A Hybrid Genetic Algorithm for the Point to Multipoint Routing Problem with Single Split Paths

Pablo Galiasso
Roger L. Wainwright

Mathematical and Computer Sciences Department
The University of Tulsa,
600 South College Avenue
Tulsa, Ok 74104-3189 USA
galiasso@euler.mcs.utulsa.edu; rogerw@utulsa.edu

Key Words: Genetic Algorithm, Steiner Trees, Point to Multipoint Routing, Telecommunications Network

ABSTRACT

A hybrid evolutionary algorithm for solving the Point to Multipoint Routing Problem (PMRP) is presented. We considered an application of the PMRP called the Message Scheduling Problem. The Message Scheduling Problem is the process of scheduling a set of requests through a network where each request has a single source and multiple destinations. Our new algorithm not only treats each request as a whole, but also allows up to two paths for transmitting the request bandwidth. We call this problem the Point to Multipoint Routing Problem with Single Split Paths. Our hybrid algorithm uses a genetic algorithm and a heuristic Steiner tree algorithm for finding near-optimal solutions to this problem. We designed the chromosome to accommodate not only the option of multiple paths for a request, but how to split the bandwidth. We ran our algorithm against results from previous researchers with superior results.

1. INTRODUCTION

With the rapid growth in the telecommunication business using network backbones and optical fiber, it is extremely important to remain competitive by making networks more efficient, reliable and cost effective in its day to day operation. Companies could save thousand of dollars by optimizing communications through their network. Consider a set of requests to be routed through a network. Each request has a single source node with perhaps multiple destination nodes. Different requests may have different source and different destination nodes. Finding a optimal routing through a network for a given set of requests is called the Point to Multipoint Routing Problem (PMRP).

In this paper we present a hybrid evolutionary algorithm for solving the Point to Multipoint Routing Problem. Our new algorithm not only treats each request as a whole, but also allows up to two paths for transmitting the request bandwidth. We call this problem the Point to Multipoint Routing Problem with Single Split Paths. Our algorithm uses a genetic algorithm and a heuristic Steiner tree algorithm for finding near-optimal solutions to this problem. One challenging aspect of this new algorithm is not only finding the path of each sub-request (up to two sub-requests per request) but also determining how much bandwidth should be transmitted for each sub-request. Other evolutionary computation research involving Telecommunications problems includes Al-Qahtani, *et. al* [2], Chakraborty, *et. al* [3], Knowles, *et. al* [7], and Ryan, *et. al* [8]. The rest of the paper is described as follows: Section 2 provides a summary of the classic routing problems and a summary of previous research for the PMRP. Section 3 presents our algorithm for the PMRP with single split paths. Section 4 reviews the Steiner Tree problem in a graph, and in Section 5 we present our methodology and implementation for the PMRP with single split paths. Results are given in Section 6, and conclusions are presented in Section 7.

2. PREVIOUS RESEARCH FOR PMRP

A classic problem in network communications is finding an optimal routing for a set of communication request through a network, where each request has *one* source and *one* destination. For example, imagine a scheduler for a network, such as the telephone system, that arranges multiple simultaneous requests from any node to any other node. Each request has a single source and single destination node. The entire signal must be routed along the same path. Each link in the network has a capacity and an associated cost for using it.

This problem is called the Point-to-Point Routing Problem (PPRP), which is known to be NP-complete [5].

Even though some work has been done on the PPRP, very little research has been done routing requests where each request originates from a single node and has more than one destination node, that is, the Point to Multipoint Routing Problem (PMRP). Historically, a simple PMRP was treated as multiple PPRPs. That is, a PMRP with *n* destination nodes was solved as *n* PPRPs with the same source node. Since results from PPRPs were computed independently, the entire solution was far from optimal, producing a costly solution. Christensen, *et. al* [4] and Zhu *et. al* [10] were the first researchers to consider each PMRP request as a whole and not as a collection of point-to-point requests. Both Christensen and Zhu used a genetic algorithm and a heuristic Steiner tree algorithm to find a near-optimal path for each point to multipoint request.

Consider the following application of the PMRP called the Message Scheduling or the Call Request Scheduling Problem. The Message Scheduling Problem is the process of scheduling a set of requests through a network where each request has a single source and multiple destinations. Each edge in the network is assumed to have a know capacity and a cost for its use. A schedule is some ordering of the requests, where each request is a PMRP. A scheduler must calculate the cost of each feasible schedule. Cost is defined as the sum of the network edge costs for carrying each request through the assigned edges in the network. Furthermore, the sum of the capacities of all of the requests assigned to a given network link cannot exceed the capacity of the link. A scheduler must find a routing schedule for as many requests a possible at the minimum cost. Clearly the order of the requests as they are assigned to the network is important. The message Scheduling Problem is a very common problem found in broadcast television using telecommunication network backbones.

