Joint Conference of 1996 International Computer Symposium
December 19~21, Kaohsiung, Taiwan, R.O.C.

Multicast Routing Based on Genetic Algorithms

Wei-Yuan Do and Ren-Hung Hwang
Department of Computer Science and Information Engineering
National Chung Cheng University, Chia-Yi, Taiwan, R.O.C.

Abstract

Due to the advent of many new multimedia applications in high speed networks, the issue of multicast routing has become more and more important. The multicast routing problem in computer networks is also known as the Stenier tree problem which has been shown to be NP-complete. In this paper, we propose a new multicast routing algorithm based on Genetic Algorithms. Computer simulations are conducted to evaluate the performance of the proposed algorithm. Our numerical results show that, under non-fully connected networks, our algorithm is able to find a better solution than the average distance heuristic (ADH) algorithm, a very promising heuristic algorithm for the Steiner tree problem.

1. Introduction

Due to the rapid advance in the switching and communication technologies and the increased demand for various kinds of communication services, many new applications in networks require support of multicast communication. Therefore, the issue of multicast routing has become more and more important.

In the literature, there are two optimization goals for the design of multicast routing algorithm. The first is to minimize the average path delay which is measured as time required to transmit a message from the source node to any destination node. The second is to minimize the cost of the multicast tree which is the sum of costs of the links in the multicast tree. The cost of a link can be a function of the capacity and/or the traffic of the link. Although these two goals often conflict with each other, in this paper, we focus on the design of multicast algorithms which try to achieve the second goal while also take the first goal into consideration.

Clearly, the least cost tree, which is called a steiner tree [1], is the optimal solution a multicast algorithm looks for. Unfortunately, the problem of finding a steiner tree is known to be NP-complete [2], even if links have unit cost [3]. Thus most previous researchers have focused on developing heuristic algorithms that take polynomial time and produce near optimal results [4, 5, 6]. Furthermore, these heuristic algorithms often guarantee that their solutions are within twice the cost of the optimal solution.

In this paper, we present a novel heuristic multicast routing algorithm based on Genetic Algorithms. The Genetic Algorithms [7-12] provide robust and efficient search in complex spaces. Survival of the fittest "genes" and structured yet randomized genetic operations are the basic philosophies behind the algorithms. The main advantages of Genetic Algorithms include (1) Solutions are coded as bit strings, referred to as chromosomes. Large problems can be easily handled by using long strings; (2) Genetic operations, such as crossover and mutation, are very easy to implement; (3) With a pool of chromosomes (candidate solutions), Genetic Algorithms search the solution space at different conners in parallel. The algorithm can be easily implemented on multiprocessors machines to do the search in parallel; (4) Randomized genetic operations, such as mutation, can avoid the search being trapped by local-optima. Genetic Algorithms have been successfully applied to control problems in ATM networks, such as bandwidth allocation [13] and buffer management [14]. Recently, it has been applied to point-to-point routing [15] and spanning tree problem [16] in communication networks.

The proposed GA-based multicast routing algorithm is evaluated through simulations under different network topologies. Our simulation results show that the proposed algorithm is able to find a multicast tree with less cost than that of a the average distance heuristic (ADH) algorithm [17,18], a very promising heuristic algorithm for the Steiner problem.

The remainder of this paper is organized as follows. Section 2 briefly describes the basics of the Genetic Algorithms. In section 3, the GA-based multicast routing algorithm is presented. In section 4, the performance of our multicast routing scheme is evaluated through simulation. The performance of our algorithm is also compared to that of the average distance heuristic (ADH) algorithm under different network configurations. Finally, conclusions and future work are given in section 5.

2. Genetic algorithms

The Genetic Algorithms(GAs) are used for solving an optimization problem based on the principle of evolution. A population of candidate solutions, called chromosomes, are maintained at each iteration of the evolution. Each chromosome consists of linearly arranged genes which are represented by binary strings.

