated. It is proved to be a NP-hard problem. Previous work separates the multicast tree construction from the channel assignment. Therefore they bear severe drawback, that is, channel assignment cannot work well with the determined multicast tree. In this paper, we integrate them together and solve it by intelligent computational methods. First, we develop a unified framework which consists of the problem formulation, the solution representation, the fitness function, and the channel assignment algorithm. Then, we propose three separate algorithms based on three representative intelligent computational methods (i.e., genetic algorithm, simulated annealing, and tabu search). These three algorithms aim to search minimum-interference multicast trees which also satisfy the end-to-end delay constraint and optimize the usage of the scarce radio network resource in wireless mesh networks. To achieve this goal, the optimization techniques based on state of the art genetic algorithm and the techniques to control the annealing process and the tabu search procedure are well developed separately. Simulation results show that the proposed three intelligent computational methods based multicast algorithms all achieve better performance in terms of both the total channel conflict and the tree cost than those comparative references.

1. Introduction

Wireless mesh networks (WMNs) [1] have emerged as a new paradigm of static multi-hop wireless networks. A typical wireless mesh network consists of two types of wireless nodes, i.e., mesh routers and mobile clients. Each mesh router functions as both a relay node and an access point. As a relay node, a mesh router can forward packets to other mesh routers according to the routing information. As an access point, a mesh router can forward packets from or to the mobile clients which are currently associated with it. Mesh routers are stationary with power supply while clients may roam and change the associated mesh routers. In the wireless mesh networks, all the mesh routers are self-organized to establish ad hoc networks and maintain the network topology. As a result, WMNs have the advantages of easy deployment, high reliability, and large coverage. There is an increasing interest in using WMNs to provide ubiquitous network connectivity in enterprises, campuses, and in metropolitan areas [2].

Multicast [3–5] is an important network service, which is the delivery of information from a source to multiple destinations simultaneously using the most efficient strategy to deliver the messages over each link of the network only once, creating copies only when the links to the destinations split. It provides underlying network support for collaborative multimedia applications such as multimedia conference, distant education and content distribution. Quality of service requirements [6] proposed by different multimedia applications are often versatile. Among them, end-to-end delay [7,8] is a pretty important QoS metric since real-time delivery of multimedia data is often required. The multicast tree cost, used to evaluate the utilization of network resource, is also an important QoS metric especially in wireless networks where limited radios and channels are available. However, little work has addressed QoS multicast in WMNs.

In WMNs, if two mesh routers falling into the radio transmission range want to enable the communication link between them, they must tune their radios to the same channel. However, the wireless interference occurs when two links whose distance is less than 2 hops away are assigned to the same channel to support the concurrent communications, which is termed as channel conflict [9]. The heavy interference caused by channel conflict degrades the performance of the wireless communication severely. Therefore, for multicast routing, each link on the multicast tree requires to be assigned to one channel and the assignment should lead to minimum interference. Therefore, the QoS multicast routing in WMNs

* Corresponding author. Tel.: +44 1895 265975; fax: +44 1895 251686.
E-mail address: mcshcheng@googlemail.com (H. Cheng).

1568-4946/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.asoc.2010.06.011

involves not only to search a routing tree but also to assign proper channels to its links. In fact, the minimum-interference channel assignment problem itself is basically the Max *K*-cut problem [2], which is known to be NP-hard. Since our QoS-MRCA problem is the routing tree construction plus minimum-interference channel assignment, it is also NP-hard.

So far the QoS multicast routing has not drawn much attention from the research community of WMNs. However, it is believed that efficient multicast, which cannot be readily achieved through combined unicast or simplified broadcast, is essential to WMNs and deserves a thorough investigation [10]. In this paper, we develop a unified framework for solving the WMN multicast problem using intelligent computational methods. This framework consists of the problem formulation, the solution representation, the fitness function, and a simple yet effective channel assignment algorithm which assigns channels to each searched multicast tree for relieving the channel conflict. Based on the framework, we propose three efficient QoS multicast routing algorithms based on genetic algorithm (GA) [11], simulated annealing (SA) [12], and tabu search (TS) [13], separately. All of them aim to search low cost routing trees on which the channel assignment can produce the minimum interference. The idea is that for each searched delay-bounded multicast tree, we first assign channels to its links by the proposed channel assignment algorithm, and then evaluate it by the total channel conflict and tree cost. Since the channel assignment strategy is fixed, intuitively by examining more candidate routing trees, we can find the one on which the minimum-interference channel assignment can be achieved. Hence, the strong search capability of GA, SA and TS can be well utilized to solve this problem. Furthermore, these algorithms integrate the multicast tree construction and channel assignment, thereby avoiding that channel assignment cannot work well with the determined multicast tree.

The rest of this paper is organized as follows. We discuss related work in Section 2. We describe the framework in Section 3. We present the proposed GA, SA and TS based QoS-MRCA algorithms in Sections 4–6, separately. We present our simulation results in Section 7 and conclude this paper in Section 8, respectively.

2. Related work

Similar as mobile ad hoc networks (MANETs) [14], a wireless mesh network is also a type of self-organizing wireless network. However, there are three main differences between them. First, nodes in MANETs are often moving while mesh routers in WMNs are normally stationary. Second, in MANETs all the mobile nodes work in a peer-to-peer fashion and each node forwards packets on behalf of other nodes, while in WMNs a hierarchy is formed where mesh routers form a backbone and mesh clients can only access their associated mesh routers. Third, a mobile node in MANETs is normally equipped with one radio while a mesh router in WMNs is equipped with at least two radios.

In MANETs, a number of multicast routing protocols, using a variety of basic routing algorithms and techniques, have been proposed over the past few years [14]. However, they mainly focus on the discovery of the optimal multicast forwarding structure (i.e., tree or mesh) spanning mobile nodes and do not need to consider the channel assignment problem. In MANETs, since a mobile node may be equipped with a Global Positioning System (GPS) device, geographical information can also be utilized for route discovery. Therefore, according to the type of the utilized information, the multicast routing protocols in MANETs can be classified as topological routing and geographical routing.

In WMNs, little work has been done on multicast routing due to its intractability. In [10], Zeng proposed the Level Channel Assignment (LCA) multicast algorithm which is a deterministic one. The LCA multicast algorithm is composed of two components. First, it constructs a multicast tree based on breadth first search (BFS) aiming to minimize the hop count distances between the source and the receivers. Second, it uses a dedicated strategy to assign channels to the tree aiming to reduce the interference. However, since LCA separates the construction of the multicast tree from the channel assignment, it bears a potential drawback, that is, channel assignment cannot work well with the determined multicast tree. Furthermore, it does not consider the delay constraint which is a common issue for multicast problems. To our best knowledge, so far LCA is the best multicast algorithm in WMNs.

Genetic algorithm is a type of stochastic meta-heuristic optimization method that models the biological principles of Darwinian theory of evolution and Mendelian principles of inheritance [15,16]. Genetic algorithm has been extensively used in solving the QoS multicast problems in various networks such as the wired multimedia networks [4] and optical networks [17].

Simulated annealing algorithm simulates the annealing process in the physics of solids. It is observed that a metal body heated to high temperature cools slowly and tends to a state with the least internal energy. SA regards the optimization problem as a physical system and the value of the objective function as its internal energy. With this analogy, annealing is the process of determining a solution with the least value of the objective function. Simulated annealing algorithm is a powerful tool to solve the combinatorial optimization problems. It has been applied to the QoS multicast routing in the wired networks such as the multimedia communication networks [4,18].

Tabu search is a meta-heuristic that can lead a local search procedure to explore the solution space beyond local optimality. Tabu search uses a local or neighborhood search procedure to iteratively move from a solution x to a solution x' in the neighborhood of x, until some stopping criterion has been satisfied. Compared with other meta-heuristics such as genetic algorithm and simulate annealing, tabu search is more general and conceptually much simpler. However, TS still shows comepeting performance when it is used for solving many combinatorial optimization problems. Tabu search has been applied to the QoS multicast routing in the wired networks such as the multimedia communication networks [4,19].

