Proceedings of the 1997 ACM/SIGAPP Symposium on Applied Computing Feb. 28 - March 2, 1997, pp.263-268, ACM Press.

A HYBRID ALGORITHM FOR THE POINT TO MULTIPOINT ROUTING PROBLEM

Heather L. Christensen
Mathematical and Computer Sciences Dept
The University of Tulsa
600 South College Avenue
Tulsa, OK 74104-3189, USA

Roger L. Wainwright
Mathematical and Computer Sciences Dept
The University of Tulsa
600 South College Avenue
Tulsa, OK 74104-3189, USA
rogerw@penguin.mcs.utulsa.edu
http://www.utulsa.edu

Dale A. Schoenefeld
Mathematical and Computer Sciences Department
The University of Tulsa
600 South College Avenue
Tulsa, OK 74104-3189, USA
dschoen@euler.mcs.utulsa.edu
http://www.utulsa.edu

Key Words: Genetic Algorithms, Steiner Trees, Point to Multipoint Routing, Telecommunications Networks

ABSTRACT

The process of finding optimal routing for a set of circuit connection requests through a communications network is known as call request scheduling or message scheduling. Each request has a single source and multiple destinations and different requests may have different source and different destination nodes. Finding optimal routing for a set of requests is called the Point to Multipoint Routing Problem (PMRP). Current practice takes each point to multipoint request and treats it as a collection of point to point requests, which is very costly. This paper presents an algorithm for the Point to Multipoint Routing Problem that uses a genetic algorithm and a heuristic Steiner tree algorithm. Our hybrid algorithm allows the scheduler to find an optimal or near-optimal path through the network for each request. In our algorithm each request is treated as a whole and not as a collection of point to point requests. We ran our hybrid PMRP algorithm on several test cases with excellent results.

INTRODUCTION

The process of finding optimal routing for a set of circuit connection requests through a communications network is known as call request scheduling or message scheduling. Each request has a single source and multiple destinations and different requests may have different source and different destination nodes. Finding an optimal routing for a set of requests is called the Point to Multipoint Routing Problem (PMRP). Discovering efficient methods to optimally route requests through a communications network can save a company thousands of dollars. Current practice takes each point to multipoint request and treats it as a collection of point to point requests. This method incurs considerable costs as it is common for multiple copies of the request to appear on the same link or on parallel links. This paper presents an algorithm for the Point to Multipoint Routing Problem that uses a genetic algorithm and a heuristic Steiner tree algorithm. Using both of these techniques in conjunction allows the scheduler to find an optimal or near-optimal path through the network for each request without having to consider the request as several point to point messages. A Steiner tree algorithm is used to find a minimal cost or near minimal cost tree in the network for a given request. Therefore, each request is treated as a whole. The Genetic algorithm is used to determine the optimal or near optimal ordering of the requests.

POINT TO POINT ROUTING PROBLEM

With the growth in size and number of communications networks, techniques for making networks more reliable, efficient, and cost effective are in high demand. Discovering efficient methods to optimally route connection requests through a communications network can save a company thousands of dollars. One problem in communication networks is finding optimal routings for a set of requests through a communications network. As an example of this type of network problem, consider a scheduler for a telecommunications network that schedules multiple simultaneous requests through a network. Each request has a single source node and a single destination node. The source and destination nodes may be any nodes in the network. The entire signal represented by a request must be routed along the same path. Each link in the network has an assigned capacity and a cost for using the link. This is called the Point To Point Routing Problem (PPRP), since each request routes a signal from one point to another point in the network. The PPRP is known to be NP-complete [4].

Cox *et al.* have studied the PPRP extensively [4]. Cox describes a scheduler for a circuit switching telecommunications network on which call requests arrive randomly at any of the nodes of the network. Each call request is specified by six attributes: source node, destination node, requested start time, requested duration, bandwidth requirement or capacity, and priority class. For each request, the scheduler assigns a single dedicated path through the network from the source node to the destination node during the entire requested time interval. That is, the connection is *circuit switched*. The scheduler must take into consideration the capacities of requests that might concurrently use a link. A feasible schedule for a set of requests is one for which the sum of the capacities of the requests assigned to a link, at any given time, is never greater than the link's capacity. If a call request is not accommodated by a particular schedule, the request is said to be blocked and the schedule is infeasible.

