Genetic Algorithms for Multicast Routing with Multiple QoS Requirements

Qijun Gu[1] and Chao-Hsien Chu[2]

School of Information Sciences and Technology, Pennsylvania State University, University Park, PA, USA
E-mail: [1]qgu@ist.psu.edu, [2]chu@ist.psu.edu

Abstract

In this paper we explore the use of genetic algorithms (GAs) for solving the multicast routing (MCR) problem with multiple quality of services (QoS) constrains. We study the impacts of encoding, initialization, repair, crossover, and selection on the performance of the GAs. An experimental design, with 32 cells and 18 data sets with 10 repetitions in each cell, is used for the empirical tests. The results indicate that encoding, selection, and repair/penalty have significant impacts on solution quality. Among the various options, the combination of determinant encoding, heuristic initialization, uniform crossover, repair, and fittest selection often provides better solution qualities than other combinations. Using this combination, the proposed GA not only obtains solutions significantly better than a shortest path based heuristic, its quality is also robust and almost the same as optimal solutions.

Keywords Genetic Algorithms, Multicast, Quality of Services (QoS)

1. Introduction

Many applications often require a source to send out a large amount of information to multiple destinations through a network, while satisfying multiple QoS requirements, such as little delay, small delay variation, low loss rate, and low cost, etc. Using a classic unicast routing, i.e. establishing a route to each user, these applications consumes tremendous bandwidth resources of the network when the number of users is large. It is also not proper to broadcast the information in the network, because only a group of selected users in the network need the information. Therefore, it is necessary that we bunch a number of users requesting the same information and use one data packet to serve them. Through such a MCR method, the source node can efficiently send information to all destination nodes. A comprehensive study of a number of unconstrained MCR algorithms for real-time communication can be found in [1].

The problem of determining an optimal multicast route that satisfies multiple QoS requirements is known as NP-complete [2]. Therefore, good heuristic algorithms are of practical interest. Many QoS-based MCR algorithms were published recently, however, most of them considered only one or two constraints and fell into one of the following problem types: delay constrained [3], bandwidth and delay constrained [4], and delay-variation constrained [5]. Finding an optimal multicast route, which satisfies multiple QoS constraints, is intractable when the network size becomes bigger. An extensive review of QoS constrained MCR problems and protocols can be referred to [6]. However, in our study, we mapped multiple QoS constraints into the fitness function of a chromosome so that these QoS constraints can be manipulated with the same formulation of routing cost.

GA has also attracted an increasing interest among scholars and engineers to solve MCR problems. However, it is worth noting that most of them only explored one or two aspects of GA and most researches only studied the unconstrained MCR problem [7] or the delay constrained MCR problem [8]. Clearly, a comprehensive study on the use of GA to solve the MCR problems with multiple QoS requirements is needed. The major objectives of this study are thus two-folds: (1) develop a GA approach to solve the MCR problem, which subjects to multiple QoS constraints; (2) determine the impacts of various factors, specifically encoding, population initialization, crossover, repair, and selection, on GA's performance.

The paper is organized as follows. In Section 2, we present the MCR problem and the detail of GA approaches to solve this problem. Section 3 covers the experimental comparison of GA operators and the selection of best GA operators for our proposed GA algorithm. Finally, in Section 4, we conclude our study.

2. QoS-Based MCR Problem

2.1. Constrained MCR Problem

A distributed network can be modeled as a connected graph, $G=(V,E)$, where V is the set of nodes and E is the set of edges. Let S be the set of source nodes, D be the set of destination nodes, and $e_{i,j}$ be the edge from node i to j in the network. $e_{i,j}$=1, if it is selected in the route; otherwise $e_{i,j}$=0. In QoS constrained MCR problems, edges and nodes of a multicast tree are assigned cost, bandwidth, delay and delay jitter and packet loss. Hence, the constrained MCR refers to the process of searching a routing tree, $G'=(V',E')$, which is a subgraph of G, so that the cost of the routing tree is minimized while other QoS constrains are satisfied. For instance, the bandwidth constraint of a connection requires that the links composing the route must have certain amount of free bandwidth available. The delay constraint of a multicast connection requires that the longest end-to-end delay from the sender to any receiver in the tree must not exceed an upper bound.

2.2. GA For QoS-Based MCR

The use of GA to any domain follows the same procedure – problem representation, population initialization, fitness evaluation, selection, reproduction (crossover and mutation), and replacement [9]. The details of the GA used for solving the QoS constrained MCR problem are described below.