In [4], the binary encoding is adopted where each bit of the binary string corresponds to a different node in the network. For each binary string, a graph G' is derived from the network topology G by including all the nodes appearing in the string and the links connecting these nodes. Then the minimum spanning tree T of G' acts as the candidate multicast tree represented by the binary string. This encoding method is a bit complicated and each binary string cannot directly represent the candidate solution. A multicast tree is a union of the routing paths from the source to each receiver. Hence, it is a more natural choice to adopt the path-oriented encoding method [17,20] than the binary encoding.

In [18], the path-oriented encoding is adopted. For each destination, a backup-path-set is constructed consisting of the k shortest (i.e., least-delay) paths from the source to it. Each time the SA algorithm generates a neighbor of a multicast tree by replacing its one path using a randomly selected backup path. Assuming m is the number of the destinations, each candidate solution is just one combination of m paths from the m backup-path-sets. Therefore, the size of the candidate solution space is limited by all the backup-path-sets. The performance of the algorithm will be hindered by the limited size of the solution space to be explored.

In [19], the path-oriented encoding is also employed. For each destination, a sink tree is constructed by connecting it to the source and all the other destinations using the shortest (i.e., least-cost) paths. On the sink tree, each path from the tree root to a leaf node is named as a superpath. Each iteration the TS algorithm first generates a few neighbors of a multicast tree by replacing its one superpath using a few randomly selected superpaths separately. Then, among these new neighbors, the one with the best cost is selected, and considered as the new solution for the next iteration. If a superpath is deleted at one iteration, then reintroducing the same superpath to the current tree is tabu. Assuming m is the number of the destinations, there are m sink trees. Each candidate solution is just one combination of m paths from the m sink trees. Therefore, the size of the candidate solution space is limited by all the sink trees. The performance of the algorithm is hindered by the limited solution space to be explored.

We are not aware of any other work that jointly considers multicast routing, which further consists of channel assignment as well as QoS in multiradio multichannel wireless mesh networks, although there are quite a few works that are related to some relevant aspects. Since GA, SA and TS show good performance in the wired networks, we believe their strong search capabilities can also help find low cost low interference routing trees in wireless mesh networks. However, to our best knowledge, none of them has been addressed to solve the QoS multicast routing and channel assignment problem in WMNs.

3. A unified framework for QoS-MRCA using intelligent computational methods

This section describes the proposed unified framework for solving the QoS-MRCA problem using intelligent computational methods. First, the network is modelled and the problem is formulated. The objective function is determined to minimize the total channel conflict. Then, two common components required by all the intelligent computational methods are provided, i.e., the solution representation and the fitness function. Finally, a simple yet effective channel assignment algorithm is proposed to produce the least channel conflict on any multicast tree.

3.1. Problem formulation

In this section, we first present our network model and then formulate the problem of joint QoS multicast routing and channel assignment.

We consider a wireless mesh network with stationary mesh routers where each router is equipped with a certain number of radio network interface cards (NICs). We model a wireless mesh network by a undirected and connected topology graph $G(V, E)$, where V represents the set of mesh routers and E represents the set of communication links connecting two neighboring mesh routers falling into the radio transmission range. A communication link (i, j) can not be used for packet transmission until both node i and node j have a radio interface each with a common channel. In addition, message transmission on a wireless communication link will experience a remarkable delay.

For clarity of presentation, we assume the *binary interference model*, i.e., two communication links either interfere or do not interfere. Given the binary interference model, the set of pairs of communication links that interfere with each other over the same channel can be represented by a conflict graph [9]. A communication link in the topology graph corresponds to a vertex in the conflict graph. With the binary interference model, the conflict graph G_c (V_c, E_c) can be easily derived from the topology graph $G(V, E)$. We assume the communication links (a, b) and (c, d) in the topology graph $G(V, E)$ are represented by the node i_c and node j_c in the conflict graph G_c (V_c, E_c), respectively. Then if the minimum distance between (a, b) and (c, d) is less than two hops, we have $(i_c, j_c) \in E_c$.

Here, we summarize some notations that we use throughout this paper.

- $G(V, E)$, the WMN topology graph.
- $G_c(V_c, E_c)$, the conflict graph derived from the WMN topology graph.
- $K = \{0, 1, 2, \ldots, k\}$, the set of available orthogonal channels.
- s, the source node of the multicast communication.
- $R = \{r_0, r_1, \ldots, r_m\}$, the set of receivers of the multicast communication.
- $T(V_T, E_T)$, a multicast tree with nodes V_T and links E_T.
- V_T^{Leaf}, the set of leaf nodes on the tree T.
- $P_T(s, r_i)$, a path from s to r_i on the tree T.
- d_l, the delay on the communication link l.
- $I_T(f)$, the total channel conflict on the tree T.
- C_T, the cost of the tree T.

The problem of joint QoS multicast routing and channel assignment in a multiradio multichannel wireless mesh network can be informally described as follows. Given a network of mesh routers with multiple radio interfaces, a delay upper bound, a source node and a set of receivers, we wish to find a delay-bounded multicast tree and assign a unique channel to each communication link on the tree. We define the *total channel conflict* as the number of pairs of communication links on the tree that are interfering (i.e., are assigned the same channel and are connected by an edge in the conflict graph). The objective of our problem is to minimize the above defined total channel conflict, as it results in improving the system throughput [10].

We also want to optimize the usage of the scarce network resources in the multicast tree. So we define the *tree cost* as the number of the radio interfaces involved in the multicast communications. We aim to find a multicast tree with low cost. There are two factors related to the tree cost. One is the number of communication links on the tree. Each communication link has one sender and one receiver, thereby occupying two radio interfaces. So we should reduce the number of links on the multicast tree, which also helps reduce the multicast end-to-end delay. The other factor is the number of broadcast nodes generated from the channel assignment. We make all the branch nodes become broadcast nodes by exploiting wireless multicast advantage (WMA) [21] and the detail is described in Section 3.4. If there are several multicast trees which have the same channel conflict value, we will choose the one with the minimum tree cost.

More formally, consider a wireless mesh network $G(V, E)$ and a multicast communication request from the source node s to a set of receivers R with the delay upper bound Δ. The *joint QoS multicast routing and channel assignment problem* is to find a multicast tree $T(V_T, E_T)$ satisfying the delay constraint as shown in (1) and compute a function f: $E_T \rightarrow K$ defined in (2) to minimize the *total channel conflict* $I_T(f)$ defined in (3).

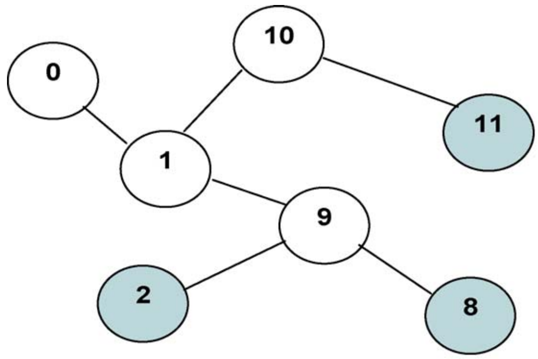
$$\max_{r_i \in R} \left\{ \sum_{l \in P_T(s, r_i)} d_l \right\} \leq \Delta, \tag{1}$$

$$f(i_c \in E_T) = \{j | j \in K\}, \tag{2}$$

$$I_T(f) = |\{(i_c, j_c) \in E_c | f(i_c) = f(j_c), i_c \in E_T, j_c \in E_T\}|. \tag{3}$$

Since the source only transmits packets and all the leaf nodes only receive packets, each of them occupies one radio interface only. All the other nodes are branch nodes which need to do both the transmission and reception. So each branch node occupies two radio interfaces. As a result, the tree cost C_T is calculated as follows:

$$C_T = |\{s\}| + |V_T^{Leaf}| + 2 \times (|V_T| - |\{s\}| - |V_T^{Leaf}|). \tag{4}$$



0	1	9	2			
0	1	9	8			
0	1	10	11			

Fig. 1. Illustration of the array representation of a multicast tree.

