Ant Colony System for Real-time Multimedia Communications

Sheng-Yuan Tseng[1], Chang-Chun Lin[1] and Yueh-Min Huang[2]
[1]Department of Information Management, Kun-Shan Universit y
[2]Department of Engineering Science, National Cheng-Kung University
tsy@mail.ksu.edu.tw huang@mail.ncku.edu.tw chanclin@ms29.hinet.net

ABSTRACT

Multicast routing can significantly optimize network resource utilization and prevent network congestion when bandwidth-consuming applications are widely implemented in a network. Multicast routing for real-time multimedia applications must construct multicast trees that minimize resources usage and meet some quality-of-service requirements. This study presents a novel algorithm based on ant colony optimization, a new meta-heuristic for combinatorial optimization problems, for solving the bandwidth-delay-constrained multicast routing problem. The simulation results indicate that the proposed method is more effective and efficient than a representative method.

1: INTRODUCTIONS

The development of high-speed network and the advance in computer technology have hastened the growth of multimedia applications [1]. Many of the multimedia applications, like video-on-demand and distance education, heavily utilize network resources and require quality-of-service (QoS) guarantees such as minimum bandwidth requirement and bounded end-to-end delay [2]. Moreover, these applications usually involve packet transmission from a source to multiple receivers. Sending packets through multiple unicast sessions to all receivers is inefficient because many duplicated packets might traverse the same link. Conversely, sending data from the source to all receivers simultaneously, that is, multicasting, may be a more efficient way to support these applications.

Multicasting consists of multicast routing (establishing a multicast tree) and data forwarding (transmitting data along that multicast tree). Transmission costs can be greatly reduced with multicasting because packets are sent to all receivers via a multicast tree such that only one copy of each packet traverses each link in that tree. The algorithms to construct multicast trees are called multicast routing algorithms. Though multicasting can reduce transmission costs, constructing multicast trees for real-time multimedia applications is very difficult. This is because the constructed multicast trees must minimize resource utilizations and at the same time meet some QoS constraints. Accordingly, multicast routing problems become difficult to solve. Among other QoS constraints, end-to-end delay constraint is very likely to be the most important one for real-time multimedia applications because data arriving beyond a deadline become useless. Bandwidth is another important QoS requirement because it is a basic requirement for transmitting real-time multimedia data. Therefore, this study focuses on the bandwidth-delay-constrained multicast routing (BDCMR) problem.

Constructing a multicast tree with the minimum cost is generally modeled as the minimal Steiner tree problem, which was shown to be NP-complete [2]. A Steiner tree is a minimum-cost tree rooted at a source node and connecting all destination nodes. Many heuristic algorithms, such as the DNH [4], ADH [5] and SPH [6], were devised to solve the problem. Stochastic methods that were based on simulated annealing [7], hop field neural network [8], and genetic algorithms (GA) [9-11] were also proposed to solve the problem.

Recently, the minimal Steiner tree problem was extended to the delay-constrained multicast routing (DCMR) problem, and several heuristic algorithms including the KPP [12], BSMA[13], and CKMB[14] were proposed. Besides heuristic algorithms, Xiang et al. [15], Zhang and Leung [16], and Ravikumar and Bajpai [17] proposed genetic algorithms to solve the DCMR problem.

Lately, many well-performed GAs for graph optimization problems used trees to directly represent solutions. Among them, the algorithm of Wang et al. [3] (denoted as WGA) is a representative method for the BDCMR problem. The WGA directly manipulated trees with specialized crossover and mutation operators, and used a penalty function to punish infeasible trees. Haghighat et al. [18] also proposed a GA (denoted as HGA) to solve the BDCMR problem and claimed that their algorithm has the best performance than other GAs. However, the HGA is more complicated than the WGA. Moreover, according to [19], the performance difference between the HGA and WGA is not significant.

This study proposes an algorithm based on ant colony optimization (ACO) to solve the BDCMR problem. Ant colony optimization, firstly developed by Dorigo et al. [20], is a new meta-heuristic for combinatorial optimization problems. The proposed algorithm was compared with the WGA. Experiments showed that the proposed algorithm is much more effective and efficient than the WGA. The remainder of this paper is organized as follows. Section 2 defines the problem. Section 3 then presents the proposed algorithm. Next, the experiments are presented in section 4. Finally, Section 5 draws conclusions.

