ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.19, No.6, June 2008, pp.1546−1554
DOI: 10.3724/SP.J.1001.2008.01546

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

可扩展的多目标最优化多播路由*

胡光岷[1+], CHANG Rocky[2]

[1](电子科技大学 通信与信息工程学院,四川 成都 610054)
[2](香港理工大学 计算科学系,香港)

Multi-Objective Optimization Multicast Routing for Forwarding State Scalability

HU Guang-Min[1+], CHANG Rocky[2]

[1](School of Communication and Information Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China)
[2](Department of Computing, Hong Kong Polytechnic University, Hong Kong, China)
+ Corresponding author: E-mail: hgm@uestc.edu.cn

Hu GM, Chang R. Multi-Objective optimization multicast routing for forwarding state scalability. *Journal of Software*, 2008,19(6):1546−1554. http://www.jos.org.cn/1000-9825/19/1546.htm

Abstract: The approach is to include forwarding state scalability as one of the optimal objective when constructing new multicast trees. This multi-objective optimization approach can be applied to many existing multicast state reduction methods. In this paper, the approach is illustrated by applying it to aggregated multicast (AM) and dynamic tunnel multicast (DTM). Both AM and DTM routing problems are formulated as multi-objective optimization problems, and both heuristic and genetic algorithms are proposed for solving them. Based on the experimental results, the approach can further improve the forwarding state scalability of both approaches by reducing the number of aggregated trees required by the AM method, and by increasing the number of non-branching nodes for the DTM method.
Key words: multi-objective optimization; multicast routing; scalability

摘 要: 提出了一种提高多播可扩展性的新思路——将多播可扩展性作为一个最优化目标引入到多播路由算法的设计中,采用多目标最优化路由算法,提高现有多播可扩展性方法的效率.采用多目标最优化路由设计方法对AM(aggregated multicast)和DTM(dynamic tunnel multicast)两种方法进行改进,给出了相应的最优化目标、启发式多目标最优化多播路由算法和多目标最优化多播路由遗传算法.对于AM方法,使用该算法可以有效地减少汇聚多播树的数量;对于DTM方法,使用该算法可以有效地增加非分枝节点的数量,减少多播状态.
关键词: 多目标最优化;多播路由;可扩展性
中图法分类号: TP393 **文献标识码**: A

* Supported by the Grant of the Hong Kong Polytechnic University of China under Grant No.COMP-H-ZJ83 (香港理工大学基金)
Received 2004-10-03; Accepted 2006-04-27

1 Introduction

Scalability is among the critical issues that delay IP multicast development. The major reason that causes the problem in scalability is that a multicast distribution tree requires all tree nodes to maintain per-group (or even per-group/source) forwarding state, which grows at least linearly with the number of concurrently active groups growth. When active groups and sources in network becomes large, the corresponding multicast forwarding table will also be very large, which will directly lead to high router cost and low forwarding performance.

Recently, much research effort has focused on the improvement of multicast scalability. A class of approaches to scaling multicast routing protocols is to eliminate unnecessary forwarding states by taking the advantage of the fact that multicast trees usually contain long and un-forked tree branches. Dynamic Tunnel Multicast (DTM), which is an example in this class, sets up IP tunnels dynamically to bypass routers on these branches[1]. RE-UNITE, another example, achieves the same effect but without requiring IP tunnel setup[2]. Since this class of approaches reduces forwarding states for individual multicast trees, we refer them collectively to as intra-group state elimination. Another class of approaches seeks to aggregate “closely related” state information for a number of multicast trees passing through a router, i.e., inter-group state aggregation. However, the unicast-like state aggregation is difficult to achieve for multicast addresses, because of the flat multicast address space and a lack of structures to assign them. Therefore, the state aggregation can be leaky or non-leaky[3,4]. This approach essentially separates the entire packet processing into input filtering and output filtering. By doing so, more states can be aggregated on each router interface. The trade-off between the leaky and non-leaky approaches has been analyzed in Ref.[5]. Aggregated multicast (AM), a third approach, employs both state elimination and aggregation for a number of multicast groups[6]. The main idea is to use a small set of multicast trees, which are called aggregated trees, to distribute packets for a large number of multicast groups.

