International Journal of Computer and Information Technology (ISSN: 2279 – 0764)
Volume 05 – Issue 06, November 2016

Tree Growth Based ACO Algorithm for Solving the Bandwidth-Delay-Constrained Least-Cost Multicast Routing Problem

Moheb R. Girgis*, Tarek M. Mahmoud, Ghada W. Hanna
Department of Computer Science, Faculty of Science, Minia University, El-Minia, Egypt
*Email: moheb.girgis [AT] mu.edu.eg

Abstract—**Quality of service (QoS) multicast routing is an NP multi-objective optimization problem. This paper presents a tree-growth based ant colony optimization (ACO) algorithm (TGACO) for solving the least-cost multicast routing problem with three QoS constraints, namely: bandwidth, delay and delay jitter. In the proposed algorithm, each ant generates a multicast tree using tree growth, such that an edge is added to the tree if it satisfies only the bandwidth constraint. Then, the fitness of the constructed multicast tree is evaluated by using a cost function that includes the delay and delay jitter constraints. Depending on the fitness of the constructed multicast trees, the local and global best multicast trees can be determined. In the TGACO algorithm, the ants perform local and global pheromone updates. In the local pheromone update, pheromone evaporation is performed by all ants after each construction step, while the global pheromone update is performed at the end of each iteration by the local best and the global best ants. The paper also presents the results of the experiments that have been conducted to evaluate the performance of the proposed algorithm.**

Keywords-QoS multicast routing; Least-cost multicast tree; Ant colony algorithm; Tree growth

I. Introduction

Due to rapid advances in the communication technologies and the increased demand for various kinds of communication services, many nowadays network applications require support of multicast communication. Therefore, the issue of multicast routing has become more and more important, especially with the emergence of distributed real-time multimedia applications, such as video conferencing, distance learning, and video on demand. These applications involve multiple users, with their own different quality of service (QoS) requirements in terms of throughput, reliability, and bounds on end-to-end delay, delay jitter, and packet loss ratio.

The main problem of QoS routing is to set up a least-cost multicast tree, i.e. a tree covering a group of destinations with the minimum total cost over all the links, which satisfies certain QoS parameters. However, the problem of constructing a multicast tree under multiple constraints is NP Complete [1]. Hence, the problem is usually solved by methods based on computational intelligence such as meta-heuristic algorithms.

In recent years, many meta-heuristic algorithms have been proposed for solving the QMR problem, such as ant colony algorithm [2-7], genetic algorithm (GA) [8-11], simulated annealing (SA) [12, 13], genetic simulated annealing [14], tabu search algorithm [15, 16], particle swarm optimization (PSO) [17, 18], and hybrid GA-PSO based algorithm [19].

However GA and SA algorithms have practical limitations in real-time multicast routing, since the GA climbing capacity is weak and premature easily, and both the efficiency and the quality of the solution for the SA algorithm depend on procedures that are sensitive to the influence of random annealing sequence.

Ant colony algorithm has high demand for parameter setting in large scale optimization. The most obvious weakness of ant colony algorithm is that it converges slowly at the initial step and takes more time to converge. Researchers have been working to improve the ant colony algorithm. For example, Dorigo and Caro [20] have proposed radically based self-adaptive ant colony algorithm, Zhao et al [21] have proposed ant colony algorithm that employs mutation and dynamic pheromone updating strategies. However, due to the complexity of network environment, these algorithms are not applicable to the multi constrained QMR.

Patel et al. [22] have proposed a hybrid ACO/PSO algorithm to optimize the multicast tree. The algorithm starts with generating a large amount of mobile agents in the search space. The ACO algorithm guides the agents' movement by pheromones in the shared environment locally, and the global maximum of the attribute values are obtained through the random interaction between the agents using PSO algorithm. Wang et al. [6] have proposed an algorithm which generates a multicast tree by using tree growth and optimizes ant colony algorithm parameters through orthogonal experiments.

The algorithm proposed by Wang et al. [6] has some drawbacks concerning the evaluation of the QoS constraints at each step during the construction of a multicast tree, which slows down the convergence of the algorithm, and the local pheromone update, which does not take into account the evaporation of pheromone on the edges.

This paper proposes a tree-growth based ACO (TGACO) algorithm for solving the least-cost multicast routing problem with three QoS constraints, namely: bandwidth, delay and delay jitter that overcomes these drawbacks.

The proposed algorithm generates each multicast tree using tree growth, such that an edge is added to the tree if it satisfies the bandwidth constraint. Then, the fitness of the constructed multicast tree is evaluated by using a cost function that includes the delay and delay jitter constraints. Depending on the fitness of the constructed multicast trees, the local and global best multicast trees can be determined. In TGACO algorithm, the ants perform local and global pheromone updates. In the local pheromone update, pheromone evaporation is performed by all ants after each construction step, while the global pheromone update is performed at the end of each iteration by the local and global best ants. Also, the paper presents the results of the experiments that have been conducted to evaluate the performance of the proposed algorithm.

