

A Modified ACO Algorithm for Multicast State Scalability Problem Based on Multicast Tree Similarity

Fangjin Zhu*, Hua Wang*

**School of Computer Science and Technology, Shandong University, Jinan, China*

zhufj@sdu.edu.cn, wanghua@sdu.edu.cn

Abstract— Traditional IP multicast technology establishes and maintains a tree for each multicast group, and this leads to a serious state scalability problem when large numbers of multicast groups exist in a network. As a novel scheme to address this scalability problem, aggregated multicast forces multiple groups which have identical or similar trees to use one deliver tree. Given the number of multicast trees, searching for an aggregation solution to minimize the average bandwidth waste rate is an important optimization problem in aggregated multicast. In this paper, we propose a modified ant colony optimization algorithm for this optimization problem. We define the similarity of multicast trees, and combine it with similarity sequence information to define the selection heuristic information of our algorithm according to the characteristics of similarity. Simulation results indicate that our algorithm has better evolutionary ability, and the selection heuristic information can overcome the initial blindness and improve the convergence time of ACO algorithm. Compared with a greedy algorithm, our algorithm has better optimization performance.

Keywords— Aggregated Multicast, Tree Similarity, Ant Colony Optimization, Greedy Algorithm, Routing

I. INTRODUCTION

Numerous distributed network applications involving group communications over the Internet have been developed during the past decades. As a mechanism for efficiently supporting group communications, traditional multicast technology [1] constructs one delivery tree for one multicast group (or even one group/source), with each tree requiring the on-tree routers to maintain a forwarding state. When large numbers of concurrent multicast groups exist in a network, the number of forwarding states in routers is expected to grow together with the number of multicast groups. This slows down the routing process and saturates the memory of the router. Additionally, the number of control messages required to maintain the forwarding states grows in the same manner. This phenomenon is referred to as the multicast state scalability problem [2]. This scalability issue has to be addressed before multicast can be viably deployed over the Internet.

There have been many attempts to address this state scalability problem. Some methods, such as Xcast [3] and application-layer multicast [4], resort to another multicast architecture in which all states are eliminated in intermediate

routers. But the former can not support multicast groups which have many members, and the latter sacrifices multicast efficiency. The other methods [5] use tunnel technology, and the non-branch nodes do not maintain forwarding states any further. But the number of multicast trees remains the same, and the overhead of control messages isn't reduced.

To reduce the forwarding states in routers and the overhead of control messages simultaneously, aggregated multicast [2] proposed by Cui *et al.* forces multiple groups which have identical or similar deliver trees to use one deliver tree. This shared deliver tree is called an aggregated tree. The routers maintain only one state for the aggregated tree, instead of upholding one state for each group, thus this scheme leads to the reduction of both the number of forwarding states and the overhead of control messages.

Aggregated multicast is a trade-off of the number of trees and extra bandwidth waste, and is a multi-objective optimization problem [2]. Given the bandwidth waste threshold, minimizing the number of aggregated trees is one optimization problem, and most researches [2], [6]-[9] focus on this optimization problem; given the number of aggregated trees, minimizing the average bandwidth waste rate is another optimization problem which this paper focuses on.

This paper firstly defines the optimization problem to be addressed. The similarity of multicast trees is defined, and is combined with similarity sequence information to define the selection heuristic information according to the characteristics of similarity. And a modified ant colony optimization (ACO) algorithm is designed based on this heuristic information. Simulation results indicate that the heuristic information help the algorithm to overcome the initial blindness and improve the convergence time. Compared with a conventional greedy algorithm, our algorithm has better optimization performance.

This paper is organized as follows: Section II describes the background of aggregated multicast, and defines the optimization problem to be addressed; Section III proposes the modified ACO algorithm to solve the discussed optimization problem; simulation results and analysis are given in Section IV and Section V gives the conclusion.