3.2. Solution representation

A routing path is encoded by a string of positive integers that represent the IDs of nodes through which the path passes. Each locus of the string represents an order of a node. The first locus is for the source and the last one is for the receiver. The length of a routing path should not exceed the maximum length $|V|$, where V is the set of nodes in the WMN.

For a multicast tree T spanning the source s and the set of receivers R, there are $|R|$ routing paths all originating from s. Therefore, we encode a tree by an integer array in which each row encodes a routing path along the tree. For example, for T spanning s and R, row i in the corresponding array A lists up node IDs on the routing path from s to r_i along T. Therefore, A is an array of $|R|$ rows. Fig. 1 illustrates a multicast tree and its representation in an array. All the solutions are encoded under the delay constraint. In case it is violated, the encoding process is usually repeated so as to satisfy the delay constraint.

3.3. Fitness function

Given a solution, we should accurately evaluate its quality (i.e., fitness value), which is determined by the fitness function. In our algorithm, we aim to find a low cost multicast tree on which the minimum-interference channel assignment can also be achieved. Our primary criterion regarding solution quality is the total channel conflict and the subsidiary one is the tree cost. Therefore, among a set of candidate solutions (i.e., multicast trees) with the same minimum channel conflict value, we choose the one with the lowest tree cost. The fitness value of chromosome Ch_i (representing multicast tree T), denoted as $F(Ch_i)$, is given by:

$$F(Ch_i) = [I_T(f) + 1.0]^{-1}. \tag{5}$$

The proposed fitness function only involves the total channel conflict. As mentioned above, The tree cost is used in the course of selecting the elitism [22] for recording the searched optimal solution.

3.4. Channel assignment algorithm

In a wireless mesh network, a link cannot be used for data transmission until it has been assigned a wireless communication channel. To support the multicast communication over the routing tree, an appropriate channel should be assigned to each link on the tree so as to achieve the minimum interference (i.e., channel conflict). In addition, the number of available channels is limited in the current network protocols. For example, in IEEE 802.11-based wireless networks, there are 11 available channels. However, at most three of them are orthogonal (non-interfering). The number of radio interfaces is also limited as a type of scarce radio network resource. Hence the channel assignment should use as small number of channels and radio interfaces as possible.

Since the minimum-interference channel assignment problem is NP-hard, we propose a heuristic algorithm which aims to reduce both the channel conflict and resource utilization. Given the set of orthogonal channels $K = 0, 1, \ldots, k$ ($k \geq 2$), the algorithm works on the multicast tree T as follows.

Algorithm 1 ChannelAssignment(T)

```
i =: 0;
while i < |R| do
   Assign channels to the routing path P_T(s, r_i) = (s, v_1, v_2, ..., v_{j-1}, r_i).
   In the following, we use v_0 to denote the source s and v_j to denote the
   receiver r_i, respectively;
   n =: 0;
   while n < j do
      if link (v_n, v_{n+1}) has not been assigned a channel then
         assign channel n%3 to it;
      end if
      n++;
   end while
   i++;
end while
```

Fig. 2 illustrates the channel assignment result over a multicast tree. For each routing path, the algorithm uses three channels to do the assignment. Since the minimum distance between two links to avoid channel conflict is two hops, three is the least number of channels to achieve conflict-free assignment on each routing path of the multicast tree. By our assignment strategy, all the links originating from the same branch node are assigned the same channel as utilizes the so-called WMA [21]. WMA refers to that a single transmission can be received by all the nodes that are within the transmission range of a transmitting node. Hence, using one radio interface only, the branch node transmits packets to all its children. This also saves the number of used radio interfaces.

4. GA based joint QoS-MRCA algorithm

This section describes the proposed GA based joint QoS multicast routing and channel assignment algorithm. The GA operations consist of several key components: genetic representation, population initialization, fitness function, selection scheme, crossover and mutation. Chromosomes (i.e., the candidate solutions) are expressed by tree data structure. The initial population explores the genetic diversity and also exploits the knowledge we have already known. Fitness function returns the total channel conflict of the multicast tree. Variation operators (i.e., crossover and mutation) efficiently promote the search capability. Note that every step guarantees that a tree does not violate the delay constraint. The population keeps evolving until it converges.

4.1. Population initialization

In GA, each chromosome corresponds to a potential solution. The initial population Q is composed of a certain number, denoted as q, of chromosomes. A general method to initialize the population is to explore the genetic diversity, that is, for each chromosome, all its routing paths are randomly generated. We start to search a random path from s to $r_i \in R$ by randomly selecting a node v_1 from $N(s)$, the neighborhood of s. Then we randomly select a node v_2 from $N(v_1)$. This process is repeated until r_i is reached. Thus, we get a random path $P_T(s, r_i) = \{s, v_1, v_2, \ldots, r_i\}$. Since no loop is allowed on the multicast tree, the nodes that are already included in the current tree are excluded, thereby avoiding reentry of the same node.

However, to exploit the knowledge that we have already known, we generate two multicast trees by the LCA multicast algorithm and the shortest path tree algorithm, respectively. Then we add these two trees into the initial population. We hope that they can help speed up the convergence. Thus, the initial population is generated as follows.

Algorithm 2 PopulationInitialization()

```
i =: 0;
while i < q do
  //Generate chromosome Ch_i
  j =: 0;
  V_T = E_T = ∅;
  while j < |R| do
    Search a random path P_T(s, r_i) which can guarantee T ∪ P_T be an
    acyclic graph;
    Add all the nodes and links in P_T into V_T and E_T, respectively;
    j++;
  end while
  i++;
end while
Replace Ch_0 by the LCA multicast tree;
Replace Ch_1 by the shortest path tree;
```

Thus, the initial population $Q = \{Ch_0, Ch_1, \ldots, Ch_{q-1}\}$ is obtained.

4.2. Selection scheme

Selection plays an important role in improving the average quality of the population by passing the high quality chromosomes to the next generation. The selection of chromosome is based on the fitness value. We adopt the scheme of pair-wise tournament selection without replacement [23] as it is simple and effective.

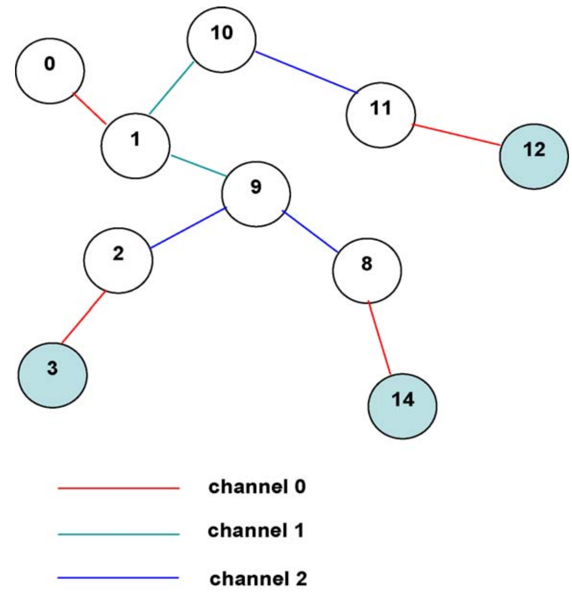


Fig. 2. Channel assignment over a multicast tree.

4.3. Crossover and mutation

Genetic algorithm relies on two basic genetic operators – crossover and mutation. Crossover processes the current solutions so as to find better ones. Mutation helps GA keep away from local optima [20]. Performance of GA very depends on them. Type and implementation of operators depends on encoding and also on a problem.