2: PROBLEM DEFINITION

A network can be modeled as an directed graph $G=(V,E)$ comprising a set of nodes $V=\{v_1,v_2,\ldots,v_n\}$ and a set of edges $E=\{(i,j)\,|\,v_i,v_j\in V\}$. Each edge (i,j) is associated a delay d_{ij}, a bandwidth b_{ij}, and a cost c_{ij}, where d_{ij} represents the data transmission delay from v_i and v_j, which might include queuing, transmission and propagation delays. b_{ij} represents the residual bandwidth of the edge. Meanwhile, c_{ij} represents the cost of transmitting a packet from v_i to v_j, and can measure monetary cost or resource requirements. Let $v_s\in V$ denote the multicast source and $D\subseteq V-\{v_s\}$ be a set of destination nodes. A Steiner tree $T=(V',E')\subset G$ is a tree rooted at v_s and spanning all the members of D. The cost of T is given by,

$$C(T)=\sum_{(i,j)\in E'}c_{ij}\,. \tag{1}$$

A path $P(t)\subset T$ represents the set of nodes and edges that constitute that path from v_s to v_t. Let λ_t denote the accumulated delay along $P(t)$, i.e.,

$$\lambda_t=\sum_{(i,j)\in P(t)}d_{ij}\,. \tag{2}$$

The bottleneck bandwidth of the path $P(t)$ is defined as the minimum residual bandwidth among edges along the path, i.e.,

$$\theta_t=\min\{b(e),e\in P(t)\} \tag{3}$$

Let δ be the delay constraint and B be the bandwidth constraint. The bandwidth-delay-constrained multicast routing problem is to find a Steiner tree $T^*=(V^*,E^*)$, which has the least tree cost among all possible Steiner trees of G and satisfies the following constraints.

$$\begin{cases}\lambda_t\le\delta & ,\forall t\in D\\ \theta_t\ge B & ,\forall t\in D\end{cases} \tag{4}$$

3: PROPOSED ALGORITHM

ACO algorithm mimics the behavior of ants in establishing paths from their net to food sources. Each ant selects a path based on the information from the pheromone trails laid by other ants. The pheromone level deposited on a particular path increases with the number of ants passing through that path. Ants use pheromones to communicate and cooperate with each another to identify shorter paths to the food source. Dorigo et al. proposed the first ACO algorithm called ant system [19] to solve the well-known traveling salesman problem (TSP), and proposed the ant colony system (ACS) [20], an enhanced version of AS, to effectively solve larger problems.

The proposed algorithm, name ABDMR, is based on the ACS and consists of three primary mechanisms, including the Steiner tree construction, the local updating rule and the global updating rule.

Before solving the problem, the network edges with bandwidths less than the requirement are removed first. If the remained sub-network contains the source and all destination nodes, the proposed algorithm can proceed with the refined network. Otherwise, the network does not meet the bandwidth requirement and no solution will be created. With the refining process, the BDCMR problem can be reduced to the DCMR problem.

3.1: Steiner tree construction

Steiner tree construction is fundamental to the proposed algorithm. An approach called APrim was proposed to incorporate Prim's algorithm [21] with the ACS to construct random Steiner trees. Let V' represent the nodes of the temporarily constructed tree, and $E_c=\{(i,j)\,|\,v_i\in V',v_j\in V-V',\lambda_i+d_{ij}\le\delta\}$ represent the set of candidate edges from which an edge is selected to expand the tree under construction. The construction process starts from adding the source node to V', which is originally empty, to initialize E_c. During the construction process, edges are randomly selected from E_c by a selection rule revised from the ACS's pseudo-random-proportional rule. Each selected edge is added to the partially constructed tree and removed from E_c before E_c is updated. The APrim stops when E_c becomes empty or when the constructed tree has included all the destination nodes. all non-destination terminals and their connected edges are then pruned from the constructed tree.

Since the original pseudo-random-proportional rule of the ACS is for constructing a linear route rather than a tree, the rule was revised to be suitable for tree construction. According to the revised rule, when $q\le q_0$, ant k selects edge (x,y) from E_c to expand the tree if

$$[\tau(x,y)][\eta(x,y)]^\beta=\max_{(i,j)\in E_c}\{[\tau(i,j)][\eta(i,j)]^\beta\} \tag{5}$$

q is a random number uniformly distributed in interval [0,1], and $0\le q_0\le 1$ is an predetermined parameter that determines the relative importance of exploitation versus exploration. $\tau(i,j)$ denotes the pheromone level on edge (i,j).