To further illustrate this problem, consider a communications company selling bandwidth on their network for the following day. Requests come in throughout the current day, and a commitment is not required until late evening. The order of the requests as they arrive may or may not be a consideration in the final schedule. Our initial interest with the Message Scheduling Problem is a result of our collaboration with two local, but well known telecommunication companies: The Williams Communication Group and MCI-WorldCom.

Christensen *et. al* [4] was the first to attempt to solve this problem. Their research assumed requests were given a priority based on the order in which they arrived. That is, their algorithm determined a schedule for the first *n* requests by finding some permutation of the *n* requests that made the "best" feasible schedule. When the *n*+1 request arrived another "best" cost feasible schedule was determined. If no such schedule could be found, by trying various permutations of the first *n*+1 requests, then the final schedule was determined with the first *n* requests, and later arriving requests were never considered due to the first come - first serve priority. This constraint led to a rather limited search space and furthermore did not represent a real world situation.

Zhu *et. al* [10] extended Christensen's work by removing the priority of requests based on arrival. Now the objective was to schedule as many of the requests as possible. That is, given at total of *n* requests to schedule, find some subset *k* of the *n* requests ($1 \leq k \leq n$), such that *k* is maximized. Among several feasible schedules for *k* requests, determine the least cost schedule.

3. POINT TO MULTIPOINT ROUTING WITH SINGLE SPLIT PATH

We extended the research started by Zhu *et. al* [10] by allowing multiple paths for transmitting a single broadcast request. That is, part of the request bandwidth can be transmitted using one path and the rest can be transmitted through another path. Using unique paths for broadcasting over a network could cause an inefficient use of its communication links in certain situations. For example, when a communication line between two routers has less capacity left than the minimal capacity needed for transmitting any communication, it basically becomes useless, decreasing the capacity of the entire network. All signals consume capacity along their paths. We allowed for a request to be split into at most two distinct routes from source to various destinations. We contend that in the future broadcast over network backbones will divide the bandwidth of each communication into several sub-communications that will carry part of the entire bandwidth. This research exploits that notion.

In our research, we considered the PMRP Message Scheduling Problem where each request could be split up into at most two paths. This presented several new challenges. First, this problem not only involved finding the path of each sub-communication, but also finding how much bandwidth should be transmitted on each path. Secondly, the chromosome needed to be designed to accommodate not only the option of multiple paths for a request, but how to split the bandwidth. As an example PMRP Message Scheduling Problem consider the communications network with eight nodes depicted in Figure 1. Each edge has an associated cost for its use and capacity. Note the cost is a per unit of capacity. For example the cost of using one unit of capacity of edge AD is five, while the cost of using its full capacity is 5*20 = 100. An example list of messages to schedule for this network is depicted in Table 1.

Request	Source	Destination(s)	Capacity
R1	G	A,C,F	10
R2	A	E,F,G	7
R3	H	B,D,F,G	12
R4	C	E	8
R5	F	A,H	10

Table 1. Five Sample Requests.

We considered only the special case of the PMRP where all requests have the same start time, duration and priority.

4. STEINER TREE PROBLEM IN A GRAPH

The Steiner Tree Problem in a Graph is a well known

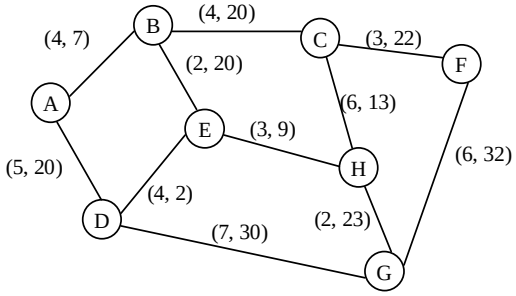

Figure 1. A Network with (cost, capacity) on each

combinatorial optimization problem that has been shown to be

a NP-complete problem. Given a graph and a subset of vertices *W* from the graph, a Steiner tree is any subtree of the graph that contains all vertices in *W*. Formally, given a weighed graph $G = (V,E)$, and a subset $W \subseteq V$ the Steiner Tree is to find a minimum cost tree $T = (V',E')$ with $W \subseteq V' \subseteq V$ and $E' \subseteq E$. A minimal Steiner tree is a Steiner tree with minimal cost. As an example, Figure 2 shows a weighted graph with a minimal Steiner tree for the set of vertices $W = \{V_1, V_6, V_9\}$.

