Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 432686, 13 pages
http://dx.doi.org/10.1155/2013/432686

Research Article

Minimum Cost Multicast Routing Using Ant Colony Optimization Algorithm

Xiao-Min Hu and Jun Zhang

Department of Computer Science, Sun Yat-sen University, Key Laboratory of Intelligent Sensor Networks, Ministry of Education, Key Laboratory of Software Technology, Education Department of Guangdong Province, Guangzhou 510006, Guangdong Province, China

Correspondence should be addressed to Xiao-Min Hu; xmhu@ieee.org

Received 7 February 2013; Accepted 18 April 2013

Academic Editor: Yuping Wang

Multicast routing (MR) is a technology for delivering network data from some source node(s) to a group of destination nodes. The objective of the minimum cost MR (MCMR) problem is to find an optimal multicast tree with the minimum cost for MR. This problem is NP complete. In order to tackle the problem, this paper proposes a novel algorithm termed the minimum cost multicast routing ant colony optimization (MCMRACO). Based on the ant colony optimization (ACO) framework, the artificial ants in the proposed algorithm use a probabilistic greedy realization of Prim's algorithm to construct multicast trees. Moving in a cost complete graph (CCG) of the network topology, the ants build solutions according to the heuristic and pheromone information. The heuristic information represents problem-specific knowledge for the ants to construct solutions. The pheromone update mechanisms coordinate the ants' activities by modulating the pheromones. The algorithm can quickly respond to the changes of multicast nodes in a dynamic MR environment. The performance of the proposed algorithm has been compared with published results available in the literature. Results show that the proposed algorithm performs well in both static and dynamic MCMR problems.

1. Introduction

Multicast routing (MR) is one of the most important communication network routing technologies. It first appeared in ARPANET as selective broadcasting from a single source node to a subset of the other nodes in the network [1]. Different from unicast routing, MR sends only one data copy from the source node(s) to multiple destination nodes. Since MR is resource efficient, it has been implemented in most of the modern communication networks, including overlay multicast protocols [2–4], wireless networks [5–7], and satellite networks [8, 9]. These networks support multicast applications such as distributed data processing, internet telephone, interactive multimedia conferencing, and real-time video broadcasting [4].

Different multicast applications have different definitions of the cost, such as minimum bandwidth [3, 5, 7] and minimum energy [6]. The objective of the minimum cost MR (MCMR) problem is to find an optimal multicast tree with minimum cost for MR. Such a tree is termed a Steiner minimal tree (SMT) in graphs [10]. Finding the SMT has been proven to be NP complete [11].

Various algorithms have been tried for solving the MCMR problem. Traditional heuristic algorithms [12–15] use greedy strategies to construct a feasible multicast tree, but they lack effective guidance to improve the results. Geographic [7] and distributed algorithms [6] are only based on the information from neighbors or nodes within a constant hop distance to compute multicast paths. However, it may be difficult to build an efficient multicast tree without complete information [2]. Computational intelligence (CI) methods such as neural networks (NNs) and genetic algorithms (GAs) have been applied to MCMR. Gelenbe et al. [16] proposed a random neural network (RNN) to optimize a multicast tree. The RNN essentially enumerated the results by iteratively adding the most potential node to the tree. Leung et al. [17] proposed a GA to train the population of individuals by simulating natural evolution. Each individual represented a multicast tree indirectly through an encoding scheme. Except for the evaluation of the fitness of individuals, no tree construction information was utilized in the evolutionary process of the GA.

In recent years, ant colony optimization (ACO) [18–20] has become an important CI method. ACO is inspired by the foraging behavior of natural ants. ACO dispatches a colony of artificial ants to cooperatively search for the optimum of the optimization problem represented on a graph. Each ant in ACO incrementally builds a solution according to the construction information in a stochastic way [18]. ACO has been applied successfully to various combinatorial optimization problems, such as the traveling salesman [19, 20], constraint satisfaction [21], allocation problems [22], scheduling [23–27], data mining [28, 29], water distribution systems [30], power electronic circuit design [31], and networks [32–37]. Using ACO for MR optimization is a promising research field, which is still under development. Singh et al. [38] proposed an ant algorithm for MR optimization. In their algorithm, one ant was initially placed at every node in the multicast group and started to move to the other node via an edge. If an ant stepped on a node that had been visited by another ant, it merged into the latter ant. When only one ant was left, the edges passed by the ants forming a multicast tree. The authors had tested three different sequences for moving the ants from one node to another and found that the random approach was the best. However, their algorithm still could not always find the optimal solutions of some of their test cases. Shen and Jaikaeo [39] and Shen et al. [40] applied swarm intelligence to a multicast protocol by connecting nodes in a multicast group through a designated node. Each node in the multicast group periodically sent a packet that behaved like an ant to explore different paths to the designated node. The designated node was not statically assigned and its location influenced the optimality of the multicast tree.

In order to make better use of ACO, this paper proposes a novel minimum cost multicast routing ant colony optimization (MCMRACO) algorithm for solving MCMR problems. The algorithm has the following features. (1) Based on the ACO framework, the proposed algorithm adopts a probabilistic greedy realization of Prim's algorithm [41] for the ants to construct multicast trees. (2) Moving in a cost complete graph (CCG) of the network topology, the ants build solutions according to the heuristic and pheromone information. (3) The heuristic information is designed to represent problem-specific knowledge for the ants to construct solutions and to bias the selection of nodes in the multicast group. (4) Representing the ants' construction experience, pheromones are modulated by the local and global pheromone update mechanisms. The local pheromone update is applied after each ant has made a construction step, whereas the global pheromone update reinforces the pheromones in the best multicast tree after each iteration. Hence, the algorithm can quickly respond to the changes of multicast nodes in a dynamic MR environment. Utilizing heuristic and pheromone information effectively, the proposed MCMRACO is more suitable for solving MCMR problems than the other heuristic and CI algorithms. The comparison results show that the algorithm works successfully in both static and dynamic MCMR problems.

The rest of the paper is organized as follows. In Section 2, background information for the proposed algorithm is presented. Section 3 describes the implementation and main techniques of the proposed MCMRACO algorithm. In Section 4, the proposed algorithm is tested on static and dynamic MCMR problems. Its performance is compared with the published results available in the literature. Section 5 concludes this paper and discusses some issues for future research.

2. Background

This section is composed of three parts. In the first part, the MCMR optimization problem is formally defined. Then a method for transforming a network graph into a CCG is illustrated. The third part describes the ACO framework.

2.1. Definition of the MCMR Optimization Problem. In an MCMR optimization problem, a network graph is denoted as $G = (V_G, E_G)$, where V_G is the set of nodes in the network and E_G is the set of edges which connect the nodes in V_G. The set V_G is divided into three subsets V_S, V_D, and V_I, where V_S is the set of source nodes, V_D is the set of destination nodes, and V_I is the set of intermediate nodes. $V_M = V_S \cup V_D$ is the set of nodes in a multicast group. Intermediate nodes can be used for relaying multicast traffic, but they do not belong to the multicast group. In backbone IP networks, the nodes in V_G generally are routers. Each edge $e \in E_G$ has a positive cost value $c(e)$ that measures the quality of the edge. If there is no edge between two nodes in V_G, the corresponding cost is denoted as ∞. The optimization objective of the problem is to find a tree $T = (V_M + V_\theta, E_\theta)$ that minimizes the cost for connecting the nodes in V_M and that satisfies the multicast constraints Ω. Formally, the MCMR problem is defined as

$$\text{minimize} \sum_{e \in E_\theta} c(e), \; T \text{ satisfies the constraints in } \Omega, \tag{1}$$

where $V_\theta \subseteq V_I$ is a subset of the intermediate nodes, $E_\theta \subseteq E_G$ is a subset of the edges in the network graph.

Figure 1 shows an example of MR in a network with twelve nodes and some edges. Node 1 is the source node. The black solid nodes 2, 4, 6, 12 are the destination nodes. A multicast tree connecting the multicast nodes via the intermediate nodes 5 and 7 is shown in the figure. In this paper, a multicast group has only one source node and multiple destination nodes.