Problem Representation: In this study, we examine the impacts of two encoding methods, edge and determinant, which are appropriate for the QoS-based MCR problem. These two encoding methods are flexible when dealing with dynamic network, because adding or removing one node or one edge only results in adding or removing one gene in the current chromosome. We still can use the solution represented by the modified chromosome to evolve to a new solution.

Initial Population: In this study, we compare the performance of two initialization methods: heuristic and random. In heuristic initialization, the shortest path from the source to every destination node is computed first by Dijkstra algorithm, and then all shortest paths are assembled to form a multicast tree. This tree may not be the best solution, however it offers good genes, which can be utilized in future generations.

Fitness Function: The fitness function is used by GA to interpret the chromosome in terms of the phenotype (physical representation) and evaluate its performance based on certain characteristics embedded in the solution. In this study, we treat the fitness function F as a summation of the objective, connectivity, and QoS requirements $F=w_CF_C+\ w_NF_{N+}w_DF_D+\ w_JF_{J+}\ w_LF_L$, where F_C is the primal fitness function as the main objective of multicast is to minimize cost, $F_{\backslash N}$ is the fitness for connectivity designed to punish a tree if it is illegal, F_D is for the delay requirements, F_J is for the jitter requirements, F_L is for the loss rate requirements, and w_* are their weights.

1) Change in-degree of every node to 1, except the source node to 0.
   a) Give a graph $G_1=(V_1,E_1)$.
   b) Check all nodes in the network.
   c) If the in-degree of node v in V_1 is more than 1, delete all except one edge flowing into this node.
   d) If the in-degree of node v in V_1 is 0, select an edge flowing into this node.
   e) Delete all edges flowing into source node 0.
   f) The new graph $G_2=(V_2,E_2)$ is formed.
2) Remove cycles in the graph.
   a) Choose a node v in V_2.
   b) Follow the flow-in edge to the upper node and record the path.
   c) If the upper node is in the recorded path, the path has a cycle.
   d) Break the cycle at $e_{i,j}$, where node j has another possible flow-in edge $e_{k,j}$ that is from an upper node k not in the path.
   e) Delete $e_{i,j}$ and select $e_{k,j}$ as the flow-in edge of node j.
   f) Repeat a) to e) until no cycle exists.
   g) The new graph $G_3=(V_3,E_3)$ is formed.
3) Remove all redundant nodes and edges in the tree.
   a) Choose a leaf node v in V_3.
   b) If it is not a destination node, delete it and its flow-in edge.
   c) Repeat a) to b) until all leaf nodes are destination nodes

Fig. 1 Repair Algorithm

Crossover: In this study we select one-point and uniform crossover for comparison. In one-point crossover, a random position is generated for a pair of chromosomes to exchange their alleles. In uniform crossover, a set of positions, called a mask, is chosen for each of the chromosomes and their alleles are exchanged with each other based on the generated positions.

Mutation: Mutation methods include inversion, insertion, displacement, reciprocal exchange, and heuristic mutation on alleles [10]. Inversion mutation, which simply flips one of the bits in a random fashion with a certain probability, has been the most popular mutation method for MCR problem. If the mutation procedure is directly performed on a tree, a subset of nodes is randomly selected and all the links that flow into these nodes are removed. We study these two in this study.

Selection: Two selection methods – fittest and probabilistic – are considered in this study. The former select the fittest individuals for the next generation. The later method is to select chromosomes based on survival probabilities, which is obtained by linear fitness ranking method.

Repair vs. Penalty: The encoding and reproduction process (crossover and mutation) may produce or turn a tree into an illegal one. There are two methods to remedy the defects of a tree – repair and penalize. One viable approach is to use the fitness function of defect as a penalty. However, this method is passive, because the fitness function won't make the tree complete. Hence, we propose a repair operator as in Fig. 1 to accelerate the convergence of the GA. the following procedure to repair a tree:

3. Experiment and Comparison

To evaluate the effectiveness of our proposed GAs, a factorial experimental design is conducted. The factors considered for the study are encoding, population initiation, crossover, methods of handling defective trees, and selection strategy. We use the total cost of the MCR tree to rate the solution quality of GAs. The following null hypotheses are formulated for empirical testing:

- H1: The mean value of solution quality is the same for the two encoding methods.
- H2: The mean value of solution quality is the same for the two population initiation methods.
- H3: The mean value of solution quality is the same for the two crossover methods.
- H4: The mean value of solution quality is the same for repair and penalty methods.
- H5: The mean value of solution quality is the same for the two selection strategies.