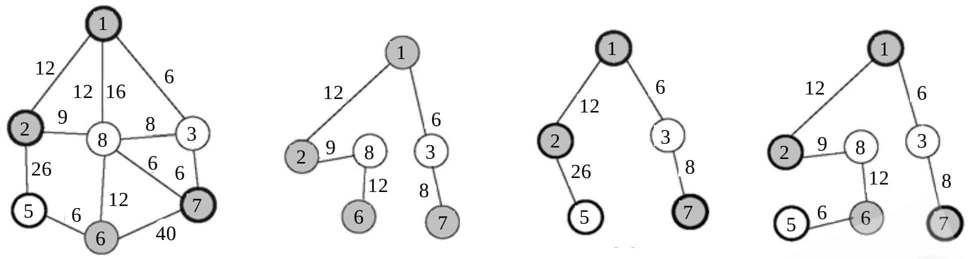
The state scalability performance achieved by the three classes of approaches discussed above is clearly dependent on the structure of the multicast trees under consideration. For example, more routers on a multicast tree can be tunnelled through by the DTM approach if the tree contains longer and more non-branching nodes. Motivated by the relationship between the structure of multicast trees and the forwarding state scalability, we make multicast scalable while building the new multicast tree. This is in stark contrast to all the previous approaches that attempt to reduce forwarding states after the new multicast trees have been built. As a result of including the state scalability as a routing criterion, the routing problem can be formulated as a multi-objective optimization (MCO) problem. Our algorithms to be presented later generate multicast trees that are more suitable for DTM and AM to reduce forwarding states. For example, in the case of AM, the MCOTs are more similar to the aggregated trees.

2 Multi-Objective Optimization Multicast Routing for Forwarding State Scalability

2.1 AM routing for forwarding state scalability

We use the example in Fig.1 to illustrate how the scalability-aware approach can be used with the AM method to reduce the multicast forwarding states. Figure 1(a) shows a network topology with two multicast groups. The nodes joining the first group are indicated by shaded circles, while the nodes joining the second group are indicated by bold-bordered circles. For example, Fig.1(b) shows that the nodes in the first group form a Steiner minimal tree (SMT) among themselves, which also serves as an aggregated tree. Another SMT is established for the second group, as shown in Fig.1(c). However, the second SMT cannot use the aggregated tree to forward its multicast packets, because the two trees are not similar enough[6]. On the other hand, the AM-aware approach can produce a tree for group 2, as shown in Fig.1(d), which is similar to the aggregated tree at the expense of a higher cost.

Therefore, the aggregated tree can forward packets for the second group by simply adding an IP tunnel from node 6 to node 5.

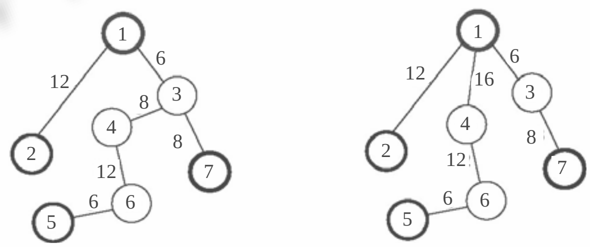


(a) (b) (c) (d)

Fig.1 An example to compare Steiner minimal trees with and without scalability awareness

2.2 DTM routing for forwarding state scalability

We use a shortest-path tree (SPT) algorithm to construct a multicast tree rooted at a source node and connected to all member nodes. A SPT algorithm would generate a tree for the second group in Fig.2(a). However, if we take state scalability into consideration, the resulting multicast tree may look like the one in Fig.2(b). Therefore, the tree generated from the DTM-aware approach contains more nonbranching nodes than the one produced by a pure SPT algorithm; thus, more routers can be tunnelled through.