In our algorithm, since chromosomes are expressed by tree data structure, we adopt single point crossover to exchange partial chromosomes (sub-trees) at positionally independent crossing sites between two chromosomes [20]. With the crossover probability, each time we select two chromosomes Ch_i and Ch_j for crossover. To at least one receiver, Ch_i and Ch_j should possess at least one common node from which one, denoted as v, is randomly selected. In Ch_i, there is a path consisting of two parts: $(s\xrightarrow{Ch_i}v)$ and $(v\xrightarrow{Ch_i}r_i)$. In Ch_j, there is a path consisting of two parts: $(s\xrightarrow{Ch_j}v)$ and $(s\xrightarrow{Ch_j}r_i)$. The crossover operation exchanges the paths $(v\xrightarrow{Ch_i}r_i)$ and $(v\xrightarrow{Ch_j}r_i)$. Fig. 3 illustrates the crossover operation. Node 13 is the selected

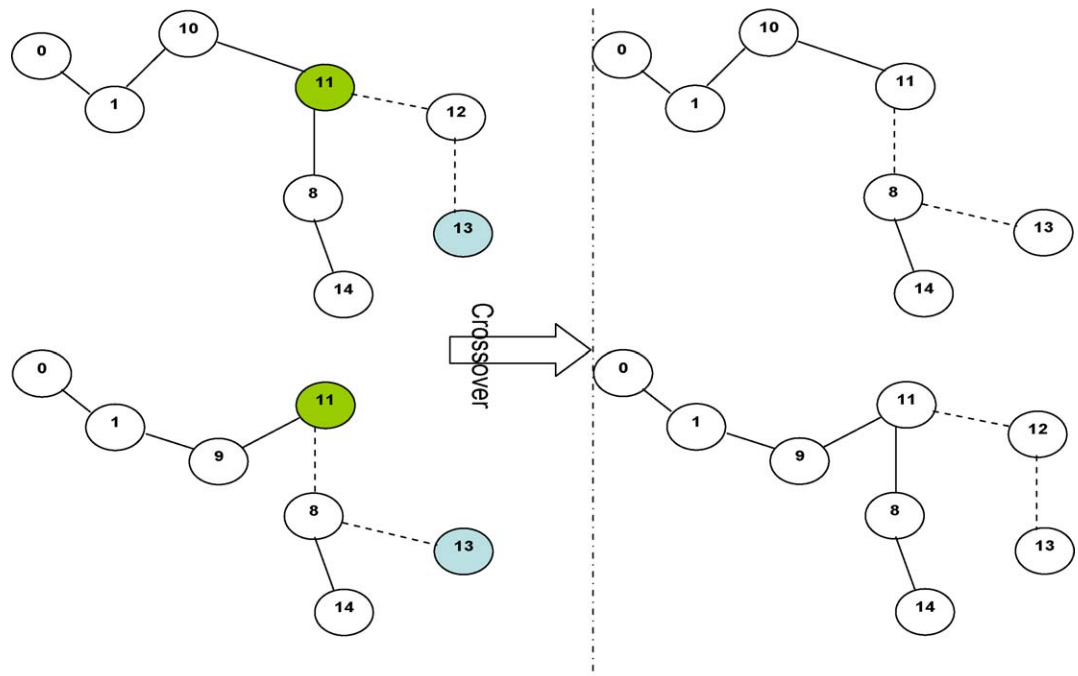


Fig. 3. Illustration of the crossover operation.

receiver and node 11 is the selected common node. The paths $(11 \rightarrow 12 \rightarrow 13)$ and $(11 \rightarrow 8 \rightarrow 13)$ are swapped.

The population will undergo the mutation operation after the crossover operation is performed. With the mutation probability, each time we select one chromosome Ch_i on which one receiver r_i is randomly selected. On the path $(s \xrightarrow{Ch_i} r_i)$ one gene is selected as the mutation point (i.e., mutation node) denoted as v. The mutation will replace the path $(v \xrightarrow{Ch_i} r_i)$ by a new random path.

Both crossover and mutation may produce new chromosomes which are infeasible solutions. Therefore, we check if the multicast trees represented by the new chromosomes are acyclic. If not, repair functions [24] will be applied to eliminate the loops. Here the detail is omitted due to the space limit. All the new chromosomes produced by crossover or mutation satisfy the delay constraint since it has already been taken into consideration.

5. SA based joint QoS-MRCA algorithm

This section describes the proposed SA based joint QoS multicast routing and channel assignment algorithm. The SA operations consist of the following key components: solution representation, neighborhood structure, initial temperature, temperature decreasing, iterative length at each temperature, and the termination rule. Note that every step also guarantees that a multicast tree does not violate the delay constraint.

We adapt SA to the joint multicast routing and channel assignment problem, and the objective function is just the fitness function, which returns the total channel conflict of the multicast tree. The fitness value just simulates the internal energy. First, the initial solution is generated by comparing the LCA tree and the SP tree in terms of the total channel conflict. Then we start the annealing process at a high temperature. As the temperature decreases, the annealing process tries to converge to the optimal solution. At each temperature, the algorithm searches a number of solutions in the solution space so that the current optimal solution stabilizes at a fitness value. When the temperature decreasing number reaches a specified upper bound and the current optimal solution keeps unchanged, the algorithm terminates and outputs the current optimal solution as the final solution.

5.1. Initial solution

Given the source and a set of receivers, both the LCA multicast algorithm and the shortest path tree algorithm can produce their own multicast trees. Intuitively, if we start the search from them, a better solution can be obtained. Therefore, we calculate the total channel conflict values for both the LCA tree and the SP tree. Then, we select the one with less value as the initial solution Q.

Algorithm 3 GenerateInitialSolution()

T_1 =:LCA tree;
T_2 :=SP tree;
$f_1 := ChannelAssignment(T_1)$;
$f_2 := ChannelAssignment(T_2)$;
if $f_1 < f_2$ **then**
  $Q := T_1$;
else
  $Q := T_2$;
end if

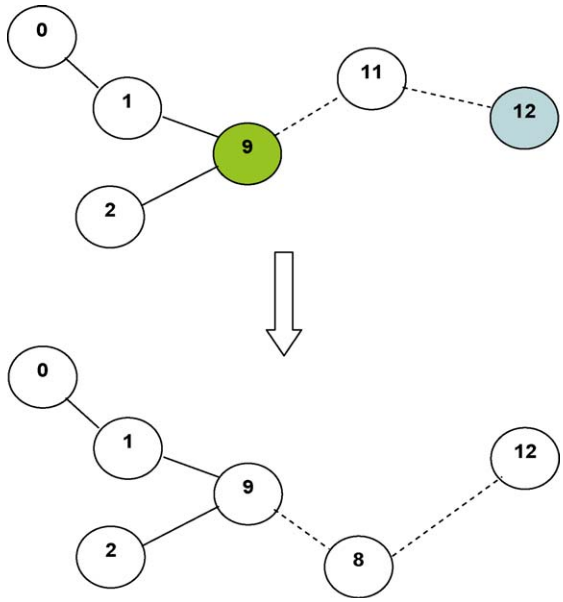


Fig. 4. Construction of a fine-grain neighborhood.

5.2. Neighborhood structure

Since SA performs searching from one solution to one of its neighbors in the solution space, we need to determine the neighborhood structure of each solution. In accordance with the solution representation, we propose two methods to construct the neighborhood.

(a) First, randomly select one receiver r_i from R, and randomly select another node v_i on the path $(s \to r_i)$. Then replace the subpath $(v_i \to r_i)$ by a new random subpath.
(b) First, randomly select two receivers r_i and r_j from R, and randomly select another two nodes v_i and v_j on the paths $(s \to r_i)$ and $(s \to r_j)$, respectively. Then replace the subpaths $(v_i \to r_i)$ and $(v_j \to r_j)$ by new random subpaths, respectively.

Given the current solution, a new neighbor solution will be produced using either of the above two methods. The first method only changes one path on the tree while the second method changes two paths at the same time. Intuitively, the adjustment to the tree is relatively smaller in (a) than in (b). So we name the first method as the fine-grain adjustment and the second method as the coarse-grain adjustment. Fig. 4 illustrates how to construct the neighborhood by the fine-grain adjustment. In the neighbor, a new path $(9 \to 8 \to 12)$ is used to replace the path $(9 \to 11 \to 12)$ in the previous solution. In the proposed algorithm, we apply the fine-grain adjustment in the first half of the temperature decreasing procedure, and then the coarse-grain adjustment in the second half of the temperature decreasing procedure. Therefore, we can ont only guarantee the algorithm converges to the optimal solution theoretically, but also accelerate the procedure to improve the efficiency.