Three basic operations, namely, reproduction, crossover, and mutation, are adopted in the evolution to generate new offspring. Reproduction is based on the Darwinian survival of the fittest among strings generated. The samples (represented as bit strings) with larger fitness function values are selected to generate new offspring bit strings by crossover operations and convert the offspring to new parameter solutions. Intuitively, a bit string with a larger fitness function value should have a higher probability of contributing one or more offspring bit strings in the next generation and vice versa. Crossover is used to cut individually two parent bit strings into two or more segments and then combine the segments undergoing crossing over to generate two offspring bit strings. Crossover can produce offspring that are radically different from their parents. Suppose the crossover operation is performed on the two bit strings, "01110001" and "10011011", and they are split at the second bit, then two new bit strings, "01011011" and "10110001" are generated, (see Figure 1). There are other ways for implementing the crossover operation, e.g. arithmetic crossover [12].

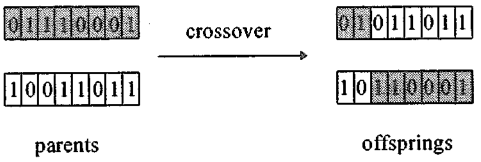


Figure 1: The crossover operation.

Mutation is to perform random alternation on bit strings by some operations, such as bit shifting, inversion, rotation,...,etc. The mutation operation will create new offspring bit strings different from those generated by the reproduction and crossover operations. Mutation can extend the scope of the solution space and reduce the possibility of falling into local extremes. In general, the possibility of mutation is very low.

The genetic algorithms are typically implemented as follows:

Step 1: Initialize a population of chromosomes (solution).
Step 2: Evaluate each chromosome in the population.
Step 3: Create new chromosomes by mating current chromosomes and apply mutation and recombination as the parent chromosomes mate.
Step 4: Delete members of the population to make room for the new chromosomes.
Step 5: Evaluate the new chromosomes and insert them into the population.
Step 6: If stopping criterion is satisfied, then Stop and output the best chromosome (solution), otherwise, go to Step 3.

3. The proposed multicast routing algorithm

3.1 Network Model and Problem Definition

For a undirected network graph, the problem of finding an optimal multicast routing (OMR) with least cost, called an optimal multicast tree (OMT), is formally defined as follows:

- GIVEN: An undirected network $G=(V,E,c)$ consists of a nonempty set V of $|V|$ vertices and a set $E, E \subseteq V \times V$, of $|E|$ edges connecting pairs of vertices. In addition, it has a cost function $c: E \rightarrow R$ and a given non-empty set $N=\{ v_0, u_1, u_2, \ldots, u_k \}$, of terminals in G, $N \subseteq V$, where v_0 is the source node and $D=\{ u_1, u_2, \ldots, u_k \}$ is the set of destination nodes.
- FIND: A subnetwork $T_G(N)$ of G such that:
  — there is a path between every pair of terminals,
  — total cost $|T_G(N)| = \sum_{e_i \in T_G(N)} c(e_i)$ is minimized.

The subnetwork $T_G(N)$ is called a Steiner minimal network for N in G. If all edges in G have positive cost, a Steiner minimal network $T_G(N)$ must be a tree, and $T_G(N)$ is called a Steiner minimal tree for N in G. In particular, $T_G(V)$ denotes a minimum spanning tree for G.

3.2 Routing table

In the network graph, $G=(V, E)$, there are $|V| \times (|V|-1)/2$ possible source- destination pairs. A source-destination node can be connected by a set of links, which is called a "route". There are usually many possible routes between any source - destination pair. For example, consider the network of Figure 2, the possible routes between v_0 to v_4 include $v_0 - v_4$, $v_0 - v_5 - v_4$, ...and so on.

Our GA-based multicast routing algorithm assumes that a routing table, consist of R possible routes, has been constructed for each source-destination pair. For example, Figure 3.2 shows the routing table for the source-destination pair (v_0, v_4). The size of the routing table, R, is a parameter of our scheme. The possible routes in the routing table is sorted according to their length (i.e. number of links) such that shorter paths are assigned with smaller route number. The reason for preferring routes with shorter length is to take the path delay into consideration on finding the multicast trees.

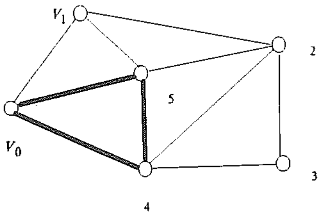


Figure 2 : A simple 6-node network.