The remainder of this paper is organized as follows: Section 2 presents the description and formulation of the QMR problem. Section 3 describes the ACO algorithm. Section 4 describes the drawbacks of the algorithm proposed by Wang et al. [6], and presents the proposed tree-growth based ACO algorithm for solving the QMR problem, which overcomes these drawbacks. Section 5 describes the operations of the proposed algorithm. Section 6 presents the steps of the proposed algorithm. Section 7 presents the results of the experiments. Finally, Section 8 presents the conclusions of this work.

II. PROBLEM DESCRIPTION AND FORMULATION

A network is modeled as a directed, connected graph G=(V, E), where V is a finite set of vertices (network nodes) and E is the set of edges (network edges) representing connection of these vertices. Each link e=(x,y) ϵ E has three weights (B(e), D(e) and C(e)) which correspond to the available bandwidth, the delay and the cost of the link, respectively.

A multicast tree T(s,M) is a sub-graph of G spanning the source node s ϵ V and the set of destination nodes $M \subset V - S$. Let m = |M| be the number of multicast destination nodes. We refer to M as the destination group and $\{\{s\} \cup M\}$ as the multicast group. In addition, T(s,M) may contain relay nodes (Steiner nodes), the nodes in the multicast tree but not in the multicast group. Let $P_T(s,d)$ be a unique path in the tree T from the source node s to a destination node d ϵ M. We now present the parameters that characterize the quality of the tree. The total cost of the tree T(s,M) is defined as sum of the cost of all links in that tree and can be given by:

$$\mathrm{Cost}\ T(s,M) = \sum_{e \in T(s,M)} C(e) \tag{1}$$

The total delay of the path $P_T(s,d)$ is simply the sum of the delay of all links along that path:

$$D\big(P_T(s,d)\big) = \sum_{e \in P_T(s,d)} D(e) \tag{2}$$

The total delay of the tree T(s,M) is defined as the maximum value of the delay on the paths from the source node to each destination node:

$$\mathrm{Delay}(T(s,M)) = \max\big(D\big(P_T(s,d)\big), \forall\, d \in M\big) \tag{3}$$

The bottleneck bandwidth of the path P_T (s,d) is defined as minimum available residual bandwidth at any link along the path:

$$B\big(P_T(s,d)\big) = \min\{B(e), e \in \big(P_T(s,d)\big)\} \tag{4}$$

The delay jitter of the tree T(s,M) is defined as the average difference of delay on the path from the source to the destination node:

$$Dj\big(T(s,M)\big) = \sqrt{\sum_{d \in M} \left(D(P_T(s,d)) - delay\_avg\right)^2} \tag{5}$$

where *delay_avg* denotes the average value of delay on the path from the source to the destination node.

Let the delay, the delay jitter and bandwidth constraints are D_{max}, D_j and B_{min}, respectively. The multi-constraint least-cost multicast problem is defined as:

Min Cost(T(s,M)) subject to:

$$\left.\begin{array}{l} D\big(P_T(s,d)\big) \le D_{max} \\ Dj\big(T(s,M)\big) \le D_j \\ B\big(P_T(s,d)\big) \ge B_{min} \end{array}\right\} \tag{6}$$

The QoS requirements described above can be classified into link constraint (e.g., bandwidth), path constraint (e.g., end to end delay) and tree constraint (e.g., delay-jitter). In our work the QoS multicast evolution is driven by the fitness function defined by (7), in which QoS constraints are considered except the bandwidth constraint, because the link that does not meet the bandwidth constraint is not chosen.

$$\begin{aligned} F\big(T(s,M)\big) = {} & Cost\big(T(s,M)\big) \\ & + \mu 1.\min\{(D_{max} - Delay\,(T\,(s,M)\,), 0\} \\ & + \mu 2.\min\{\big(D_j - Dj\big(T(s,M)\big), 0\} \end{aligned} \tag{7}$$

where µ1 and µ2 are punishment coefficients, their values determine the punishment extent.

III. ACO ALGORITHM

The basic ideas of ACO are from the social search behavior of biological ant colonies. In nature, ants move around in their environment in a rather random way, but they have certain tendency to follow the walk of other ants. They can recognize these walks because, while moving, each ant leaves a chemical substance called pheromone on the ground. Sensing pheromone on a path increases the probability of an ant to follow it, which further reinforces this path. This mechanism has the effect that short paths between a starting point and a goal point are favored, leading to a kind of heuristic optimization behavior. [7]

The described principle is exploited in ACO algorithms for optimizing arbitrary objective functions of combinatorial problems by simulating the walks of conceptual ants and by doing the re-enforcement of good walks based on an evaluation of the objective. Such ACO algorithms are based on the following ideas. First, each path followed by an ant is associated with a candidate solution for the given problem. Second, when an ant follows a path, the amount of pheromone deposited on that path is proportional to the quality of the corresponding candidate solution for the given problem. Third, when an ant has to choose between two or more paths, the path(s) with a larger amount of pheromone are more attractive to the ant. After some iteration, eventually, the ants will converge to the path, which is expected to be the optimum or a near-optimum solution for the target problem. [22]

IV. THE PROPOSED TREE-GROWTH BASED ACO ALGORITHM FOR SOLVING THE QMR PROBLEM

Wang et al. [6] have proposed a tree growth based ACO algorithm (TGBACA). It generates a multicast tree in the way of tree growth and optimizes the ant colony parameters through their most efficient combinations.