The scheduler calculates the cost of each feasible schedule. The cost is the sum of the net costs of successfully carrying scheduled requests through their assigned paths. For an infeasible schedule, one includes penalties associated with requests that are blocked. The scheduler's task is to choose a set of paths for all call requests that minimizes total cost while taking into account the time varying nature of the call requests. Cox addresses this problem through the use of a statistical approach that makes rapid initial assignments of incoming call requests to paths, based upon statistical information about the current network traffic. He then uses evolutionary programming techniques to find alternate routing assignments that are more nearly optimal or that can accommodate additional requests [4].

POINT TO MULTIPOINT ROUTING PROBLEM

Though some work has been done on the Point To Point Routing problem, little attention has been given to routing requests where each request originates at a single source node and may have several destination nodes. This is called the Point to Multipoint Routing Problem (PMRP). The Point to Multipoint Routing Problem is similar to the Point to Point Routing Problem. The difference is that in the PMRP there are several destination nodes instead of a single destination node for each request. The remaining attributes of a request as described by Cox (source node, requested start time, requested duration, bandwidth requirement or capacity, and priority class) are the same for the PMRP as for the PPRP.

Our investigation into the PMRP is a result of our collaboration with LDDS WorldCom (formally WilTel), a locally based internationally prominent telecommunications vendor. Frequently, a telecommunications company needs to transmit the same signal to many different destinations over a telecommunications network backbone; that is, it must solve the PMRP. Most current algorithms for implementing the PMRP take each point to multipoint request and treat it as a collection of point to point requests. This technique incurs considerable costs as it is common for multiple copies of the signal to appear on the same link or on parallel links.

Our technique for the PMRP finds a circuit switched minimal spanning tree (MST) assignment for each of several point to multipoint requests so that the largest possible number of requests can be accommodated or so that the routings incur the least possible cost. The MST determines a point to multipoint circuit in the network. Instead of several copies of the signal on a single link, a single copy would be sent. When this signal reaches an intermediate node that is linked to two or more destination nodes or other intermediate nodes, the intermediate node duplicates the signal as determines by the MST. Compared to treating each point to multipoint request as several point to point requests, our method reduces the network traffic caused by each request, and allows for more messages to be sent simultaneously.

As an example of the PMRP, consider a communications network consisting of 8 nodes as shown in Figure 1. The network can be modeled as a graph consisting of eight vertices representing the eight nodes and eleven edges representing the available transmission lines between them. Each edge has associated with it a cost for use and an available capacity. Note the cost is a per unit of capacity cost. For example, using one unit of capacity on link AB in Figure 1 costs seven, and using five units of capacity costs 35. Now consider four message requests. Each request contains a source node, one or more destination nodes, and a message size or capacity. This is shown in Table 1. To route these four messages using a point to multipoint routing technique, the scheduler must find, for each message, a tree in the network that contains the source and each destination. In this case, the scheduler must find four different trees (one tree for each source and its destinations in Table 1). For the routing to be optimal, the trees must cost as little as possible and no request can be blocked. The requests can be routed in any order. However, the order one chooses for the requests may greatly effect the solution.

Consider R2, R3, R4, and R1 as an example order for routing the requests messages. Suppose in order to route R2 (D to B and D to H) we use a capacity of 3 units on edges DE, BE, and EH, which by observation, represents the least cost routing for message R2. Next, suppose in order to route R3 (G to E, G to F, and G to B) we use a capacity of 4 units on edges GH, EH, FG, and BE. Next suppose in order to route R4 (A to G) we use a capacity of 2 units on edges AC, CD, and DG). Figure 2 shows the results of starting with the network in Figure 1 and routing R2, R3, and R4 in that order. Note after R4 is routed, we can route part of R1 (5 units from B to C), but we are unable to route 5 units from B to H, because there is no available path from B to H with enough capacity for the request. Thus, R1 becomes *blocked*. Though there may be a successful ordering for considering the requests in a given network, the scheduler is operating under time constraints.