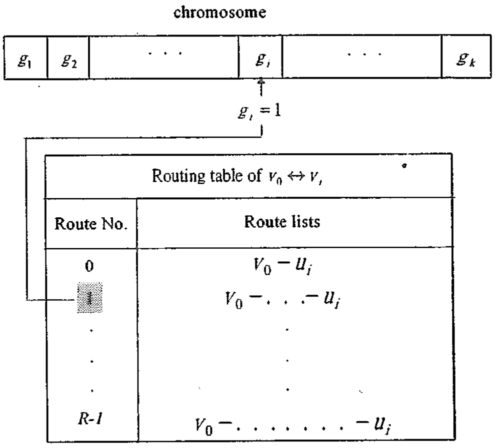
Routing table of $v_0 \leftrightarrow v_4$	
Route No.	Route lists
0	$v_0 - v_4$
1	$v_0 - v_5 - v_4$
.	.
.	.
.	.
R-1	$v_0 - v_5 - v_1 - v_2 - v_3 - v_4$

Figure 3: Example of a routing table

3.3 Representation of chromosomes

For a given source node v_0 and destination set $D = \{ u_1, u_2, \ldots, u_k \}$, a chromosome can be presented by a string of integers with length k. An gene, g_i, $1 \le i \le k$, of the chromosome is an integer in $\{0, 1, \ldots, R-1\}$ which represents a possible route between v_0 and u_i, where $u_i \in D$. The relationship of chromosome, gene, and routing table is explained in Figure 4.

This coding method was proposed in [15] for point-to-point routing problem. Obviously, a chromosome represents a candidate solution for the multicast routing problem since it guarantees a path between the source node to any of the destination nodes. However, a chromosome does not necessary represent a tree. This is not a serious problem since a graph (chromosome) with cycles will have bad fitness value and, thus, will be deleted from the population. It is possible to choose a coding method in which chromosomes only represent trees, e.g. [19]. The major advantage of using the coding method of [15] is that given a chromosome, the links of the multicast tree (graph) can be easily identified. On the other hand, the methods proposed in [19] all require complex transformation in order to obtain the links of the tree of a chromosome. Another advantage of using the coding method of [15] is that the path delay can be taken into consideration through the proper selection of routes in routing tables.



Routing table of $v_0 \leftrightarrow v_i$	
Route No.	Route lists
0	$v_0 - u_i$
1	$v_0 -. . .- u_i$
.	.
.	.
.	.
R-1	$v_0 -. . . . . - u_i$

Figure 4: A chromosome. In this example, the ith gene g_i=1 in the chromosome means the second route of the routing table of (v_0, v_i) is chosen.

When the network is large, the number of possible routes between two nodes may become very huge. Therefore, we must limit the number of candidate routes to a reasonable amount, say R. That is, only the R shortest routes will be listed in the routing table.

3.4 Description of the algorithm

The GAs maintain a population of chromosomes, each of which has a fitness value. The fitness value defines the quality of the chromosome. Beginning with a set of random chromosomes, a process of evolution is simulated. The main components of this process are crossover, which mimics propagation, and mutation, which mimics the random changes occurring in nature. After a number of generations, highly fit chromosomes will emerge corresponding to good solutions.

The outline of our GA-based multicast routing algorithm, schematically illustrated in Figure 5, is given as follows.

3.4.1 Initialization of chromosomes

In a chromosome, each gene corresponds to a specific route of a routing table. Thus a chromosome is a series of integers which initial values are taken at random. The initial procedure generates P different chromosomes at random which form the first generation. This set of chromosomes is called the gene pool, and P is the population size of the gene pool.

3.4.2 Evaluation of chromosomes

The fitness value of a chromosome is a value of the objective(fitness) function for the solution (e.g. a multicast tree) represented by the chromosome. Given a population $H = \{ h_0, h_1, \cdots, h_{P-1} \}$, the fitness value of each chromosome is computed as follows. Let $C(h_i)$ be the sum of the costs of the links of the graph represented by the chromosome h_i. The fitness value of the chromosome h_i, $F(h_i)$, is given by

$$F(h_i) = 1 - \frac{C(h_i)}{C(L)}$$

where $C(L)$ is the sum of costs of all edges.

After the fitness values of all chromosomes are computed, they are then sorted according to their fitness values such that $C(h_0) \le C(h_1) \le \cdots \le C(h_{P-1})$.