2.2. Cost Complete Graph (CCG). The network graph is usually a noncomplete but connected graph. The edges in a network graph are termed physical edges. In a CCG, however, each pair of nodes is connected by a logical edge with the minimum cost between the two nodes. One of the classical algorithms for transforming a network graph into a CCG is Floyd's algorithm [42]. The pseudocode of the algorithm is shown in Algorithm 1. The input cost matrix of the network graph is denoted by $\mathbf{C}_{|V_G| \times |V_G|}$, where its element c_{ij} is the cost

```
Floyd's Algorithm
Input:  $\mathbf{C}_{|V_G|\times|V_G|}[c_{ij}]$: the cost matrix of the network graph
Output: $\widehat{\mathbf{C}}_{|V_G|\times|V_G|}[\hat{c}_{ij}]$: the cost matrix of the cost complete graph (CCG)
        $\mathbf{O}_{|V_G|\times|V_G|}[o_{ij}]$: records the next node on the minimum cost route
        from node $i$ to node $j$
For $i := 1$ to $|V_G|$
    For $j := 1$ to $|V_G|$
        $\hat{c}_{ij} = c_{ij}$;
        If $c_{ij} = \infty$
            $o_{ij} := -1$;
        Else
            $o_{ij} := j$;
        End If
    End For
End For
For $k := 1$ to $|V_G|$
    For $i := 1$ to $|V_G|$
        For $j := 1$ to $|V_G|$
            If $i \neq j$ and $\hat{c}_{ik} + \hat{c}_{kj} < \hat{c}_{ij}$
                $\hat{c}_{ij} := \hat{c}_{ik} + \hat{c}_{kj}$;
                $o_{ij} := o_{ik}$;
            End If
        End For
    End For
End For
```

Algorithm 1: Pseudocode of Floyd's algorithm.

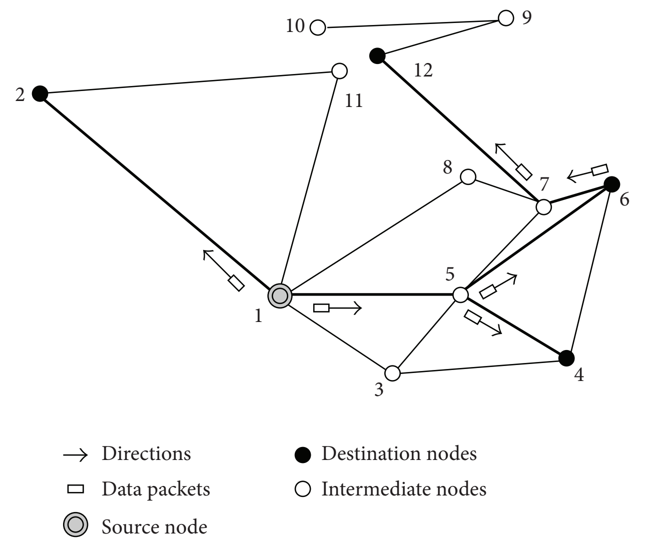

Figure 1: An example of multicast routing. The source node and the destination nodes belong to a multicast group. The heavy black lines indicate the multicast tree.

of the edge connecting nodes i and $j (i, j = 1, 2, \ldots, |V_G|)$. The output of the Floyd's algorithm consists of two matrixes. One is the cost matrix $\widehat{\mathbf{C}}_{|V_G|\times|V_G|}$ of the CCG, where the value of its elements $\hat{c}_{ij}$ denotes the minimum cost from node i to node j. The other is $\mathbf{O}_{|V_G|\times|V_G|}$, where each element o_{ij} records the next node of node i on the minimum cost route from node i to node j in the network graph. Figure 2(a) presents an undirected network graph. The values in the figure indicate the cost of the corresponding physical edges. Suppose $\Omega = \varnothing$ and take the edges (b_1, b_2) and (b_1, b_4) as examples. Nodes b_1 and b_2 are not adjacent, but they are connected via node b_3 or b_4 or nodes b_5, b_6, b_4. In the CCG (Figure 2(b)), nodes b_1 and b_2 are logically adjacent with $\hat{c}_{b_1 b_2} = \hat{c}_{b_2 b_1} = 2.5$ and $o_{b_1 b_2} = o_{b_2 b_1} = b_3$. For nodes b_1 and b_4, although they are already adjacent, there is a shorter path via nodes b_5 and b_6. So $\hat{c}_{b_1 b_4} = \hat{c}_{b_4 b_1} = 1.5$, whereas $o_{b_1 b_4} = b_5$, $o_{b_4 b_1} = b_6$.

The CCG of the network is obtained before the proposed MCMRACO algorithm starts. Generally, the CCG is maintained by the network routers or a central network controlling apparatus. For example, in [2], master multicast routers are used to deal with all the multicast-related tasks. They gather routing information, make MR decisions, and manage the other routers to perform MR. The proposed MCMRACO algorithm can be implemented by the master multicast router to find a high-quality multicast tree for MR.

2.3. Ant Colony Optimization. Once the CCG is available, the ants can be dispatched to construct multicast trees. The ants in ACO can move in parallel or sequentially to construct their solutions [18]. Before introducing the proposed algorithm, the ACO framework is briefly presented.

The ACO algorithm used in this paper is ant colony system (ACS) [20], which is characterized by its state transition rule and the pheromone update mechanisms. At each iteration, a colony of ants is dispatched to search for solutions.

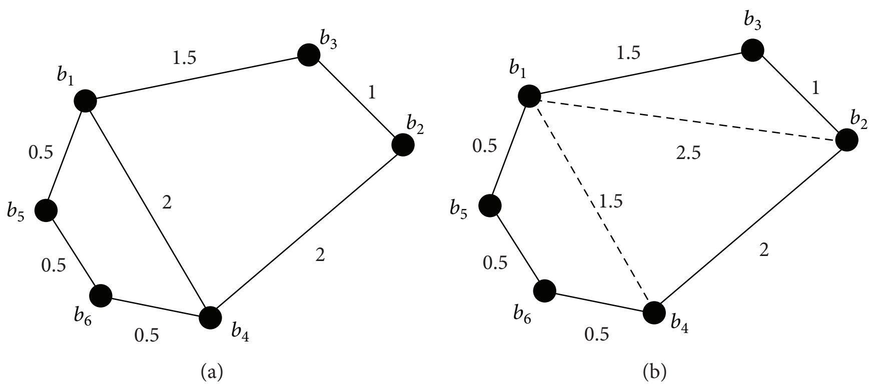

FIGURE 2: An example of transforming edge (b_1, b_2) and edge (b_1, b_4) in the CCG. (a) The original undirected graph showing the physical edges and the edge cost. (b) The graph showing the transformed edges (b_1, b_2) and (b_1, b_4) and the corresponding edge cost in the CCG.

Choosing an edge for the completion of the solution is termed a construction step. Each ant makes every construction step according to the state transition rule. After an ant makes a construction step, the local pheromone update is applied to the selected edge. The local pheromone update reduces the chances for the other ants to choose the same edge repetitively and thus ensures diversity in the search. After every ant has constructed a complete solution, the global pheromone update mechanism is applied to the edges of the best-so-far solution in order to intensify the attraction of the edges.

Figure 3 illustrates the ants' behavior in one iteration by showing the local and global pheromone update operations on the edges of a graph. When the ants are moving, the pheromones on the visited edges are changed by the local pheromone update, resulting in the reduction of pheromones. After all the ants have completed their solutions, the global pheromone update reinforces the pheromones on the edges of the best-so-far solution. In a complex search environment with multiple ramifications, the previous mechanisms were shown to be able to find high-quality solutions [18].

3. ACO for Minimum Cost Multicast Routing

In this section, the minimum cost multicast routing ant colony optimization (MCMRACO) algorithm is proposed for solving the unconstrained MCMR problems. A complete flowchart of the algorithm is shown in Figure 4(a).

3.1. Prim's Algorithm, Pheromone, and Heuristic Mechanisms. The ant's construction behavior in MCMRACO is similar to the generation of a minimum spanning tree (MST) by Prim's algorithm [41]. Suppose that V is the set of nodes in an undirected connected graph and W is a set containing only one node. The classical Prim's algorithm continuously moves a node d from $V - W$ to W, provided that $s \in W$ and the cost of edge (s, d) is minimum. In the proposed algorithm, the selection criterion for the next node is not simply based on the cost of the edges but is based on the product of the pheromone and heuristic values.

The pheromone and heuristic values of an edge (s, d) are denoted by $\tau(s, d)$ and $\eta(s, d)$, respectively. The initial pheromone value is $\tau_0 = 1/c(T_{\text{KMB}})$, where $c\ (T_{\text{KMB}})$ is the cost of the initial tree generated by the Kou-Markowsky-Berman (KMB) method [12].