The basic idea of this algorithm is as follows: initially, the multicast tree has only the multicast source node. Then, the ant selects one link and adds it to the current tree according to an edge selection probability. After adding the selected edge, the path is checked to see whether it satisfies the specified QoS constraints. If not, another edge is selected. When the tree covers all the multicast members it stops growing. The tree obtained is then pruned and rendezvous links are removed to get the real multicast tree, then the delay jitter of the multicast tree is calculated. If it is greater than the delay jitter bound, then the multicast tree is rejected and another one is regenerated. Then, pheromones on the links that have been visited by the obtained local and global best multicast tree are updated. The above mentioned steps are repeated until the algorithm converges.

The drawbacks of the algorithm proposed by Wang et al. [6] are as follows:

- The rejection of the multicast tree after its construction, if it does not satisfy the delay jitter constraint, and re-generation of another one, slows down the convergence of the algorithm.
- During the construction of a multicast tree, each time an edge e(i, j) is to be added, the values of delay, packet loss ratio, and bandwidth, on the path from multicast source to node j, are modified. If the new values do not satisfy the related constraints, another edge e(i, j) is selected, and the calculations are repeated. Here, two odd situations may occur, which slow down the convergence of the algorithm:
    - If node j was a destination node, then the path to that destination has to be reconstructed.
    - If node j was the last destination node, i.e., at the end of a multicast tree construction, then the whole multicast tree has to be reconstructed.
- The local pheromone update is performed by computing the total pheromone on the candidate edge each time that edge is traversed. It does not take into account the evaporation of pheromone on the edges, which is very important to increase the exploration of edges that have not been visited yet and to prevent ants from producing identical solutions during one iteration [23].

The proposed tree-growth based ACO (TGACO) algorithm for solving the QMR problem overcomes these drawbacks.

In our TGACO algorithm, during the construction of a multicast tree, we check only the bandwidth of the edge that has been chosen from the candidate edge set to be sure that it satisfies the bandwidth constraint. The other QoS constraints, i.e. delay and delay jitter, are included with the cost in the fitness function (7), which is used to evaluate the quality of the constructed multicast tree. Depending on the fitness of the constructed multicast trees, the local and global best multicast trees can be determined. This way we avoid the overhead of calculating the QoS parameters for each path when a new edge is to be added to it. Accordingly, no path or tree rejection occurs. This speeds up the convergence of the algorithm.

In addition, our algorithm performs pheromone evaporation on each traversed edge in the local pheromone update step, as described below, to improve its performance.

V. THE OPERATIONS OF THE PROPOSED ALGORITHM

The proposed TGCAO algorithm for solving the QMR problem has the same three operations: tree growth, tree pruning and pheromone update, as TGBACA [6], but here they are carried out differently, as described below.

A. Tree Growth

This operation creates a tree $T(E_T, V_T)$, where E_T and V_T are the sets of edges and nodes of the tree, respectively, such that each edge in E_T satisfies the bandwidth constraint. Initially, E_T and V_T are set as follows: E_T = NULL, V_T ={s}, where s is the multicast source. Then, a link set E' is created. Initially, E' = {e(s, i)}, where each link e(s, i) belongs to the given network and satisfies the bandwidth constraint: $B(e(s, i)) \geq B_{min}$. Then, the following steps are performed:

Step 1: Select an edge from the set E' according to the following selection probability equation:

$$p_i = \left\{ \frac{[\tau_i]^\alpha \, [\lambda_i]^\beta}{\sum_{e_i \in E} [\tau_i]^\alpha \, [\lambda_i]^\beta} \right\} \tag{8}$$

where e_i is the i^{th} link of E', τ_i is the pheromone intensity of e_i, and λ_i is the heuristic function of e_i. We have selected $\lambda_i = 1/cost_i$, where $cost_i$ is the cost value of e_i, α and β are parameters used to adjust the effect of the pheromone intensity and the heuristic function. Suppose the edge (i, j) is selected, then it is added to the set E_T of tree T, and node j is added to the node set V_T of T.

Step 2: Update the links set E' to be $E' = E' - E_1 + E_2$, where $E_1 = \{e(k, j) | e(k, j) \in E'\}$ denotes the link set which takes node j as the destination and belongs to E', and $E_2 = \{e(j, k) | k \notin V_T\}$ denotes the link set which takes node j as starting node and satisfy the bandwidth constraint. It can be seen that one link at a time is added to tree T.

Step 3: Repeat the previous two steps of tree growth process until tree T covers all the destination group nodes.

B. Tree Pruning

Although the tree found covers all the destination group nodes, it may not be a multicast tree, because it contains some leaf nodes which are not destination nodes. Therefore, a pruning process is performed to remove those leaf nodes. Then, the delay, cost and delay jitter of the pruned multicast tree are modified accordingly, and its fitness is calculated using (7).