(a) (b)

Fig.2 An example to compare shortest-path trees with and without scalability awareness

3 Heuristic Multi-Objective Optimization Multicast Routing Algorithm for State Scalability

3.1 Multi-Objective optimization DTM routing

A multicast tree based on SPT is constructed to minimize the end-end path cost between a source and each member node. Using the DTM-aware routing, we introduce a new objective of minimizing the total number of branching nodes. A branching node is one with more than one nodal degree. As a result, the DTM-aware routing algorithm could eliminate more multicast states by producing more non-branching nodes.

We let $T=(V_{SPT}\subset V, E_{SPT}\subset E)$ that defines a shortest-path tree from a source node $u_s\in V$ to a given set of member nodes $V_m\subset V$. The DTM-aware routing problem can be formulated as an MCO problem

$$\text{minimize}\begin{cases}\sum\limits_{e\in E(T_m)} C(e)\\ \sum\limits_{v\in V(T_m)} X_v,\ X_v=\begin{cases}1, & v\text{ is a branching node}\\ 0, & \text{otherwise}\end{cases}\end{cases}\tag{1}$$

As discussed in Section 1, we use a weighted sum to combine the two conflicting objectives. Denote the weights for the second objective in Eq.(1) by λ thus, Eq.(1) becomes

$$\text{minimize} \sum_{e\in E(T_m)} C(e) + \lambda \times \sum_{v\in V(T_m)} X_v \tag{2}$$

Moreover, we define a weighted-sum cost of a tree path P as

$$C_w(P) = \sum_{e\in P} C(e) + \lambda \times DC_p \tag{3}$$

3.2 A heuristic algorithm for multi-objective optimization DTM routing

We use the following notations when describing the DTM-aware routing algorithm.

$M(i)$: A set of nodes and edges that have already been included in the SPT at the end of the ith iteration.

$R(i)$: A set of nodes in V_m that have not been included in the SPT at the end of the ith iteration.

$C(P)$: The cost of a path P based on the cost function C that is the sum of $C(e)$ for all the edges on this path.

DTM-aware routing algorithm:

1. For each node $v\in V_m$,
   (a) Compute k shortest paths from the root node us to v based on the cost function C, and denote this set by SP_v.
   (b) Find the shortest path in SP_v and denote it by $P_{\min,v}$.
   (c) Compute $C_w(P)$ for each path $P\in SP_v$, as defined in Eq.(3).
   (d) Find from SP_v a shortest path from u_s to v based on $C_w(P)$, and denote this path by P_v.
   (e) Compute $C_{diff}(P_v)=C_w(P_v)-C(P_{\min,v})$.
2. Set i=1, $R(i)=V_m$ and $M(i)=\{u_s\}$.
3. While $R(i{+}{+})\neq\varnothing$,
   (a) Find a node from $R(i)$, such that the path from u_s to this node has a smallest C_{diff} , and denote this node by u_m. If there is a tie, pick the one that has a smallest C_w.
   (b) Insert u_m, and the edges and other nodes on P_{u_m} into $M(i)$, and remove u_m from $R(i)$.
   (c) Check each node $v\in R(i)$ to see whether there is another path that has a smaller weighted-sum cost than the current $C_wP(v)$. If so, update P_v and $C_wP(v)$.

The DTM-aware routing algorithm requires setting up of k shortest paths at the beginning of the algorithm[7]. This is because the costs introduced by the branching nodes in the middle of the algorithm (under Step 3) may alter the shortest paths computed at the beginning of the algorithm (under Step 1). If k is sufficiently large, it is very likely that the shortest paths computed by each node to other nodes based on the weighted-sum costs are truly shortest. We use k=3 in our simulation experiments to be presented in Section 4. The computation complexity of the DTM-aware routing algorithm is given by $O(|E|+|V|\log|V|+k|V|+|V_m|^2)$.