We present a hybrid algorithm for the Point to Multipoint Problem that uses genetic algorithms and Steiner tree heuristics. Used in conjunction, these two algorithms allow the scheduler to find an optimal or near-optimal path through the network for each request without treating the request as several point to point messages. A Steiner tree algorithm is used to find a minimal or near minimal cost tree in the network for a given request. Therefore, each request is treated as a whole, where previous algorithms considered a multiple destination request as several point to point requests. The genetic algorithm allows a search through all possible orderings of the requests attempting to find the least cost ordering. To the authors' knowledge this paper is the first attempt to formulate the PMRP problem and to solve the PMRP using genetic algorithms.

Table 1: Four Example Message Requests

Request	Source	Destination(s)	Capacity
R1	B	C, H	5
R2	D	B, H	3
R3	G	E, F, B	4
R4	A	G	2

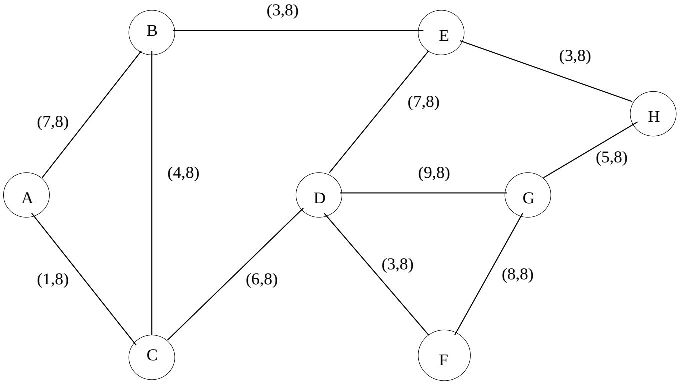

Figure 1: A Communications Network with (Cost, Capacity) associated with each edge

GENETIC ALGORITHMS

Genetic Algorithms (GA) are based on the principles of natural genetics and survival of the fittest. Genetic algorithms search for solutions by emulating biological selection and reproduction. The genetic algorithm operates on a fixed size population of chromosomes. A chromosome is a string of genes that represents an encoding of a candidate solution. An allele is a value assumed by a gene. Associated with each chromosome is a fitness value. The fitness of a chromosome corresponds to its ability to survive and reproduce offspring. The least fit or weakest chromosomes of the population are displaced by more fit chromosomes.

Several researchers have investigated the benefits of solving combinatorial problems using genetic algorithms. Davis [5], Goldberg [7], Rawlins [9], and Mitchell [8] provide excellent in-depth studies of genetic algorithms. This research uses an order based genetic algorithm where a chromosome

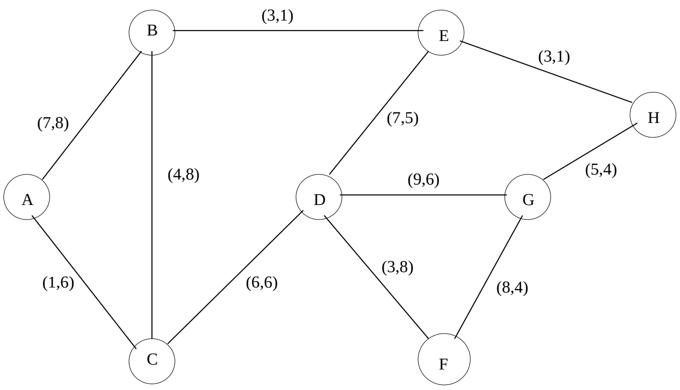

Figure 2: Communications Network after Routing R2, R3, and R4 using PMRP

is a permutation of integers. The genetic algorithm implementation used in this research is LibGA [3]. It is assumed the reader is generally familiar with the fundamentals of genetic algorithms.