II. BACKGROUND AND DEFINITIONS

A. Description of Aggregated Multicast

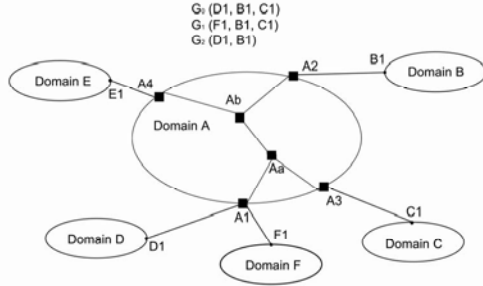


Figure 1. Illustration of aggregated multicast

Figure 1 illustrates a hierarchical inter-domain network peering. Suppose Domain A is the backbone network of an ISP and Domains B, C, D, E, and F are customer networks. There are three multicast groups: G_0 , G_1 , and G_2 . The sub-tree of these groups are (A1, Aa, Ab, A2, A3), (A1, Aa, Ab, A2, A3), and (A1, Aa, Ab, A2), respectively. The three groups in Domain A have identical or overlapping intra-domain subtrees; hence, they can use a shared tree—(A1, Aa, Ab, A2, A3)—called an aggregated tree. In this system, core routers Aa and Ab are required to maintain only one state for the aggregated tree, instead of upholding one state for each group, as in traditional multicast technology. This scheme reduces forwarding states.

Using the aggregated tree to deliver data for G_0 and G_1 is an ideal approach, but for G_2 , bandwidth is wasted on the link from Aa to A3 because A3 is not a member of G_2 . So aggregated multicast induces some extra bandwidth waste while it reduces the number of forwarding states, thus it trades off bandwidth waste for state reduction.

B. Network Model and Definitions

The network is modelled as an undirected graph $N(V, E)$. Each edge (i, j) is assigned a positive cost $c_{ij} = c_{ji}$. A multicast routing algorithm is given to establish multicast trees.

For a multicast group g , its native multicast tree $T^n(g)$ is computed using the given multicast routing algorithm, and the cost of $T^n(g)$ is:

$$\text{cost}(T^n(g)) = \sum c_{ij}, \quad \text{link}(i, j) \in T^n(g) \quad (1)$$

For two multicast groups g and g' , their multicast trees are $T^n(g)$ and $T^n(g')$, respectively. Define the similarity of the two multicast groups as the similarity of their multicast trees:

$$\text{sim}(g, g') = \frac{\text{cost}(T^n(g) \cap T^n(g'))}{\text{cost}(T^n(g) \cup T^n(g'))} \quad (2)$$

where $T^n(g) \cap T^n(g')$ is the shared intersect edges of the two trees, and $T^n(g) \cup T^n(g')$ is the union of their edges. In this paper, the similarity of two groups has the same meaning of the similarity of their multicast trees. They are interchangeable.

The above definition (2) reflects the similarity of two multicast trees, and the aggregation of two trees which have higher similarity leads to smaller bandwidth waste. On the other hand, the definition (2) of similarity has another property: given two items A and B , if item A is the most similar item of item B , item B may not be the most similar one of item A . Due to this property, when B wants to aggregate to A due to its highest similarity, A may not want to aggregate with B . The reason is A may have some neighbours which are more similar to A than B , and the aggregation with B may lead to more bandwidth waste for those neighbours. So, when selecting a group to aggregate to one tree, not only the similarity of them but also the similarity sequence informations are considered.

When multiple groups $\{g_1, g_2, \dots, g_k\}$ aggregate together, i.e., when they use one tree to deliver their data, every group must satisfy with the condition $\text{mem}(g_i) \subseteq \text{leaf}(T)$, where $\text{mem}(g_i)$ is the set of members of g_i , and $\text{leaf}(T)$ is the set of leaves of T . The bandwidth waste rate of each group g_i is:

$$\delta(T, g_i) = \frac{\text{cost}(T) - \text{cost}(T^n(g_i))}{\text{cost}(T^n(g_i))} \quad (3)$$

So, the total bandwidth waste rate of a tree T is defined as the sum of bandwidth waste rate of every group which uses T as its delivery tree:

$$\delta(T) = \sum_{i=1}^k \delta(T, g_i) \quad (4)$$

C. Problem Definition

Given N_g multicast groups and the number L of the aggregated trees, the task is to search for a partition $G = \{G_1, G_2, \dots, G_L\}$. Each group $g_i (i=1, 2, \dots, k)$ in subset $G_l = \{g_1, g_2, \dots, g_k\}$ uses the same aggregated tree T_l as their deliver tree. The goal is to minimize the average bandwidth waste rate.