3.3 Multi-Objective optimization AM routing

Assume that an aggregated tree has already been constructed, denoted by T_A. Now a new multicast group becomes active and we use the AM-aware routing approach to a tree for this new group, denoted by T_m. Besides the criterion of minimizing the total edge cost, the second criterion is to minimize the use of the edges that are not on T_A. However, sometimes it may be necessary to use the edges not on T_A. In this case, the tree construction should try to use edges that have been used by other trees other than T_A. Therefore, a third criterion is to minimize the use of the edges that have not been used by any existing multicast trees, and we denote this set of edges by $\bar{E}$.

We let $T:=(V_{SMT}\subset V,E_{SMT}\subset E)$, which defines a Steiner tree for a given set of member nodes. Thus, the AM-aware routing problem can be formulated as an MCO problem

$$\text{minimize}\begin{cases}\sum_{e\in E(T_m)} C(e)\\ \sum_{e\in E(T_m)} Z_e\times C(e),\ Z_e=\begin{cases}1, & e\notin E(T_A)\\ 0, & \text{otherwise}\end{cases}\\ \sum_{e\in E(T_m)} W_e\times C(e),\ W_e=\begin{cases}1, & e\in \overline{E}\\ 0, & \text{otherwise}\end{cases}\end{cases} \tag{4}$$

Since the MCO-Steiner problem is a generalization of the Steiner problem, which is known to be NP-complete[8], the MCO problem formulated in Eq.(4) must also be NP-complete.

Same as for the DTM, we use a weighted sum to combine the three conflicting objectives. Denote the weights for the second and third objectives in Eq.(4) by σ_1 and σ_2, respectively. Thus, Eq.(4) becomes

$$\text{minimize}\sum_{e\in E(T_m)}(C(e)+\sigma_1\times Z_e\times C(e)+\sigma_2\times W_e\times C(e)) \tag{5}$$

Similar to Eq.(3), we define a weighted-sum cost of a tree path P for the AM-aware routing algorithm as

$$C_w(P)=\sum_{e\in P}(C(e)+\sigma_1\times Z_e\times C(e)+\sigma_2\times W_e\times C(e)) \tag{6}$$

3.4 A heuristic algorithm for multi-objective optimization AM routing

We use the same definitions of $M(i)$ and $R(i)$ as for the DTM-aware routing algorithm.

AM-aware routing algorithm:

1. For each node $u\in V_m$,
   (a) Compute k shortest paths trees from itself to each other node in the network based on the cost function C, and denote this set of shortest paths by SP_u.
   (b) Compute $C_w(P)$ for each path $P\in SP_u$, as defined in Eq.(6).
   (c) Find a shortest path based on $C_w(P)$ from itself to each other node in the network.
2. Pick a node from V_m to be a source node, denoted by u_s. Set i=1, $R(i)=V_m-\{u_s\}$ and $M(i)=\{u_s\}$.
3. While ($R(i{+}{+})\neq\varnothing$),
   (a) Find a node in $R(i)$, such that the path from this node to the nodes in $M(i)$ is the shortest in terms of the weighted-sum costs. Denote this node by u_m.
   (b) Insert u_m and the edges on this shortest path into $M(i)$, and remove u_m from $R(i)$.
   (c) For each node $u\in V_m$, if the inclusion of u_m and the associated edges changes $C_w(P)$, re-compute $C_w(P)$ from u to other nodes not in $M(i)$ for each path $P\in SP_u$, and find a shortest path based on $C_w(P)$.
   (d) Check the nodes in $M(i)$ other than um and us to see whether they should switch their paths. If so, update $M(i)$ accordingly.

The algorithm above can be used to solve general MC-SMT problems. Under certain objective, the shortest path computed based on the original costs (in Step 1(a)) may not be congruent with the one based on the weighted-sum costs (in Step 3(a)). To overcome this problem, we compute k shortest paths[7], instead of just one in the beginning of the algorithm. The complexity of this algorithm above is therefore given by $O(k|V_m|(|E|+|V|\log|V|+k|V|))$. However, this problem does not occur in our problem formulation for the MCO-AM approach in Eq.(1). Therefore, we can set k to 1, and the complexity of the algorithm is reduced to $O(|V_m|(|E|+|V|\log|V|))$.

