A Genetic Algorithm for Minimizing Delay in Dynamic Overlay Networks

Chae Y. Lee and Sang Kun Seo
Dept. of Industrial Engineering, KAIST,
373-1 Kusung Dong, Taejon, Korea
Tel: +82-42-869-2916, FAX: +82-42-869-3110
Email: {chae, ssanggn}@kaist.ac.kr

Abstract

Overlay multicast is an emerging technology for next generation Internet service to various groups of multicast members. It will take the place of traditional IP multicast which is not widely deployed due to the complex nature of its technology. The overlay multicast which effectively reduces processing at IP routers can be easily deployed on top of a densely connected IP network. An end-to-end delay problem is considered which is serious in the multicast service. To periodically optimize the route in the overlay network and to minimize the maximum end-to-end delay, overlay multicast tree is investigated with genetic Algorithm. Outstanding experimental results are obtained which is comparable to the optimal solution and the tabu search.

1. Introduction

Overlay multicast is highlighted as an alternative for providing multicast services on the Internet [1,2, 3, 4, 5, 6, 7, 8]. In overlay multicast, data packets are replicated at end hosts. Each multicast member is responsible for forwarding the packets. Thus, overlay multicast does not change the network infrastructure but performs forwarding at each end host. The logical tree structure of overlay multicast is built to form an overlay multicast network regardless of the underlying physical Internet.

An important practical problem in this paper is to determine how to construct a multicast tree which minimizes the maximum end-to-end delay. There are a number of studies on overlay multicast services in recent literature, mostly due to the efficiency of overlay networks. [6] proposed a heuristic multicast routing algorithm for overlay networks that optimizes the access bandwidth usage at the Multicast Service Nodes' (MSNs) interfaces, while satisfying the end-to-end delay requirements of applications.

The rest of the paper is organized as follows. In Section 2, we formulate the construction of overlay multicast tree that minimize the maximum end-to-end delay. In Section 3, a genetic algorithm is introduced to solve the multicast tree. The performance of suggested algorithms is analyzed in Section 4 with conclusion in Section 5.

2. IP Formulation of Overlay Multicast Trees

We want to find an overlay multicast tree that minimizes maximum overlay delay for all multicast members. To guarantee the quality of data delivery path, the packet processing ability of each member node is also considered with a degree bound. Therefore, this problem is formulated into a degree-bounded minimum delay spanning tree.

Consider a graph $G=(V,E)$ where V is the set of multicast nodes and a source and E is the set of links which constitute a full-meshed overlay network. A node $m \in V$ represents a multicast member that wants to receive services from the source. If no path exists between the source and the multicast member m, the tree recovery is impossible. By assuming at least one path between the source and node m, the failed multicast tree is recovered to connect every multicast member.

Let x_{ij}^{m} be a binary variable to represent a link (i,j) that is employed for a path between the source and multicast member m. If link (i,j) is employed to connect the source and node m, then $x_{ij}^{m}=1$. Otherwise, $x_{ij}^{m}=0$. Then, the following relations hold for every node including the source s.

$$\sum_{j\neq i} x_{ij}^{m} - \sum_{j\neq i} x_{ji}^{m} = 1, \quad \text{for all } m \in V \text{ and } i = s$$

$$\sum_{j\neq i} x_{ij}^{m} - \sum_{j\neq i} x_{ji}^{m} = 0, \quad \text{for all } m \in V \text{ and } i \in V/\{s,m\}$$

$$\sum_{j\neq i} x_{ij}^{m} - \sum_{j\neq i} x_{ji}^{m} = -1, \quad \text{for all } m \in V \text{ and } i = m$$

Let y_{ij} be a binary variable to represent the adoption of link (i,j) for paths in the overlay multicast tree. Then, the following equation holds.

$$x_{ij}^{m} \leq y_{ij}, \quad \text{for all } m \in V \text{ and } i \neq j$$

The above constraint represents that link (i,j) has to be selected in a tree to create the path between the source and multicast member m.

Now, we consider the degree constraint of a multicast member node. By letting w_i be the degree constraint of node i, we have

$$\sum_{j\neq i} y_{ij} + \sum_{j\neq i} y_{ji} \leq w_i, \quad \text{for all } i \in V$$

To build up a spanning tree with n member nodes, we have

$$\sum_{j\neq i} y_{ij} = n-1, \quad \text{for all } i \in V$$

Our objective in the multicast tree is to minimize maximum overlay delay for all end hosts. By letting d_{ij} be the link delay and Z be the maximum end-to-end delay, we have the following equation to minimize Z.

$$\sum_{(i,j)\in E} d_{ij} x_{ij}^{m} \leq Z, \quad \text{for all } m \in V$$

Note in the above expression that the link delay in an overlay network reflects number of hops in the corresponding physical network and the processing time at each physical node. From the above discussion, we have the following binary integer programming formulation.