So the optimization problem is defined by the following grouping mathematical model:

$$\text{minimize} \quad \frac{1}{N_g} \sum_{l=1}^L \delta(T_l) \quad (5)$$

$$\text{subject to} \quad \begin{cases} \bigcup_{l=1}^L G_l = G & (6) \\ G_m \cap G_n = \emptyset, & 1 \leq m \neq n \leq L & (7) \\ G_l \neq \emptyset, & l=1, 2, \dots, L & (8) \end{cases}$$

III. A MODIFIED ACO ALGORITHM

Faure *et al.* define a special example of the above optimization problem and prove it to be NP-Complete [10]. So our problem is also NP-Complete, and needs an approximation algorithm. In this paper a modified ACO algorithm is designed to tackle this problem. Firstly proposed by Dorigo, the ACO algorithm [11] has been successfully

used in many combinational optimization problems. To overcome the initial blindness and improve the convergence time of a classical ACO algorithm, the similarity defined in Section II and the similarity sequence information are used together to define the selection heuristic information.

Now the solution construction process of an ant, the pheromone update rules and the terminal condition of our algorithm are described in the following sub-sections.

A. Solution Construction Process of an Ant

Each ant constructs an entire solution individually. At first selects L groups according to the probability defined in (12) and uses their native multicast trees as the L starting trees. Then aggregates the left groups to these L trees according to the probability defined in (14). After all ants finish their solution construction processes, updates the pheromone and starts the next iteration, until the terminal condition of the algorithm is reached. The probabilities are determined by the combination of pheromone and heuristic information. The definition of pheromone is given firstly.

1) Pheromone: Based on the idea of grouping, define the pheromone $\tau_{g,g'}$ between two groups g and g' as their expectation to aggregate to a tree. The value of $\tau_{g,g'}$ is larger when groups g and g' aggregate to one tree and a better solution is gained. Otherwise, the values of $\tau_{g,g'}$ is smaller. Based on the definition of pheromone between two groups, the pheromone between a group g and a tree T_l , the corresponding subset is $G_l = (g_1, g_2, \dots, g_k)$, is the average value of pheromones amongst group g and the groups $g_i (i = 1, 2, \dots, k)$:

$$\tau_{g,T_l} = \frac{1}{k} \sum_{i=1}^k \tau_{g,g_i} \quad (9)$$

2) Probability of Selecting L Starting Trees: Randomly select a group and use its native multicast tree as the first starting tree. In each step select the most 'distant' tree from the existing trees as the successive starting tree, until L trees are selected. In this paper the 'distance' is reflected by combination of dissimilarity and inverse pheromone between trees.

Based on similarity definition (2), the average dissimilarity amongst a group g and j existing groups $g_i (i = 1, 2, \dots, j)$ is defined as:

$$dis^{(j)}(g) = \frac{1}{j} \sum_{i=1}^j (1 - sim(g, g_i)) \quad (10)$$

Because of pheromone reflecting the expectation of two groups to aggregate to one tree, the group which does not expect to aggregate to currently selected trees should be selected and use its native multicast tree as the next starting tree. So the inverse pheromone is used in this stage. The average inverse pheromone amongst a group g and j existing groups $g_i (i = 1, 2, \dots, j)$ is defined as:

$$int\tau^{(j)}(g) = \frac{1}{j} \sum_{i=1}^j (\tau_{g,g_i})^{-1} \quad (11)$$

The probability of selecting a group g and using its native multicast tree as the $j+1^{th}$ starting tree is defined as:

$$p_g^{j+1} = \frac{[dis^{(j)}(g)]^\alpha [int\tau^{(j)}(g)]^\beta}{\sum_{h \in UnSelected} [dis^{(j)}(h)]^\alpha [int\tau^{(j)}(h)]^\beta} \quad (12)$$

where $UnSelected$ is the set of unselected groups, and α and β reflect the relative importance of dissimilarity and inverse pheromone, respectively.