STEINER TREE PROBLEM IN A GRAPH

The Steiner Problem in a Graph (SPG) is a classic combinatorial optimization problem. The SPG has been shown to be NP-complete. Given a graph and a required subset of vertices from the graph, W, any subtree containing all vertices of W, and possibly additional vertices from the graph, is called a Steiner tree. A minimal Steiner tree is a Steiner tree of minimal cost. Thus, given a weighted graph $G = (V,E)$, and a subset $W \subseteq V$, the SPG is to find a minimum cost spanning tree $T = (V', E')$ with $W \subseteq V' \subseteq V$ and $E' \subseteq E$. An example Steiner tree where the required vertices are W = $\{V_0, V_4, V_8\}$ is shown in Figure 3. Steiner trees are applied to network design, circuit layout and design, as well as many other network scheduling and routing problems. There are several techniques for constructing near-optimal Steiner trees in the literature [1, 6]. We assume the reader is familiar with the fundamental concepts of Steiner trees.

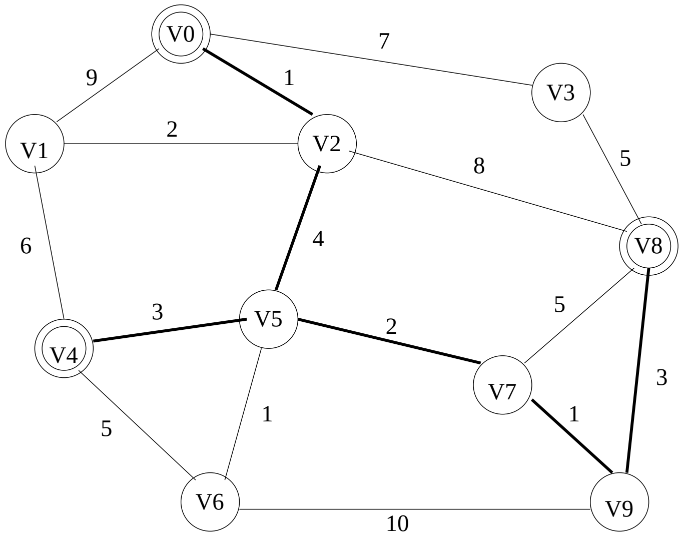

Figure 3: An SPG Example, W = $\{V_0, V_4, V_8\}$. The bold lines indicate the Minimum Cost Steiner Tree

METHODOLOGY AND IMPLEMENTATION

We consider only the special case of the PMRP where all requests have the same start time, duration, and priority. The first step to finding a solution is to model the network as a graph. Each edge is assigned a cost and a capacity. Each request is identified by the source node, the destination node(s), and the capacity of the request. In our hybrid algorithm, a non-GA (deterministic) method initially selects a Steiner tree for each request and then a GA searches for a good ordering of the requests. In our GA, the chromosome length is equal to the total number of requests to be routed through the network, and the allele type is integer. Consequently, the chromosome is a permutation of integers where each integer is a request number. The genetic algorithm initializes the population with random permutations.

During execution, the genetic algorithm calls an evaluation function to assign fitness values to the chromosomes. Each message is routed through the network in the order it appears in the chromosome. To route a given request, the algorithm first inspects the edges of the entire graph for any edges with a capacity less than the capacity of the request and temporarily changes the cost on each such edge to an artificially high cost value. Then, a near-optimal or optimal Steiner tree is found for the current request. After the Steiner tree has been found, the capacities of the edges used in the Steiner tree are decremented by the capacity of the request. This is repeated for each request, in the order they are found in the chromosome, until every request has been routed. The evaluation function then assigns the fitness value to the chromosome. The fitness value of a chromosome is the cost of using the network. That is, the sum of the capacity used on each link times its cost.

Consider the network in Figure 1 and the requests in Table 1. Assume that the genetic algorithm is ready to evaluate a chromosome with an allele pattern of 2,1,4,3. First a Steiner tree is found to route R2. Then, Steiner trees will be determined for R1, R4, and R3. For each request the algorithm verifies that capacities on all edges are not less than the capacity of the request. Consider the route for R2. The algorithm finds a Steiner tree for R2. The result is shown in Figure 4. The Steiner tree is illustrated with the dashed lines and the changes to the available capacities are shown.