3.1. Experiment Design

As discussed, our experiment takes into considerations of five factors. Each factor contains two levels as shown in the footnote of Tab. 2. For example, we compare two kinds (termed as levels) of encoding, determinant encoding, marked by E0, and edge encoding, marked by E1. Hence, an experiment design with 32 cells (2 x 2 x 2 x 2 x 2) is used to represent the combinations of all the factors. For each cell, we run ten repetitions with different seeds for each data set.

The data sets are test set B in the public database - Steinlib [11] as in shown in Tab. 1, where |V| is the number of nodes, |E| is the number of edges, |T| is the number of destination nodes, and Opt is the minimum MCR cost. The test set consists of 18 data sets with the number of nodes ranged from 50 to 100 and up to 200 edges. Since these data sets only contain information of network and cost for each edge, we generate the other QoS information, including delay, bandwidth, jitter and loss rate, for every edge. The constraints are set so that the minimum multicasting trees in the original unconstrained problems still satisfy the QoS constraints (for benchmark purpose).

Tab. 1 Test Set

Data sets	|V|	|E|	|T|	Opt	Heu-ristic	Proposed GA*	
						Best	Mean
1	50	63	9	82	82	82	82
2	50	63	13	83	97	83	83
3	50	63	25	138	148	138	138
4	50	100	9	59	77	59	59.5
5	50	100	13	61	65	61	61
6	50	100	25	122	176	124	124.2
7	75	94	13	111	123	111	111
8	75	94	19	104	116	104	104
9	75	94	38	220	234	220	220
10	75	150	13	86	134	86	86
11	75	150	19	88	127	88	88.1
12	75	150	38	174	223	174	175.5
13	100	125	17	165	192	165	165
14	100	125	25	235	277	235	235.1
15	100	125	50	318	353	318	318
16	100	200	17	127	166	127	127
17	100	200	25	131	152	131	131.1
18	100	200	50	218	293	223	226.9

* Using determinant encoding, heuristic initialization, uniform crossover, repair method, and fittest selection strategy

3.2. Comparison of GA Operators

The results of the experiment are in Tab. 2, where the first and fourth columns show the various combinations of GA operators. The values in the cell report the average and standard deviation of relative value (individual value/optimal value) from the 180 data points (18 data sets x 10 repetitions) in each cell. The highlighted cells - determinant encoding, heuristic initialization, uniform (or one-point) crossover, repair method and fittest selection strategy represent the best solutions with little variation in quality.

Tab. 2 Comparison of GA Operators

Factors	Mean	Factors	Mean
1. E0I0C0L0S0	1.008	17. E0I0C0L0S1	1.080
2. E1I0C0L0S0	1.080	18. E1I0C0L0S1	1.273
3. E0I1C0L0S0	**1.006**	19. E0I1C0L0S1	1.084
4. E1I1C0L0S0	1.076	20. E1I1C0L0S1	1.278
5. E0I0C1L0S0	1.006	21. E0I0C1L0S1	1.089
6. E1I0C1L0S0	1.086	22. E1I0C1L0S1	1.270
7. E0I1C1L0S0	**1.004**	23. E0I1C1L0S1	1.084
8. E1I1C1L0S0	1.081	24. E1I1C1L0S1	1.267
9. E0I0C0L1S0	1.337	25. E0I0C0L1S1	17.98
10. E1I0C0L1S0	1.412	26. E1I0C0L1S1	1821
11. E0I1C0L1S0	1.167	27. E0I1C0L1S1	16.27
12. E1I1C0L1S0	1.209	28. E1I1C0L1S1	1822
13. E0I0C1L1S0	1.359	29. E0I0C1L1S1	15.22
14. E1I0C1L1S0	1.407	30. E1I0C1L1S1	1837
15. E0I1C1L1S0	1.184	31. E0I1C1L1S1	15.23
16. E1I1C1L1S0	1.209	32. E1I1C1L1S1	1831

E0: Determinant encoding; E1: Edge encoding;
I0: Random initialization; I1: Heuristic initialization;
C0: One-point crossover; C1: Uniform crossover;
L0: Repair; L1: Penalty;

S0: Fittest selection; S1: Probabilistic selection.