C. Pheromone Update

The most interesting contribution of ant colony system (ACS), [24], is the introduction of a local pheromone update in addition to the global pheromone update performed at the end of the construction process for all ants, which is called *offline pheromone update*.

The *local pheromone update* is performed by all ants after each construction step. Each ant applies it only to the last edge traversed. The main goal of the local update is to diversify the search performed by subsequent ants during one iteration. In fact, decreasing the pheromone concentration on the edges as they are traversed during one iteration encourages subsequent ants to choose other edges and hence to produce different solutions. This makes less likely that several ants produce identical solutions during one iteration. [25]

In our algorithm, after obtaining a multicast tree, the ant reduces the pheromone trial τ_{ij} of each traversed edge e(i, j), by using the following local pheromone update equation:

$$\tau_{ij} = (1 - \rho).\tau_{ij} \tag{9}$$

where $\rho \in [0,1]$ is the pheromone evaporation parameter, which is used to control the evaporating speed of pheromone.

Then, the *offline pheromone update* is performed at the end of each iteration by two ants, the iteration-best (local-best) ant and the best-so-far (global-best) ant using the following equation:

$$\tau_{ij} = (1 - \rho).\tau_{ij} + \rho.\Delta\tau_{ij}, \tag{10}$$

where $\Delta\tau_{ij} = Q / L$ denotes the pheromone increase for each edge e(i, j) that belongs to the local best and global best tree obtained, Q is the pheromone strength coefficient, and L can be either the cost of the local or global best tree.

VI. THE OVERALL TGACO ALGORITHM

The steps of the proposed TGACO algorithm are as follows:

Input: A network G= (V, E), s (multicast source), M (destination group), QoS bounds (D_{max}, D_j and B_{min}) maxIteration (maximum number of iterations), nAnts (number of ants), constants (ρ, Q, α, β, $\mu1$ and $\mu2$)

Output: A bandwidth-delay-constrained least-cost multicast tree (T_{best})

Begin

1. For iter = 1 to maxIteration Do
2. For ant = 1 to nAnts Do
   - 2.1 Tree Growth
     - a) Assign an initial value 1 to the pheromone trial τ_{ij} on each edge $e(i, j) \in E$.
     - b) Initialize set of nodes of tree T_{ant}: $V_T = \{s\}$, and its set of edges: E_T = NULL
     - c) Create a link set E'. Initially, set $E' = \{e(s, i)\}$, where each link e(s, i) satisfies the bandwidth constraint
     - d) Initialize number of covered destination nodes: mCount = 0
     - e) For each node n in the given network, set visited[n] = false
     - f) Select one edge e(s, i) from set E', according to the selection probability (8)
     - g) if $i \in M$ then mCount ++
     - h) visited[i] = true
     - i) Add node v_i to V_T and e(s, i) to E_T
     - j) While mCount < |M| do
       - Update the set E' as described in Sec. V.A
       - Select one edge e(i, j) from set E', according to the selection probability (8)
       - if $j \in M$ then mCount ++
       - visited[j] = true
       - Add node v_j to V_T and e(i, j) to E_T

       End While
   - 2.2 Prune tree T_{ant}
   - 2.3 Evaporate pheromone on the edges used by T_{ant} using (9)
   - 2.4 Calculate cost, delay, delay jitter, and bandwidth of the tree T_{ant} using equations (1) to (5)
   - 2.5 Calculate the fitness $F(T_{ant})$ using (7)
   - 2.6 Get local best tree T_{lbest}:

     if ant = 1 or $F(T_{ant}) < F(T_{lbest})$ then $T_{lbest} = T_{ant}$

   End For
3. Get global best tree T_{best} :

   if iter = 1 or $F(T_{lbest}) < F(T_{best})$ then $T_{best} = T_{lbest}$
4. Update pheromone on edges used by T_{lbest} and T_{best} using (10)

End For

End.

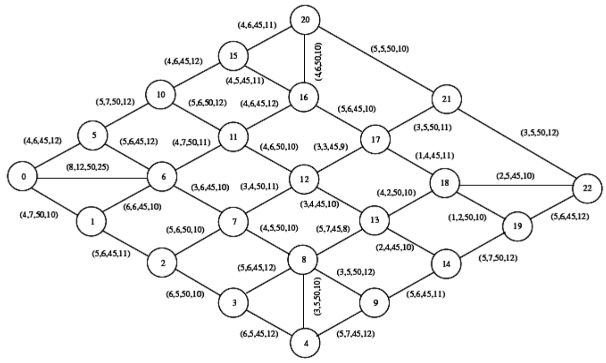

Figure 1. Network Model 1

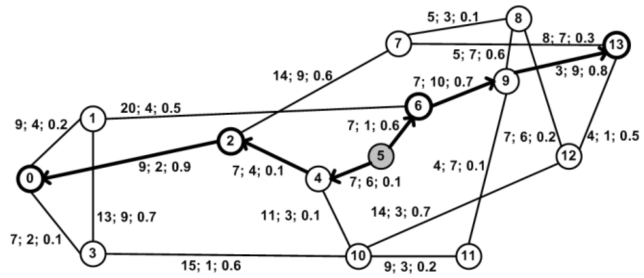

Figure 2. Network Model 2, showing the multicast tree obtained using our algorithm