4 A Genetic Algorithm for Multi-Objective Optimization AM Routing

4.1 NSGA-II algorithm

Simple, robust, and widely used Genetic Algorithms are adapted to find the optimal solution in complex and large search space. This paper adopts multi-objective optimization selection routing algorithm which is based on NSGA-II algorithm. The basic idea is to classify all the individuals in terms of their levels. Before using the selection operator, classification and sort has been finished in population according to the relationship of dominating and being dominated. Each individual is denoted by a virtual fitness evaluation value (generally in proportion to scale of population), and individuals at the same level have the same virtual fitness evaluation value, which guarantees their same copy probability. To maintain diversity of population, the classified individuals share their virtual fitness evaluation value. NSGA, turning computing of multi-objective function into computing of virtual fitness evaluation, can deal with the optimization of the multi-objective function, maximization and minimization. Restricted by the length of the paper, we don't describe NSGA-II algorithm[9] in detail.

4.2 A genetic algorithm for multi-objective optimization AM routing

We use the same definitions of $M(i)$ and $R(i)$ as for the DTM-aware routing algorithm described in Section 3.2.

Genetic Algorithm for Multi-objective optimization DTM routing:

(1) For each node $u \in V_m$, Compute k shortest paths trees from itself to each other node in the network based on the cost function C, and denote this set of shortest paths by SP_u.

(2) Pick a node from V_m to be a source node, denoted by u_s. Based on root node u_s, select the shortest path from SP_u to generate the initial multicast tree P_0 (that is, initial population) whose value is N.

(3) The population P_0 is sorted into different non-domination levels. Every multicast tree in P_0 is assigned a fitness equal to its non-domination level.

(4) Binary tournament selection, recombination and mutation operators are used to create an offspring population of Q_0 size N.

(5) Combine parent and offspring populations and create $R_t = P_t \cup Q_t$, perform a non-dominated sorting to R_t and identify different fronts: F_i, $i=1,2,\ldots,etc$.

(6) Set new population $P_{t+1}=$. Set a counter $i=1$. Until $|P_{t+1}|+|F_i|<N$, perform $P_{t+1}=P_{t+1}\cup F_i$ and $i=i+1$.

(7) Perform the sorting procedure and include the most widely spread solution. Create offspring Q_{t+1} from P_{t+1} by using tournament selection, crossover and mutation operators. Turn to Step 4 until the termination condition is met.

5 Performance Evaluation

In this section, we evaluate the additional forwarding state reduction after applying the scalability-aware approach to the AM and DTM schemes. For this purpose, we generate random graphs based on Waxman's algorithm in which the probability of having an edge between nodes u and v is given by $\lambda^*\exp((-d(u,v))/(\rho L)$, where $d(u,v)$ is the distance between nodes u and v, and L is the maximum possible distance between any pair of nodes[10]. In our simulations, λ and ρ are set to 0.25 and 0.2, respectively. All the simulation results are presented as averages obtained from 100 independent experiments.

5.1 Heuristic algorithm for dynamic tunnel multicast-aware routing

To evaluate the efficacy of DTM-aware routing, we use a random graph of 500 nodes with 150 member nodes, and consider the value of λ between 0 and 12. The case of $\lambda=0$ is equivalent to the pure DTM approach. Here we measure the forwarding table size based on the number of entries in the table. Figure 3 shows the results for the shortest-path trees generated by SPT, SPT/DTM, and SPT/DTM-aware routing algorithms. The results show that the DTM-aware approach is able to further reduce the forwarding table size by a noticeable amount, and the gain also increases with the number of concurrent multicast groups. When the number of groups reaches around 1 000, the gain of using the DTM-aware routing is approximately 25% for an λ of 12. Clearly, a higher λ facilities even more state reduction, but the rate of increase would eventually flatten out. Fig.3 also shows the increase in the number of non-branching nodes as λ increases, which is responsible for the reduction in the forwarding table size.