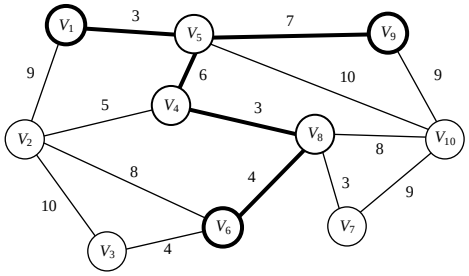

Figure 2. A weighted graph with a minimal Steiner tree for the set of vertices $W = \{V_1, V_6, V_9\}$

Steiner trees are applied to networks design, circuit layout and design, and many other scheduling and routing problems. Our hybrid evolutionary algorithm makes use of the Steiner tree to assist in the point to multipoint routing of each request. For example, if there is a need to route a request from V_1 to both V_6 and V_9, we could use the results from the Steiner tree in Figure 2 to suggest near-optimal routes. It is assumed the reader is familiar with Steiner trees. There are several techniques for constructing near-optimal Steiner trees found in the literature [1,6]. It is also assumed the reader is familiar with evolutionary algorithms and specifically genetic algorithms.

5. METHODOLOGY AND IMPLEMENTATION

The first step in implementing our algorithm was to design the chromosome encoding. We created a chromosome with two segments: the left segment called *request segment* is a permutation of the partial requests, and the right segment call the *percentage segment* is a string of percentage codes for the requests. See an example shown in Figure 3. The *request segment* represents the order in which partial requests are placed in the network. Each gene in the *request segment* represents one part of a request. For example, assume we have five requests (like those in Table 1) to route in some order. Each request, R1, R2, ... R5 is split into two parts, thus the *request order* segment will contain a permutation of 1, 1', 2, 2', ... 5, 5', where 1, and 1' for example, represent the two sub-requests of the original request R1. Thus if the problem is to route *n* requests, then the *request segment* portion of the chromosome will contain 2*n* genes. The *percentage segment* contains *n* genes, one for each request. The first gene in the *percentage segment* always refers to request R1, the second gene to request R2, and so forth. The value indicates how much of the transmission is assigned to the "prime" (second) sub-request of the original request in multiples of 10%. In Figure 3, the *percentage segment*: 1,3,5,0,1 means the 1' request in the *request segment* get 10% of the transmission (1 unit in this case) and consequentially request 1 in the *request segment* gets 90% (9 units) of the original request, R1. Request 2' gets 30% of the transmission for the original request R2 (2 units) and request 2 in the *request segment* gets 70% (5 units), and so forth where the original request R5 (10 units) is split as 10% to 5' and 90% to 5. Note the values in the *percentage segment* range from zero to five. It is not necessary to have values greater that 5 in the *percentage segment*. A value of 7, for example, is identical to a value of 3 after switching the two sub-requests around. Note further that a value of zero is allowed as in request 4. This means 4' is void and all eight units of the original request R4 are routed by request 4 in the *request segment*.

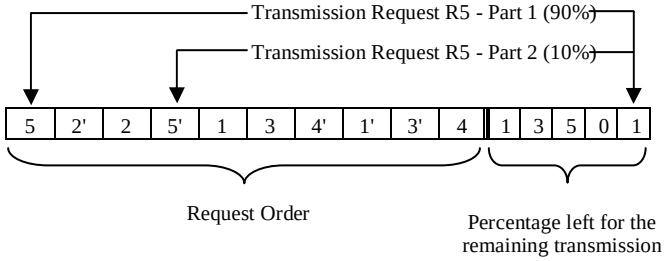

Figure 3. Example Gene Encoding

We solved the PMRP with single split using a hybrid algorithm based on a minimum Steiner tree algorithm we obtained from [6], and a Genetic Algorithm (GA). The GA determines the order in which requests are processed and how the bandwidth is divided into the two communication paths. For a given chromosome indicating a request order and transmission percentages, we route each request in the order given into the network using a Steiner tree algorithm to determine "best" possible route. After each request is placed into the network, the bandwidth of the request is subtracted from the capacity of the used edges (communication lines), and the Steiner tree algorithm is used again to place the next partial request. This continues until all of the requests are

placed in which case we have a feasible solution, or until we are unable to place a request (infeasible solution). Our algorithm is described below.

