

The Memetic Algorithm for The Minimum Spanning Tree Problem with Degree and Delay Constraints

Minying Sun*, Hua Wang*

*Department of Computer Science and Technology, Shandong University, China

sunminying87@gmail.com, wanghua@sdu.edu.cn

Abstract— As the critical technology of many real-time applications, application layer multicast requirements limit the transmission time and the number of receivers to which each node can transmit. Such a communication scheme in an overlay network can be regarded as a degree- and delay-constrained minimum spanning tree (DDCMST) problem, and appears to be NP-complete. This paper proposes a novel Memetic algorithm(MA) to solve the DDCMST problem and compares it with some related methods. The results of a series of simulations show the efficiency and effectiveness of the proposed algorithm.

Keywords— Application layer multicast, Degree and delay constraints, Memetic algorithm

I. INTRODUCTION

With the development of Internet, various multimedia businesses have emerged, such as telephone conference, the remote teaching, and TV conference. However, streaming media technology based on C/S style cannot meet the demand of people, especially when a large number of users are online meanwhile, the server performance will plummet. In order to solve the transmission problem of streaming media network, people put forward network multicast[1] communication mode that a source node can send information to multiple destination nodes. Because of its own defect[2], such as the rapid expansion of routing table with users increasing, and the overweight of router load, network multicast technology has not been widely applied. In order to improve the efficiency of multicast, application layer multicast[3], which is a multi-user data distribution scheme based on application layer, has become the research hot spot.

When the application layer multicast services are implemented, an important problem is the design of multicast tree[4]. Building a multicast tree in overlay is somewhat different from the underlying network layer solution. Since overlay network is a virtual structure via existing unicast connections,

each node may reach everybody else directly. Therefore, the ideal minimum delay multicast tree is a star overlay rooted at the source obviously. Unfortunately, access link bandwidth and processing power capacity of an end host limit the number of peers it could serve. Another side, a main application field of the application layer multicast technique is real-time transmission, so it has strict restriction to network delay and node performance. Consider these factors, many different schemes have already been proposed, including the minimum-delay application-layer multicast trees problem[5], the degree-constrained minimum spanning tree problem[6][7], the degree-constrained low delay application layer multicast tree problem[8] and the degree- and delay-constrained minimum spanning tree(DDCMST) problem. Among them DDCMST directly considered node degree and delays from the source node to destination nodes, and solved the problems existing the application layer multicast, such as big delay, overweight of terminal nodes. So DDCMST problem is focused in this paper. Finding DDCMST has been proven to be NP-hard in [9] and [10] respectively. This paper proposed a new optimization algorithm to solve DDCMST problem using the basic idea of the MA, compares it with some other heuristic algorithms that were modified for the DDCMST problem.

The rest of this paper is organized as follows: Section 2 gives the formal definition of the problem to be addressed. Section 3 describes the details of the proposed algorithm. Section 4 shows the experimental results. Finally, we summarize our work in Section 5.

II. PROBLEM FORMULATION

According to the graph theory, a network can be modeled as an undirected graph $G=(V,E)$ where V

is a set of vertices representing end-hosts and E is a set of overlay edges denoting the unicast paths. In a DDCMST problem, Every edge $e(i, j)$ in set E possesses some QoS metrics, which include cost c_{ij} , and delay d_{ij} , where d_{ij} represents the data transmission delay and c_{ij} represents the cost of transmitting a packet from v_i to v_j . A spanning tree $T(s, D)$, where $s \in V$ is the source node of multicast application, $D \subseteq V - \{s\}$ is a group of destination nodes. The total cost of $T(s, D)$ is the sum of the cost of all the links, marked as $C(T)$. The objective function can be formulated as follows:

$$C(T) = \sum_{(i,j) \in E} c_{ij} \quad (1)$$

If v_i is the parent of v_j , the delay from the source node to v_j is:

$$D(j) = D(i) + d_{ij} \quad (2)$$

$O(i)$ represents the out-degree of a node v_i in T which is the number of child nodes connecting to node v_j , while $D(i)$ represents the delay from the source to v_i . σ represents the overall allowable delay from the source to destination nodes, and φ represents the allowable out-degree for each node. The DDCMST problem is then to find a minimum spanning tree $T(s, D)$ that has the least cost among all possible spanning trees and that satisfies the following constraints:

$$D(i) \leq \sigma, \forall v_i \in V \quad (3)$$

$$O(i) \leq \varphi, \forall v_i \in V \quad (4)$$

III. MEMETIC ALGORITHM

A. Basic thought of the algorithm

Inspired from Darwinism principles in natural evolution, the term "Memetic Algorithm" (MA) was proposed in 1989 by Moscato[11] and was defined as a hybrid of an evolutionary algorithm with an individual learning procedure that has the capability of local improvement. MA has been successfully applied to many NP-hard problems[12]. In the area of optimization, there are two kinds of search methods: global search and local search. One hand, MA uses operators found in natural genetics to guide the trek through a search space. On the

other hand, local improvement procedures can quickly find the exact local optimum of a small region of the search space. Also an important decision should be made about which members of the population to select for further improvement via local search. After defining these main rules, MA is well-defined. The operations of the algorithm process and the specific steps are as follows:

Step1: First of all, randTree() randomly generate num trees as a candidate set, and record the optimal tree;

Step2: Enter the iterative process, selects tree in the candidate trees set, and cross it with the optimal tree, then create new tree, and replace the original tree if the new tree is better;

Step3: Execute mutation operation, randomly select a node of a tree, and change the parent node of the selected node;

Step4: Execute local update operation, update the longest delay path, namely select parent node of the node that have the largest delay. If update successfully, check whether the new tree is the optimal tree, otherwise proceed to next step;

step5: Repeat from step2 to step 4, until it reaches the maximum iterations.

Num is the number of population.

Niter is the number of iteration.

N is the number of nodes.

proc MA_DDCMST()

For k = 1 to Num do

T(k) = randomTree();

End for

/* Enter the iterative process. */

For iter = 1 to Niter

For i = 0 to N-1 ;

Crossover(T(i), optimal);

j = rand() % Num;

Mutation(T(j));

For i = 0 to N-1;

Local(T(i)); /* local update operation */

End for

Output T(optimal);

End

B. Operation of the algorithm

1) Initialization algorithm

The storage method of trees is shown in figure 1 in this paper.

$Tree(Nd, Nf)$ —the structure of tree, Nd storages destination nodes, Nf storages parent nodes of Nd ;

$T[num]$ —storage all spanning trees;

Optimal—record the optimal tree;

New—record temporary spanning tree;

First a tree $Tree(Nd, Nf)$ is created. Nd storages nodes of the tree. At the initial step, we select v_i from the node set V randomly, select s as its parent node, and then join them in Nd and Nf respectively. For each node $v \in V$ which not in Nd , we select one node in Nd or s which satisfy constraints as the parent node of v . If they are joined successfully, the parent node's degree and the node delay are modified. Repeat the process until all nodes join the tree.

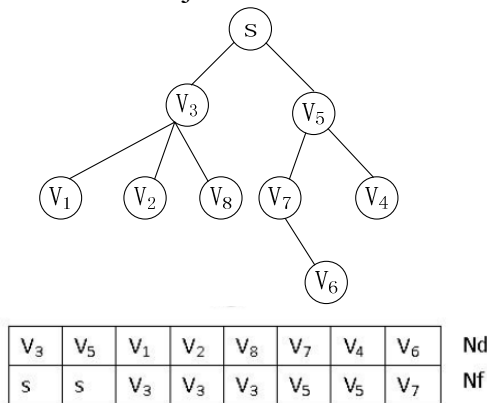


Figure 1. Tree structure

2) Crossover operator

Crossover operator is to use an individual, which is selected in the population, combining with the optimal tree. According to the order of joining, the nodes are sorted according to the order of joining, that is to say, more near to source node, more front. In the crossover, each node is examined sequentially. Scanning array $T[i]$ and *Optimal*, and executing execute the following operation: for each node which is not in *New*, if its parent node is s which is in $T[i]$ or *Optimal* and s has the remaining out-degree, it is selected to add into *New*, and then update the value of delay and out-degree; if not, check that whether the parent node has been added to Nd of *New* array and has the remaining degree or not. If so, the node and its parent node can be added

to the *New*, update the value of corresponding node's delay and parent node's out-degree. If above all are not satisfied, ignore it and go to the next. The operation executes until all nodes are added to *New*. If the total cost of *New* is smaller than $T[i]$, $T[i]$ would be replaced by *New*.

3) Mutation

Mutation operator selects a node, and alters the parent node of the selected node. New parent node must meet the constraints, then update the value of delay and out-degree; otherwise to choose again. As shown in the figure 2, the parent node of v_8 is v_3 , the parent node becomes v_5 after executing mutation operator.