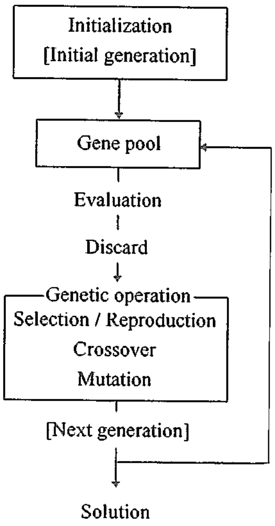


Figure 5: The proposed Genetic Algorithm

3.4.3 Discard the duplicate chromosomes

In the gene pool, there may exist two or more duplicate chromosomes. Apply some of the genetic operations, e.g., crossover, on two duplicate chromosomes will yield the same chromosome. Therefore, too many duplicated chromosomes in the gene pool will reduce the ability of searching. Once this situation occurs, the duplicated chromosomes must be discarded. In our algorithm, they are replaced by new randomly generated chromosomes, as shown in Figure 6.

3.4.4 Reproduction

According to the computed fitness values, some of chromosomes are selected to generate more offspring through crossover and mutation operations, and others will be removed from the gene pool. By this way, chromosomes with large fitness values will survive and reproduce more. On the other hand, chromosomes with small fitness values die off.

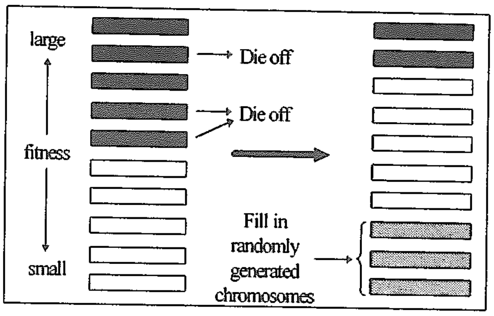


Figure 6: Discard duplication chromosomes.

The reproduction process selects certain number of chromosomes with the best fitness values from the current generation for reproduction. Another number of chromosomes, again with the best fitness values, are selected to reproduce offspring through crossover operation. Note that the number of the chromosomes in gene pool is always restricted to P, as illustrated in Figure 7.

3.4.5 Crossover

Crossover operation is used to exchange genetic information between two chromosomes. In this process, two chromosomes strings with larger fitness values are picked from the gene pool first. The start point and length of the portion to exchange are randomly selected. Two new offsprings are created and put back to the gene pool, as illustrated in Figure 8.

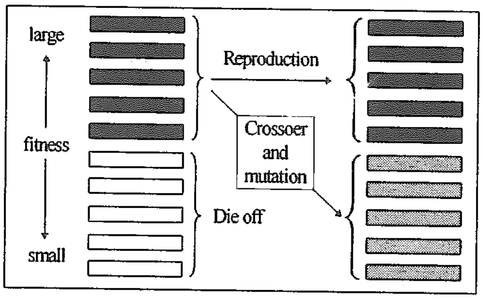


Figure 7: Select chromosomes for genetic operation.

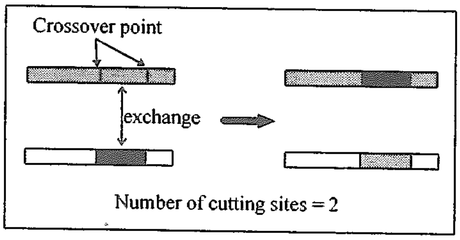


Figure 8: Crossover operator

3.4.6 Mutation

The mutation operation is one kind of random change in the chromosome. In our algorithm, pointwise mutation is adopted in which one bit in the chromosome string is changed with a probability, called mutation probability. The mutation operation gives the Genetic Algorithm an opportunity to search for new more feasible chromosomes in new corners of the solution spaces.

4. Numerical results

In this section, performance of the proposed multicast routing scheme is evaluated via simulations. The performance is then compared to that of the optimal solution (based on [20]), as well as the average distance heuristic steiner tree method. The performance of multicast routing algorithms heavily depends on the underlying network topology and the volume of traffic carried in the network. Therefore, three different network configurations are used in simulation.