5.3. Initial temperature

We start the SA algorithm from a high temperature (T_0) in order to allow acceptance of any new neighbor solution. A reasonable setting of the initial temperature will reduce the waste of the search time and still allow virtually all proposed uphill or downhill moves to be accepted [18]. In this algorithm, we set $T_0 = 100$.

5.4. Temperature decreasing

We employ the following method:

$$T_{k+1} = \alpha \times T_k (0 \le k, 0 < \alpha < 1). \tag{6}$$

This method is widely used, simple but effective. By this method, the temperature decreases at the same ratio.

5.5. Iterative length at each temperature

In our algorithm, the iterative length at one temperature is proportional to the number of temperature decreasing counted so far. We use L_i to denote the maximum iteration number allowed at temperature T_i, and M_i to denote the maximum number of continuous iterations without improving the present optimal solution allowed at T_i. As the temperature gradually decreases to T_i, both L_i and M_i should become larger simultaneously to explore more candidate solutions in the solution space.

We employ the method of linear increasing, that is, the maximum iteration number allowed at temperature T_i is in direct proportion to the up-to-now times of temperature decreasing, and the maximum number of continuous iterations without improving the present optimal solution allowed at T_i is in direct proportion to the maximum iteration number allowed at the same temperature. The method is formulated as follows:

$$L_i = (i + 1) \times \delta \times \tau, \tag{7}$$

$$M_i = \omega \times L_i, \tag{8}$$

where τ is the size of the receiver set, serving as the cardinal number. Since in each iteration, we need to change the path to one receiver. Ideally, we hope the paths to all the receivers will undergo the change at the same temperature. L_i limits the iteration number at the same temperature to speed up the convergence, and M_i helps stop the iteration at T_i since the search may be stuck in the local optimum.

5.6. Termination rule

The termination rule employed in this algorithm is to control the maximum number of continuous temperature decreasing without improving the present optimal solution. Let the maximum number of temperature decreasing be I, and the upper bound of the continuous temperature decreasing without improving the present optimal solution be U. They have the following relationship:

$$U = \lambda \times I \quad (0 < \lambda < 1). \tag{9}$$

In the proposed algorithm, during the first half period of temperature decreasing, i.e., from T_0 to $T_{\lfloor I/2 \rfloor}$, we generate a neighbor solution by the coarse-grain method; during the second half period of temperature decreasing, i.e., from $T_{\lfloor I/2 \rfloor+1}$ to T_I, we generate a neighbor solution by the fine-grain method. During the first half period, it is more likely that the difference between the current solution and the global optimal solution is relatively large. So we change two paths to two receivers at each iteration. During the second half period, the difference may become smaller. So we change only one path at each iteration. This design philosophy can help reduce the overhead of the fitness function calculation. Moreover, the algorithm can be theoretically assured to find the global optimal solution as the iteration approach infinity.

6. TS based joint QoS-MRCA algorithm

This section describes the proposed TS based joint QoS multicast routing and channel assignment algorithm. The TS operations consist of the following key components: solution representation, initial solution, neighborhood structure, fitness function, tabu move, tabu list, aspiration criterion, and termination rule. Note that every step guarantees that a multicast tree does not violate the delay constraint.

We adapt TS to the joint multicast routing and channel assignment problem, and the objective function is just the fitness function, which returns the total channel conflict of the multicast tree. First, the initial solution is generated. For the current solution, one of its neighbors is determined by the random path replacement. Then TS moves from the current solution to its neighbor, even this move deteriorates the fitness value. To explore more unvisisted solutions, solutions that have been recently visited are tabu for a few iterations. An aspiration criterion is proposed to free the solutions in tabu status to continue the search. When the number of continuous iterations without improving the current optimal solution reaches the specified upper bound, the algorithm ends and outputs the best solution that TS has ever visited as the final solution.

6.1. Initial solution

The method to generate the initial solution is the same as in the SA based algorithm.

6.2. Neighborhood structure

Since TS performs searching from one solution to one of its neighbors in the neighborhood, we need to determine the neighborhood structure of each solution. In accordance with the solution representation, we propose the following method to construct the neighborhood. First, randomly select one receiver r_i from R, and randomly select another node v_i on the path $(s \to r_i)$. Then replace the subpath $(v_i \to r_i)$ by a new random subpath to generate a neighbor solution. However, the replacement should guarantee that the delay constraint is not violated. It is similar as the fine-grain adjustment method in the SA based algorithm.

6.3. Tabu move

According to the solution representation and the neighborhood structure, each tabu move is a replacement of a subpath from a non-leaf node to a receiver. A new solution is reached after a move. Three cases may appear after each move.

(a) The fitness value of the new solution is greater than that of the original solution. That is, the new solution is superior to the original one.
(b) The fitness value of the new solution is equal to that of the original solution. That is, the new solution has the same quality as the original one in terms of the total channel conflict. However, they may still have different tree costs.
(c) The fitness value of the new solution is less than that of the original solution. That is, the new solution is inferior to the original one.

In the algorithm, each iteration we randomly select one node pair $\{v_1, r_1\}$. Then we replace the subpath $(v_1 \to r_1)$ by another different random subpath. Thus, a new solution is generated as a neighbor and its fitness value is calculated.

6.4. Tabu list

A tabu list is maintained to prevent returning to previously visited solutions. Each iteration we generate one neighbor. Without loss of generality, we assume that the neighbor is generated by replacing $(v_1 \to r_1)$. Then we push the subpath $(v_1 \to r_1)$ into the tabu list. As a result, one subpath is tabu each time. Since the new neighbor is selected, it is necessary to forbid the addition of the subpath $(v_1 \to r_1)$, otherwise the solution may return to the previously visited one in the following iterations.

The size of the tabu list is set to $\lfloor |R|/2 \rfloor$, where R is the set of receivers.

6.5. Aspiration criterion

Aspiration criterion is a device used to override the tabu status of moves whenever appropriate [19]. It temporarily overrides the tabu status if the move is sufficiently good. In our algorithm, at each iteration a new subpath is generated randomly. However, if the new path is currently in the tabu list, it cannot be used. Then we generate another new subpath randomly. If this new subpath is also in the tabu list, of these two tabu subpaths we will free the one which lies closer to the tabu list head.

6.6. Termination rule

In the algorithm, we record the current optimal solution and we also record the number of continuous iterations without improving it. Therefore, the termination rule employed is to control the maximum number of continuous iterations without improving the present optimal solution. We calculate the ratio of this number to the total iteration number. If the ratio exceeds the specified upper bound γ, we believe that to run the algorithm further will not contribute any improvement to the optimal solution. Therefore, we terminate the search to reduce the overhead. In the algorithm, we set γ to 0.3.

The maximum number of iterations is given to guarantee that the algorithm will terminate after sufficient search has been done. We denote W as the total number of iterations. As suggested in [19], we set W to 500. We denote U as the upper bound of the continuous iterations without improving the current optimal solution. We have

$$U = \eta \times W \quad (0 < \eta < 1). \tag{10}$$

So when U is reached, the algorithm will terminate. In the algorithm, $\eta \le 0.3$.

7. Performance evaluation

In this section, we compare the proposed three joint QoS-MRCA algorithms with the LCA multicast algorithm [10] and the shortest path tree algorithm. LCA separates the multicast tree construction from the channel assignment. If the channel assignment strategy cannot work well on the determined multicast tree, the LCA algorithm can do nothing while our algorithms can search other trees. The shortest path tree algorithm also provides a deterministic tree without considering the proper channel assignment.