V3	V5	V1	V2	V8	V7	V4	V6
s	s	V3	V3	V3	V5	V5	V7

↓

V3	V5	V1	V2	V8	V7	V4	V6
s	s	V3	V3	V5	V5	V5	V7

Figure 2. Mutation

4) Local optimization

Corresponding to each tree, there is an array to store node delay. To optimize node with the maximum delay, that is to try to select the node with smaller delay as parent node under the constraint conditions, until the updating stops.

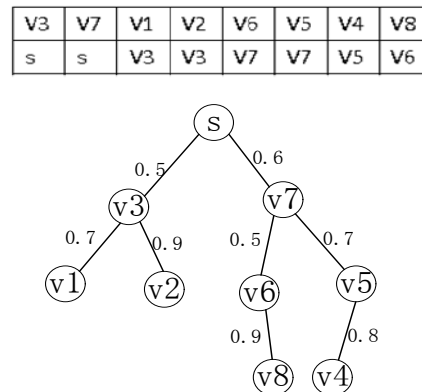


Figure 3. The spanning tree before optimization

For the tree in the figure 3, the node with the maximum delay is v_4 . For the source node has no the remaining degree, it is compared sequentially according to the order of node in the array except

itself. We compare the next node v_3 , then v_3 can be selected as the parent node of v_4 because of the delay decreases. Of course, the spanning tree will be adjusted. After updating, the node with maximum delay is v_8 , the parent node of which is re-selected. At this time, the node v_3 and node v_7 have no remaining degrees, so v_1 is compared with the parent node of v_8 , and then the node v_1 is selected as the parent node of v_8 because of the delay decreases. The spanning tree is updated. At this time, the node with the maximum delay is still v_8 . The spanning tree can no longer be updated, therefore, local search stops. The result is shown in figure 4.

v3	v7	v1	v2	v4	v6	v5	v8
s	s	v3	v3	v3	v7	v7	v1

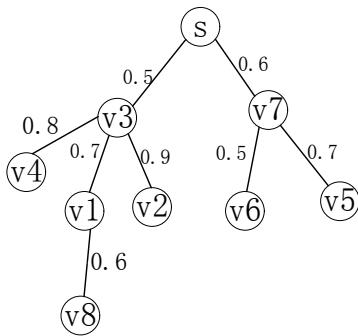


Figure 4. The spanning tree after optimization

IV. SIMULATION RESULTS

In this section, we evaluate the performance of MA. In our study we adopted MRSIM which is developed by Salama et al. of Advanced Computational Communication Center in North Carolina University. MA and several other algorithms were realized in MRSIM[13]. The network topology used in the experiment was randomly generated according to the approach of Waxman. The generator first randomly distributes nodes over a rectangle area with size $[800*800]$ and then creates links between these nodes at random. The edge delay is set to the distance between its corresponding nodes. End-hosts are randomly selected. Two simulations were conducted in terms of the experimental purposes.

Figure 5 shows the total cost of MA, GA and ACO. We can see that the total cost increases with

the increase of the node number. The curves in the figure show that it spends similar cost for MA to construct multicast tree with ACO algorithms although there is difference between the two algorithms in finding the best solution, approximate sameness is shown in different node number. Therefore, it indicates that the solution MA finds is not worse than ACO. While the total cost of GA is always larger than other algorithms.

Convergence time is an important performance index of algorithm. It can be seen from the figure 6 that with the growth of node number, the convergence time of all algorithms increase as well. But it takes shorter time for MA to converge than ACO algorithms. This is more obvious when there are more than 50 nodes in the network. The convergence time of MA no have obvious gap compared with the convergence time of GA.

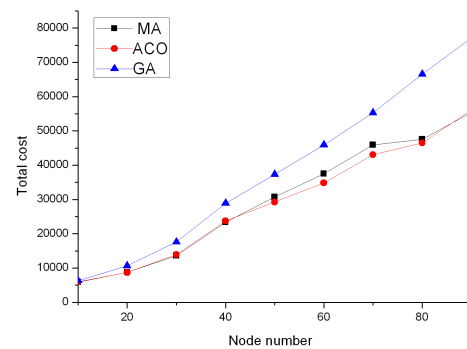


Figure 5. The total cost

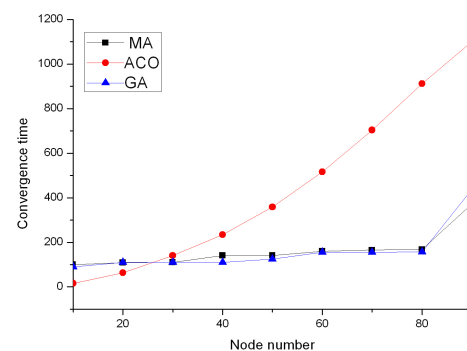


Figure 6. The convergence time

Based on the above experimental results, it can be found that MA is much faster in convergence speed than the ACO, and that the total cost of their optimal trees is similar. Meanwhile, the total cost of MA is smaller than GA, and their convergence time

is similar. So we can conclude that the MA is a more effective algorithm considered both the total cost and the convergence time.