Next we measure the additional cost incurred by the scalability-aware approach in terms of the increase in the total cost of the edges. As shown in Fig.4, the tradeoff for the improvement in the state scalability is the increase in the total cost. When λ=12, the total cost could be increased by 50%, but the increase would also eventually flatten out as λ increases further.

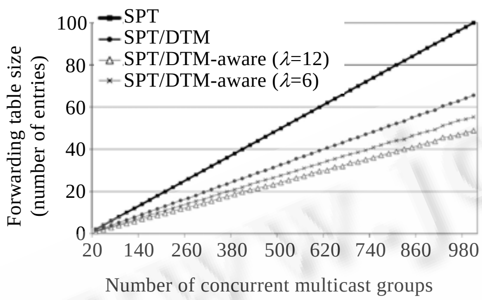


Fig.3 Forwarding table sizes for the DTM-aware multicast routing

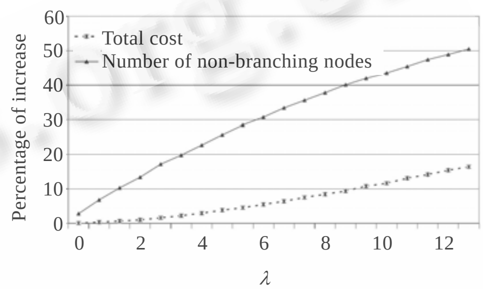


Fig.4 Increase in the total tree cost and the number of non-branching nodes for the DTM-aware multicast routing

5.2 Heuristic algorithm for aggregated multicast-aware routing

In evaluating the AM-aware routing, we generate a random graph consisting of 100 nodes of which the number of member nodes is 30. We consider σ_1=0, 0.2, 0.4, 0.6 and $\sigma_2=\sigma_1/2$. The case of $\sigma_2=\sigma_1=0$ will produce the classic SMT.

The AM scheme always maintains a set of aggregated trees. After a new multicast tree has been constructed, an AM manager attempts to match it to one of the aggregated trees in the set. If the matching is unsuccessful, this tree will be added to the set. As a result, the AM approach will eventually yield a set of aggregated trees, some of which are used to forward packets for multiple multicast groups. Thus, one way to evaluate the total number of forwarding states required for a given number of multicast groups is based on the number of the aggregated trees required. A small set means a better tree aggregation across the groups. Figure 5 shows the results of the AM-aware routing. Compared with the AM scheme alone ($\sigma_2=\sigma_1=0$), the scalability-aware approach can significantly reduce the number of aggregated trees. For 9 000 multicast groups, the number of aggregated trees can be reduced by half.

On the other hand, the scalability-aware approach inevitably increases the total cost of the multicast trees. In the AM approach, the total cost is computed by summing the total cost of the aggregated tree and tunnel cost. Figure 6 shows the percentage of increase in the total cost as compared with the original set of SMTs. The figures show that the AM scheme incurs at most 16.3% in the total cost, but this increase declines with the number of multicast groups. Similar to the AM scheme, the AM-aware scheme incurs additional costs which are slightly higher than that of AM (within 2% at most).