A random WMN topology is generated using the following method. We first specify a square region with the area of 200×200 that has the width [0, 200] on the x axis and the height [0, 200] on the y axis. Then we generate a certain number of nodes and the position (x, y) of each node is randomly specified within the square area. If the distance between two nodes falls into the radio transmission range D, a link will be added to connect them and the delay of this link is randomly assigned within the range [1,5]. Finally, we check if the generated topology is connected. If not, the above process is repeated until a connected topology is generated. In the experiments, D is given a reasonable value 50. In GA, SA, and TS, we have a few algorithmic parameters and we list their suggested values in Table 1.

Without loss of generality, we assume that each mesh router has two radio network interface cards: one for transmission and

Table 1
Algorithmic parameters and their suggested values.

Parameter variable	Parameter description	Suggested value
p (GA)	Population size	50
ρ_c (GA)	Crossover probability	0.8
ρ_m (GA)	Mutation probability	0.05
T_0 (SA)	The initial temperature	100
α (SA)	The coefficient of temperature decreasing	0.95
δ (SA)	The coefficient of the maximum iteration number allowed at one temperature	1
ω (SA)	The coefficient of the maximum number of continuous iterations without improving the present optimal solution allowed at one temperature	0.50
λ (SA)	The coefficient of the maximum number of continuous temperature decreasing without improving the present optimal solution	0.30
γ (TS)	The ratio of the number of continuous iterations without improving the current optimal solution to the total iteration number	0.30
η (TS)	The coefficient of the number of continuous iterations without improving the current optimal solution	0.30
Δ	Delay upperbound	30

the other for reception. We assume that there are three orthogonal channels as the case in 802.11 wireless network. We compare the GA, SA and TS multicast algorithms with the LCA multicast algorithm and the shortest path tree algorithm on two different network topologies. One is small scale consisting of 23 nodes and 34 links and the other is larger consisting of 50 nodes and 201 links. The topology for the small scale network is shown in Fig. 5. The metrics that we evaluate include the total channel conflict, the tree cost, the average tree delay, and the maximum tree delay. Each experiment is terminated when the population converges in GA or the termination condition is satisfied in either SA or TS.

In Section 3.3, we have mentioned that our primary optimization objective regarding solution quality is the total channel conflict and the subsidiary one is the tree cost. In Formula (5), the fitness function is related to the total channel conflict value only. The tree cost value is used only when two multicast trees have the same channel conflict values. In such a case, the tree with less cost will be selected. However, it is interesting to investigate the use of both optimization objectives in the fitness function. In Section 7.1, the fitness function follows the one in Formula (5) and there is only one optimization objectives. In Section 7.2, we develop a new fitness function which combines these two optimization objectives linearly.

7.1. Results under single-objective optimization

In the WMN of 23 nodes, the size of the multicast group ranges from 3 to 11 while in the WMN of 50 nodes it ranges from 9 to 17. Fig. 6 shows the comparison results in terms of the total channel conflict. It shows that in both networks, our GA, SA and TS multicast algorithms can find the multicast trees with less channel conflict than the trees obtained by the LCA multicast algorithm and the SPT multicast algorithm. In the network of 23 nodes, all the three proposed algorithms can find the conflict-free multicast trees when the multicast group size is less than or equal to 7. When the number of multicast nodes is beyond 7, GA multicast can still find the conflict-free multicast trees.

Fig. 7 shows the comparison results in terms of the tree cost. It shows that the LCA and SPT multicast trees always have higher cost than any of the tree intelligent methods. It means that the GA, SA and TS multicast trees consume less radio network resources than both the LCA and SPT multicast trees. In the network of 23 nodes, when the multicast group size is less than 6, the GA multicast trees have the same cost as the LCA multicast trees, and the SA multicast trees have the same cost as the TS multicast trees. However, when the multicast group size is equal to or greater than 6, the cost of the TS multicast trees is higher than the GA and SA multicast trees. In

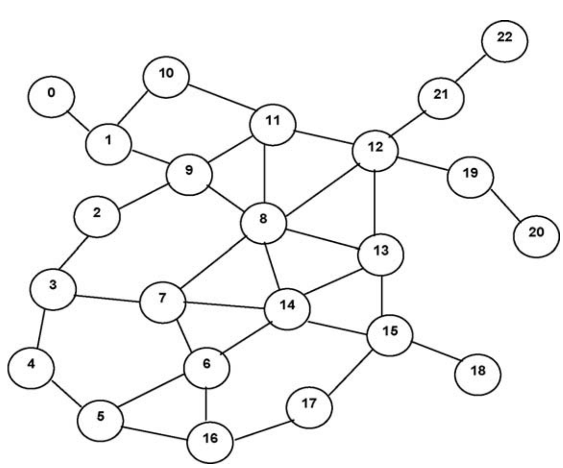


Fig. 5. The topology of the WMN with 23 nodes.

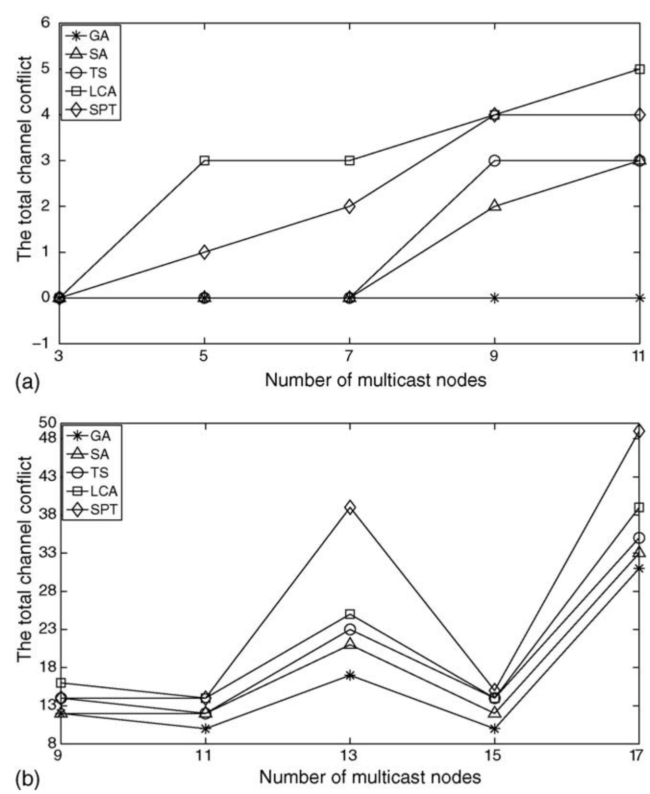


Fig. 6. Comparison of GA, SA, TS multicast and LCA multicast, SPT multicast in terms of the total channel conflict in: (a) a WMN of 23 nodes; (b) a WMN of 50 nodes.

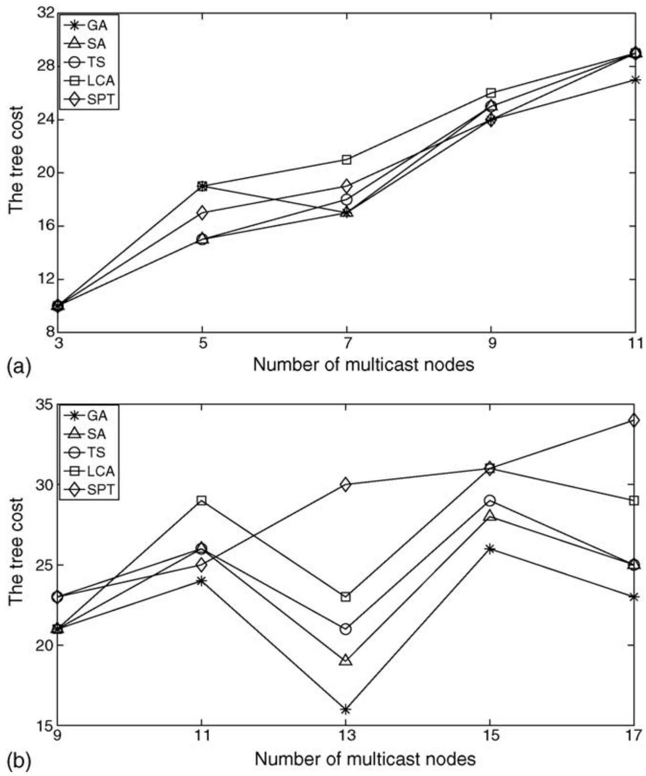


Fig. 7. Comparison of GA, SA, TS multicast and LCA multicast, SPT multicast in terms of the tree cost in: (a) a WMN of 23 nodes; (b) a WMN of 50 nodes.