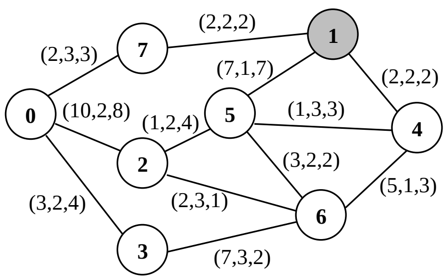

Figure 3. Network Model 3

VII. EXPERIMENTAL RESULTS

This section presents the results of the experiments that have been conducted to evaluate the performance of the proposed TGACO algorithm in solving the QMR problem compared to the TGBACA proposed by Wang et al. [6], and to study the effect of the number of ants used, the network size, and the destination group size, on the convergence of our algorithm. The algorithm has been implemented using C++.

In these experiments we have used three network models shown in Fig. 1, 2 and 3, where each edge is labeled with (delay, bandwidth, and cost). The first network model, shown in Fig.1, has 23 nodes with node 0 being the source node, the second network model, shown in Fig.2, has 14 nodes with node 5 being the source node, and the third network model, shown in Fig.3, has 8 nodes with node 1 being the source node.

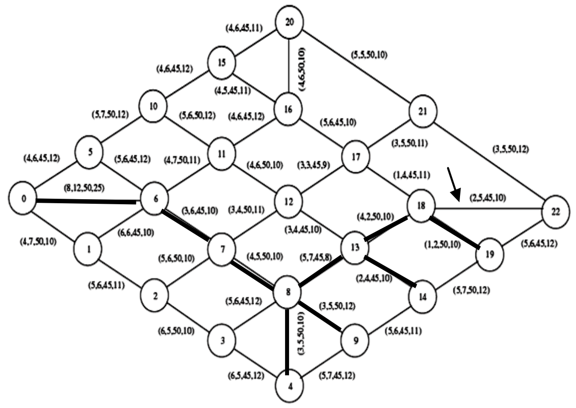

Figure 4. An example of multicast tree construction using the TGBACA [6], which led to violation of the specified bounds when the last destination node was to be added

The experiments configurations were as follows: Number of ants is 15 and number of iterations is 20; the value of α and β, which are used in (8), were set to 0.9 and 3.0, respectively; and the value of ρ which is used in (9) and (10) was set to 0.01. The performance of the algorithm was measured from perspective of the best multicast tree obtained, the fitness value and run time.

Before describing the experiments, we show an example of the case that may occur during the multicast tree construction using the TGBACA algorithm [6], where the specified bounds are violated when the last destination node is to be added, which requires the reconstruction of the multicast tree, (see Section IV). The network model used was the first one, with the destination group M = {4, 9, 14, 19, 22}, and the delay bound equals to 25. As shown in Fig. 4, the ant started at the source node 0, and reached the destination node 4 along the path (0-6-7-8-4), with path delay equals 18, then it reached the destination node 9 along the path (0-6-7-8-9), with path delay equals 18, then it reached the destination node 14 along the path (0-6-7-8-13-14), with path delay equals 22, and it reached destination node 19 along the path (0-6-7-8-13-18-19), with path delay equals 25. Finally, when the ant tried to reach the last destination node 22 along the path (0-6-7-8-13-18-22), the delay of this path was 26, which violates the delay bound. Thus, the ant must search for different paths to reconstruct a multicast tree, although it has reached 80% of the destination nodes.

In our algorithm, during the construction of a multicast tree, we only make sure that each selected edge satisfies the bandwidth constraint. This way, the ant constructs a multicast tree, then its fitness is evaluated using the fitness function, Eq. (7), which includes the other QoS constraints, i.e. delay and delay jitter, with the cost.

In the first experiment, we have applied the two algorithms to the three network models. The destination groups for the first, second and third network models were {4, 9, 12, 14, 19, 22}, {0, 2, 6, 13} and {3, 5, 7}, respectively. Table I shows the best multicast trees obtained by using the two algorithms for the three network models, with the cost, delay, delay jitter, the fitness value and running time in minutes. The table shows that, for the first two network models, the proposed TGACO algorithm produced multicast trees with less cost than the TGBACA algorithm [6], and for the third network model both algorithms produced multicast trees with same cost. But, in all cases, our algorithm has taken less time.

Fig. 5 and 6 shows the multicast trees obtained for Network Model 1 (Fig. 1), in the first experiment, by using the proposed TGACO and TGBACA [6] algorithms, respectively.

In the second experiment, we have applied the two algorithms to the first network model (Fig. 1) with four different destination groups: {3, 7,19}, {4, 9, 14, 19, 22}, {4, 9, 14, 19, 22, 12, 6}and{4, 9, 14, 22, 12, 16, 3, 21, 17}, representing 13%, 21%, 30% and 43% of the network nodes, respectively. Table II shows the best multicast trees obtained using the two algorithms with their fitness, cost, delay and delay jitter. It can be seen that our proposed algorithm produced multicast trees with less cost in all cases.