For each **chromosome** in the **population**
  repeat for each **sub-request** in the **request segment**
    for each **edge** in the **network**
      if **sub-request capacity** > **edge capacity**
        remove **edge** from consideration in the **network**
    run Steiner Tree Algorithm for the **sub-request** on **network**
    if the Steiner Tree Algorithm find a tree then
      for each **edge** used in the **Steiner Tree**
        subtract **sub-request capacity** from **edge capacity**
    add **all removed edges**
  until (**all sub-requests have been processed** or the
    Steiner Tree Algorithm did not find any tree (path))

Table 2 represents an example routing for the chromosome in Figure 3, using the requests in Table 1 and Network in Figure 1. Each row in Table 2 represents how each sub-request was routed and what percentage of the transmission. In this contrived instance every request was able to be routed, thus this represents a feasible solution to this particular PMRP Message Scheduling Problem.

Request	Source	Destination	Steiner Tree	Capacity
5	F	A,H	AD, DG, GH, FG	9
2'	A	E,F,G	AD, DE, EH, FG, GH	2
2	A	E,F,G	AB, BC, BE, CF, CH, GH	5
5'	F	A,H	AB, BC, CH, CF	1
1	G	A,C,F	AD, DG, CF, FG	9
3	H	B,D,G,F	BE, EH, DG, GH, FG	6
4'	skip	skip	skip	0
1'	G	A,C,F	AB, BE, EH, CH, CF, GH	1
3'	H	B,D,G,F	BC, CH, CF, FG, DG	6
4	C	E	BC, BE	8

Table 2. Example routing for Chromosome in Fig 3, using requests in Table 1 and Network in Figure 1.

For some chromosomes, the request order and associate transmission percentage may be such that not all requests can be routed. Assume in Figure 4 that only the first 8 partial-requests (out of 10) can fit in a network, then only 4 requests out of a possible 5 are complete (1,2,4, and 5.) Requests R1, R2, and R4 are complete because all their partial requests are placed into the network. Although partial request *5'* is not placed into the network, request R5 is complete because its main transmission carries the entire bandwidth (fifth gene in the *percentage* segment is 0.) Furthermore, request R3 is incomplete because 10% of its transmission (partial request *3'*) does not fit into the network.

5	**2'**	**2**	**4**	**1**	**3**	**4'**	**1'**	3'	5'	1	3	5	1	0

Figure 4. An example of a chromosome completing 8 sub-requests

We designed a fitness function that takes into account feasible and infeasible chromosomes (routing schedules) and at the same time rewards schedules that could route as many of the original requests as possible using only one path, that is the route was not split. Our fitness function for a chromosome is given by the following equation:

$$Fitness = (\#\text{ of Complete Requests}) + C \times (\#\text{ of Complete requests with Unique Path})$$

where $C < \frac{3}{\text{Chromosome Length}}$. This makes

$$C \times (\#\text{ of Complete requests with Unique Path}) < 1$$

For example, assuming C = 0.1, the fitness of the chromosome in Figure 4 is 4.1 (four completed requests where one was not split). The fitness for the chromosome depicted in Figure 3 is 5.1 (all five requests routed and one of them was not split). In our example case any chromosome with a fitness ≥ 5 is a feasible solution.

We used roulette wheel as our means for selection. The GA stopped after a maximum number of 200 generations or when the population converged. The population in each generation was composed of 198 offspring of the previous generation and the best two individuals from the previous generation.

We developed three crossover operators for our GA. The first crossover denoted as X1 is a combination of two classic crossovers, PMX and double point crossover. See Figure 5. The PMX crossover was used on the request segment of the chromosome to rearrange the sequence in which the requests are placed. The double point crossover was used in the percentage segment of the chromosome to change the percentage of the second part of the transmission.

Parents:

1	**2'**	**2**	**5'**	**3**	**5**	**4'**	**1'**	**3'**	**4**	**1**	**3**	**5**	**1**	**0**

4'	3'	2	1'	5	3	1	5'	4	2'	4	1	2	3	0

Children:

Figure 5. An example of PMX crossover and double point crossover combination (Crossover X1)

The second crossover denoted as X2, was based on the well known Order1 crossover for the request segment. The Order 1 crossover was developed by L. Davis and is described in great detail by Starkweather, *et. al* [9]. The crossover for the percentage segment directly depended on the result of the request segment. If both parts of the request R5, for instance, (*5* and *5'*) were between the crossover marks in one of the parents, their associated percentage genes were exchanged; otherwise, they remained the same. For example, in Figure 6 requests *5* and *5'* are between the crossover marks in the request segment hence the fifth genes in the percentage segment are exchanged (2 and 0).