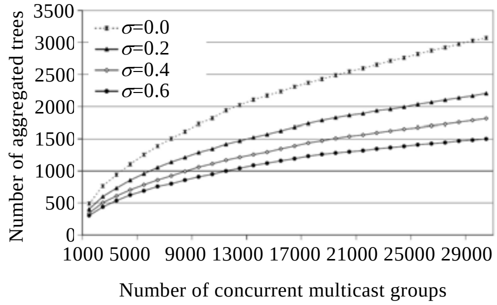


Fig.5 Number of aggregated trees required by the AM-aware routing

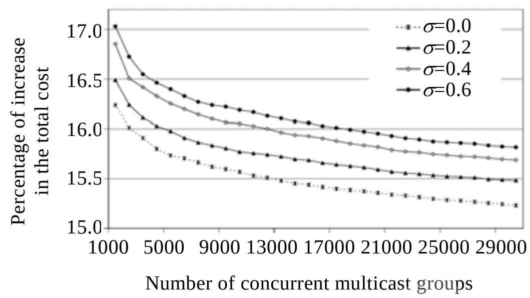


Fig.6 Increase in the total cost of the multicast trees by the AM-aware routing

5.3 Genetic algorithm for aggregated multicast-aware routing

To evaluating the Genetic Algorithm for multi-objective optimization routing, we generate a random graph same as in the Section 5.2. All the parameter used in this simulation is same as that in the Section 5.2. Compared with the AM scheme alone, the Genetic Algorithm for Multi-objective optimization routing can significantly reduce the number of aggregated trees as the heuristic algorithm for aggregated multicast-aware routing algorithm (Fig.7). In most cases, the genetic algorithm can achieve better results. But for different number of multicast group, Genetic Algorithm is not as stable as the heuristic algorithm is.

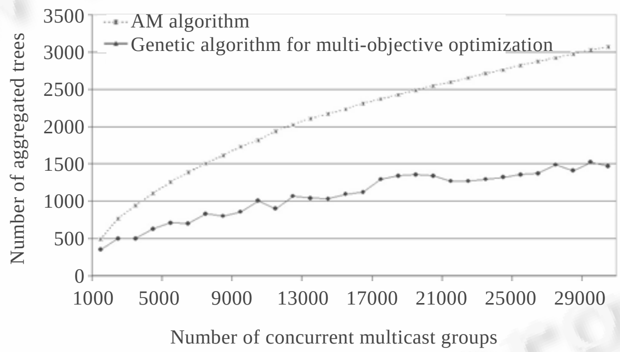


Fig.7 Increase in the total cost of the multicast trees by the genetic algorithm

6 Conclusions

In this paper, we have proposed a new algorithm to build multicast tree, which can generate multicast tree with multi-objective. We have illustrated how the approach can be applied to Aggregated Multicast and Dynamic Tunnel Multicast. Moreover, we have formulated both AM-aware and DTM-aware routing problems as multi-objective optimization problems, and obtained an efficient heuristic algorithm and genetic algorithms for solving them. The simulation results have shown that this approach can significantly improve the results obtained from the AM and DTM approaches alone. Although the total costs of the multi-objective optimized trees are higher, but these additional costs can be easily adjusted by tuning the parameters in the algorithms.

References:

[1] Tian JN, Neufeld G. Forwarding state reduction for sparse mode multicast communication. In: Proc. of the IEEE INFOCOM. 1998.

[2] Stoica I, Eugene Ng T, Zhang H. Reunite: A recursive unicast approach to multicast. In: Proc. of the IEEE INFOCOM. 2000.

[3] Thaler D, Handley M. On the aggregatability of multicast forwarding state. In: Proc. of the IEEE INFOCOM. 2000.

[4] Boudani A, Cousin B. A new approach to construct multicast trees in MPLS networks. In: Proc. of the IEEE ISCC. 2002.

[5] Radoslavov PI, Govindan R, Estrin D. Exploiting the bandwidth-memory tradeoff in multicast state aggregation. Technical Report, USC, 1999. 99−697.

[6] Cui JH, Maggiorini D, Kim JY, Boussetta K, Gerla M. A protocol to improve the state scalability of source specific multicast. In: Proc. of the IEEE Globecom. 2002.

[7] Eppstein D. Finding the *k* shortest paths. SIAM Journal on Computing, 1999,28(2):652−673.

[8] Garey T, Johnson D. Computers and Intractability—A Guide to the Theory of NP-Completeness. New York: Freeman, 1979.

[9] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans. on Computation, 2002,6(2):182−197.

[10] Waxman BM. Routing of multipoint connections. IEEE Journal Selected Areas in Communications, 1988,6(9):1617−1622.

HU Guang-Min was born in 1966. He is a professor at the University of Electronic Science and Technology of China. His current research areas are network behavior and protocol, etc.

CHANG Rocky was born in 1955. He is an associate professor at the Hong Kong Polytechnic University. His current research areas are computer network, etc.