V. CONCLUSION

Based on the principles of MA, we put forward a new algorithm and apply it to solving DDCMST problem. MA combines the global optimize operation and local optimize operation, namely the algorithm not only has a strong ability of global optimization, but also execution local search, so it can eliminates bad individuals as soon as possible, accelerates the speed of the algorithm, and assures the high convergence performance. To test the effectiveness of our algorithm, we compared it with other commonly used multicast optimization algorithms on the criteria of convergence time and total cost. The simulation results show that the performance of new algorithm is better in solving DDCMST problem than the other two algorithms. In the future, we will try to make the algorithm more adaptable by applying it in practice.

ACKNOWLEDGMENT

The study is supported by Natural Science Foundation of Shandong Province (Grant No. ZR2011FM021), and Science and Technology Development Program of Jinan (Grant No.201102010).

REFERENCES

- [1] S. E. Deering, D. R. Cheriton, "Multicast routing in internetworks and extended LANs," *ACM Transactions on Computer System*, 1990, 8(2): pp. 85-110.
- [2] A. El-Sayed, V. Roca, L. Mathy, "A Survey of Proposals for all Alternative Group Communication Service," *IEEE NETWORK*, 2003, 17(1):pp. 46-51.
- [3] J. Saltzer, D. Reed, D. Clark, "End-to-end arguments in system design," *ACM Transactions on Computer System*, 1984, 2(4):pp. 195-206.
- [4] A. El-Sayed, V. Roca, L. Mathy, "A survey of proposals for an alternative group communication service," *IEEE Networks*, 2003, 17(1):pp. 46-51.
- [5] E. Brosh, A. Levin, and Y. Shavitt, "Approximation and Heuristic Algorithms for Minimum-Delay Application-Layer Multicast Trees," *IEEE/ACM TRANSACTIONS ON NETWORKING*, VOL. 15, NO. 2, APRIL 2007:pp. 473-484.
- [6] A.B. Ning, L. Ma, X.H. Xiong, "Solving Degree-constrained Minimum Spanning Tree with a New Algorithm," 2007 International Conference on Management Science & Engineering (14th), August 20-22, 2007:3pp. 81-886.
- [7] Baoliu Ye, Minyi Guo, Daoxu Chen, Sanglu Lu, "A degree-constrained QoS-aware routing algorithm for application layer multicast," *Information Sciences* 177 (2007) pp. 3613 - 3626.
- [8] T. Chen, B.S. Shan, X. Xiong, "Constructing a Degree-Constrained Low Delay Application Layer Multicast Tree," 978-1-4244-2108-4/08/© 2008 IEEE: pp. 1-4.
- [9] S. Y. Tseng, Y. M. Huang, C. C. Lin, "Genetic algorithm for delay and degree-constrained multimedia broadcasting on overlay networks," *Computer Communications*. 2006, 29(17): pp. 3625-3632.
- [10] S. Y. Tseng, Y. M. Huang, C. C. Lin, "Ant colony-based algorithm for constructing broadcasting tree with degree and delay constraints"[J]. *Expert Systems with Applications*, 2008,35(3):1473-1481.
- [11] P. Moscato, M.G. Norman, "A Memetic Approach for the Travelling Salesman Problem-implementation of a Computational Ecology for Combinatorial Optimization on Message-passing Systems," In *Proceedings of the International Conference on Parallel Computing and Transport Applications*.1992.
- [12] Q. H. Nguyen, Y. S. Ong, and N. Krasnogor. "A study on the design issues of Memetic Algorithm," *IEEE Congress on Evolutionary Computation*. Singapore, 2007: pp. 2390-2397.
- [13] H. F. Salama, D. S. Reeves, and Y. Viniotis, "Evaluation of multicast routing algorithms for real-time communication on high-speed networks," *Selected Areas in Communications*, *IEEE Journal on*, vol. 15, 1997: pp.332-345.