3.3. Selection of GA Operators

We further conduct ANOVA data analysis on each factors as in Tab. 3. The results indicate that: 1) determinant encoding is better than edge encoding; 2) fittest selection is better than probabilistic selection; and 3) repair method is better than penalize method. We also notice that although 1) heuristic initialization is generally better than random initialization and 2) uniform crossover is generally better than one-point crossover; their performance differences are not statistically significant. Hence, GA using determinant encoding, heuristic initialization, uniform crossover, repair method, and fittest selection strategy performs significantly better than the shortest path based heuristic. Its solution quality is robust and comparable with the optimal solutions as shown in Tab. 1.

Tab. 3 ANOVA Analysis

	Sum of Square	DF	Mean Square	F value	P value
Main effects					
Encoding (E)	12.408	1	12.408	717.74	0.0000
Initialization (I)	0.0017	1	0.0017	0.1000	0.7570
Crossover (C)	0.0000	1	0.0000	0.0000	0.9690
Selection (S)	13.012	1	13.012	752.70	0.0000
Two-Way Interactions					
E*I	0.0001	1	0.0001	0.0000	0.9540
E*C	0.0012	1	0.0012	0.0700	0.7950
E*S	2.2808	1	2.2808	131.93	0.0000
I*C	0.0037	1	0.0037	0.2100	0.6440
I*S	0.0028	1	0.0028	0.1600	0.6880
C*S	0.0018	1	0.0018	0.1000	0.7490
Error	2869	49.598	49.598	0.0173	
Total	2879	77.310			

4. CONCLUSIONS

Developing an efficient algorithm to find the optimal MCR tree subjected to multiple QoS constraints is a complex and challenging problem to be explored. In this paper, we applied GA to solve this problem. We identified five factors -- encoding, initialization, crossover, repair/penalty, and selection -- that are critical to the GA design and examined their influence on performance. A factorial experimental design with 32 cells to represent different options (2 levels for each factor) was used to study the factors and their interactions. We simulated the experiment using 18 public available data sets and run each data set for 10 random repetitions (with different seeds). The results highlight the importance of choosing the right operators to get the best performance: (1) encoding, selection, and repair/penalty methods have significant impact on the GA performance, while the impacts of initialization and crossover operators are less significant; (2) the combination of determinant encoding, heuristic initialization, uniform crossover, repair and fittest selection provides better results than other combinations most of the time; (3) determinant encoding performs better than edge encoding; (4) fittest selection performs better than probabilistic selection; and (5) repair method performs better than penalty method. We also notice that when both penalty and probability selection are applied, the GA cannot properly converge within an acceptable number of generations.

5. References

[1] H. F. Salama, D. S. Reeves, and Y. Viniotis, "Evaluation of multicast routing algorithms for real-time communication on high-speed networks," IEEE JSAC, vol. 15, pp. 332-345, 1997.

[2] V. P. Kompella, J. C. Pasquale, and G. C. Polyzos, "Multicast routing for multimedia communication," IEEE/ACM TON, vol. 1, pp. 286-292, 1993.

[3] M. Parsa, Q. Zhu, and J. J. Garcia-Luna-Aceves, "An iterative algorithm for delay-constrained minimum-cost multicasting," IEEE/ACM TON, vol. 6, pp. 461-474, 1998.

[4] J. Dai, H. K. Pung, and T. Angchuan, "QROUTE: an integrated framework for QoS-guaranteed multicast," Proc. of 27th Annual IEEE LCN, 2002.

[5] G. N. Rouskas and I. Baldine, "Multicast routing with end-to-end delay and delay variation constraints," IEEE JSAC, vol. 15, pp. 346-356, 1997.

[6] B. Wang and J. C. Hou, "Multicast routing and its QoS extension: problems, algorithms, and protocols," IEEE Network, vol. 14, pp. 22-36, 2000.

[7] H. Esbensen, "Computing near-optimal solutions to the Steiner problem in a graph using a genetic algorithm," Networks, vol. 26, pp. 173-185, 1995.

[8] Q. Zhang and Y.-W. Leung, "An orthogonal genetic algorithm for multimedia multicast routing," IEEE Transactions on Evolutionary Computation, vol. 3, pp. 53-62, 1999.

[9] D. E. Goldberg, Genetic algorithms in search, optimization & machine learning. New York: Addison-Wesley, 1989.

[10] G. Premkuma, C. H. Chu, and H. Chou, "Telecommunications network design comparison of alternative approaches," Decision Sciences, vol. 31, pp. 483-506, 2000.

[11] SteinLib Testdata Library, http://elib.zib.de/steinlib/steinlib.php