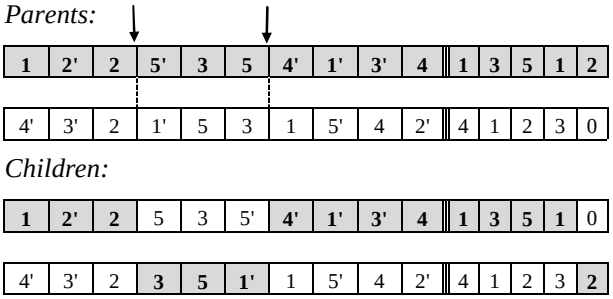

Figure 6. An example of modified Order1 (Crossover X2)

The third crossover denoted as X3 was also based on the Order1 crossover [9] for the request segment. Crossover on the percentage segment also depended on the result of the request segment. If the first part of a request was located between the crossover marks in both parents, their percentage genes were exchanged; otherwise, they remained the same. In Figure 7, requests *3* and *5* are between the crossover marks in the request segment hence the third and fifth genes in the percentage segment are exchanged between parents.

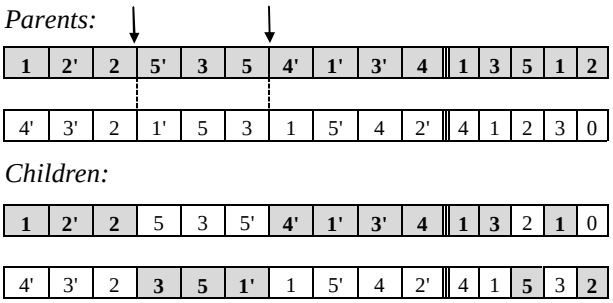

Figure 7. An example of modified Order1 (Crossover X3)

We used two types of mutation operations. One mutation operated on the request segment and the other operated on the percentage segment. The mutation used on the request segment involved swapping two random genes. For example in Figure 8 the third and the seventh genes in the chromosome are exchanged. The mutation operator for the percentage segment randomly selects a gene which is then altered to be a random value from 0 to 5. See Figure 8 where the second gene in the percentage segment is altered from 3 to 5.

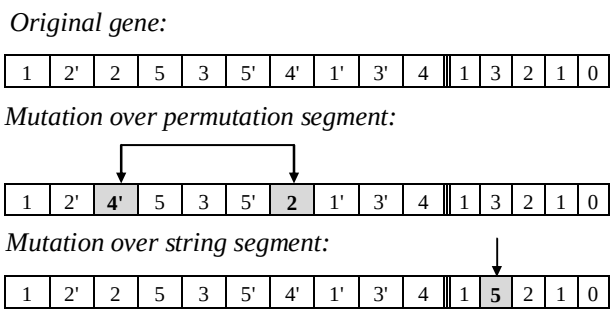

Figure 8. Two examples of chromosome mutations

6. RESULTS

The previous research on PMRP was done by Zhu *et. al* [10]. They did not allow any split transmissions. Since our algorithm allows for both full and split transmissions, our conjecture was that we should be able to at least reproduce Zhu's work and perhaps obtain even better results by using some split transmissions. We used two test datasets for our research, a network of 61 nodes with 133 edges (Dataset 1), and another network of 61 nodes with 100 edges (Dataset 2). These datasets were randomly generated. Each edge was assigned a random cost from 1 to 10 and a fix capacity of 12. A set of 20 randomly generated requests was used on both networks; each request had no more than 8 destinations. For comparison purposes, we ran our algorithms on the same two datasets use in Zhu's research.

Table 3 shows the number of requests routed by Zhu's algorithm using both Datasets. That is, Zhu's best effort was to route a subset of 18 of the 20 requests from Dataset1, and a subset of 12 of 20 requests from Dataset2.