In the third experiment, we have applied our TGACO algorithm to the first network model (Fig. 1) with destination group {3, 7, 9}, to study the relationship between the number of ants used and the fitness of the best multicast tree obtained. Fig. 7 shows the results of this experiment. It indicates that as the number of ants increases, the fitness of the best multicast trees decreases, until it reaches a certain number, 10 ants in this case, where the fitness value starts to stabilize.

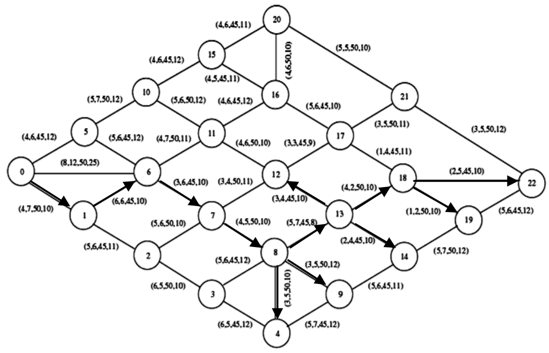

Figure 5. The multicast tree obtained by using the proposed TGACO

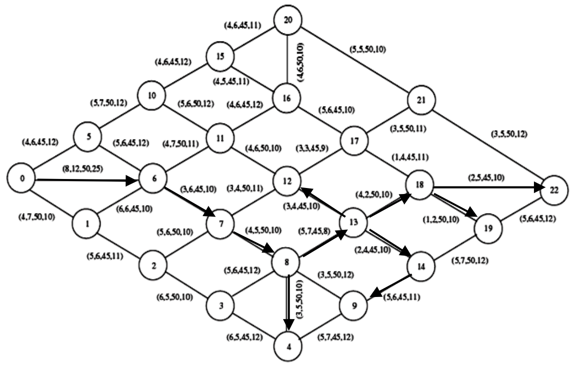

Figure 6. The multicast tree obtained by using TGBACA [6].

TABLE I. COMPARISON BETWEEN BEST MULTICAST TREES, WITH QOS CONSTRAINTS VALUES AND RUNNING TIME, OBTAINED USING THE TWO ALGORITHMS FOR THE THREE NETWORK MODELS.

Network model	Algorithm used	Multicast tree	Cost	Delay	Delay jitter	Fitness function	Run time (min)
Network Model 1 (Fig. 1)	Proposed TGACO	0-1-6-7-8-4. 0-1-6-7-8-9. 0-1-6-7-8-13-12. 0-1-6-7-8-13-14. 0-1-6-7-8-13-18-19. 0-1-6-7-8-13-18-22.	120	28	49.8	120	1.64
	TGBACA [6]	0-6-7-8-4. 0-6-7-8-13-14-9. 0-6-7-8-13-12. 0-6-7-8-13-18-22. 0-6-7-8-13-18-19.	124	27	47.37	124	4.56
Network Model 2 (Fig. 2)	Proposed TGACO	5-4-2-0. 5-6-9-13.	32	23	20.3	32	0.187
	TGBACA [6]	5-6-9-8-12-13. 5-4-2-0.	37	30	29.32	37	0.48
Network Model 3 (Fig. 3)	Proposed TGACO	1-4-5 1-7-0-3	10	7	5.2	10	0.15
	TGBACA [6]	1-4-5 1-7-0-3	10	7	5.2	10	0.209

TABLE II. COMPARISON BETWEEN THE RESULTS OBTAINED BY USING THE TWO ALGORITHMS FOR NETWORK MODEL 1 WITH DIFFERENT DESTINATION GROUPS

Destination Group	Algorithm	Cost	Delay	Delay jitter	Fitness function
{3, 7,19}	Proposed TGACO	79	28	27.6	79
	TGBACA [6]	88	27	31.2	88
{4, 9, 14, 19, 22}	Proposed TGACO	104	33	46.9	104
	TGBACA [6]	109	29	48.8	109
{4, 9, 14, 19, 22, 12, 6}	Proposed TGACO	120	27	48.8	120
	TGBACA [6]	124	27	47.37	124
{4, 9, 14, 22, 12, 16, 3, 21, 17}	Proposed TGACO	163	24	49.7	163
	TGBACA [6]	171	35	56.4	171

In the fourth experiment, we have applied the two algorithms, our TGACO algorithm and the TGBACA algorithm [6], to the three network models, to show the effect of the variation of the network size on the cost of the obtained multicast tree. Fig.8 shows this relationship, with network sizes, 23, 14 and 8, and the multicast group with ratio 43%. It can be seen that as the network size increases the cost increases. It can be seen also that the costs of the multicast trees obtained by our algorithm were less than those obtained by TGBACA [6].

In the fifth experiment, we have used our TGACO algorithm to show the effect of the size of distention group on the multicast tree cost. Fig. 9 shows the relationship between the percentage of destination group and cost for the three network models. It can be seen that as the percentage of destination group increases the cost of the multicast tree increases.