The algorithm now finds a routing for R1. Because the capacity of R1 is five and all links in the network still have a capacity of at least five, all links are declared usable. The resulting Steiner tree is shown in Figure 5. Note that the remaining capacities on edges BE and EH are now zero. Next, R4 is routed. Because the capacity of R4 is greater than the remaining capacities on edges BE and EH, those edges have their costs changed to an extremely high value to minimize the possibility of being used by the Steiner tree algorithm. After the Steiner tree function finishes execution, the cost values of edges BE and EH are returned to their original value. Figure 6 shows the Steiner tree and resulting capacities after routing R4. Finally, R3 is routed. As before, edges BE and EH are assigned high cost values. Edge BC must be given a high cost as well. The result of routing R3 is shown in Figure 7. Hence the routing schedule for chromosome (2 1 4 3) using our PMRP genetic algorithm is shown in Figure 7 with a fitness value (cost of using the network) of 253.

For comparison, consider routing R2, R1, R4, R3 using the traditional PPRP algorithm that considers each of the four requests as a collection of point to point requests. Since point to multipoint routes are broken up into independent point to point routes, R2 becomes $R2_a$ (D to B), and $R2_b$ (D to H). R1 becomes $R1_a$ (B to C) and $R1_b$ (B to H). R4 remains unchanged, and R3 becomes $R3_a$ (G to E), $R3_b$ (G to F), and $R3_c$ (G to B). A routing is as follows: $R2_a$ used 3 units on DE and BE. $R2_b$ uses 3 units on DE and EH. $R1_a$ uses 5 units on BC. $R1_b$ uses 5 units on BE and EH. R4 uses 2 units on AC, CD, and DG. The results after routing $R2_a$, $R2_b$, $R1_a$, $R1_b$, and R4 are shown in Figure 8. Note at this point $R3_a$ is blocked, since there is no way to route 4 units from G to E. Of course the problem is that 3 units were double counted on DE for routing the same message for $R2_a$ and $R2_b$.

In this research, a generational genetic algorithm was used. The pool size was fifty. The PMX crossover method and roulette wheel selection method were used. The GA terminated after 100 iterations or upon convergence. Our program to find Steiner trees was an adaptation of the code developed by Alexander and Robins [1]. It uses the Kou, Markowsky, Berman (KMB) algorithm for solving the Steiner problem in a graph.

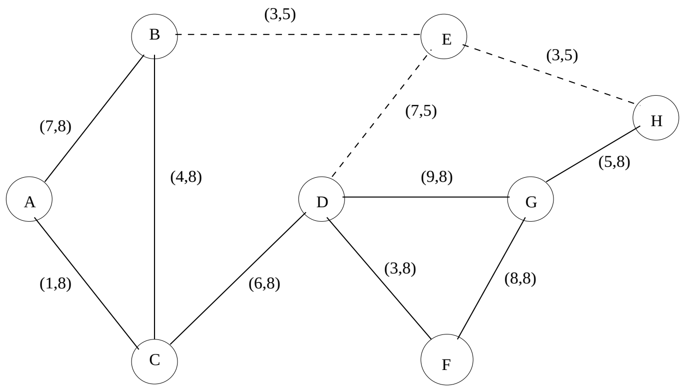

Figure 4: The Steiner Tree for R2 (dashed lines) and the changes in available capacities