The heuristic value $\eta(s, d)$ of the edge connecting nodes s and d is a function of the cost of the edge (s, d) and the type of the node d, that is

$$\eta(s,d) = \begin{cases} \dfrac{\mu}{\widehat{c}_{sd}}, & \text{if } d \in V_M \\ \dfrac{1}{\widehat{c}_{sd}}, & \text{otherwise}, \end{cases} \tag{2}$$

where μ ($\mu > 1$) is a reinforcement rate to the nodes in the multicast group. Heuristic values represent the quality of the candidate edges. Lower cost edges in the CCG are preferred. Moreover, multicast nodes have higher probabilities to be selected than intermediate nodes. If $\mu \to \infty$, the ants perform similarly to KMB, which only considers multicast nodes during the construction. If $\mu = 1$, the ants cannot differentiate multicast nodes from the intermediate nodes. So the value of μ influences the ants' sensitivity to the multicast nodes in the network.

3.2. The Ants' Search Behavior. In this subsection, we describe an ant's search behavior step by step. The corresponding flowchart is shown in Figure 4(b).

Step 1 (initialization). Initially, an ant k is placed on a randomly chosen multicast node s. The visited node set of ant k is denoted by $W^{(k)} = \{s\}$. The unvisited node set is $U^{(k)} = V_G - W^{(k)}$. The product of the pheromone and heuristic values for every unvisited node $i \in U^{(k)}$ is computed as

$$\omega_i = \max_{j \in W^{(k)}} \left(\tau(j,i) \cdot \eta(j,i)\right). \tag{3}$$

That is, $\omega_i = \tau(s, i) \cdot \eta(s, i)$ as $W^{(k)} = \{s\}$. By using χ_i to denote the corresponding visited node that connects to node $i \in U^{(k)}$ with the maximum product, we have

$$\chi_i = \arg \max_{j \in W^{(k)}} \left(\tau(j,i) \cdot \eta(j,i)\right) = s. \tag{4}$$

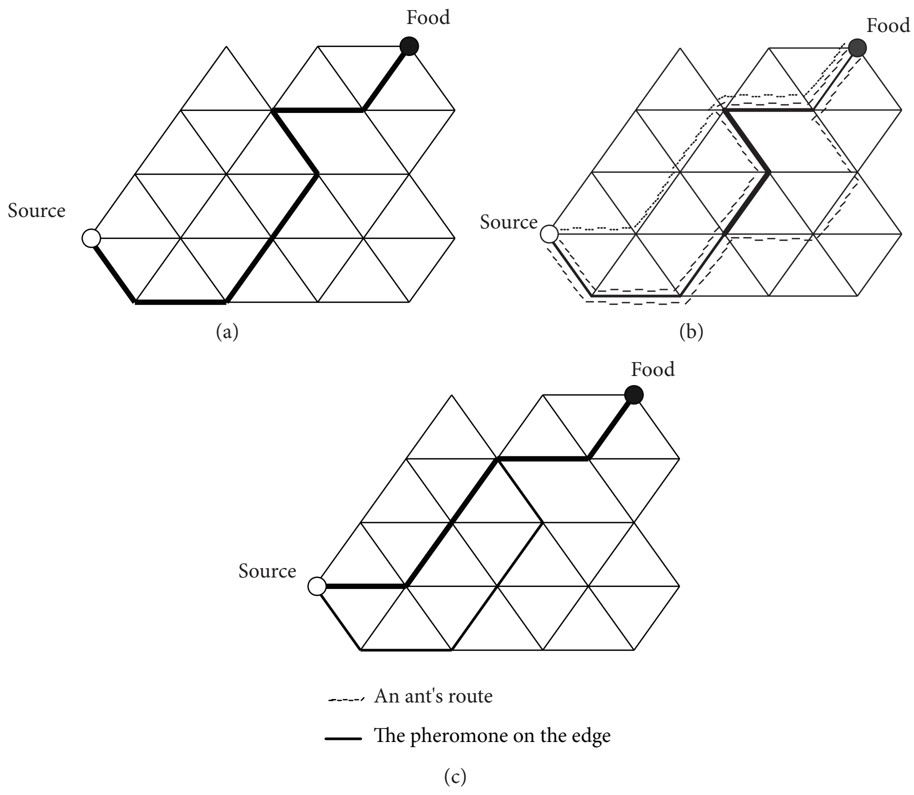

Figure 3: An illustration of the ants' behavior in one iteration. The thickness of the edges denotes the pheromone density. (a) The distribution of pheromones from the last iteration. (b) When the three ants are moving, the pheromones on the passed edges are changed by the local pheromone update, resulting in the reduction of pheromones. (c) The global pheromone update reinforces the pheromones on the edges of the best-so-far solution.

Step 2 (exploitation or exploration). Based on the state transition rule [20], the ant probabilistically chooses to do exploitation or exploration, which is controlled by

$$d = \begin{cases} \arg \max\limits_{r \in U^{(k)}} \omega_r, & \text{if } q \le q_0 \ (\text{exploitation}) \\ D, & \text{otherwise } (\text{biased exploration}), \end{cases} \quad (5)$$

where q_0 is a predefined parameter in $[0, 1]$ controlling the proportion of exploitation to exploration, q is a uniform random number in $[0, 1)$, and D is a random number selected according to a probability distribution as (6). If $q \le q_0$, the ant will choose the next unvisited node d that has the maximum product of the pheromone and heuristic values. This is called the exploitation step. Otherwise, the next node d is chosen according to (6), which is called the exploration step.

$$p_k(d) = \begin{cases} \dfrac{\omega_d}{\sum_{r \in U^{(k)}} \omega_r}, & \text{if } d \in U^{(k)} \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

Step 3 (edge extension and local pheromone update). When an ant is building a solution, the pheromone values on the visited edges are reduced. After choosing the next node d, ant k moves from node $s = \chi_d$ to node d. Since the ant moves in the CCG, the logical edge (s, d) can be extended to a physical route $\langle b_0, b_1, b_2, \ldots, b_\psi \rangle$, where $b_0 = s$, $b_\psi = d$, $b_l = o_{b_{l-1}, d}$, and $l = 1, \ldots, \psi - 1$. The edges (b_i, b_j) $(i = 0, \ldots, \psi, j = 0, \ldots, \psi, i \ne j)$ have the pheromones updated by

$$\tau(b_i, b_j) \longleftarrow (1 - \rho) \cdot \tau(b_i, b_j) + \rho \cdot \tau_{\min}, \quad (7)$$

where $\rho \in (0, 1)$ is the pheromone evaporation rate. The larger the value of ρ, the more pheromones are evaporated on the edges that the ant has just passed. The lower boundary of the pheromone value is set as $\tau_{\min} = \tau_0$ so that the updated value will not be smaller than the initial pheromone value. The unvisited nodes in $b_1, b_2, \ldots, b_\psi$ are marked as visited by moving them from $U^{(k)}$ to $W^{(k)}$.

Step 4 (has the ant finished the mission?). If an ant has visited all the multicast nodes ($V_M \subseteq W^{(k)}$), the ant has finished constructing a multicast tree. Otherwise, for all $i \in U^{(k)}$, update the values of ω_i and χ_i as

$$\begin{aligned} \omega_i &\longleftarrow \max_{j \in \{\chi_i, b_1, b_2, \ldots, b_\psi\}} (\tau(j, i) \cdot \eta(j, i)). \\ \chi_i &\longleftarrow \arg \max_{j \in \{\chi_i, b_1, b_2, \ldots, b_\psi\}} (\tau(j, i) \cdot \eta(j, i)). \end{aligned} \quad (8)$$

Then return to Step 2 for a further search.