Finally, we have evaluated the two algorithms in terms of run time. The last column of Table I and Fig.10 show the run time of the two algorithms with the three network models with destination group size equal to 30% of the total nodes of each network. It can be seen that our algorithm takes considerably less time than TGBACA [6]. It can be seen also that as the size of the network increases the time increases.

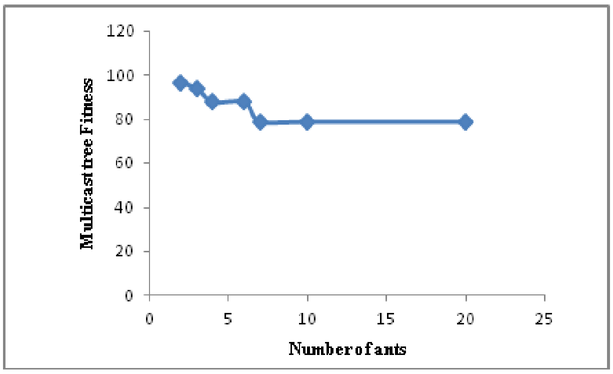

Figure 7. Multicast tree fitness values with different number of ants

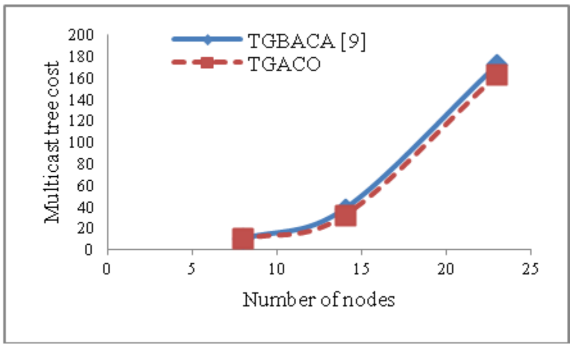

Figure 8. The relation between number of nodes and cost of the multicast tree obtained by using the two algorithms.

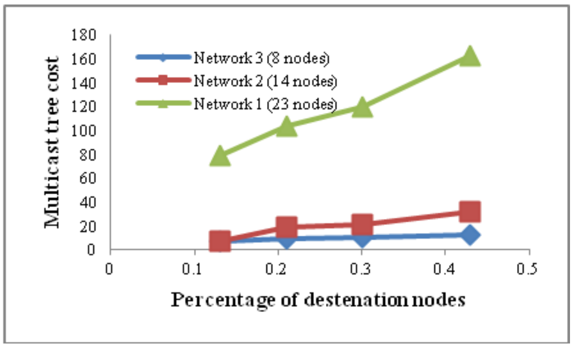

Figure 9. The relation between percentage of destination nodes and cost

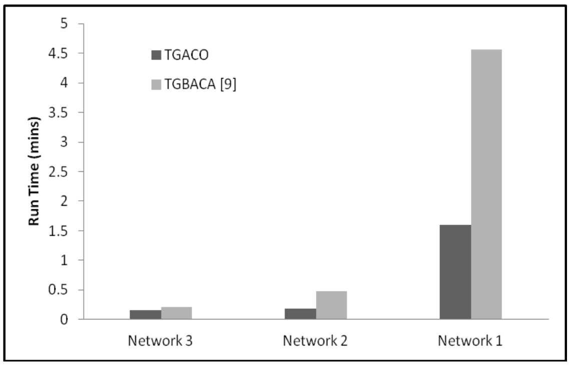

Figure 10. A comparison between the time taken by our TGACO algorithm and the TGBACA algorithm [6] for the three network models

VIII. CONCLUSION

This paper presented a proposed tree-growth based ACO (TGACO) algorithm for solving the QMR problem. In this algorithm, during the construction of a multicast tree, we only make sure that each selected edge satisfies the bandwidth constraint. This way, the ant constructs a multicast tree, then its fitness is evaluated using a proposed fitness function, which includes the other QoS constraints, i.e. delay and delay jitter, with the cost. Also, in addition to the pheromone update performed at the end of the construction process, the algorithm performs a local pheromone update. The effect of the local pheromone updating rule is that each time an ant uses a link its pheromone trial is reduced, so that the link becomes less desirable for the following ants, this allows an increase in the exploration of arcs that have not been visited. So, the algorithm does not show a stagnation behavior.

The performance of the algorithm has been evaluated through experiments, which showed that it is efficient in producing least cost multicast trees that satisfy the specified QoS constraints.

The experiments showed that the costs of the multicast trees obtained by our algorithm were less than those obtained by TGBACA [6], and our algorithm takes considerably less time.

It showed also that as the number of ants increases, the fitness of the best multicast trees decreases, and stabilizes after reaching a certain number of ants. It showed also that as the network size and the percentage of destination group increase the cost of the multicast tree increases, and as the size of the network increases the time increases.

REFERENCES

[1] S. L. Hakimi, "Steiner problem in graphs and its implications", Networks, vol. 1, pp. 113-133, 1971.

[2] G. Lu and Z. Liu, "Multicast routing based on ant-algorithm with delay and delay variation constraints", The 2000 IEEE Asia-Pacific Conference on Circuits and Systems, APCCAS 2000, February 2000, pp. 243-246.