Minimize Z

subject to

$$\sum_{(i,j)\in E} d_{ij} x_{ij}^{m} \leq Z \quad \text{for all } m \in V$$

$$\sum_{j\neq i} x_{ij}^{m} - \sum_{j\neq i} x_{ji}^{m} = 1 \quad \text{for all } m \in V \text{ and } i = s$$

$$\sum_{j\neq i} x_{ij}^{m} - \sum_{j\neq i} x_{ji}^{m} = 0 \quad \text{for all } m \in V \text{ and } i \in V/\{s,m\}$$

$$\sum_{j\neq i} x_{ij}^{m} - \sum_{j\neq i} x_{ji}^{m} = -1 \quad \text{for all } m \in V \text{ and } i = m$$

$$x_{ij}^{m} \leq y_{ij} \quad \text{for all } m \in V \text{ and } i \neq j$$

$$\sum_{j\neq i} y_{ij} + \sum_{j\neq i} y_{ji} \leq w_i \quad \text{for all } i \in V$$

$$\sum_{j\neq i} y_{ij} = n-1 \quad \text{for all } i \in V$$

$$x_{ij}^{m}, y_{ij} \in \{0,1\}$$

The above overlay multicast tree construction is a well-known degree bounded minimum spanning tree problem, which is NP-hard [9, 10]. Considering the NP-hardness of the overlay multicast tree problem, the proposed linear integer programming may not be effectively solved by any conventional optimization techniques. We consider genetic algorithms as a promising solution procedure for the above tree recovery problem [11].

3. Genetic Algorithms

Genetic algorithms, which are based on the genetic process of biological organism, are adaptive methods and they may be used to solve search and optimization problems [12]. Fitness value, which represents chromosome's performance, is used to determine whether the chromosomes survive or not in next generation.

In this section, we will look over genetic operations that are used in genetic algorithms by its step.

3.1 Tree Representation

To represent an overlay multicast tree as a chromosome, encoding scheme is required. One of the classical and popular encoding scheme is prüfer encoding, which is node and link based encoding scheme and it represents n nodes with $n-2$ digit.

One disadvantage of prüfer encoding scheme is limited locality. By any one change in the tree, prüfer number can be changed dramatically [13].

[14] suggested determinant encoding which is node-based encoding scheme and it represents n nodes with n-1 digit. In determinant encoding, nth digit in chromosome represents the node that node n connect to. [14] shows that determinant encoding outperforms than prüfer encoding in MST problem.

3.2 Initial Population

Initial population procedure is performed at the beginning of the genetic operation and generates initial trees. In this paper, two methods are used – Random Population and Breadth-First Population. Random population is the method which assigns random number in an array so that it can be decoded as an overlay multicast tree. Breadth-First Population (BFP) is based on the assumption that if the node-to-node delay in the network is same, parallel transmission will perform better than series transmission. BFP makes full use of upper level node's degree, and gives higher priority to the node which has higher degree. BFP procedure is as follows.

First connect source to nodes, which have minimum source-to-node delay, among highest degree while source's degree constraint is satisfied. If the nodes which have highest degree are exhausted, the left degrees are assigned to the second highest degree nodes. The nodes, which are connected to the source, connect with the remainder with the same way. This procedure continues until the node-to-be-connected is exhausted. For diversity of tree, each node is selected with probability 0.7.

3.3 Selection

In selection procedure two parents are selected to perform crossover operation. In this paper, two selection procedure is used – tournament selection and ranking selection. In tournament selection, randomly chosen two chromosomes are compared and better one is selected. In this way, two chromosomes are selected so that crossover operation can be performed. In ranking selection, good performing chromosome has higher selection probability. In this paper, we use linear ranking selection, which increase selection probability linearly as rank increase. [15] shows that tournament selection has better time complexity than linear ranking selection, while they have identical performance.

3.4 Crossover & Mutation

Two parents which are selected by above operation generate two offspring by crossover operation. In this paper, we use one-point crossover and uniform crossover. One-point crossover, which is one of the most widely used crossover operation, randomly choose one point of two parents, and swap the fragment.

In uniform crossover, each node is changed with probability 0.5.

Each offspring generated by crossover operation is mutated with very low probability, such as 0.1%~1%[15]. In mutation operation randomly choose two nodes of an offspring and swap in a selected offspring.

3.5 Repair

When performing crossover operation, offspring's degree constraint can be violated because fragments of different chromosomes are mixed. In that case discard degree-violated offspring and perform crossover operation again. And when performing genetic operation, cycle and tree break can happen and they happen at the same time. Figure shows an example when cycle and tree break happen.

When the cycle and tree break happen, discard the offspring and perform genetic operation again.

3.6 Overall Genetic Algorithms procedure

Based on the genetic operation explained above, we will describe overall genetic algorithms procedure. The fitness value of a chromosome is maximum end-to-end delay of overlay multicast tree. The minimum fitness value of chromosomes represents the fitness value in a generation. First, create 100 initial populations with BFP. Second, select two chromosomes with tournament selection. Third, perform crossover operation to the selected two chromosomes in the second step. Fourth, each offspring is mutated with probability 0.01. From the second to fourth steps construct one generation. The generation is terminated when there is no improvement in best fitness value in a generation for cumulative 100 generation.