3) Probability of Aggregating a Group to a Tree: After selecting the L starting trees, the left trees need to aggregate to an appropriate starting tree according to the aggregation probability. The aggregation probability is determined by the similarity and pheromone between a group and a tree.

Given a starting tree T_l , there are already k groups aggregating to it, and its corresponding subset is $G_l = \{g_1, g_2, \dots, g_k\}$. Based on the analysis of similarity in Section II, the heuristic information is determined by both the similarity and similarity sequence information. The heuristic information of a group g aggregating to a tree T_l is defined as:

$$\eta_{g,T_l} = \frac{1}{k} \sum_{i=1}^k \frac{sim(g, g_i)}{s_{g,g_i}} \quad (13)$$

where s_{g,g_i} is the descendant similarity sequence number of group g to g_i . This similarity sequence number resists a less similar group g to aggregate to the tree T_l , thus avoid more total bandwidth waste of the existing groups induced by g .

Based on the definitions (9) and (13), the probability of an ant ω to select a group $g \in UnSelected$ to aggregate to a tree $T_l (l = 1, 2, \dots, L)$ is defined as:

$$p_{g,T_l}^\omega = \begin{cases} \frac{\tau_{g,T_l} [\eta_{g,T_l}]^\gamma}{\sum_{i=1}^L \tau_{g,T_i} [\eta_{g,T_i}]^\gamma}, & g \in UnSelected, \quad l = 1, 2, \dots, L \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

where γ reflects the relative importance of heuristic information to pheromone.

B. Pheromone Update Rules

In this paper the objective function (5) of the optimization problem is used as the fitness function:

$$f(S) = \frac{1}{N_g} \sum_{l=1}^L \delta(T_l) \quad (15)$$

where S is a solution of the optimization problem.

A pheromone update strategy of elite ants is adopted. Each ant constructs an entire solution, and the solution set constructed by all ants is denoted by S^a . Only the solution S^* , $S^* = \arg \min_{S \in S^a} (f(S))$, releases pheromone. The pheromone update rule is defined as:

$$\tau_{g,g'} = (1-\rho)\tau_{g,g'} + \nabla\tau_{g,g'} \quad (16)$$

where $\nabla\tau_{g,g'}$ is the increase of pheromone amongst the groups which aggregate to a tree in the solution S^* :

$$\nabla\tau_{g,g'} = \begin{cases} \frac{Q}{f(S^*) \times \lambda}, & \text{if both } g, g' \in T_l \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

λ ($\lambda > 0$) is an adjustment parameter, and Q ($Q > 0$) is a constant about pheromone increase.

At the same time, to avoid convergence to a local minimum and maintain the exploration ability, a minimum value is set to the pheromone.

C. Terminal Condition

The ACO algorithm is an evolutionary algorithm, and requires multiple iterations. Given a suitable windows size, the most optimal (minimal) values obtained in each iteration are recorded in this window. The variance of these values is used to define terminal condition. When it is less than a given threshold, the algorithm is assumed to reach convergence.

Based on the above definitions, the ACO algorithm, LAT-ACO (L aggregated trees with ACO algorithm), which is used to tackle the optimization problem, is given below:

- Step 1. initialize the parameters
- Step 2. if the terminal condition is reached, stop the algorithm; otherwise, start the next iteration
- Step 3. for each ant, selects L groups successively according to the probability defined in (12) and uses their native multicast trees as the initial L starting trees
- Step 4. aggregates each left group to an appropriate tree according to the probability defined in (14)
- Step 5. update the global optimal solution, and update pheromone based on the global optimal solution and rules defined in (16). Proceed to Step 2
- Step 6. construct the final L aggregated trees based on the global optimal solution and output

IV. SIMULATION AND ANALYSIS

A. Simulation Settings

All the network topologies used by our algorithm are generated by Power Law rule [12].

The random node-weighted model was adopted to distribute multicast group members [13]. Simulation time was fixed at 600 s. The average life span of the multicast groups was 100 s. We adjusted the joining rate of the multicast groups to change the number of groups in the network. We supposed that after 400 s, the state of the multicast becomes stable, and conducted sampling every 10 s. When the simulation time reached 600 s, 20 samples could be obtained.