[3] Chao-Hsien Chu, JunHua Gu, Xiang Dan Hou, Qijun Gu, "A Hesristic Ant Algorithm for Solving QoS Multicast Routing Problem", Proceedings of the 2002 Congress on Evolutionary Computation, CEC '02, 12-17 May 2002.

[4] B. Gong, L. Li, and X. Wang, "Multicast routing based on ant algorithm with multiple constraints", International Conference on Wireless Communications, Networking and Mobile Computing, WiCom 2007, 21-25 Sept. 2007, Shanghai, pp. 1945-1948.

[5] H. Wang, Z. Shi, S. Li, "Multicast routing for delay variation bound using a modified ant colony algorithm", Journal of Network and Computer Applications, vol. 32, pp. 258–272, 2009.

[6] H. Wang, H. Xu, S. Yi, and Z. Shi, "A tree-growth ant colony algorithm for QoS multicast routing problem", Expert Systems with Applications, vol. 38, no. 9, pp. 11787–11795, September 2011.

[7] D. Dhull and S. Dhull, "An Improved Ant Colony Otimization (IACO) Based Multicasting in MANET", International Journal of Inventive Engineering and Sciences (IJIES), vol. 1, no. 3, pp. 8-12, February 2013.

[8] W. Zhengying, S. Bingxin, and Z. Erdun, "Bandwidth-delay-constrained least-cost multicast routing based on heuristic genetic algorithm", Computer Communications, vol. 24, pp. 685-692, 2001.

[9] A. T. Haghighat, K. Faez, M. Dehghan, A. Mowlaei, and Y. Ghahremani, "GA-based heuristic algorithms for bandwidth-delay-constrained least-cost multicast routing", Computer Communications, vol. 27, no. 1, pp. 111–127, 2004.

[10] F. De Rango, M. Tropea, A. Santamaria, and S. Marano, "Multicast QoS corebased tree routing protocol and genetic algorithm over an HAP-satellite architecture", IEEE Transactions on Vehicular Technology, vol. 58, no. 8, pp. 4447-4461, Oct. 2009.

[11] A. Younes, "Multicast routing with bandwidth and delay constraints based on genetic algorithms", Egyptian Informatics Journal, vol. 12, pp. 107–114, 2011.

[12] Z. Kun, W. Heng, and L. Feng-Yu, "Distributed multicast routing for delay and delay variation-bounded Steiner tree using simulated annealing", Computer Communications, vol. 28, pp. 1356–1370, 2005.

[13] Z/ Kun, Q. Yong, and Z. Hong, "Dynamic multicast routing algorithm for delay and delay variation-bounded Steiner tree problem", Knowledge-Based Systems, vol. 19, pp. 554–564, 2006.

[14] L. Zhang, L. Cai, M. Li, and F. Wang, "A method for least-cost QoS multicast routing based on genetic simulated annealing algorithm", Computer Communications, vol. 32, pp. 105–110, 2009.

[15] N. Ghaboosi and A. T. Haghighat, "Tabu search based algorithms for bandwidth-delay-constrained least-cost multicast routing", Telecommunication Systems, vol. 34, no. 3, pp. 147-166, 2007.

[16] X. Wang, J. Cao, H. Cheng, and M. Huang, "QoS multicast routing for multimedia group communications using intelligent computational methods", Computer Communications, vol. 29, pp. 2217–2229, 2006.

[17] J. Liu, J. Sun, and W. Xu, "QoS Multicast Routing Based on Particle Swarm Optimization", Proceedings of 7th International Conference on Intelligent Data Engineering and Automated Learning, IDEAL 2006, Burgos, Spain, September 20-23, LNCS 4224, pp. 936- 943, Springer-Verlag Berlin Heidelberg, 2006.

[18] C. Mala and R. Narendran, "Simulated Study of QoS Multicast Routing Using Particle Swarm Optimization", World Applied Programming, vol. 1, no 3, pp. 176-182, August 2011.

[19] M. R. Girgis, T. M. Mahmoud, and G. W. Hanna, "Using GA, PSO and hybrid GA-PSO based algorithms for solving the least-cost multicast problem with QoS constraints", International Journal of Computer Systems, vol. 3, no. 3, pp. 166-172, March, 2016

[20] M. Dorigo and G. Caro, The ant colony optimization meta-heuristic, new ideas in optimization, McGraw-Hill, 1999.

[21] N. Zhao, Z. Wu, Y. Zhao, and T. Quan, "Ant colony optimization algorithm with mutation mechanism and its applications", Expert Systems with Applications, vol. 37, pp. 4805-4810, 2010.

[22] M. K. Patel, M. R. Kabat, and C. R. Tripathy, "A hybrid ACO/PSO based algorithm for QoS multicast routing problem", Ain Shams Engineering Journal, vol. 5, no. 1, pp. 113–120, March 2014.

[23] M. Dorigo and T. Stützle, Ant Colony Optimization, MIT Press, Massachusetts Institute of Technology, 2004.

[24] 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.

[25] M. Dorigo, "Ant colony optimization", Scholarpedia, vol. 2, no. 3, doi:10.4249/scholarpedia.1461, 2007.