4. Computational Results

Computational results of the proposed genetic algorithms shows the gap between proposed genetic algorithm and CPLEX and Tabu search.. CPLEX[16] and tabu search[17] are employed to compare the solutions. Due to the exponential growth of the branches in the process of CPLEX, it fails to obtain the optimal solution even with a running time of 10,000 seconds for networks with 30, 50 and 100 nodes. With 10 nodes problem, the proposed genetic algorithms generated optimal solutions in all cases with 10 nodes. With 30 and 50 nodes respectively the average gap from the upper bound by CPLEX is 3% in 30 nodes problem, and no difference in 50 nodes problem. For the network with 100 nodes CPLEX experiences memory problem and fails to generate a solution. Proposed genetic algorithm shows better performance from 0 to 4% than tabu search.

	10	30	50	100
(GA-TS)/TS	0.00	-0.02	-0.04	-0.04
(GA-CPLEX)/CPLEX	0.00	0.03	0.00	N/A

5. Conclusion

An end-to-end delay problem in overlay networks is considered for next generation multicast service in Internet. To minimize the maximum delay from a source to multicast group members, overlay multicast tree is constructed on the overlay network. The problem is formulated as a degree bounded minimum spanning tree which is a well-known NP-hard problem.

A genetic algorithm is proposed to solve the problem. For each step of genetic algorithm, computational results are used to compare various operations. Genetic operations are chosen to increase the performance and decrease the complexity of operation. An iterative operation for generations create better chromosome.

Computational experiments of the proposed genetic algorithm are performed for overlay networks with 10, 30, 50 and 100 multicast nodes. Outstanding performance is illustrated by the proposed heuristic. The average gap from the solution by CPLEX is within 3% in problems with 30 and 50 nodes. Solutions for problems with 100 nodes are also obtained in a reasonable time while those of CPLEX are not available due to memory problem.

References

[1]P. Francis, Yoid: Your Own Internet Distribution, UC Berkeley ACIRI Tech Report, April, 2000.

[2]Y. Chawathe, Scattercast: An Architecture for Internet Broadcast Distribution as an Infrastructure Service, Ph. D. thesis, Department of EECS, UC Berkeley, December, 2000.

[3]Y. Chu, S. Rao, and H. Zhang,. A Case for End System Multicast, ACM Sigmetrics, 2000.

[4]J. Jannotti, D.K. Gifford, K. Jhonson, and M. Kaashock, Overcast: Reliable Multicasting with an Overlay Network, In Proceedings of the 5th Symposium on Opening System Design and Implementation, December, 2000.

[5]D. Helder and S. Jamin, Banana Tree Protocol, an End-host Multicast Protocol, Technical Report, July, 2000.

[6]S. Shi and J. Turner, Routing in Overlay Multicast Networks, IEEE INFOCOM, 2002.

[7]Hee K. Cho and Chae Y. Lee, Multicast Tree Rearrangement to Recover Node Failure in Overlay Multicast Network, Computers and Operations Research, Vol. 33, No.3, pp. 581-594, 2006.

[8]Z. Li, and P. Mohapatra, QRON: QoS-Aware Routing in Overlay Networks, IEEE Journal on Selected Areas in Communications, Vol. 22, No. 1, January, 2004.

[9]S. Shi, J. Turner and M. Waldvogel, Dimensioning Server Access Bandwidth and Multicast and Multicast Routing in Overlay Networks, In Proceedings of NOSSDAV, June, 2001.

[10]E. Nardelli and G. Proietti, Finding All the Best Swaps of a Minimum Diameter Spanning Tree under Transient Edge Failures, Journal of Graph Algorithms and Applications, Vol. 5, No.5, pp. 39-57, 2001.

[11]Goldberg, Genetic Algorithms, Addison Wesley, 1989.

[12]F. Busetti, Genetic algorithms overview, citeseer.ist.psu.edu/busetti01genetic.html, 2002.

[13]Hsinghua Chou, G. Premkumar, and Chao-Hsien Chu, Genetic Algorithms for Communications Network Design—An Empirical Study of the Factors that Influence Performance, IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 5, NO. 3, 2001.

[14]F.N. Abuali, R.L.Wainwright and D.A. Schoenefeld, Determinant Factorization: A New Encoding Scheme for Spanning Trees Applied to the Probabilistic Minimum Spanning Tree Problem, in Proceeding of the Sixth International Conference on Genetic Algorithms, Larry J. Eshelman, Ed. pp. 470–477, 1995.

[15]D. E. Goldberg and K. Deb, A comparative analysis of selection schemes used in genetic algorithms, Foundations of Genetic Algorithms, pages 69-93, Morgan Kaufmann, 1991

[16]CPLEX 9.1, CPLEX Optimization Inc., 2005.

[21]Chae Y. Lee, Hyo Jung Park and Jin Woo Baek, A Multicast Tree to Minimize Mazimum Delay in Dynamic Overlay Networks, M.S. thesis, Industrial Engineering Department, KAIST, 2007.