If $q>q_0$, edge (x,y) is randomly selected from E_c using roulette wheel selection based on the probability distribution given as follows.

$$p_k(x,y)=\begin{cases}\dfrac{[\tau(x,y)][\eta(x,y)]^{\beta}}{\sum\limits_{(i,j)\in E_c}[\tau(i,j)][\eta(i,j)]^{\beta}}, & \text{if } (x,y)\in E_c,\\ 0, & \text{otherwise.}\end{cases} \tag{6}$$

The heuristic function $\eta(i,j)$ has alternative definitions,

$$\eta(i,j)=\frac{1}{c_{ij}} \tag{7}$$

or

$$\eta(i,j)=\frac{1}{c_{ij}\dfrac{\lambda_i+d_{ij}}{\delta}} \tag{8}$$

The definition of $\eta(i,j)$ switches to the other one when one ant has completed its tree construction. Both Eqns. (7) and (8) penalize edges that have large costs, while the second $\eta(i,j)$ also penalizes edges that would lead to long delays. Equation (7) aims to find low cost solutions, while Eqn. (8) aims to find feasible ones. The alternative use of these two definitions is a compromise between the conflicting goals of finding low-cost and feasible solutions. The APrim algorithm is summarized as follows.

```
Procedure APrim
  $V' \leftarrow \phi$, $E' \leftarrow \phi$;
  $V' \leftarrow V' + \{v_s\}$ and update $E_c$;
  While $E_c \neq \phi$ and $D \not\subset V'$
    Select an edge $(x,y)$ from $E_c$ according the revised pseudo-random-proportional rule;
    $E' \leftarrow E' + \{(x,y)\}$; $V' \leftarrow V' + \{v_y\}$;
    $V \leftarrow V - \{v_y\}$;
    Update $E_c$;
  End while
  Prune all non-destination terminals and their connected edges from $T$
  Return $T = [V', E']$
End
```

3.2: Pheromone updates and cost evaluation

After an ant has found a Steiner tree $T_k=(V'_k,E'_k)$, the pheromone levels on the edges of that Steiner tree are updated using the local updating rule (9).

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

where $0<\rho<1$ is a parameter representing the local pheromone evaporation rate, and $\Delta\tau(i,j)=\tau_0$, the initial pheromone level.

The Steiner tree might not include all of the destination nodes such that $|V'_k|\leq|D|$. For avoiding that infeasible trees might have smaller costs than feasible trees, the constructed trees are evaluated with a penalty technique like that of [3]. The cost function is defined as follows.

$$C(T_k)=\gamma^{|D|-|V'_k|}\left(\sum_{(i,j)\in E'_k}c_{ij}\right), \tag{10}$$

Parameter γ controls the degree of penalty and was set to 2 in this study

When all the ants have found their respective Steiner trees, the pheromone levels on the edges of the globally best Steiner tree $T^*=(V^*,E^*)$ are updated using the global updating rules (11) and (12).

$$\tau(i,j)\leftarrow(1-\alpha)\tau(i,j)+\alpha\tau_{gb}(i,j) \tag{11}$$

where $0<\alpha<1$ is a parameter representing the global pheromone evaporation rate, and

$$\tau_{gb}(i,j)=\begin{cases}\dfrac{|V|}{L_{gb}}, & \text{if}(i,j)\in E^*\\ 0, & \text{otherwise}\end{cases} \tag{12}$$

L_{gb} represents the cost of the up-to-now best Steiner tree from the beginning of the trial.

3.3: Algorithm of ABDMR

The ABDMR algorithm repeatedly constructs Steiner trees and updates pheromones locally and globally until the stop condition is met. The algorithm is summarized as follows.

```
Procedure ABDMR
  While the stop condition is not met
    For each ant $k$
      Use APrim to construct a Steiner tree $T_k=(V'_k,E'_k)$;
      Apply the penalty technique to evaluate $T_k$;
      Update the pheromones on the edges of $T_k$ using the local updating rule;
    End for
    Update the pheromones on the edges of the up-to-now best Steiner tree $T^*$ using the global updating rule;
  End while
End.
```

4: EXPERIMENAL RESULTS

A series of simulations were conducted to compare the ABDMR with the WGA. The test problems were generated using the random graph generator of Waxman [21]. Graphs of four sizes, 50, 100, 200 and 500 nodes were generated. The source node was set to node 1, and the destination nodes were randomly selected. The number of destination nodes was set to 20% of the node number. Table 1 shows some characteristics of the test problems.