4.1 Network model and traffic distribution

There are three network models employed in our simulations. The first model is a 10-node fully connected network consisting of 10 nodes and 45 bi-directional physical links. Figure 9 shows the physical layout of the 10-node network. The second and third models, taken from [21], are 25-node and 21-node sparsely connected networks, respectively. The 25-node grid network consists of 25 nodes and 40 bi-directional links, and the 21-node network, which is the topology of the ARPA, consists of 21 nodes and 26 bi-directional links. The physical layouts of these two networks are shown in Figure 10 and Figure 11.

We assume that all networks are well-dimensioned and the cost of a link is normalized to the unity. The cost may represent the cost of carrying a connection on the link. For example, in [22], the routing problem is formulated as an Markov decision process (MDP) and the cost of a link, which is state dependent, represents the expected revenue loss due to the carry of the new call. Since we assume that the network is well-dimensioned, the costs of the links are uniformly distributed between 0.3 to 1.

4.2 Computer simulation results

4.2.1 Off-line setup

In our proposed algorithm, the GA approach uses a gene to represent a route from the source node to one of the destinations. Hence, the routing table for each source-destination pair needs to be constructed first. Obviously, the number of possible routes between two nodes heavily depends on the network topology. If the network is densely connected or the size of the network is large, the number of possible routes of a source-destination pair becomes huge. Hence, it is impossible to list all the possible routings in the routing table. In the simulation, we have set the size of routing table to 64. An algorithm has been designed to automatically generate the routing tables. Recall that we prefer routes with short length.

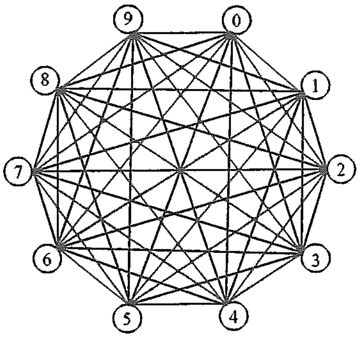


Figure 9: 10-node fully connected network

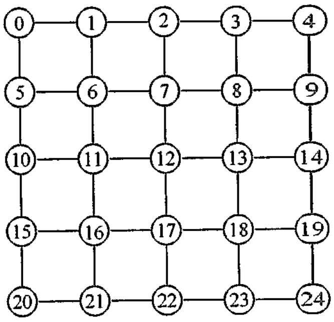


Figure 10: 25-node grid network model

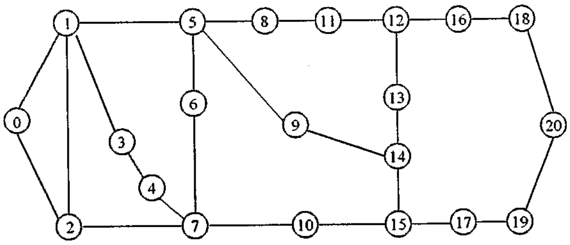


Figure 11 : 21-node ARPA network model

4.2.2 Parameter selection

Parameters of the genetic algorithm, such as the size of gene pool, crossover rate, mutation rate, and the number of generations, must be properly selected to yield the best performance. In the following, simulations are conducted to obtain proper values of these parameters.

Let us examine the effect of the number of generations on the performance of the algorithm. Figure

12 shows the minimum cost of each generation for a multicast connection with node 0 as the source node and all the other nodes as the destination nodes under the 10-node fully connected network model. The mutation rate is set to 0.4. The three curves in Figure 12 show the minimum cost of each generation when the size of the gene pool (population) is set to 16, 32, and 64, respectively. As expected, the cost does not decrease much further after the 80th generation. Therefore, in our later simulations, we set the number of generations to 100.

Figure 13 shows the cost observed after 200 generations under different population sizes. The network configuration and other parameters are the same as above. As expected, larger population size leads to better performance because more diverse chromosomes available in a large population can protect the genetic algorithm from falling into the local optimum. From Figure 13, we can observe that after 200 generations, the population size of 64 yields the best performance. As the population size is larger than 64, the cost is even slightly increased. This may due to the slow convergent rate of larger population sizes. That is, the process has not converged to the optimal solution yet.

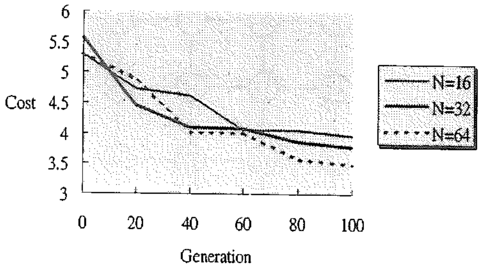


Figure 12 : The effects of the number of generations. (Mutation rate=0.4).