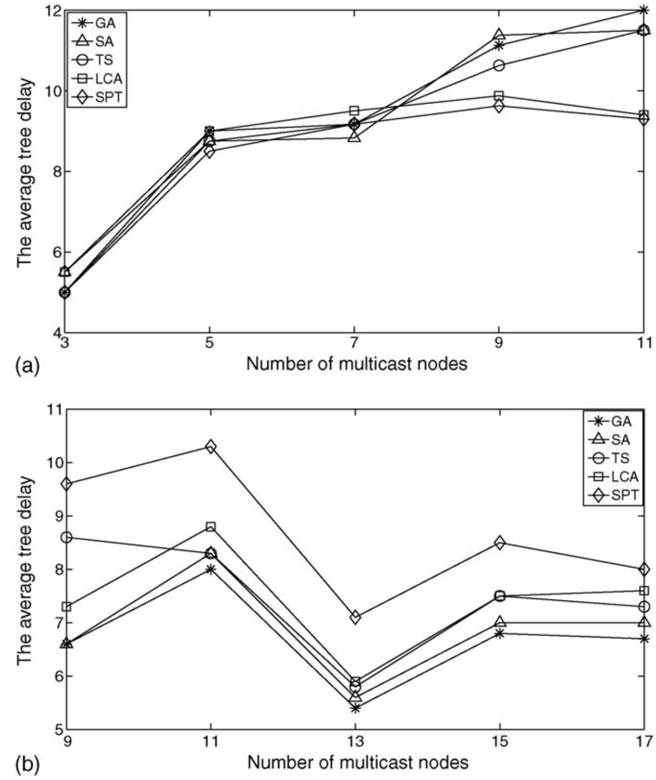


Fig. 8. Comparison of GA, SA, TS multicast and LCA multicast, SPT multicast in terms of the average tree delay in: (a) a WMN of 23 nodes; (b) a WMN of 50 nodes.

the network of 50 nodes, among the three intelligent methods, TS trees have the highest cost and GA trees have the lowest.

Fig. 8 shows the comparison results in terms of the average tree delay. The average tree delay is defined as the average delay of all the paths from the source to all the receivers on the tree. It shows that in the network of 23 nodes, the SPT multicast trees almost always have the lowest average delay. In the network of 50 nodes, only when the multicast group size is 9, the average delay of the LCA multicast tree is a bit lower than the TS tree. In all the other cases, both the LCA and SPT trees have higher cost than others. Therefore, these five algorithms have competing performance in terms of the average delay. Fig. 9 shows the comparison results in terms of the maximum tree delay. The maximum tree delay is defined as the maximum delay among all the paths from the source to all the receivers on the tree. Similar as the average delay comparison results, in the network of 50 nodes the SPT multicast trees almost always have the highest end-to-end delay, and in the network of 23 nodes, the five algorithms have the competing performance. From Figs. 8 and 9, the GA, SA and TS multicast algorithms do not improve the delay performance no matter in the average delay or in the maximum delay. The reason is that they use long paths to avoiding the channel conflict. However, they still can find the trees which satisfy the end-to-end delay constraint.

7.2. Results under multi-objective optimization

Although both the total channel conflict and the tree cost have been mentioned as the optimization objectives, only the total channel conflict is used in the search procedure. The tree cost plays less important role since it is just used for breaking the tie. To further investigate the effects of both objectives on the algorithm performance, we modify the fitness function by adding the tree cost value into it. These two optimization objectives are linearly combined together and each has a different weight factor. The new fitness

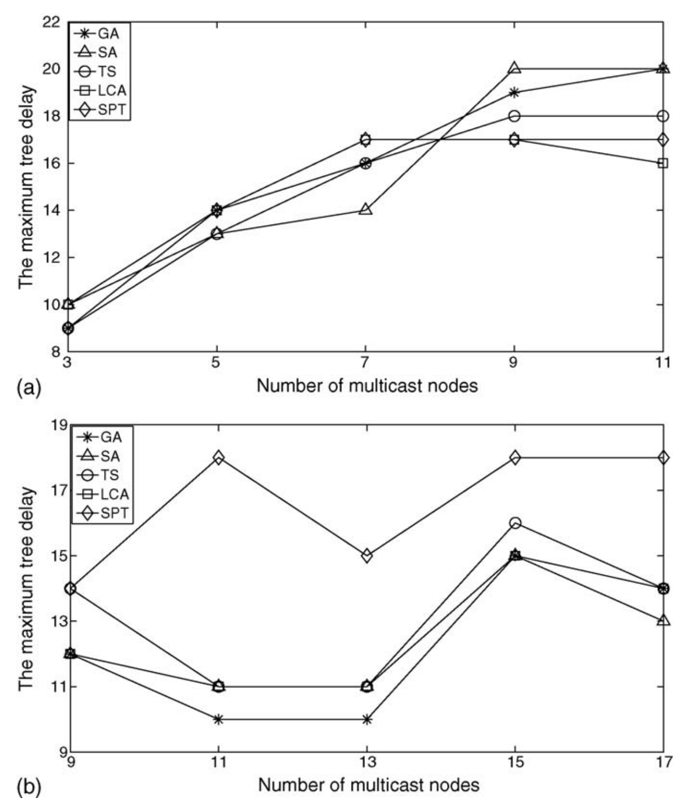


Fig. 9. Comparison of GA, SA, TS multicast and LCA multicast, SPT multicast in terms of the maximum tree delay in: (a) a WMN of 23 nodes; (b) a WMN of 50 nodes.

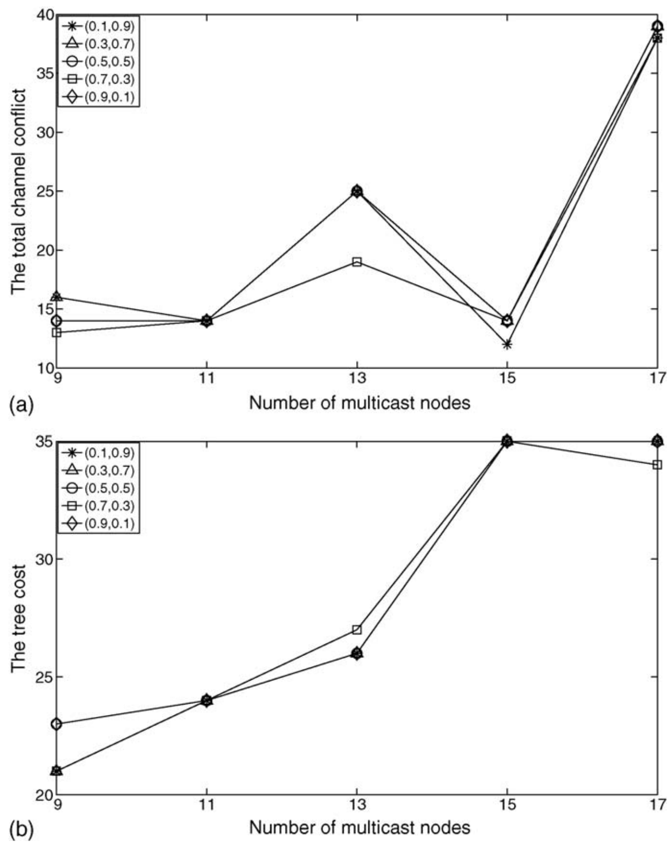


Fig. 10. Comparison of GA multicast in a WMN of 50 nodes under various weight combinations in terms of: (a) the total channel conflict; (b) the tree cost.

function is shown below.

$$F(Ch_i) = [\alpha \times I_T + \beta \times C_T]^{-1}. \tag{11}$$

Since $\alpha + \beta = 1.0$, we can have different combinations for these two weight factors by varying their values. Intuitively, the larger the weight factor, the higher the importance of the corresponding optimization objective. In the following experiments, we propose five different combinations for (α, β), i.e., (0.1, 0.9), (0.3, 0.7), (0.5, 0.5), (0.7, 0.3), and (0.9, 0.1). The weight factor for the total channel conflict is increased gradually and oppositely, the factor for the tree cost is reduced gradually. We have tested the GA multicast in the WMN of 50 nodes under these five combinations. In each run, the optimal individual regarding the fitness value is output as the final solution. Then its channel conflict and tree cost are recorded for comparison. The results are presented in Fig. 10.