The average bandwidth waste rate defined in (5) is used to evaluate the optimization performance of the algorithms. We evaluate the average bandwidth waste rate under different number of aggregated trees. But the range of the number of aggregated trees changes with the given number of multicast

groups. We define the *Rate of Aggregated Trees (RAT)* to eliminate this change:

$$RAT = \frac{L}{N_g} \quad (18)$$

B. Evolutionary Ability and Optimization Performance

Figure 2 gives the average bandwidth waste rates when the number of groups is 100 and 1000, and the number of iteration is 1 ($N=1$) and 100 ($N=100$), respectively.

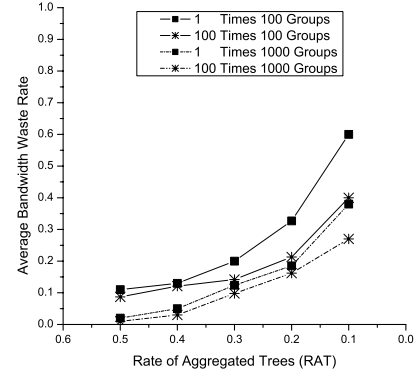


Figure 2. Average bandwidth waste rate when $N=1$ and $N=100$

A trend is shown in Figure 2 that when the *RAT* becoming smaller (the number of aggregated trees becomes smaller), the average bandwidth waste rate becomes higher. This is true because when less trees are allowed in a network, each group has to aggregate with some less similar groups, thus induces more bandwidth waste.

Another trend is that with more iteration, the average bandwidth waste rate becomes lower. This illustrates that our algorithm obtains better solutions with more iteration, thus it has good evolutionary ability.

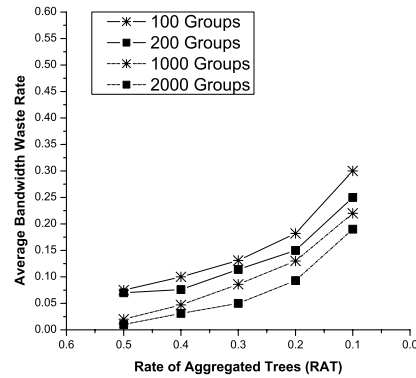


Figure 3. Average bandwidth waste rate with different number of groups

Figure 3 shows the average bandwidth waste rate when the number of groups is 100, 200, 1000, 2000, respectively.

The same trend also holds in figure 3: if the number of aggregated trees becomes smaller (the *RAT* is smaller), the average bandwidth waste rate becomes higher.

Figure 3 also tells us that more groups will obtain less bandwidth waste rate. The reason behind this is that when more groups exist in a network, the groups have more opportunities to share the same members, and this leads to less bandwidth waste rate when they aggregate together.

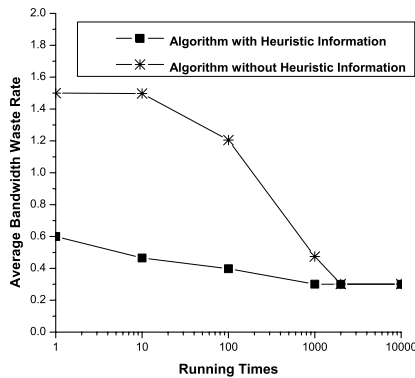


Figure 4. The influence of heuristic information

To show how the heuristic information defined in our algorithm to improve the convergence time, we devise two algorithms: the first one is our LAT-ACO algorithm, in which the heuristic information is combined; the second one is the compared algorithm, which is without the heuristic information. The simulation results shown in figure 4 verify our assumption: the heuristic information can overcome the initial blindness and improve the convergence time of our ACO algorithm.