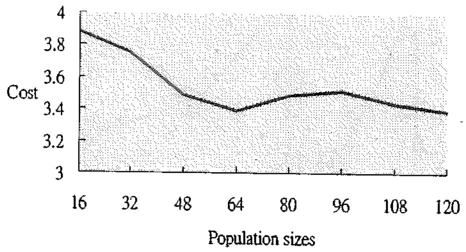


Figure 13: The effects of the population size.

Figures 14 and 15 show the effect of mutation rate on the performance of the genetic algorithm. The four curves in Figure 14 show the cost observed after 200 generations under various population sizes for different mutation rates, namely 0.001, 0.01, 0.1, 0.25. Figure 15 shows the cost under various mutation rates when population size is set to 32. Although previous researches in genetic algorithms suggest that a small mutation rate usually yields a better performance, we have observed a contrary result. Figures 14 and 15 suggest that any mutation rate that is larger than 0.1 will yield good performance. Unfortunately, it seems impossible to observe a mutation rate that yields the best performance under differ network configurations and parameters.

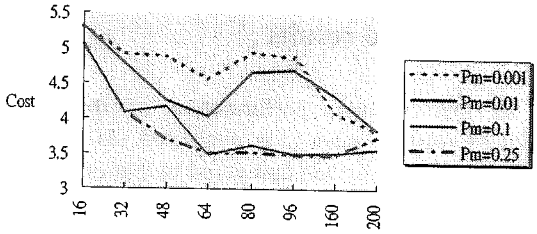


Figure 14: The effects of mutation rate .

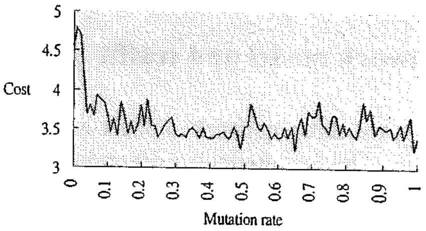


Figure 15: The effects of mutation rate

4.2.3 Comparison of performance

In this section, we evaluate the performance of three multicast routing algorithms through simulations. The simulations on the 10-node fully connected network are run on SUN IPX workstations while simulations on the other two network configurations are run on IBM PC's.

Table 1 shows the performance of the three multicast algorithms under different sizes of destination set and GA parameters. The last column of Table 1 shows the relative error rate, i.e. the ratio of the cost difference between the proposed algorithm and the optimal solution to the optimal solution. Figure 16 shows the obtained solution from our proposed method for the broadcast case, where node 0 is the source node. In this example, the proposed method cannot find the optimal solution, even both the population size and the number of generations are set to large. Thus, it seems to fall into local-optimum. However, we observe that the difference between the costs obtained from the proposed method and the optimal method is always less than the minimum cost of one physical link, i.e. 0.3. By carefully examine the solutions, we found that the selected links from the two methods are almost the same.

The reason why our algorithm cannot find the optimal solution may due to the fact that the number of routes in routing tables is limited. Therefore, the optimal solution may not be presented by any combinations of genes. Since increases the size of routing table also increases the computation complexity of our algorithm, we conclude that our proposed method is not very suitable for solving the multicast routing problem in fully connected networks.

multicast method multicast nodes	Optimal method	ADH method	Propose algorithm	error rate
Complete multicast (broadcast)	3.090889	3.090889	3.235910	4.692%
8-destination nodes	2.754019	2.754019	2.899040	5.265%
7-destination nodes	2.412504	2.412504	2.483104	2.926%
6-destination nodes	2.242692	2.433600	2.295119	2.338%

Table 1: The performance of three methods under the 10-node fully connected network model.

Algorithm &Time # of destinations	ADH method	ADH time (sec)	Propose algorithm	GA time (sec)	improvement
24	13.810000	0.22	13.968166	0.33	-1.15%
20	12.571197	0.17	10.716147	0.28	14.76%
16	12.571197	0.22	11.324875	0.33	9.91%
12	10.673905	0.11	10.2558900	0.22	3.92%
8	7.062323	0.11	7.081897	0.16	-0.28%
5	5.234097	0.11	4.902666	0.11	6.33%

Table 2: The performance of ADH and our proposed algorithm under the 25-node network model and GA parameters set 1.