	Routed
Dataset1	18
Dataset2	12

Table 3. Zhu's PMRP results for Dataset1 and Dataset2

Table 4 shows results from our hybrid algorithm using Dataset1 and Dataset2 with our three crossovers operators X1, X2, X3. In each of these six cases, we ran our algorithm forcing transmissions splits at five different percentages. For example, 0% means no splits were allowed in our algorithm (simulation of Zhu's work), 31% means in all instances the probability of a transmission being split into two parts was 31%. After that, the percentage of the transmission split was determined randomly. 100% means in every instance every

request was forced into a split of two parts. In each test case the number of routed requests is given and how many routed requests were unique (not split). We ran each case five different times with little variation in the results. The best result (which always occurred in at least three of the five runs) is recorded in Table 4.

		Dataset1		Dataset2	
		Routed	Unique	Routed	Unique
X1	0%	19	19	14	14
	31%	20	18	15	15
	45%	20	18	16	13
	83%	20	19	16	10
	100%	20	0	16	0
X2	0%	19	19	14	14
	31%	20	15	15	13
	45%	20	14	14	12
	83%	20	7	**17**	4
	100%	20	0	16	0
X3	0%	19	19	14	14
	31%	20	9	15	12
	45%	20	13	15	13
	83%	20	7	16	3
	100%	20	0	16	0

Table 4. Our PMRP with single split paths results for Dataset1 and Dataset2 using various crossovers and routing probabilities

7. CONCLUSION

We were able obtain better results than that reported by Zhu [10]. Note the 0% entries for all crossovers in Table 4. This represents all unique paths (no splits). We were able to route 19 requests using Dataset1 compared to 18 requests by Zhu, and route 14 requests using Dataset2 compared to 12 requests by Zhu. One would expect that the results of our 0% split would be identical to Zhu's results. In fact our results turned out to be superior. The reason for this is that we used a different chromosome representation of the problem. As expected, our results also showed that allowing transmission requests to be routed using 2 paths improved the usability the network. For Dataset1 we were able to route all 20 requests using all three of our crossover operators. Furthermore, using Dataset2 we were able to route 16, 17, 16 requests out of 20 using crossover operators X1, X2, X3, respectively.

We can also see in Table 4 that the best solution was found using crossover X2, but over all, better solutions were found using crossover X1. It appears from our test cases that the crossover that worked independently on both segments of the chromosome, (that is X1), obtained better results that the other crossover operators that had dependencies between both segments of the chromosome.

Acknowledgements

The authors gratefully acknowledge Michael Alexander and Gabriel Robins for providing code for several Steiner tree algorithm 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] T. Al-Qahtani, M. Abedin, S. Ahson, "Dynamic Routing in Homogenous ATM Networks using Genetic Algorithms*", Proceeding of the 1998 IEEE International Conference on Evolutionary Computing* (ICEC'98), part of WCCI, Anchorage, Alaska, May 4-9, 1998.

[3] G. Chakraborty, Y. Hirano, "Genetic Algorithm for Broadcast Scheduling in Packet Radio Networks", *Proceeding of the 1998 IEEE International Conference on Evolutionary Computing* (ICEC'98), part of WCCI, Anchorage, Alaska, May 4-9, 1998.

[4] H.L. Christensen, R.L. Wainwright and D.A. Schoenefeld, "A Hybrid Algorithm for The Point to Multipoint Routing Problem", *Proceedings of the 1997 ACM Symposium on Applied Computing*, ACM Press, *1997, pp 263-268.*

[5] 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.*

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

[7] Joshua Knowles and David Corne, "Heuristics for Evolutionary Off-Line Routing in Telecommunications Networks", in *Proceedings of the Genetic and Evolutionary Computation Conference* (GECCO 2000), Morgan Kaufmann Publishers, 2000, pp. 574 – 581.

[8] M. Ryan, J. Debuse, G. Smith, and I. Whittley, "A Hybrid Algorithm for the Fixed Channell Assignment Problem*", in Proceedings of the Genetic and Evolutionary Computation Conference* (GECCO 1999), Morgan Kaufmann Publishers, 1999, pp. 1707-1714.

[9] T. Starkweather, S.McDaniel, K. Mathis, D. Whitley, and C. Whitley, "A Comparison of genetic sequencing operators", in *Proceedings of the Fourth International Conference in Genetic Algorithms,* San Diego, California, 1991, Morgan Kaufmann, publishers, pp. 69-76.

[10] L. Zhu, R. Wainwright, D. Schoenefeld, "A Genetic Algorithm for the Point to Multipoint Routing Problem with Varying Number of Requests", *Proceeding of the 1998 IEEE International Conference on Evolutionary Computing* (ICEC'98), part of WCCI, Anchorage, Alaska, May 4-9, 1998.