C. Compared with a Greedy Algorithm

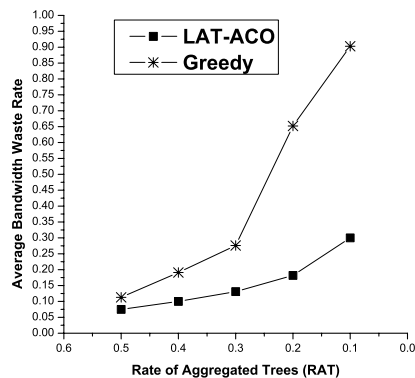


Figure 5. Comparison between LAT-ACO and a greedy algorithm

We compare the optimization performance of our algorithm with a greedy algorithm [6]. The simulation results are shown in figure 5. From figure 5, our algorithm has better optimization performance than that of the greedy algorithm, particularly when the number of aggregated trees is smaller.

V. CONCLUSIONS

In this paper an important optimization problem in aggregated multicast is defined: given N_g multicast groups

and the number L of aggregated trees in a network, $L \ll N_g$, searching for an aggregation scheme to minimize the average bandwidth waste rate. This is an NPC problem and a modified ACO algorithm is proposed. In our algorithm, the similarity of two multicast groups (their multicast trees) is defined, and is combined with the similarity sequence information to define the selection heuristic information according to the characteristics of the similarity. Simulation results indicate that the introduction of the heuristic information overcome the initial blindness and improve the convergence time of ACO algorithm. Compared with a greedy algorithm, our modified ACO algorithm has better optimization performance.

Veering from traditional ACO algorithms, we use the heuristic information and pheromone in two different ways: in the stage of selecting L starting trees, we use the dissimilarity and inverse pheromone; in the stage of aggregating a group to a tree, we use the normal similarity and pheromone. Considering the similarity sequence information can efficiently improve the convergence time of our algorithm. Simulation results verify our modification.

ACKNOWLEDGMENT

This study is supported by CNGI (China Next Generation Internet), China, Grant No. CNGI-04-13-2T, National Natural Science Foundation of China (NSFC No. 60773101).

REFERENCES

- [1] S. E. Deering and D. R. Cheriton, "Multicast routing in datagram internetworks and extended LANs," *ACM Transactions on Computer Systems*, vol. 8, pp. 85-110, 1990.
- [2] L. Li, J.H. Cui, and M. Gerla, "Tackling group-tree matching in large scale group communications," *Computer Networks*, vol. 51, pp. 3069-3089, 2007.
- [3] R. Boivie, N. Feldman, Y. Imai, W. Livens, and M. Dario, "Explicit multicast (Xcast) basic specification," 2001.
- [4] Y. Chu, S. Rao, and H. Zhang, "A case for end system multicast," in *Proc. ACM SIGMETRICS*, 2000.
- [5] B.X. Zhang and H. Mouftah, "Forwarding state scalability for multicast provisioning in IP networks," *IEEE Communications Magazine*, pp. 46-51, 2003.
- [6] L. Li, J.H. Cui, and M. Gerla, "A scalable overlay multicast architecture for large-scale applications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, pp. 449-459, 2007.
- [7] J.H. Cui, L. Li, M. Faloutsos, and M. Gerla, "AQoS: Scalable QoS multicast provisioning in Diff-Serv networks," *Computer Networks*, vol. 50, pp. 80-105, Jan. 2006.
- [8] N. B. Ali, A. Belghith, J. Moulhierac, and M. Molnar, "QoS multicast aggregation under multiple additive constraints," *Computer Communications*, vol. 31, pp. 3564-3578, 2008.
- [9] H. Wang, Z.Q. Ge, C.Y. Yu, and S.W. Yi, "A modified genetic algorithm for the optimization of aggregated multicast," in *Proc. 11th ICACT*, Korea, 2009, pp. 83-88.
- [10] N. Faure, P. Chretienne, E. Gourdin, and F. Sourd, "Biclique completion problems for multicast network design," *Discrete Optimization*, vol. 4, pp. 360-377, 2007.
- [11] M. Dorigo, V. Maniezzo, and A. Colomi, "The ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, Cybernetics, Part B*, vol. 26, pp. 29-41, 1996.
- [12] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the Internet topology," *ACM SIGCOMM Computer Communication Review*, vol. 29, pp. 251-262, 1999.
- [13] J.H. Cui, L. Li, M. Gerla, and M. Faloutsos, "Modelling the spatial properties of multicast group members in the Internet," *UCLA CSD Research Review*, 2003.