Algorithm &Time # of destinations	ADH method	ADH time (sec)	Propose algorithm	GA time (sec)	improvement
24	13.810000	0.22	11.601373	0.55	15.99%
20	12.571197	0.17	10.716147	0.39	14.76%
16	12.571197	0.22	11.564104	0.38	8.01%
12	10.673905	0.11	10.000676	0.33	6.31%
8	7.062323	0.11	6.980517	0.16	1.16%
5	5.234097	0.11	4.336138	0.22	6.33%

Table 3: The performance of ADH and our proposed algorithm under the 25-node network model and GA parameters set 2.

The performance of the ADH algorithm and our proposed algorithm under the 25-node network model is compared in Tables 2 and 3. In table 2, the size of the gene pool is set to 24, the crossover rate is set to 1. The first 8 chromosomes are selected for reproduction. The mutation rate is 0.39. The costs shown in Table 2 are obtained after 100 generations. As we can see from Table 2, our proposed algorithm outperforms the ADH algorithm under most multicast settings. However, it also takes more computation time.

The performance of our proposed algorithm can yield even better performance if we increase the population size. In table 3, the size of the gene pool is increased to 32, the crossover rate is still set to 1. The first 8 chromosomes are selected for reproduction. The mutation rate is 0.58. Again, the performance is obtained after 100 generations. As we can observe from Table 3, our proposed algorithm yields much better performance than the ADH algorithm under any multicast settings.

Finally, these two algorithms are evaluated under the network model of Figure 11. Tables 4 and 5 show the comparison of the performance of the ADH method and the proposed method. The parameters in all experiments are set as follows. In table 4, the size of gene pool is set to 24, the crossover rate is set to 1, and the mutation rate is set to 0.81. The first 6 chromosomes are selected for reproduction. Again, the performance is obtained after 100 generations. As we can observe from Table 4, our proposed algorithm yields slightly better performance than the ADH algorithm.

The performance of our proposed algorithm yields better performance with larger population size. In table 5, the size of the gene pool is increased to 64, the crossover rate is still set to 1, and the mutation rate is set to 0.92. The first 12 chromosomes are selected for reproduction. Again, the performance is obtained after 100 generations. As we can observe from Table 5, our proposed algorithm yields much better performance than the ADH algorithm under any multicast settings.

Algorithm &Time # of destinations	ADH method	ADH time (sec)	Propose algorithm	GA time (sec)	improvement
20	11.039253	0.33	10.832664	0.44	1.87%
15	9.318338	0.27	9.941274	0.33	-6.69%
10	9.758270	0.22	9.050517	0.22	7.25%
8	7.272851	0.16	7.370375	0.18	-1.34%
5	4.983870	0.16	4.983870	0.11	0%

Table 4: The performance of ADH and our proposed algorithm under the 21-node network model and GA parameters set 3.

Algorithm &Time # of destinations	ADH method	ADH time (sec)	Propose algorithm	GA time (sec)	improvement
20	11.039253	0.33	10.832664	0.73	1.87%
15	9.318338	0.27	8.612569	0.60	7.57%
10	9.758270	0.22	9.050517	0.50	7.25%
8	7.272851	0.16	7.122356	0.36	2.07%
5	4.983870	0.16	4.983870	0.34	0%

Table 5: The performance of ADH and our proposed algorithm under the 21-node network model and GA parameters set 4.

5. Summary and future work

In this paper, we have proposed a new multicast routing algorithm based on GAs. Current existing multicast routing algorithms emphasis either on minimum cost or path delay. Our new multicast routing algorithm tries to minimize the multicast cost while maintaining a reasonable path delay.