From Fig. 10(a), we can see that with the gradual increase in the weight factor for the total channel conflict, the solution quality is improved in terms of this optimization objective. However, when the size of multicast group is larger than 13, there is no improvement. From Fig. 10(b), we can see that basically the tree cost has no response to the change of the weight factor. These results are worse than the ones in the previous section where only the total channel conflict is used in the fitness function. We have also tested the SA multicast and the TS multicast and found similar results. In summary, the weight factors have no significant effect on the algorithm performance. The reason is due to the intrinsic drawbacks as a scalar objective function to provide solution for multi-objective optimization.

It is known that a single scalar objective function on ad hoc basis not only makes the solution highly sensitive to the chosen weight vector but also requires the user to have some knowledge about the priority or influence of a particular objective parameter over another [25]. For multi-objective multicast, the same problem occurs because different optimization objectives evaluate different properties of the trees. Moreover, the evaluation criterion is different for different objectives. Hence, it is difficult to determine the weight factors for the objectives in the linear combination formula. If an algorithm uses the weighted sum as a single objective, in our opinion, it is still a single-objective multicast approach since it results in only one final solution. This solution cannot always optimize both objectives simultaneously. If we really want to reflect the impact of both objectives, we need to seek help from multi-objective optimization algorithms, e.g., multi-objective evolutionary algorithm (MOEA). This will be investigated in the future work.

8. Conclusions

The wireless mesh networks have seen various collaborative multimedia applications which require efficient information delivery service from a designated source to multiple receivers. A multicast tree with orthogonal channels appropriately assigned is preferred to support this service. However, the optimal multicast routing and channel assignment problem is proved to be NP-hard. Unfortunately, so far little work has been done on it. This paper presents three joint multicast routing and channel assignment algorithm for wireless mesh networks. These algorithms are based on different intelligent computational methods. They apply GA, SA and TS separately to discover delay-bounded minimum-interference low cost multicast trees. We believe that the synergy achieved by combining the strong search capabilities of the three intelligent computational methods and the effective channel assignment results in the improved quality of solution. We compare the performance of the proposed algorithms with the prestigious LCA multicast algorithm. Experimental results demonstrated that all our algorithms are capable of finding the multicast trees which have both less channel conflict and lower cost (i.e., consuming less radio network interfaces) than the shortest path trees and the trees produced by the LCA multicast algorithm. Although they do not improve the delay performance, they still can find the delay constrained multicast trees.

Acknowledgements

The authors would like to thank anonymous reviewers for their valuable comments to improve the paper. This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) of UK under Grant EP/E060722/1.

References

[1] I. Akyildiz, X. Wang, W. Wang, Wireless mesh networks: a survey, Comput. Netw. 47 (March(4)) (2005) 445–487.
[2] A. Subramanian, H. Gupta, S. Das, Minimum interference channel assignment in multi-radio wireless mesh networks, in: Proc. 4th Annu. IEEE Commun. Soc. Conf. Sensor, Mesh, Ad Hoc Commun. Netw. (SECON 2007), San Diego, CA, 2007, pp. 481–490.
[3] B. Wang, J. Hou, A survey on multicast routing and its QoS extensions: problems, algorithms, and protocols, IEEE Netw. 14 (January(1)) (2000) 22–36.
[4] X. Wang, J. Cao, H. Cheng, M. Huang, QoS multicast routing for multimedia group communications using intelligent computational methods, Comput. Commun. 29 (August(12)) (2006) 2217–2229.
[5] C. Oliveira, P. Pardalos, A survey of combinatorial optimization problems in multicast routing, Comput. Oper. Res. 32 (August(8)) (2005) 1953–1981.
[6] X. Xiao, L. Ni, Internet QoS: a big picture, IEEE Netw. 13 (March(2)) (1999) 8–18.
[7] M. Parsa, Q. Zhu, J. Garcia-Luna-Aceves, An iterative algorithm for delay-constrained minimum-cost multicasting, IEEE/ACM Trans. Netw. 6 (August(4)) (1998) 461–474.
[8] X. Jia, A distributed algorithm of delay-bounded multicast routing for multimedia applications in wide area networks, IEEE/ACM Trans. Netw. 6 (December(6)) (1998) 828–837.

[9] K. Jain, J. Padhye, V. Padmanabhan, L. Qiu, Impact of interference on multi-hop wireless network performance, in: 9th ACM Annu. Int. Conf. Mobile Comput. Netw. (MobiCom 2003), San Diego, CA, 2003, pp. 66–80.
[10] G. Zeng, B. Wang, Y. Ding, L. Xiao, M. Mutka, Multicast algorithms for multi-channel wireless mesh networks, in: 15th IEEE Int. Conf. Netw. Proto. (ICNP 2007), Beijing, China, 2007, pp. 1–10.
[11] M. Gen, R. Cheng, Genetic Algorithms and Engineering Optimization, Wiley, New York, 2000.
[12] T. Levanova, M. Loresh, Algorithms of ant system and simulated annealing for the p-median problem, Autom. Rem. Contr. 65 (March(3)) (2004) 431–438.
[13] F. Glover, Tabu search. Part I, ORSA J. Comput. 1 (3) (1989) 190–206.
[14] C. Cordeiro, H. Gossain, D. Agrawal, Multicast over wireless mobile ad hoc networks: present and future directions, IEEE Netw. 17 (January(1)) (2003) 52–59.
[15] D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, USA, 1989.
[16] J. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, 1975.
[17] D. Din, Anycast routing and wavelength assignment problem on WDM network, IEICE Trans. Commun. E88–B (October(10)) (2005) 3941–3951.
[18] K. Zhang, Y. Qi, H. Zhang, Dynamic multicast routing algorithm for delay and delay variation-bounded Steiner tree problem, Knowl.-Based Syst. 19 (November(7)) (2006) 554–564.
[19] Y. Habib, A. Abdulaziz, M. Sadiq, A. Muhammad, QoS-driven multicast tree generation using tabu search, Comput. Commun. 25 (July(11–12)) (2002) 1140–1149.
[20] C. Ahn, R. Ramakrishna, A genetic algorithm for shortest path routing problem and the sizing of populations, IEEE Trans. Evol. Comput. 6 (December(6)) (2002) 566–579.
[21] M. Thai, Y. Li, D. Du, A combination of wireless multicast advantage and hitch-hiking, IEEE Commun. Lett. 9 (December(12)) (2005) 1037–1039.
[22] J. Majumdar, A. Bhunia, Elitist genetic algorithm for assignment problem with imprecise goal, Eur. J. Oper. Res. 177 (March(2)) (2007) 684–692.
[23] S. Lee, S. Soak, K. Kim, H. Park, M. Jeon, Statistical properties analysis of real world tournament selection in genetic algorithms, Appl. Intell. 28 (April(2)) (2008) 195–2205.
[24] S. Oh, C. Ahn, R. Ramakrishna, A genetic-inspired multicast routing optimization algorithm with bandwidth and end-to-end delay constraints, Lecture Notes Comput. Sci. 4234 (2006) 807–816.
[25] A. Roy, S. Das, QM^2RP: a QoS-based mobile multicast routing protocol using multi-objective genetic algorithm, Wireless Netw. 10 (May(3)) (2004) 271–286.