3.3. Redundancy Trimming. After an ant has finished building a multicast tree connecting all multicast nodes,

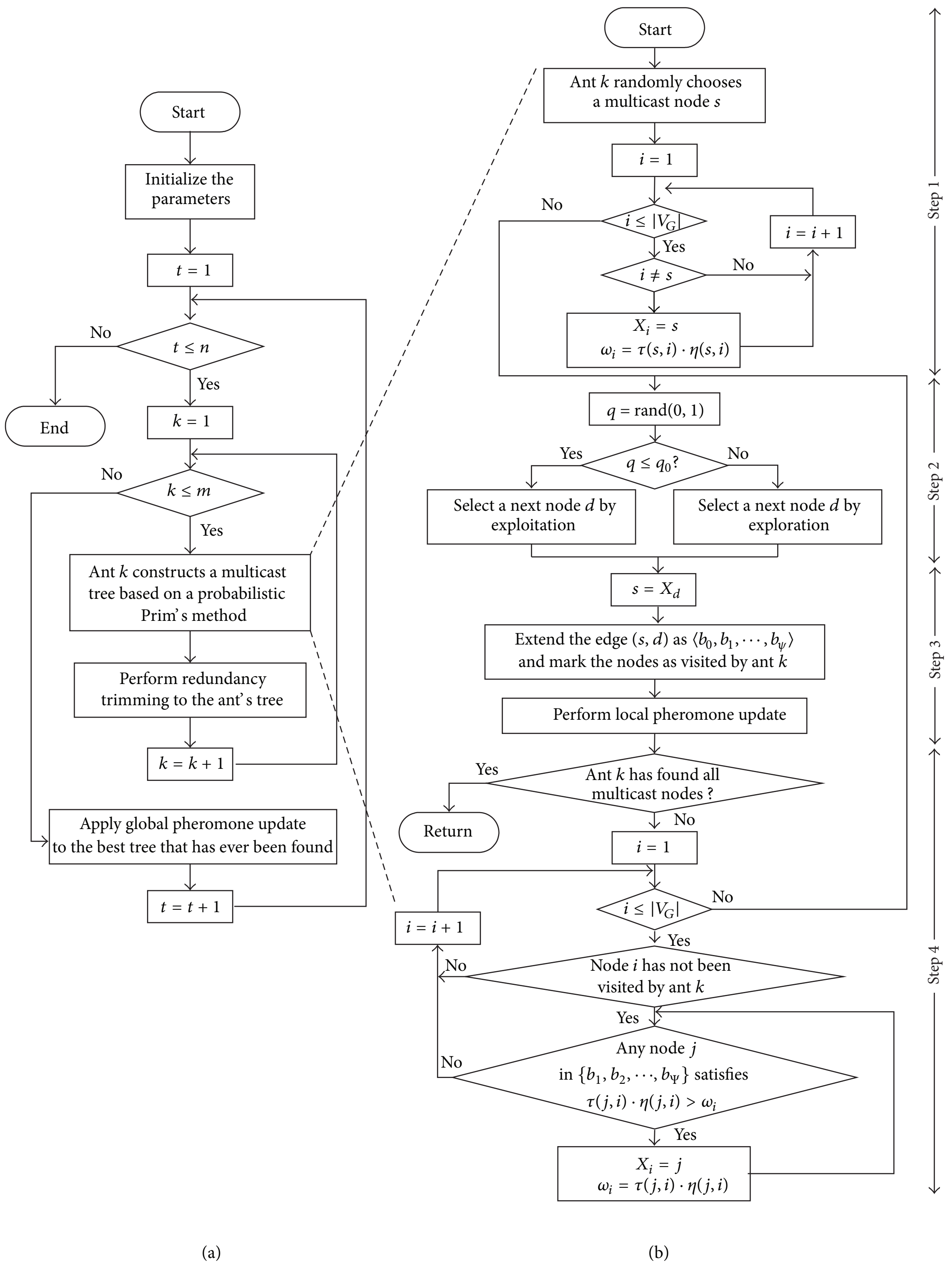
Start
Initialize the parameters
$t = 1$
$t \leq n$
No
Yes
End
$k = 1$
$k \leq m$
Ant k constructs a multicast tree based on a probabilistic Prim's method
Perform redundancy trimming to the ant's tree
$k = k + 1$
Apply global pheromone update to the best tree that has ever been found
$t = t + 1$
(a)
Start
Ant k randomly chooses a multicast node s
$i = 1$
$i \leq |V_G|$
$i = i + 1$
$i \neq s$
$X_i = s$
$\omega_i = \tau(s, i) \cdot \eta(s, i)$
$q = \text{rand}(0, 1)$
$q \leq q_0$?
Select a next node d by exploitation
Select a next node d by exploration
$s = X_d$
Extend the edge (s, d) as $\langle b_0, b_1, \cdots, b_\psi \rangle$ and mark the nodes as visited by ant k
Perform local pheromone update
Ant k has found all multicast nodes ?
Return
Node i has not been visited by ant k
Any node j in $\{b_1, b_2, \cdots, b_\Psi\}$ satisfies $\tau(j, i) \cdot \eta(j, i) > \omega_i$
$X_i = j$
$\omega_i = \tau(j, i) \cdot \eta(j, i)$
Step 1
Step 2
Step 3
Step 4
(b)

FIGURE 4: Flowchart of the proposed minimum cost multicast routing ant colony optimization (MCMRACO).

the tree must be checked for redundancy. The flowchart of this process is given in Figure 5.

Firstly, apply the classical Prim's algorithm to the visited nodes in the network graph. If the cost of the generated MST is smaller than that of the tree built by the ant, the ant's solution is replaced by the MST.

Secondly, check for useless intermediate nodes. Delete the one-degree intermediate nodes and their connected edges.

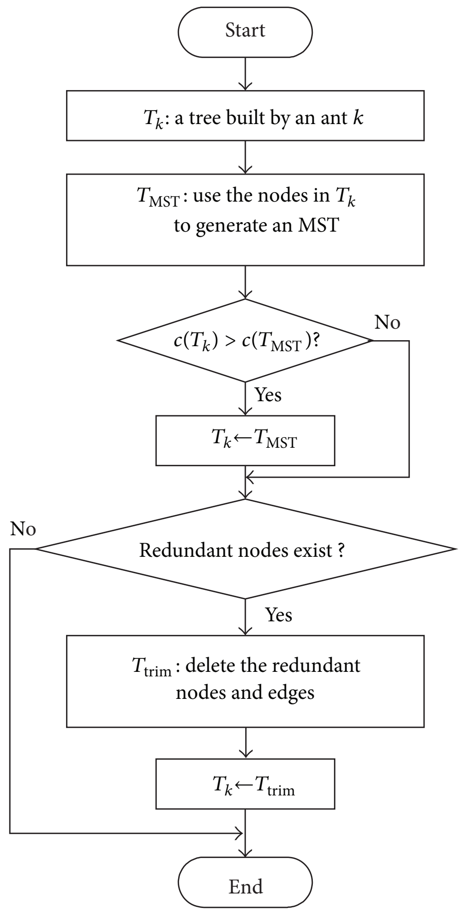
Start
T_k: a tree built by an ant k
T_{MST}: use the nodes in T_k to generate an MST
$c(T_k) > c(T_{\mathrm{MST}})$?
No
Yes
$T_k \leftarrow T_{\mathrm{MST}}$
Redundant nodes exist ?
No
Yes
T_{trim}: delete the redundant nodes and edges
$T_k \leftarrow T_{\mathrm{trim}}$
End

Figure 5: Flowchart of the redundancy trimming.

3.4. Global Pheromone Update. After all the m ants have finished the tree construction, the ants' solutions are evaluated and the best-so-far tree is updated. The global pheromone update is applied only to the best-so-far tree. The pheromone values on the edges (i, j) of the best-so-far (bsf) tree T_{bsf} are updated as

$$\tau(i, j) \longleftarrow (1-\rho)\cdot\tau(i, j) + \alpha\cdot\Delta\tau, \tag{9}$$

where $\Delta\tau = 1/(c(T_{\mathrm{bsf}}))c\,(T_{\mathrm{bsf}})$ is the cost of T_{bsf} and α is the pheromone reinforcement rate with $\alpha \geq \rho$.

Moreover, the pheromone values on some logical edges in the tree T_{bsf} are also updated. If the extended physical route between a pair of nodes i and $j(i, j \in T_{\mathrm{bsf}})$ is $\langle b_0, b_1, b_2, \ldots, b_\psi\rangle$, which satisfies $b_0 = i$, $b_\psi = j$, $b_r = o_{b_{r-1},j}$, $r = 1, \ldots, \psi-1$, $(b_l, b_l+1) \in T_{\mathrm{bsf}}$, $l = 0, 1, \ldots, \psi-1$, and $\psi > 1$ is the number of edges in the route, then the new pheromone value of edge (i, j) becomes

$$\tau(i, j) \longleftarrow \frac{\sum_{l=0}^{\psi-1}\tau(b_l, b_{l+1})}{\psi}. \tag{10}$$

The updated pheromone value of edge (i, j) is in accordance with the average pheromone value of the corresponding physical edges in the tree.

3.5. The Complexity and Convergence of MCMRACO. The time complexity of the proposed MCMRACO can be estimated by counting the number of multicast trees that are generated during the optimization process. In each iteration of MCMRACO, a colony of m ants construct m trees and the classical Prim's algorithm is performed m times for redundancy trimming. As the time complexity for constructing a multicast tree is $O(|V_G|^2)$ and $2m$ trees are constructed in each iteration, the time complexity of MCMRACO is approximately $O(2nm|V_G|^2)$, where n is the predefined maximum iteration number.