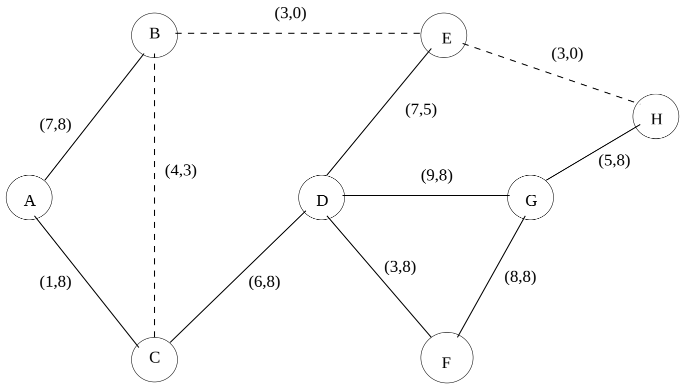

Figure 5: The Steiner Tree for R1 (dashed lines) and the available capacities

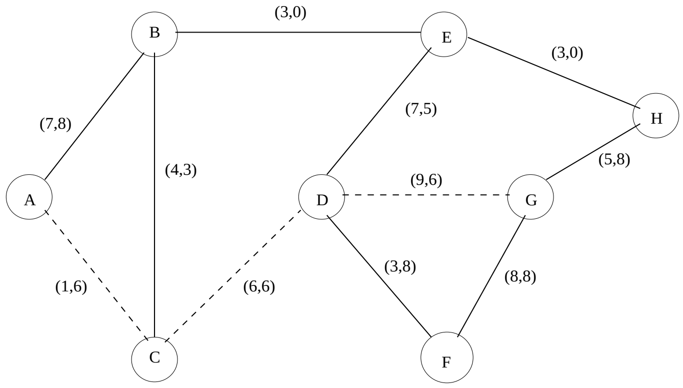

Figure 6: The Steiner Tree for R4 (dashed lines) and the available capacities

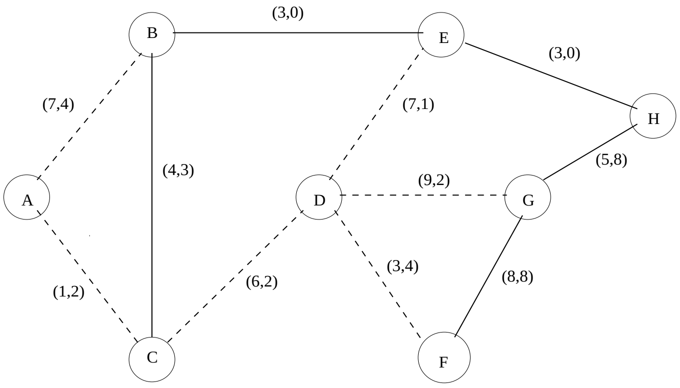

Figure 7: The Steiner Tree for R3 (dashed lines) and the available capacities

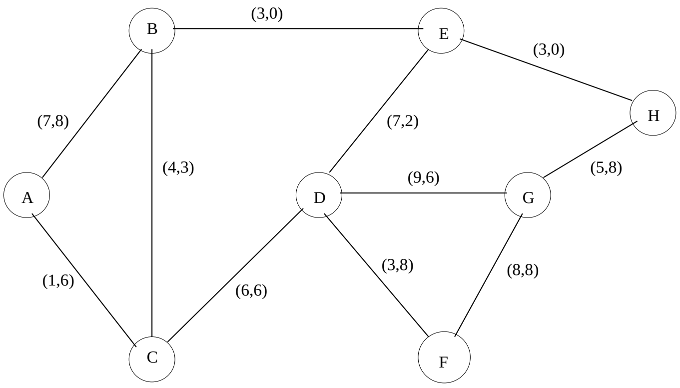

Figure 8: A Blocked Communications Network after Routing $R2_a$, $R2_b$, $R1_a$, $R1_b$, and R4 using PPRP

TEST CASES

We used test cases with four, five, six, seven, and eight routing requests on a single network of fifty nodes and eighty-three edges. The data set was obtained by modifying one of the benchmark *steinb* data sets that are used for the Graphical Steiner Tree Problem [2]. Each edge was assigned a random cost from one to ten and a fixed capacity of twelve. Each request contained no more than ten nodes and had a capacity between five and nine. The requests are listed in Table 2. To make the implementation as close as possible to Cox's simulation, the data sets all began with the same requests. That is, the five-request data set contains the same requests as the four-request data set with one additional request. The six-request data set has the same requests as the five-request data set with one more request, and so on. This simulates, to some extent, the random arrival of requests for the scheduler. This also simulates, in practice, how LDDS WorldCom schedules their requests. First a set of requests is scheduled for the network for a specific time period, then later another request arrives and a new schedule is determined, then another request comes in, a new schedule is determined, and so on until no more requests can be scheduled or the time has arrived to execute the schedule. In LDDS WorldCom's case, their algorithm for scheduling the request is based on a combination of simple point to point routings and hand scheduling based on experience. Our GA implementation automates this process with good results.