From our simulation results, it can be found that with properly setting of the GA's parameters, such as the population size, the generation time, the mutation rate and the crossover rate, the proposed multicast routing algorithm is able to obtain a better solution than the ADH method in the two sparsely connected network models we examined. However, in the 10-node fully connected network, it does not seem to yield a better performance than the ADH method. We believe that if the routing table is large enough to contain all possible routings, it is possible to obtain the optimal solution. Although the proposed algorithm requires more computation time than the ADH algorithm, it is still very promising for the use in real networks. The computation time can be reduced if the number of generations can be decreased. In real network, the solutions obtained for setting up a multicast connection can be used as initial chromosomes for the next multicast request, either with the same source and destination set or not. Since the network configuration (e.g. traffic on each link) may not change significantly between two consecutive multicast requests, a near optimal solution can be obtained after few generations. Therefore, in the future, we will examine the number of generations require if network configuration does not change significantly and the previous solution can be used as the initial solution.

References

[1] S. L. Hakimi, Steiner's "Steiner problem in graphs and its implications," *Networks,* Vol. 1, pp. 113-133, 1971.

[2] R. M. Karp, "Reducibility among combinatorial problems," Complexity of Computer Communications, R. E. Miller and J. W. Thatcher, Eds, New York, NY: Plenum, 1972, pp. 85-103.

[3] M. R. Gareg and D. S. Johnson, Computer and Intractability: A guide to the Theory of NP-Completeness. New York, NY: Freeman, 1979.

[4] C. H. Chow, "On multicast path finding algorithms," IEEE INFOCOM'91, pp. 1274-1283, 1991.

[5] H. Takahashi and A. Mutsuyama, "An approximate solution for the Steiner problem in graphs," *Math. Japonica.* Vol. 6, pp. 573-577, 1980.

[6] B. M. Waxman, "Routing of multipoint connections," IEEE J. Selected Areas on Communication. Vol. 6 No. 9, pp. 1617-1622, Dec 1988.

[7] D. E. Goldberg, *Genetic Algorithms: Search, Optimization and Machine Learning, Addison Wesley,* Reading, MA, (1989).

[8] J. H. Holland, "Genetic algorithms," *Scientific American,* July, 1992, pp. 66-72.

[9] L. Davies (Ed.), *Genetic Algorithms and Simulated Annealing,* Morgan Kaufmann, (1987).

[10] G. Syswerda, "Uniform Crossover in Genetic Algorithm," *Proc, 3rd. Int. Conf. On GA,* pp. 2-9, 1989.

[11] H. Muhlenbein, "Parallel Genetic Algorithm, population genetics and combinatorial optimization," *Proc, 3rd. Int. Conf. On GA,* pp. 416-421, 1989.

[12] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs,* Springer-Verlag, New York, 1992.

[13] H. Pan, and I. Y. Wang, "The bandwidth allocation of ATM through Genetic Algorithm," IEEE GLOBECOM'91, pp. 125-129, 1991.

[14] L.-D. Chou, and J.-L. C. WU, "Buffer Management Using Genetic Algorithms and Neural Networks," IEEE GLOBECOM'95, pp. 1333-1337, 1995.

[15] N. Shimamoto, A. Hiramatu, and K. Yamasaki, "A dynamic routing control based on a genetic algorithm," 1993 IEEE International Conference on Neural Network, pp. 1123-1128, 1993.

[16] C. C. Palmer and A. Kershenbaum, "Two algorithms for finding optimal communication spanning trees," Technical Report, IBM T. J. Watson Research Center, Yorktown Heights, NY, 1993.

[17] V. J. Rayward-Smith, "The computation of nearly minimal Steiner trees in graphs," *Int. J. Math. Educ. Sci. Technol.* Vol. 14, pp. 15-23, 1983.

[18] V. J. Rayward-Smith and A. Clare, "On Finding Steiner vertices," *Networks,* Vol. 16, pp. 283-294, 1986.

[19] C. C. Palmer, A. Kershenbaum, "Representing Trees in Genetic Algorithms," IEEE World Congress on Computational Intelligence, Vol. 1, pp. 379-384, 1993.

[20] S. E. Dreyfus and R. A. Wagner, "The Steiner problem in graphs," *Networks,* Vol. 1, pp. 195-207, 1971.

[21] C.-S. Wu, S. Lee, and K.-T. Wu, "Study of Working and Backup Path Routing Policies in Self-healing ATM Networks," 1995 International Symposium on Communications, Taipei, Dec. 1995, pp. 807-814.

[22] T. J. and K. R. Krishnan, "State Dependent Routing of Telephone Traffic and the use of Separable Routing Schemes," ITC11, Kyoto, Japan, 1985, pp. 867-872.