According to [43], the convergence condition for the proposed algorithm is to satisfy $0 < \tau_{\min} < \tau_{\max} < +\infty$, where $\tau_{\min}$ and $\tau_{\max}$ are the lower and upper boundaries of the pheromone value, respectively. Although the heuristic and pheromone mechanisms have been redesigned in the proposed algorithm, the convergence of the algorithm is still maintained. The lower boundary of the pheromone value is $\tau_{\min} = \tau_0 > 0$. The upper boundary of the pheromone value is $(\alpha/\rho)\cdot(1/c(T_{\mathrm{opt}}))$, where $c(T_{\mathrm{opt}})$ is the cost of the optimal tree. Furthermore, every ant in MCMRACO builds a feasible multicast solution. Therefore, the proposed MCMRACO can converge to an optimal solution.

4. Experiments and Discussions

The experiments in this paper are composed of two parts. The first part is the experiment on the static MR cases, whereas the second part is the one on the dynamic MR cases. All the results in the experiments are computed by a computer with CPU Pentium IV 2.8 GHZ, memory 256 MB.

4.1. Static Multicast Routing Cases. The static MR cases are the Steiner problems in group B from the OR-Library [44]. The eighteen problems are tabulated in Table 1, with the graph size from 50 to 100. In the table, $|V_G|$ stands for the number of nodes, $|V_M|$ is the number of multicast nodes, $|E_G|$ is the number of physical edges in the graph, and $c(T_{\mathrm{opt}})$ is the cost of the optimal tree. The performance of the proposed algorithm is compared with the KMB heuristic algorithm in [12], the random neural network (RNN) algorithm in [16], the genetic algorithm (GA) in [17], and the ant algorithm in [38].

Firstly, the parameter settings of the proposed MCMRACO algorithm are analyzed. The proposed algorithm has five parameters, which are the number of ants m, the reinforcement rate to destination nodes μ, the proportion of exploitation q_0, the pheromone evaporation rate ρ, and the pheromone reinforcement rate α. The number of ants m depends on the number of nodes in the network. More ants may perform better in a large network, but it will take longer time in one iteration. If the number of ants is not enough, the algorithm may be trapped easily in suboptimal results. The value $m = 50$ is suitable for most of the networks in our empirical study.

The other four parameters μ, q_0, ρ, and α are analyzed by testing $\mu = 5, 6, \ldots, 19$, $q_0 = 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.85, 0.9, 0.95$, $\rho = 0.1, 0.2, \ldots, 0.5$, and $\alpha = 0.1, 0.2, \ldots, 1.1$ with $\alpha \geq \rho$. Each combination of parameter values (μ, q_0, ρ, α)

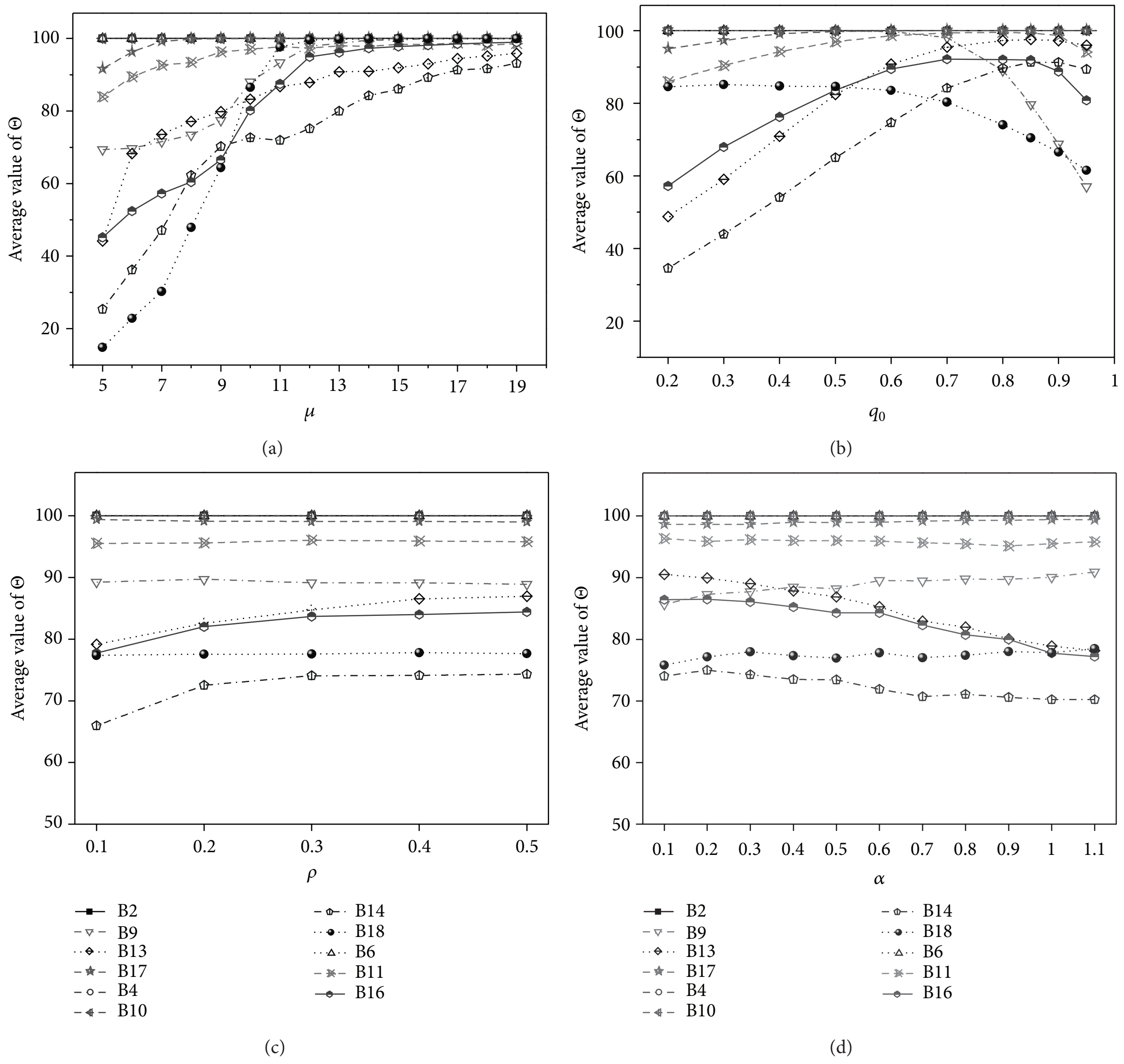

Figure 6: Average performance of MCMRACO with different parameter values in solving the static MR cases. (a) Average value of Θ with $\mu = 5, 6, \ldots, 19$. (b) Average value of Θ with $q_0 = 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.85, 0.9, 0.95$. (c) Average value of Θ with $\rho = 0.1, 0.2, \ldots, 0.5$. (d) Average value of Θ with $\alpha = 0.1, 0.2, \ldots, 1.1$.

is termed a parameter configuration. Each configuration is tested in ten independent runs by MCMRACO on each of the eighteen instances. Each run of MCMRACO terminates when it cannot find a better result in three consecutive iterations. The performance of MCMRACO by configuration θ on instance i is measured as

$$\Theta(\theta, i) = 100\sigma - \bar{t}, \tag{11}$$

where σ is the successful percentage and $\bar{t}$ denotes the average time in seconds of the ten independent runs by configuration θ on instance i for obtaining the optimal solution.

Figure 6 shows the average performance of MCMRACO with different parameter values in solving some static MR instances. Each point in the figure is drawn as follows. For example, the total measurements of the performance Θ for $\mu = 5$ of B9 are 31235.4. When μ equals to 5, there are 10 choices for q_0 and 45 choices for the combinations of (ρ, α). So the total configuration number of (μ, q_0, ρ, α) with $\mu = 5$ is 450. The average measurement of performance Θ for $\mu = 5$ of B9 is thus computed as $31235.4/450 = 69.412$, which has been plotted as a point in Figure 6(a). Only the instances that cannot be solved successfully by the KMB algorithm are considered in the figure. The curves show that the parameter values have similar influences on different instances. (1) The successful percentage increases when the value of μ becomes bigger. It means that the ants should have a strong bias towards the multicast nodes. (2) The desirable value of q_0 is in the range of [0.6,0.9]. The results indicate that the probability for performing exploitation is better to

Table 1: Eighteen Steiner B test cases.