Pilot experiments were conducted to optimize the parameters α , q_0 and ρ for the proposed ABDMR before applying the algorithm to solve the test problems. The pilot experiments indicated that $\alpha = 0.06$, $q_0 = 0.3$ and $\rho = 0.06$ were the best setting. Moreover, the number of ants and β were set to 30 and 2, respectively. The initial pheromone level τ_0 was set to $1/g_0$. In this study g_0 was initially set to 1 before being replaced with the best cost found in the first iteration. The ABDMR stops after 20,000 evaluations, and the WGA stops after 1,000,000 evaluations, because the WGA converges very slowly on dense networks. The population size and mutation rate for the WGA were set to 50 and 0.005, respectively, as in [17]. The simulations were conducted on a PC with a P4 3.0 GHz CPU. Thirty trials were performed on each test problem, and the best solution, average run time, the success rate, and average cost of each 30 trials were recorded. A trial that found no feasible solution was reckoned as failed. The success rate means the percentage of success trials.

The results in table 2 show that the ABDMR performed stably on both sparse and dense networks. However, the WGA is very unstably on large and dense networks for its low success rates on some problems. The ABDMRA achieved 100% success rates on all the test problems. Meanwhile, the average and best costs obtained by the ABDMR for the problems were better than those by the WGA. Moreover, the computation times of the ABDMR were significantly shorter than those of the WGA.

Figure 1 shows the ratio of the cost achieved by the WGA over that by the proposed algorithm on sparse and dense networks. The proposed algorithm achieved a smaller average cost than the WGA. The performance difference between the ABDMR and WGA became more significant on dense and large networks.

Network	Nodes	Edges	Destinations	δ
P50-100	50	100	10	294
P50-200	50	200	10	230
P100-200	100	200	20	488
P100-1000	100	1000	20	144
P200-400	200	400	40	608
P200-2000	200	2000	40	200
P500-1000	500	1000	100	808
P500-5000	500	5000	100	256

Table 1 Characteristics of test networks

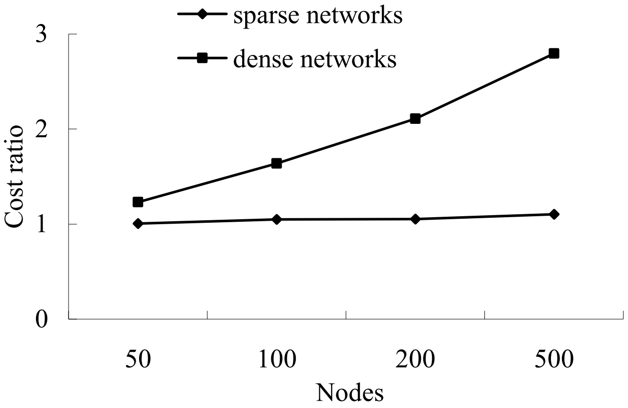

Fig. 1 Ratio between the costs achieved by the WGA and ABDMRA

Network	WGA				ABDMR			
	Average cost	Best cost	Runtime (sec)	Success rate(%)	Average cost	Best cost	Runtime (sec)	Success rate(%)
P50-100	666.5	661	2.48	100	661	661	0.01	100
P50-200	709.5	650	3.57	100	575	575	0.11	100
P100-200	921.8	878	6.05	100	877.7	875	0.32	100
P100-1000	1585.2	1453	11.73	100	968.6	964	1.25	100
P200-400	1794.1	1719	15.59	100	1701.1	1694	3.26	100
P200-2000	2822.6	2637	32.94	70	1339.1	1325	11.72	100
P500-1000	5556.6	5382	128.8	100	5037.8	4997	36.05	100
P500-5000	9693	9467	215.67	20	3467.7	3400	77.24	100

Table 2 Simulation results

5: CONCLUSIONS

This study investigates the bandwidth-delay-constrained multicast routing (BDCMR) problem, and proposes an ant colony-based algorithm (ABDMR) to solve it. Simulations were conducted to compare the ABDMR with the methods of Wang et al. (WGA) The ABDMR found better solutions to the test problems than those by WGA, with much shorter computation time than does the WGA. The simulation results suggested that the ABDMR algorithm is an efficient and effective approach to the BDCMR problem.