Table 2: Test Case Requests

Request Number	Source Node	Destination Node(s)	Capacity
1	36	7,23,25,40	8
2	17	15,30,31,40,41,46	5
3	48	3,9	9
4	41	13,22,27,35,50	6
5	2	6,14,18,23,27,33,47,49	5
6	13	28	6
7	50	5,12,28,31,44,45	7
8	20	17,25,41	8

RESULTS

Our algorithm found feasible solutions for seven of the eight test cases. On the eighth test case, the Steiner code fails to find an optimal tree without employing an edge that was unusable. This is due to the high capacity assigned to request number eight and the lack of available capacity left on the edges of the network. Indeed there may not be a feasible solution for the case of eight call requests in this network. Because eight requests failed, we did not try a higher number of requests. Having a network model with more edges between nodes may increase the possibility of more requests being scheduled. Table 3 contains the resulting ordering of the requests and their fitness values found by the genetic algorithm for each of the test cases. Because there is no previous research similar to this project, there is nothing with which to compare our results. However, the results exceed expectations and give an excellent indication of the potential success of future genetic algorithm implementations of the PMRP problem. The GA implementation for the test case with eight requests took under 10 minutes of CPU time running on a Sun Sparc 10 workstation.

Table 3: Results of the Five Test Cases

Test Case	Number of Requests	Final Ordering(s)	Fitness Value
1	4	3,4,1,2	1638
2	5	3,5,1,4,2	2114
3	6	3,5,6,1,2,4	2206
4	7	7,4,3,6,5,1,2 and	2560
		4,7,3,6,5,1,2	
5	8	7,8,3,1,4,2,5,6	4235

This research successfully demonstrates the potential for a point to multipoint request scheduler. By using a Steiner tree to describe the routing of a signal from the source to the destinations, a near-optimal routing can be found. By using Steiner tree based routing instead of treating the point to multipoint requests as several point to point requests, there will generally be fewer simutaneous copies of a signal on a link, resulting in less network traffic and a potential for more requests to be sent. We did consider using a GA to determine the "best" Steiner tree for each call request (a GA within a GA implementation). However, we determined that the GA driving the order of the requests was the crucial factor, so a deterministic Steiner tree implementation was used. In addition, by using genetic algorithms to find an optimal or near-optimal ordering of the nodes, a series of message requests can be routed concurrently at a reasonable cost without blocking as many requests. The result is a network that can accommodate more requests at a lower cost and offer increased profit for industry.

Acknowledgements

The authors gratefully acknowledge Michael Alexander and Gabriel Robins for providing code for several Steiner tree algorithms that they developed in [1].

References

[1] Michael J. Alexander and Gabriel Robins, "New Performance-Driven FPGA Routing Algorithms", Proceedings of ACM/SIGDA Design Automation Conference, June 1995.

[2] J.E. Beasley, "OR-Library: distributing test problems by electronic mail," *J. Opl. Res. Soc.*, vol. 41, pp 1069-1072, 1990.

[3] A.L. Corcoran and R.L. Wainwright, "Using LibGA to Develop Genetic Algorithms for Solving Combinatorial Optimization Problems", *Practical Handbook of Genetic Algorithms*, Vol. 1, Lance Chambers, ed., CRC Press, 1995, pp 143- 172.

[4] Louis Anthony Cox, Jr., Lawrence Davis and Yuping Qiu, "Dynamic Anticipatory Routing In Circuit-Switched Telecommunications Networks", *Handbook of Genetic Algorithms*, Lawrence Davis, ed., Van Nostrand Reinhold, 1991, pp 124 - 143.

[5] L. Davis, editor, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1991.

[6] Henrik Esbensen, "Finding (Near-) Optimal Steiner Trees in Large Graphs", Proceedings of the Sixth International Conference on Genetic Algorithms, Morgan Kaufmann Publishers, Inc., 1995, pp 485 - 492.

[7] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.

[8] Melaine Mitchell, *An Introduction to Genetic Algorithms*, The MIT Press, 1996.

[9] Gregory Rawlins, editor, *Foundations of Genetic Algorithms*, Morgan Kaufman, 1991.