No.	$\|V_G\|$	$\|E_G\|$	$\|V_M\|$	$c(T_{\text{opt}})$
1	50	63	9	82
2	50	63	13	83
3	50	63	25	138
4	50	100	9	59
5	50	100	13	61
6	50	100	25	122
7	75	94	13	111
8	75	94	19	104
9	75	94	38	220
10	75	150	13	86
11	75	150	19	88
12	75	150	38	174
13	100	125	17	165
14	100	125	25	235
15	100	125	50	318
16	100	200	17	127
17	100	200	25	131
18	100	200	50	218

be higher than that of the exploration. (3) With $\alpha \geq \rho$, the influences for choosing different values of α and ρ are quite small for the same instance.

After analyzing the influence of different parameter values, the values of the four parameters μ, q_0, ρ, and α are selected based on a statistically sound approach, that is, a racing algorithm termed F-Race [45]. According to the results by F-Race, the parameter values of the proposed MCMRACO are set as $m = 50$, $\mu = 19$, $q_0 = 0.8$, $\rho = 0.3$, and $\alpha = 0.5$.

The results for solving the Steiner B instances are tabulated in Table 2. As there is no stochastic factor in KMB and RNN, the results are unique in different runs. The ant algorithm in [38], the GA in [17], and the proposed MCMRACO are probabilistic algorithms, so they are tested ten times independently for each instance. If the results are equal to the optima, they are bold in the table.

The results show that KMB only solves seven instances successfully, whereas RNN obtains eleven successful results out of the eighteen instances. The ant algorithm in [37] can find the best multicast trees of the instances at least once except for B16 and B18. However, it can only achieve 100% success in eight instances. GA terminates when it cannot find a better result in twenty consecutive generations with a population size of 50. Note that the termination condition of GA is more stringent than the one used in [17] but is looser than that of MCMRACO. The results show that GA can find the optima of all the eighteen instances, but it can only solve eleven instances with 100% success. The proposed MCMRACO has the best performance among the algorithms, for it can solve all the static instances with 100% success. To further study the performance of the algorithms, a dynamic environment is designed in the next subsection.

4.2. Dynamic Multicast Routing Cases. Nodes in the network may join or leave the multicast group. Suppose the network topology is stable without failure. We design the following dynamic scenario to test the stability of the algorithm. When the multicast nodes are changed, the algorithm is invoked to find a new multicast tree.

Suppose that v multicast nodes in the multicast group leave and v new nodes join the group alternatively, forming a dynamic situation. Note that the nodes in the multicast group remain unchanged when the algorithm is still running. Initially, there is a network $G = (V_G, E_G)$ including a multicast group $V_M(0) \subseteq V_G$. At time 1, v multicast nodes are chosen randomly to become intermediate nodes, indicating that v nodes leave the multicast group. Then the multicast group becomes $V_M(1) = V_M(0) - \Delta V_M(1)$ and a multicast tree $T(1)$ is computed. At time 2, v intermediate nodes are chosen randomly to become multicast nodes, indicating that v nodes join the multicast group. Then the multicast group becomes $V_M(2) = V_M(1) + \Delta V_I(2)$ and a new multicast tree $T(2)$ is computed. Given the value of v, the nodes in the multicast group are changed $2K$ times and the objective of the experiment is to minimize the total cost of the multicast trees. The formal definition is

$$\Phi_v = \min \sum_{i=1}^{2K} \sum_{e \in E(i)} c(e),$$

$$T(i) = (V_M(i) + V_\theta(i), E(i)), \quad i = 1, 2, \ldots, 2K,$$

$$V_M(i) = V_M(i-1) - \Delta V_M(i), \quad \text{if } i = 1, 3, 5, \ldots, 2K-1,$$

$$V_M(i) = V_M(i-1) + \Delta V_I(i), \quad \text{if } i = 2, 4, 6, \ldots, 2K, \tag{12}$$

where $V_\theta(i) \subseteq V_G - V_M(i)$, $E(i) \subseteq E_G$. $\Delta V_M(i)$ is the set of the v randomly chosen multicast nodes to leave the multicast group when $i = 1, 3, 5, \ldots, 2K-1$. $\Delta V_I(i)$ is the set of the v randomly chosen intermediate nodes to join the multicast group when $i = 2, 4, 6, \ldots, 2K$.

There are eighteen dynamic multicast routing instances, which use the static Steiner B instances as their initial MR networks, respectively. For each instance, the values $v = 1, 2, \ldots, |V_M| - 3$ are tested. When the value of v grows, the multicast group becomes more and more unstable.

The proposed MCMRACO takes advantage of pheromones to encode a memory about the ants' search process. After the search for a multicast tree is finished, the pheromones are still maintained in the network, reflecting the previous routing information. Once the multicast nodes are changed, the ants' construction is restarted. The pheromones still bias the ants to select the previous edges. As the ants reduce pheromones on edges by the local pheromone update, the influences of the obsolete routing information decrease.

The results of the summation cost of the $2K$ multicast trees, which are computed by MCMRACO, RNN, and GA, are denoted by $\Phi_v^{(M)}$, $\Phi_v^{(R)}$, and $\Phi_v^{(G)}$, respectively. The comparison result of the tree cost between RNN and MCMRACO is denoted by $\delta^{(R)}$, whereas the one between GA

Table 2: Optimization results for the static experiment.

No.	$c(T_{\text{opt}})$	KMB	RNN	Ant		GA		MCMRACO	
				Best	Mean	Best	Mean	Best	Mean
1	82	**82**	**82**	**82**	**82**	**82**	**82**	**82**	**82**
2	83	88	**83**	**83**	**83**	**83**	**83**	**83**	**83**
3	138	**138**	**138**	**138**	**138**	**138**	**138**	**138**	**138**
4	59	64	**59**	**59**	**59**	**59**	**59**	**59**	**59**
5	61	**61**	**61**	**61**	**61**	**61**	**61**	**61**	**61**
6	122	127	**122**	—	—	**122**	**122**	**122**	**122**
7	111	**111**	**111**	—	—	**111**	**111**	**111**	**111**
8	104	**104**	**104**	**104**	**104**	**104**	**104**	**104**	**104**
9	220	221	**220**	**220**	**220**	**220**	**220**	**220**	**220**
10	86	91	87	**86**	87.4	**86**	**86**	**86**	**86**
11	88	92	90	**88**	89	**88**	88.2	**88**	**88**
12	174	**174**	**174**	—	—	**174**	**174**	**174**	**174**
13	165	175	175	**165**	167.3	**165**	165.8	**165**	**165**
14	235	237	237	**235**	235.3	**235**	237.3	**235**	**235**
15	318	**318**	**318**	**318**	**318**	**318**	318.6	**318**	**318**
16	127	135	135	132	133	**127**	127.3	**127**	**127**
17	131	134	132	—	—	**131**	131.1	**131**	**131**
18	218	221	219	224	225.5	**218**	218.4	**218**	**218**

and MCMRACO is denoted by $\delta^{(G)}$. The definitions of $\delta^{(R)}$ and $\delta^{(G)}$ are

$$\delta^{(R)} = \frac{\Phi_v^{(R)} - \Phi_v^{(M)}}{\Phi_v^{(M)}}, \qquad \delta^{(G)} = \frac{\Phi_v^{(G)} - \Phi_v^{(M)}}{\Phi_v^{(M)}}. \tag{13}$$

Table 3 lists the tree cost of MCMRACO and the comparison results of the eighteen initial MR networks when $v = 1$, $\lfloor(|V_M| - 3)/2 + 1\rfloor$, $|V_M| - 3$, and $K = 100$. As the comparison results are all positive, the proposed MCMRACO can obtain multicast trees with less cost than RNN and GA. Figure 7 shows the comparison results of the tree cost with B18 as the initial MR network when the values of v are changed from 1 to 47. The differences between RNN and MCMRACO are always larger than those between GA and MCMRACO. The figure presents that the proposed MCMRACO can steadily achieve better results than RNN and GA.

When the multicast group is changed, the time used for finding the solution is the response delay. The time needed by RNN is determined by its enumeration subset. The smaller the multicast group, the more nodes that may be used for enumeration, resulting in longer computation time. However, for GA and MCMRACO, the time used for a small group is generally shorter than a large group in the same termination condition. Figure 8 illustrates the average time used by RNN, GA, and MCMRACO with different values of v when the initial multicast network is B18. The curves show that the average time needed by GA and MCMRACO reduces as the value of v becomes bigger, but the time used by RNN is growing upwards.