REFERENCES

[1] J.W. Ding, S.Y. Tseng, Y.M. Huang, Packet Permutation: A Robust Transmission Scheme for Continuous Media Streams over the Internet, Multimedia Tools and Applications, 21(3) (1993) 281-305.

[2] B.M. Waxman, Routing of Multipoint Connections, IEEE Journal of Selected Area in Communications, 6(9) (1988) 1617-1622.

[3] Z. Wang, B. Shi, E. Zhao, Bandwidth-delay-constrained least-cost multicast routing based on heuristic genetic algorithm, Computer Communications 24 (2001) 685-692.

[4] L. Kou, G. Markowsky and L. Berman, A fast algorithm for Steiner trees, Acta Informatica, 15(2) (1981) 141-145.

[5] V. Rayward-Smith, The computation of nearly minimal Steiner trees in graphs, International Journal of Mathematical Education in Science and Technology, 14(1) (1983) 15-23.

[6] H. Takahashi and A. Matsuyama, An approximate solution for the Steiner problem in graphs, Mathematica Japonica, 22(6) (1980) 573-577.

[7] E. Gelenbe, A. Ghanwani and V. Srinivasan, Improved neural heuristics for multicast routing, IEEE Journal on Selected Area of Communications, 15(2) (1997) 147-155.

[8] P.Chotipat, C. Goutam, S. Norio, Neural network approach to multicast routing in real-time communication networks, Proc. ICNP, (1995) 332-339.

[9] A. Kapsalis, V.J. Rayward-Smith and G.D. Smith, Solving the graphical Steiner tree problem using genetic algorithms, Journal of the Operational Research Society, (44) (1993) 397-406.

[10] H. Esbensen, Computing near-optimal solutions to the Steiner problem in a graph using a genetic algorithm, Networks, 26 (1995) 173-185.

[11] Y. Leung, G. Li, Z. Xu, A genetic algorithm for the multiple destination routing problems, IEEE Transaction on Evolutionary Computation, 2(4) (1998) 150-161.

[12] Vachaspathi P. Kompella, Joseph C. Pasquale, and George C. Polyzos. Multicast Routing for Multimedia Communication. IEEE/ACM Transactions on Networking, 1(3) (1993) 286-292.

[13] M. Parsa , Q. Zhu, J.J. Garcia-Luna-Aceves , An Iterative Algorithm for Delay-Constrained Minimum Cost Multicasting, IEEE/ACM Trans. on Networking, 6(4) (1998) 461-474.

[14] Quan Sun, Horst Langendörfer, An efficient delay-constrained multicast routing algorithm, Journal of High Speed Networks, 7(1) (1998) 43-55.

[15] F. Xiang, L. Junzhou, W. Jieyi, G. Guanqun, QoS routing based on genetic algorithm, Computer Communications 22 (1999) 1394-1399.

[16] Q. Zhang, Y.W. Lenug, An orthogonal genetic algorithm for multimedia multicast routing, IEEE Transaction on Evolutionary Computation 3 (1) (1999) 53-62

[17] C.P. Ravikumar, R. Bajpai, Source based delay-bounded multicasting in multimedia networks, Computer communications 21 (1998) 126-132.

[18] A. T. Haghighat, Karim Faez, Mehdi Dehghan, A. Mowlaei, Y. Ghahremani, GA-based heuristic algorithms for bandwidth-delay-constrained least-cost multicast routing, Computer Communications 27(2004) 111-127.

[19] M. Dorigo, V. Maniezzo, A. Colorni, The ant system: Optimization by a colony of cooperating agents, IEEE Transaction on System, Man, and Cybernetics, Part B, 26(1) (1996) 1-13.

[20] M. Dorigo, L.M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, IEEE Transactions on Evolutionary Computation, 1(1) (1997) 53-66.

[21] Waxman BM (1988) Routing of multipoint connections. IEEE Journal on Selected Area on Communications 1988; 6(9): 1617-1622