Table 4 tabulates the average time used for finding a multicast solution in the 18 instances when = 1, $\lfloor(|V_M| - 3)/2 + 1\rfloor$, $|V_M| - 3$. Although the average time used by

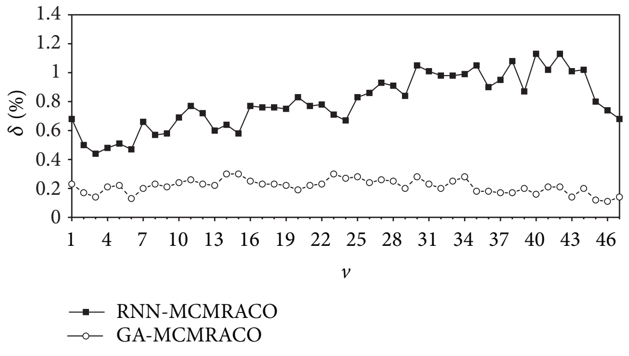

Figure 7: Comparison results of the tree cost between RNN and MCMRACO and between GA and MCMRACO with different values of v when the initial MR network is B18.

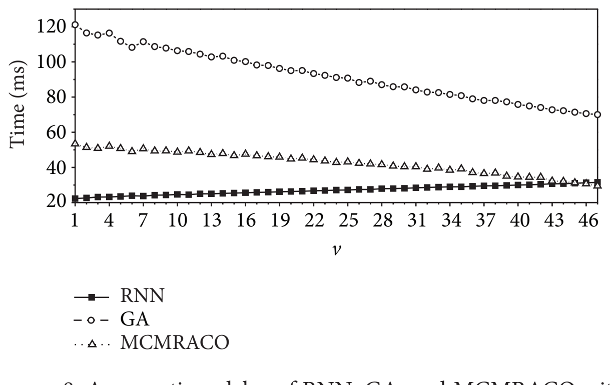

Figure 8: Average time delay of RNN, GA, and MCMRACO with different values of v when the initial MR network is B18.

Table 3: Comparison results of the tree cost for the dynamic experiment.

No.	v	$\Phi_v^{(M)}$	$\delta^{(R)}$	$\delta^{(G)}$
1	1	16381	1.33%	0.00%
	4	12570	0.99%	0.00%
	6	10426	0.55%	0.00%
2	1	17345	1.32%	0.00%
	6	17553	1.88%	0.01%
	10	14399	1.51%	0.03%
3	1	25656	0.27%	0.00%
	12	23588	0.45%	0.01%
	22	17162	0.41%	0.02%
4	1	10756	1.78%	0.01%
	4	9243	1.36%	0.02%
	6	7736	1.31%	0.00%
5	1	12687	1.14%	0.00%
	6	10374	1.33%	0.03%
	10	8320	0.91%	0.02%
6	1	22206	0.43%	0.04%
	12	18168	0.62%	0.04%
	22	13142	0.69%	0.02%
7	1	22392	0.64%	0.02%
	6	18322	1.66%	0.04%
	10	14574	1.22%	0.06%
8	1	25664	1.48%	0.00%
	9	22945	1.16%	0.02%
	16	17236	0.68%	0.02%
9	1	44687	0.41%	0.01%
	18	38390	0.51%	0.02%
	35	26371	0.40%	0.04%
10	1	16525	1.11%	0.11%
	6	13361	2.63%	0.08%
	10	10803	2.02%	0.19%
11	1	16823	0.81%	0.08%
	9	15501	1.90%	0.19%
	16	11748	2.03%	0.14%
12	1	31642	0.63%	0.02%
	18	27443	0.89%	0.08%
	35	19069	0.79%	0.06%
13	1	30795	1.90%	0.25%
	8	25326	3.03%	0.08%
	14	21049	2.91%	0.11%
14	1	40368	0.97%	0.13%
	12	33929	1.45%	0.18%
	22	24475	1.03%	0.24%
15	1	62009	0.84%	0.09%
	24	52308	1.13%	0.10%
	47	35118	0.81%	0.06%
16	1	19557	1.71%	0.25%
	8	17001	2.06%	0.38%
	14	13228	1.81%	0.26%
17	1	24107	1.08%	0.17%
	12	18215	1.18%	0.15%
	22	13003	1.12%	0.20%
18	1	45115	0.68%	0.23%
	24	36000	0.67%	0.27%
	47	24589	0.69%	0.15%

Table 4: Computation time delay for the dynamic experiment.

No.	v	Time (microsecond)		
		RNN	GA	MCMRACO
1	1	4.69	17.42	7.42
	4	5.00	15.08	5.39
	6	5.08	14.07	5.00
2	1	4.38	19.92	8.60
	6	4.61	18.60	7.34
	10	4.85	16.80	6.02
3	1	3.05	33.05	14.06
	12	3.52	28.83	12.27
	22	4.22	23.44	8.91
4	1	5.08	16.88	6.10
	4	5.32	15.00	4.92
	6	5.39	14.30	4.30
5	1	4.69	20.71	8.44
	6	5.00	18.05	7.03
	10	5.16	16.17	5.47
6	1	3.36	35.78	13.91
	12	3.91	29.77	11.80
	22	4.45	23.91	8.52
7	1	15.08	27.42	13.28
	6	15.55	24.61	11.25
	10	16.02	22.03	8.91
8	1	13.75	36.25	17.42
	9	14.53	31.33	15.08
	16	15.47	26.80	11.48
9	1	9.14	68.05	30.16
	18	10.94	56.48	26.56
	35	13.05	43.91	18.29
10	1	15.63	27.74	11.88
	6	16.10	24.54	10.00
	10	16.64	22.11	7.89
11	1	14.38	35.86	15.86
	9	15.16	31.80	14.22
	16	15.86	26.96	10.71
12	1	10.47	68.44	28.99
	18	12.11	56.72	24.92
	35	13.99	44.38	17.50
13	1	40.16	42.11	22.74
	8	36.48	36.25	19.69
	14	38.20	32.97	16.49
14	1	31.10	59.85	33.60
	12	33.05	49.84	27.50
	22	35.40	40.94	20.16
15	1	21.95	116.88	51.80
	24	26.02	91.96	45.32
	47	30.31	69.92	30.39
16	1	35.08	42.50	21.18
	8	36.33	38.52	18.68
	14	37.50	33.36	14.53
17	1	31.64	58.44	30.47
	12	33.75	47.74	25.63
	22	35.78	39.46	18.36
18	1	22.27	121.09	53.44
	24	27.11	91.10	42.50
	47	31.49	70.00	29.61

RNN is shorter than that of MCMRACO in some cases, the results obtained by MCMRACO are always better than those of RNN (as Table 3). Moreover, MCMRACO performs faster than RNN in instances Nos. 7, 10, 13, 14, 16, and 17. For GA and MCMRACO, the results show that even if GA takes longer time than MCMRACO, the results obtained by GA are still worse than those of the proposed algorithm. For example, the proposed MCMRACO uses only 7.42 microseconds (ms) on average to find the result of instance No. 1 when $v = 1$, whereas the GA needs 17.42 ms. For the instance No. 18 when $v = 47$, the proposed MCMRACO uses only 29.61 ms to obtain the result, whereas GA needs 70.00 ms and the result is still 0.15% worse than the proposed algorithm. Overall, the proposed MCMRACO performs better than RNN and GA in solving the MCMR problems.

5. Conclusion

This paper proposes a minimum cost multicast routing ant colony optimization (MCMRACO) algorithm for solving the minimum cost multicast routing (MCMR) problems. Different from the traditional algorithms for MCMR, the proposed MCMRACO utilizes the ant colony optimization (ACO) technique to search for an optimal multicast tree in the network graph. The artificial ants in the algorithm are based on a modified Prim's algorithm to build a tree. The heuristic information represents problem-specific knowledge for the ants to construct solutions, whereas the pheromones on edges reserve the previous routing information. By designing effective heuristic and pheromone mechanisms, the proposed MCMRACO is very competitive for solving MCMR problems. The performance of MCMRACO has been compared with the published results available in the literature. The comparison results show that the proposed MCMRACO works successfully in both static and dynamic MCMR cases. The proposed algorithm is protocol independent so that it can be implemented conveniently in different network environments. Extending the proposed algorithm to heterogeneous networks is a promising future research topic.

Acknowledgments

This work was supported in part by the National High-Technology Research and Development Program ("863" Program) of China under Grant no. 2013AA01A212, by the National Science Fund for Distinguished Young Scholars under Grant 61125205, by the National Natural Science Foundation of China under Grants 61070004, 61202130, by the NSFC Joint Fund with Guangdong under Key Projects U1201258 and U1135005, by the Guangdong Natural Science Foundation S2012040007948, by the Fundamental Research Funds for the Central Universities no. 12lgpy47, and by the Specialized Research Fund for the Doctoral Program of Higher Education 20120171120027.

References

[1] K. Bharath-Kumar and J. M. Jaffe, "Routing to multiple destinations in computer networks," *IEEE Transactions on Communications*, vol. 31, no. 3, pp. 343–351, 1983.

[2] Y. Yang, J. Wang, and M. Yang, "A service-centric multicast architecture and routing protocol," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 1, pp. 35–51, 2008.

[3] D.-N. Yang and W. J. Liao, "On bandwidth-efficient overlay multicast," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 11, pp. 1503–1515, 2007.

[4] A. Ganjam and H. Zhang, "Internet multicast video delivery," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 159–170, 2005.

[5] D.-N. Yang and M.-S. Chen, "Efficient resource allocation for wireless multicast," *IEEE Transactions on Mobile Computing*, vol. 7, no. 4, pp. 387–400, 2008.

[6] S. Guo and O. Yang, "Localized operations for distributed minimum energy multicast algorithm in mobile ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 2, pp. 186–198, 2007.

[7] J. A. Sanchez, P. M. Ruiz, J. Liu, and I. Stojmenovic, "Bandwidth-efficient geographic multicast routing protocol for wireless sensor networks," *IEEE Sensors Journal*, vol. 7, no. 5, pp. 627–636, 2007.

[8] F. Filali, G. Aniba, and W. Dabbous, "Efficient support of IP multicast in the next generation of GEO satellites," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 2, pp. 413–425, 2004.

[9] E. Ekici, I. F. Akyildiz, and M. D. Bender, "A multicast routing algorithm for LEO satellite IP networks," *IEEE/ACM Transactions on Networking*, vol. 10, no. 2, pp. 183–192, 2002.

[10] E. N. Gilbert and H. O. Pollak, "Steiner minimal trees," *SIAM Journal on Applied Mathematics*, vol. 16, pp. 1–29, 1968.

[11] M. R. Garey and D. S. Johnson, *Computers and Intractability*, W. H. Freeman and Co., San Francisco, Calif, USA, 1979.

[12] L. Kou, G. Markowsky, and L. Berman, "A fast algorithm for Steiner trees," *Acta Informatica*, vol. 15, no. 2, pp. 141–145, 1981.

[13] V. J. Rayward-Smith and A. Clare, "On finding Steiner vertices," *Networks*, vol. 16, no. 3, pp. 283–294, 1986.

[14] V. J. Rayward-Smith, "The computation of nearly minimal Steiner trees in graphs," *International Journal of Mathematical Education in Science and Technology*, vol. 14, no. 1, pp. 15–23, 1983.

[15] P. Winter and J. M. Smith, "Path-distance heuristics for the Steiner problem in undirected networks," *Algorithmica*, vol. 7, no. 2-3, pp. 309–327, 1992.

[16] E. Gelenbe, A. Ghanwani, and V. Srinivasan, "Improved neural heuristics for multicast routing," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 2, pp. 147–155, 1997.

[17] Y. Leung, G. Li, and Z.-B. Xu, "A genetic algorithm for the multiple destination routing problems," *IEEE Transactions on Evolutionary Computation*, vol. 2, no. 4, pp. 150–161, 1998.

[18] M. Dorigo and T. Stützle, *Ant Colony Optimization*, MIT Press, Cambridge, Mass, USA, 2004.

[19] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 26, no. 1, pp. 29–41, 1996.

[20] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.

[21] C. Solnon, "Ants can solve constraint satisfaction problems," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 347–357, 2002.

[22] Y.-C. Liang and A. E. Smith, "An ant colony optimization algorithm for the redundancy allocation problem (RAP)," *IEEE Transactions on Reliability*, vol. 53, no. 3, pp. 417–423, 2004.

[23] W.-N. Chen and J. Zhang, "Ant colony optimization for software project scheduling and staffing with an event-based scheduler," *IEEE Transactions on Software Engineering*, vol. 39, no. 1, pp. 1–17, 2013.

[24] W. N. Chen, J. Zhang, H. S. H. Chung, R. Z. Huang, and O. Liu, "Optimizing discounted cash flows in project scheduling-an ant colony optimization approach," *IEEE Transactions on Systems, Man and Cybernetics C*, vol. 40, no. 1, pp. 64–77, 2010.

[25] S. Guo, H.-Z. Huang, Z. Wang, and M. Xie, "Grid service reliability modeling and optimal task scheduling considering fault recovery," *IEEE Transactions on Reliability*, vol. 60, no. 1, pp. 263–274, 2011.

[26] W.-N. Chen and J. Zhang, "An ant colony optimization approach to a grid workflow scheduling problem with various QoS requirements," *IEEE Transactions on Systems, Man and Cybernetics C*, vol. 39, no. 1, pp. 29–43, 2009.

[27] D. Merkle, M. Middendorf, and H. Schmeck, "Ant colony optimization for resource-constrained project scheduling," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 333–346, 2002.

[28] R. S. Parpinelli, H. S. Lopes, and A. A. Freitas, "Data mining with an ant colony optimization algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 321–332, 2002.

[29] D. Martens, M. de Backer, R. Haesen, J. Vanthienen, M. Snoeck, and B. Baesens, "Classification with ant colony optimization," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 5, pp. 651–665, 2007.

[30] A. C. Zecchin, A. R. Simpson, H. R. Maier, and J. B. Nixon, "Parametric study for an ant algorithm applied to water distribution system optimization," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 2, pp. 175–191, 2005.

[31] J. Zhang, H. S.-H. Chung, A. W.-L. Lo, and T. Huang, "Extended ant colony optimization algorithm for power electronic circuit design," *IEEE Transactions on Power Electronics*, vol. 24, no. 1, pp. 147–162, 2009.

[32] A. Konak and S. Kulturel-Konak, "Reliable server assignment in networks using nature inspired metaheuristics," *IEEE Transactions on Reliability*, vol. 60, no. 2, pp. 381–393, 2011.

[33] K. M. Sim and W. H. Sun, "Ant colony optimization for routing and load-balancing: survey and new directions," *IEEE Transactions on Systems, Man, and Cybernetics A*, vol. 33, no. 5, pp. 560–572, 2003.

[34] R. Schoonderwoerd, O. Holland, and J. Bruten, "Ant-like agents for load balancing in telecommunications networks," in *Proceedings of the 1st International Conference on Autonomous Agents (Agents '97)*, pp. 209–216, New York, NY, USA, February 1997.

[35] G. di Caro and M. Dorigo, "AntNet: distributed stigmergetic control for communications networks," *Journal of Artificial Intelligence Research*, vol. 9, pp. 317–365, 1998.

[36] S. S. Iyengar, H.-C. Wu, N. Balakrishnan, and S. Y. Chang, "Biologically inspired cooperative routing for wireless mobile sensor networks," *IEEE Systems Journal*, vol. 1, no. 1, pp. 29–37, 2007.

[37] O. H. Hussein, T. N. Saadawi, and M. J. Lee, "Probability routing algorithm for mobile ad hoc networks' resources management," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 12, pp. 2248–2259, 2005.

[38] G. Singh, S. Das, S. V. Gosavi, and S. Pujar, "Ant colony algorithms for Steiner trees: an application to routing in sensor networks," in *Recent Developments in Biologically Inspired Computing*, L. N. de Castro and F. J. von Zuben, Eds., pp. 181–206, Idea Group Publishing, Hershey, Pa, USA, 2004.

[39] C.-C. Shen and C. Jaikaeo, "Ad hoc multicast routing algorithm with swarm intelligence," *Mobile Networks and Applications*, vol. 10, no. 1, pp. 47–59, 2005.

[40] C.-C. Shen, K. Li, C. Jaikaeo, and V. Sridhara, "Ant-based distributed constrained steiner tree algorithm for jointly conserving energy and bounding delay in ad hoc multicast routing," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 3, no. 1, article 3, 2008.

[41] R. C. Prim, "Shortest connection networks and some generalizations," *Bell System Technical Journal*, vol. 36, pp. 1389–1401, 1957.

[42] R. W. Floyd, "Shortest path," *Communications of the ACM*, vol. 5, no. 6, p. 345, 1962.

[43] T. Stützle and M. Dorigo, "A short convergence proof for a class of ant colony optimization algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 358–365, 2002.

[44] J. E. Beasley, "OR-Library: distributing test problems by electronic mail," *Journal of Operational Research Society*, vol. 41, no. 11, pp. 1069–1072, 1990.

[45] M. Birattari, T. Stützle, L. Paquete, and K. Varrentrapp, "A racing algorithm for configuring metaheuristics," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '02)*, pp. 11–18, 2002.